DERS KİTABI

Algoritmik Oyun Kuramı

Oyunlar, denge, mekanizmalar ve hesaplama

Sercan Külcü

Giresun Üniversitesi · Bilgisayar Mühendisliği Bölümü

Algoritmik Oyun Kuramı

Sercan Külcü
Giresun Üniversitesi, Bilgisayar Mühendisliği Bölümü
Giresun, Türkiye

Bu kitap, şu adreste yayımlanan ders notlarının basılı hâlidir:
`sercankulcu.github.io/teaching/algorithmic-game-theory/`

Sunumlar, çalışma soruları, canlı sınıf oyunları ve notlardaki etkileşimli şekillerin çalışan hâlleri o sayfadadır. Kitaptaki etkileşimli şekiller varsayılan değerleriyle basılmıştır.

© Sercan Külcü. Tüm hakları saklıdır.

Notların 14 Kasım 2026 tarihli hâlinden üretilmiştir.

Önsöz

Algoritmik oyun kuramı, kendi çıkarını gözeten tarafların etkileşimini hesaplama gözüyle inceler: oyun kuramının modellerini algoritmalar ve karmaşıklık çözümlemesiyle birleştirir. Bu kitap, Giresun Üniversitesi Bilgisayar Mühendisliği Bölümü'nde verilen Algoritmik Oyun Kuramı dersinin notlarından oluşur ve bölümleri dersteki sırayla izler.

Kitap beş kısımdan oluşur. Kısımlar sırasıyla şunlardır: **Araçlar ve Temeller** (Bölüm 0–3), **Mekanizmalar** (Bölüm 4–7), **Ağlar ve Verimlilik** (Bölüm 8–9), **İşbirliği ve Toplum** (Bölüm 10–12) ve **Dinamikler ve Gelecek** (Bölüm 13–14). Bölüm 0, derste gerçekten kullanılan matematiği oyun örnekleri üzerinden yeniden kurar; bu konulara hâkim okur doğrudan Bölüm 1'den başlayabilir.

Her bölüm aynı düzeni izler. Başta öğrenme çıktıları yer alır. Biçimsel tanımlar, teoremler ve çözülmüş örnekler adlandırılmış kutulardadır. *Sık yapılan hata* kutuları, öğrencilerin en sık düştüğü yanılgıları gösterir. Kod izlemeleri, hesapların kısa Python fonksiyonlarıyla nasıl yapıldığını adım adım gösterir. Bölüm; bir özet, bir kavram listesi, cevaplarıyla birlikte *Kendini Dene* soruları, ileri okuma önerileri ve kaynakçayla kapanır. Kitabın sonundaki terim sözlüğü, her İngilizce terim için notlarda kullanılan tek Türkçe karşılığı verir.

Bazı şekiller web sayfasında kaydırıcıyla değiştirilebilen etkileşimli çizimlerdir; kitapta varsayılan değerleriyle basılmıştır. Sunumlar, çalışma soruları ve sınıfta oynanan oyunlar ders sayfasındadır.

Sercan Külcü
Giresun

İçindekiler

Dersin Haritası 1

KISIM I · ARAÇLAR VE TEMELLER

0 Hazırlık 4

0.1 Giriş 5
0.2 Kümeler ve fonksiyonlar 6
0.3 Toplam ve en büyük 12
0.4 Üs, logaritma ve türev kuralları 21
0.5 Olasılık ve beklenen değer 28
0.6 Karışık strateji 44
0.7 Kazanç matrisi 48
0.8 En iyi tepki 52
0.9 Denklemler ve kayıtsızlık 55
0.10 Oyun ağacı ve büyüme hızı 62
0.11 Sık yapılan hatalar 66
0.12 Özet 67
0.13 Kavram listesi 68
0.14 Kendini Dene 69
0.15 İleri okuma 70
0.16 Kaynaklar 71

1 Karar Kuramı 72

1.1 Beklenen fayda 72
1.2 Risk tutumları 83
1.3 Bayesçi karar 93
1.4 Karar ağaçları ve bilginin değeri 102
1.5 Tartışma: risk ve belirsizlik altında daha iyi karar 109
1.6 Özet 111
1.7 Kavram listesi 112
1.8 Kendini Dene 113
1.9 İleri okuma 115
1.10 Kaynaklar 115

2 Oyunlar ve Temsiller 116

2.1 Temel kavramlar 117
2.2 Oyunların temsili 128
2.3 Oyunların sınıflandırılması 137
2.4 Denge ve Mahkûmun İkilemi 140
2.5 Klasik 2×2 oyunlar 146
2.6 2/3 ortalaması oyunu 150
2.7 Kombinatoryal oyunlar 156

2.8 Sıralı oyunlar ve geriye doğru çıkarım 166
2.9 Minimaks ve alfa-beta budaması 177
2.10 Algoritmik bakış ve mekanizma tasarımı 204
2.11 Çözümlü problemler 207
2.12 Özet 208
2.13 Kavram listesi 209
2.14 Kendini Dene 210
2.15 İleri okuma 213
2.16 Kaynaklar 214

3 Nash Dengesi 216

3.1 Temeller 216
3.2 En iyi tepki 222
3.3 Baskın strateji eleme 234
3.4 Karışık strateji 243
3.5 Varlık ve denge seçimi 253
3.6 Minimaks ve karmaşıklık 260
3.7 Tartışma ve kontrol listesi 270
3.8 Özet 271
3.9 Kavram listesi 272
3.10 Kendini Dene 273
3.11 İleri okuma 275
3.12 Kaynaklar 276

KISIM II · MEKANİZMALAR

4 Mekanizma Tasarımı 278

4.1 Temel kavramlar 278
4.2 Doğrudan mekanizmalar 283
4.3 VCG mekanizması 299
4.4 Parasız mekanizmalar 307
4.5 Bayesçi oyunlar 325
4.6 Değerlendirme 330
4.7 Özet 333
4.8 Kavram listesi 334
4.9 Kendini Dene 334
4.10 İleri okuma 337
4.11 Kaynaklar 337

5 Açık Artırmalar 339

5.1 Temel kavramlar 339
5.2 Dört temel tür 345
5.3 Kapalı zarf stratejileri 351
5.4 Gelir ve bilgi 361
5.5 Çift taraflı piyasalar 373
5.6 Değerlendirme 380

5.7 Özet 382
5.8 Kavram listesi 383
5.9 Kendini Dene 383
5.10 İleri okuma 385
5.11 Kaynaklar 386

6 Çevrim İçi Platformlar 387

6.1 Platformlar ve pazarlar 388
6.2 Ağ etkileri ve kritik kitle 392
6.3 Platform fiyatlandırması 400
6.4 Sponsorlu arama 409
6.5 İtibar ve rekabet 425
6.6 Tartışma 431
6.7 Özet 432
6.8 Kavram listesi 433
6.9 Kendini Dene 434
6.10 İleri okuma 437
6.11 Kaynaklar 437

7 Sosyal Seçim ve Oylama 439

7.1 Sosyal seçim 440
7.2 Oylama kuralları 444
7.3 Condorcet paradoksu 459
7.4 Stratejik oy 468
7.5 İmkânsızlık teoremleri 470
7.6 Hesaplamalı sosyal seçim ve uygulamalar 479
7.7 Özet 483
7.8 Kavram listesi 484
7.9 Kendini Dene 485
7.10 İleri okuma 487
7.11 Kaynaklar 487

KISIM III · AĞLAR VE VERİMLİLİK

8 Ağ Tasarımı ve Yönlendirme 490

8.1 Ağlar ve yönlendirme 490
8.2 Bencil yönlendirme 497
8.3 Braess paradoksu 508
8.4 Tıkanıklık oyunları 516
8.5 Ağ oluşturma 523
8.6 Kontrol ve teşvikler 533
8.7 Özet 537
8.8 Kavram listesi 538
8.9 Kendini Dene 538
8.10 İleri okuma 541
8.11 Kaynaklar 541

9 Anarşinin Bedeli 543

9.1 Temel kavramlar 543
9.2 Pigou ağı 550
9.3 Doğrusal gecikme ve Braess 556
9.4 Yük dengeleme 567
9.5 İstikrarın Bedeli 576
9.6 PoA'yı azaltma 582
9.7 Çözümlü problemler 587
9.8 Özet 589
9.9 Kavram listesi 590
9.10 Kendini Dene 591
9.11 İleri okuma 593
9.12 Kaynaklar 593

KISIM IV · İŞBİRLİĞİ VE TOPLUM

10 Çoklu Ajan Sistemleri 596

10.1 Ajanlar ve çoklu ajan sistemleri 596
10.2 İşbirliği ve koordinasyon 602
10.3 Görev tahsisi 610
10.4 Koalisyon oyunları 621
10.5 Kamu malı oyunu 642
10.6 Özet 647
10.7 Kavram listesi 648
10.8 Kendini Dene 649
10.9 İleri okuma 652
10.10 Kaynaklar 652

11 Algoritmik Adalet ve Etik 654

11.1 Giriş 654
11.2 Adalet ölçütleri 657
11.3 Ölçütlerin uyumsuzluğu 672
11.4 Adil bölüşüm 680
11.5 Etik ve yönetişim 697
11.6 Özet 704
11.7 Kavram listesi 705
11.8 Kendini Dene 706
11.9 İleri okuma 708
11.10 Kaynaklar 709

12 Davranışsal ve Kolektif Karar 710

12.1 Davranışsal karar 711
12.2 Sosyal tercihler 725
12.3 Kolektif kararlar ve ortak kaynaklar 731
12.4 Kamu malı ve ceza 737
12.5 Eşikli projeler 745

12.6 Toplu bilgi 747
12.7 Kolektif kararda sık hatalar ve örnek problemler 756
12.8 Özet 757
12.9 Kavram listesi 758
12.10 Kendini Dene 759
12.11 İleri okuma 761
12.12 Kaynaklar 762

KISIM V · DİNAMİKLER VE GELECEK

13 Öğrenme ve Adaptasyon 765
13.1 Giriş 766
13.2 En iyi tepki ve kurgusal oyun 769
13.3 Pişmanlık ve Hedge 778
13.4 Çok kollu haydut 788
13.5 Pekiştirmeli öğrenme 797
13.6 Evrimsel dinamik 801
13.7 Tekrarlanan oyunlar 806
13.8 Tartışma: öğrenme, insan ve etik 813
13.9 Özet 814
13.10 Kavram listesi 815
13.11 Kendini Dene 816
13.12 İleri okuma 818
13.13 Kaynaklar 818

14 Gelecek Yönelimler 820
14.1 Giriş 820
14.2 Yapay zekâ ajanları 823
14.3 Öğrenen mekanizmalar 836
14.4 Blokzincir mekanizmaları 846
14.5 Ölçek ve açık sorular 856
14.6 Değerlendirme: sık hatalar ve örnek problemler 861
14.7 Özet 864
14.8 Kavram listesi 865
14.9 Kendini Dene 866
14.10 İleri okuma 867
14.11 Kaynaklar 868

EKLER

Ek A Terim Sözlüğü 869

Ek B Algoritmik Oyun Kuramı İçin Ön Bilgiler 879

Ek C Algoritmik Oyun Kuramında Önemli İsimler 900

ALGORİTMİK OYUN KURAMI

Dersin Haritası

Bu sayfa dersin bütün bölümlerini tek bir haritada birleştirir. Her bölüm bir düğümdür; bir ok, sonraki konunun önceki konunun hangi fikrine dayandığını gösterir. Derse başlamadan önce okunursa konuların nereye gittiğini, ders sırasında okunursa bulunulan yeri gösterir.



Şekil 1: Dersin haritası. Oklar, bir konunun hangi konunun üzerine kurulduğunu gösterir. Bütün yollar Hazırlık'tan başlar; ana dallar Gelecek Yönelimler'de birleşir.

Şekil 1 konular arasındaki bağları gösterir. Her düğüm bir bölümdür; bir ok, sonraki konunun önceki-nin hangi fikrine dayandığını belirtir.

Konu (bölüm)	Dayandığı konu	Bağ
Hazırlık (0)	—	Σ ve argmax, olasılık ve beklenen değer, kazanç matrisini okuma, kayıtsızlık denklemi; bütün bölümler bu araçları kullanır.
Karar Kuramı (1)	Hazırlık	Tek kişilik karar: beklenen fayda, risk tutumu, Bayesçi karar ve bilginin değeri.
Oyunlar ve Temsiller (2)	Karar Kuramı	Karar verenler çoğalınca oyun doğar: normal ve genişletilmiş form, geriye doğru çıkarım.
Nash Dengesi (3)	Oyunlar ve Temsiller	2×2 oyunlarda işaretlenen dengeler genel Nash tanımına ve varlık teoremine götürür.
Mekanizma Tasarımı (4)	Nash Dengesi	Denge + teşvik: oyunu çözmek yerine istenen dengeyi veren kuralı tasarlamak.

Konu (bölüm)	Dayandığı konu	Bağ
Açık Artırmalar (5)	Mekanizma Tasarımı	VCG'nin tek nesneli hâli ikinci fiyat (Vickrey) açık artırmasıdır.
Çevrim İçi Platformlar (6)	Açık Artırmalar	Reklam yuvası açık artırmaları: GSP, Vickrey fikrinin çok yuvalı genellemesidir.
Sosyal Seçim ve Oylama (7)	Mekanizma Tasarımı	Sosyal seçim fonksiyonu → oylama kuralları; Gibbard–Satterthwaite.
Ağ Tasarımı ve Yönlendirme (8)	Nash Dengesi	Tıkanıklık oyunlarında Nash dengesi = Wardrop dengesi; Rosenthal potansiyeli.
Anarşinin Bedeli (9)	Ağ Tasarımı ve Yönlendirme	Pigou ve Braess ağlarında denge kalitesi → anarşinin bedeli 4/3; tıkanıklık ücreti.
Çoklu Ajan Sistemleri (10)	Oyunlar ve Temsiller	İşbirlikçi / işbirlikçi olmayan ayrımı → koalisyon oyunları, Shapley değeri.
Algoritmik Adalet ve Etik (11)	Sosyal Seçim ve Oylama	Kolektif karardan adalete: ölçütler ve adil bölüşüm.
Davranışsal ve Kolektif Karar (12)	Karar Kuramı	İnsanlar beklenen faydadan sapar (kayıptan kaçınma); ortak mallarda bireysel ve toplu çıkar ayrışır.
Öğrenme ve Adaptasyon (13)	Nash Dengesi	Denge nasıl bulunur? Oyuncular öğrenerek dengeye yakınsayabilir.
Gelecek Yönelimler (14)	Anarşinin Bedeli, Öğrenme ve Adaptasyon, Algoritmik Adalet ve Etik	Ölçek, öğrenen ajanlar ve etik bugünün açık sorularında birleşir.

Tablo 1: Haritadaki bağlar.

Haritanın ana omurgası şöyle okunabilir: **denge → tasarım → uygulama → öğrenme → gelecek**. Tek kişilik karardan oyunlara, oyunlardan dengeye geçilir. Denge kavramı iki yöne açılır: kuralları tasarlamak (mekanizma, açık artırma, oylama) ve var olan kuralların sonucunu ölçmek (ağlar, anarşinin bedeli). Öğrenme, dengeye nasıl ulaşıldığını sorar. Son bölümün (Gelecek Yönelimler) konuları, bu dalların birleştiği yerdedir. Örneğin algoritmik danışıklılık hem tekrarlanan oyunlara hem öğrenmeye, işlem ücreti mekanizmaları hem açık artırmalara hem tıkanıklık fiyatlandırmasına dayanır.

KISIM I

Araçlar ve Temeller

Bölüm 0–3

KISIM I · BÖLÜM 0

Hazırlık

Oyun kuramı, birden çok karar vericinin birbirini etkileyen seçimlerini inceler. Bu seçimleri kesin bir dille yazmak ve sonuçlarını hesaplamak için az ama sağlam bir matematik gerekir: kümeler ve fonksiyonlar, toplam ve en büyük değer, türev, olasılık ve beklenen değer, matris okuma, basit denklemler ve ağaçlar. Bu not, dersin bütün bölümlerinde kullanılan bu araçları tek tek, sıfırdan ve oyun örnekleriyle anlatır. Her kavram önce sade bir dille, sonra biçimsel olarak tanımlanır; ardından ara adımları atlanmadan çözülmüş bir örnek ve sık yapılan bir hata gelir. Matematik temeli zayıf olan bir okur için not tek başına yeterli olacak biçimde yazılmıştır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum00`

ÖĞRENME ÇIKTILARI

Gösterim
Küme, Kartezyen çarpım ve $u_1(s_1, s_2)$ gibi fonksiyon gösterimini okumak.

Σ, max, argmax
Toplamı ve geometrik seriyi hesaplamak; en büyük değeri ve onu veren seçeneği, sürekli seçimde türevle (zincir kuralı, kısmi türev) bulmak.

Olasılık
Sayarak, çarparak ve koşullu olasılıkla (Bayes) olasılık bulmak; $U[0, 1]$ ile çalışmak.

Beklenen değer
Bir piyangonun ve karışık stratejinin beklenen kazancını hesaplamak; üs, ln ve softmax ifadelerini okumak.

Matris ve en iyi tepki
(a, b) çiftini doğru okumak, en iyi tepkileri matriste işaretlemek.

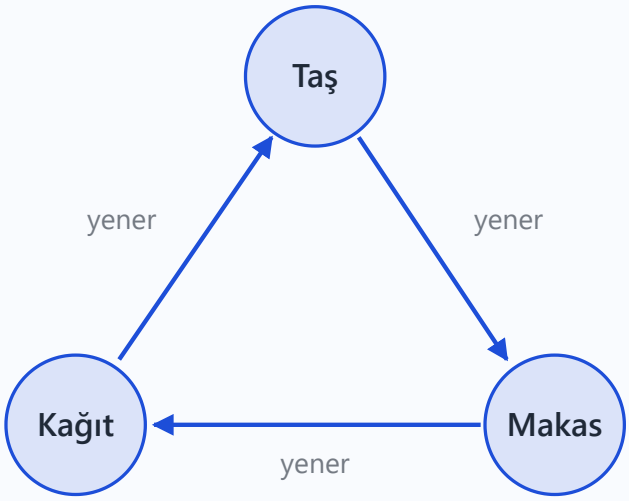
Denklem ve ağaç
İkinci derece denklemi ve kayıtsızlık denklemini çözmek; oyun ağacını yapraktan köke çözmek; O ve Θ ile büyüme hızını okumak.

0.1 Giriş

Hazırlık bölümünde yeni bir oyun kuramı kavramı yoktur. Amaç, oyun kuramının konuştuğu dili tanıtmaktır. Bu dil birkaç sembolden, birkaç hesap kuralından ve birkaç okuma alışkanlığından oluşur. Hepsi küçük bir oyun üzerinden somutlaşır: Taş-Kağıt-Makas.

0.1.1 Bir oyunla başlangıç: Taş-Kağıt-Makas

İki oyuncu aynı anda Taş, Kağıt ya da Makas seçer. Taş Makas'ı, Makas Kağıt'ı, Kağıt da Taş'ı yener; aynı seçim berabere biter. Kazanan 1 puan alır, kaybeden 1 puan verir, beraberlikte kimse puan almaz. Şekil 0.1 bu yenme ilişkisini bir döngü olarak gösterir: her ok, başladığı hamlenin ucundaki hamleyi yendiğini gösterir.



Şekil 0.1: Taş-Kağıt-Makas'ın yenme döngüsü. Taş Makas'ı, Makas Kağıt'ı, Kağıt Taş'ı yener. Hiçbir hamle ötekilerin hepsini yenmez.

Döngünün önemli bir sonucu vardır: “en iyi hamle” diye tek bir hamle yoktur. Taş, Makas'a karşı iyidir ama Kağıt'a karşı kötüdür. Hangi hamlenin iyi olduğu tamamen rakibin ne oynadığına bağlıdır. Oyun kuramının bütün konusu bu bağımlılıktır: bir oyuncunun kazancı yalnız kendi seçimine değil, rakibin seçimine de bağlıdır.

Oyunu bir bilgisayara karşı oynadığımızı düşünelim. Bilgisayar iki türlü davranabilir. Birinci türde her turda rastgele bir hamle seçer. İkinci türde bizim hamlelerimizi izler: hangi hamleyi ne sıklıkla yaptığımızı ya da hangi hamleden sonra hangisini yapmaya eğilimli olduğumuzu öğrenir ve buna göre oynar. İkinci tür bilgisayar karşısında iki soru doğal olarak ortaya çıkar:

- Öğrenen bir bilgisayarın uzun vadede yenemeyeceği bir oynayış var mı?
- Hep Taş oynayan biri, rakibi bunu fark edince tur başına ortalama kaç puan kaybeder?

Bu iki soruyu cevaplamak için gereken araçlar şunlardır: hamleleri ve sonuçları yazmak için küme ve fonksiyon gösterimi; "ortalama kaç puan" sorusu için olasılık ve beklenen değer; "en iyi karşılık" sorusu için argmax ve en iyi tepki; "yenilmeyen oynayış" için kayıtsızlık denklemi. Bu araçlar sırayla kurulunca iki soru da tam olarak cevaplanır (0.9.5).

0.1.2 Oyun kuramında matematiğin rolü

Matematiğin oyun kuramındaki ilk rolü bir *yazım düzeni* olmasıdır. "Ayşe Taş, Bora Makas oynarsa Ayşe 1 puan alır" cümlesi uzun ve belirsizdir; aynı bilgi $u_1(Taş, Makas) = 1$ olarak kısa ve kesin yazılır. Oyun büyüdükçe bu kısalık bir zorunluluk olur: on oyunculu bir oyunun bütün sonuçlarını cümlelerle anlatmak mümkün değildir.

İkinci rol *hesaplamaktır*. Bir oyuncu "rakibim yarı yarıya Taş oynuyorsa ben ne yapmalıyım?" diye sorduğunda cevap bir sayı karşılaştırmasıdır: her hamlenin ortalama getirisi hesaplanır ve büyük olan seçilir. Bunun için olasılık, toplam ve en büyük değer bulma yeterlidir. Seçim sürekli olduğunda (bir firmanın kaç birim üreteceği gibi) en büyük değer türevle bulunur.

Üçüncü rol *ölçmektir*. Bir yöntemin bilgisayarda ne kadar süreceği, girdinin büyüklüğüne göre büyüme hızıyla anlatılır. Algoritmik oyun kuramının "algoritmik" kısmı buradan gelir: bir dengeyi bulmak mümkün mü, mümkünse ne kadar hızlı? Bu soruların dili O ve Θ gösterimidir (0.10.3).

Notun her kesimi aynı sırayı izler: önce kavramın sade bir anlatımı, sonra biçimsel tanımı, ardından neden önemli olduğu, çözülmüş bir örnek ve en sık yapılan hata. Örneklerdeki sayılar sunumdaki sayılarla aynıdır.

0.2 Kümeler ve fonksiyonlar

Bir oyunu yazmak için üç şey gerekir: kim oynuyor, her oyuncunun seçenekleri neler ve her durumda kim ne kazanıyor. Seçenekler bir küme olarak, "her durum" Kartezyen çarpım olarak, "kim ne kazanıyor" da bir fonksiyon olarak yazılır. Sayma kuralları da bu üç yapıyla yakından ilgilidir.

0.2.1 Küme ve Kartezyen çarpım

Küme, nesnelerin sırasız bir topluluğudur. Oyun kuramında en sık kullanılan küme, bir oyuncunun seçeneklerinin listesidir. Küme süslü parantezle yazılır: *{Taş, Kağıt, Makas}*. Kümede sıra önemsizdir; *{Taş, Kağıt}* ile *{Kağıt, Taş}* aynı kümedir. Bir eleman bir kümede iki kez yazılmaz.

Tanım 0.1

Bir A kümesinin elemanı olan a için $a \in A$ yazılır ve "a, A'nın elemanıdır" diye okunur. Kümenin eleman sayısı $|A|$ ile gösterilir.

İki küme A ve B'nin **Kartezyen çarpımı** (*Cartesian product*), birinci elemanı A'dan, ikinci elemanı B'den gelen bütün sıralı çiftlerin kümesidir:

$$A \times B = \{(a, b) : a \in A, b \in B\}$$

Eleman sayısı çarpılır: $|A \times B| = |A| \cdot |B|$.

Tanımdaki "sıralı" sözcüğü çok önemlidir. Bir kümede sıra önemsizdir ama bir çiftte sıra önemlidir: (a, b) çiftinde birinci yer A'ya, ikinci yer B'ye aittir. İki çift, ancak her iki yerdeki elemanları aynıysa eşittir.

Oyun kuramında oyuncuların seçeneklerine **strateji**, bir oyuncunun bütün seçeneklerinin kümesine **strateji kümesi** denir. Oyuncu 1'in strateji kümesi S_1, oyuncu 2'ninki S_2 ile gösterilir. Oyunun bir turunda her oyuncu bir strateji seçer. İki oyuncunun seçimlerinin birlikte oluşturduğu çifte **strateji profili** (*strategy profile*) denir. Bütün olası profillerin kümesi tam olarak Kartezyen çarpımdır: $S_1 \times S_2$. Böylece Kartezyen çarpım, "oyunda neler olabilir?" sorusunun eksiksiz cevabı olur.

Örnek 0.1

Ayşe ile Bora Taş-Kağıt-Makas oynuyor. Ayşe'nin strateji kümesi $S_1 = \{\text{Taş}, \text{Kağıt}, \text{Makas}\}$'tır.

1. Taş, Ayşe'nin stratejilerinden biridir: $\text{Taş} \in S_1$. Kümede üç eleman vardır: $|S_1| = 3$.
2. Bora'nın seçenekleri de aynıdır: $S_2 = \{\text{Taş}, \text{Kağıt}, \text{Makas}\}$, $|S_2| = 3$.
3. Profiller birinci yere Ayşe'nin, ikinci yere Bora'nın seçimi yazılarak kurulur. Ayşe'nin her seçimi Bora'nın her seçimiyle eşleşir: Taş ile üç çift, Kağıt ile üç çift, Makas ile üç çift.
4. Toplam profil sayısı $|S_1 \times S_2| = 3 \cdot 3 = 9$'dur.

Dokuz profil, Şekil 0.2'de üçe üç bir tablo olarak dizilmiştir. Satırlar Ayşe'nin, sütunlar Bora'nın seçimidir; her hücre bir profildir.



Şekil 0.2: $S_1 \times S_2$: Taş-Kağıt-Makas'ın dokuz strateji profili. Satırlar Ayşe'nin (kırmızı), sütunlar Bora'nın (mavi) seçimidir. Turuncu iki hücre (Taş, Kağıt) ve (Kağıt, Taş) farklı profillerdir.

Tablo biçimi rastlantı değildir. İki oyunculu her oyunun profilleri böyle bir tabloya dizilir; hücrelere kazançlar yazıldığında bu tablo **kazanç matrisi** olur (0.7). Üç oyunculu bir oyunda profiller üç elemanlı listelerdir: $(s_1, s_2, s_3) \in S_1 \times S_2 \times S_3$. Bu tür listelere üçlü, genel olarak n'li denir. Sayma kuralı aynıdır: eleman sayıları çarpılır.

Sık yapılan hata

(Taş, Kağıt) ile (Kağıt, Taş) aynı sanılır. Çiftte sıra önemlidir: ilk eleman Ayşe'nin, ikinci eleman Bora'nın seçimidir. (Taş, Kağıt) profilinde Ayşe Taş, Bora Kağıt oynamıştır ve Bora kazanır. (Kağıt, Taş) profilinde roller yer değiştirmiştir ve Ayşe kazanır. Biri Ayşe'nin, öteki Bora'nın kazandığı iki ayrı sonuçtur (Şekil 0.2'deki turuncu hücreler). Bu yanılgı, profil sayısını da yanlış saydırır: sırasız düşünen biri dokuz yerine altı sonuç bulur.

0.2.2 Fonksiyon gösterimi

Fonksiyon bir makine gibi düşünülebilir: bir girdi alır ve *tek* bir çıktı verir. Aynı girdi her zaman aynı çıktıyı verir. Oyun kuramında en önemli fonksiyon **kazanç fonksiyonudur** (*payoff function*). Girdisi bir strateji profili, çıktısı bir sayıdır: o profil oynandığında bir oyuncunun aldığı puan.

Tanım 0.2

$f : X \to Y$ yazımı, f'nin X kümesindeki her elemana Y kümesinden tam olarak bir eleman karşılık getiren bir fonksiyon olduğunu söyler. x'e karşılık gelen eleman $f(x)$ ile gösterilir.

Oyuncu i'nin kazanç fonksiyonu $u_i : S_1 \times S_2 \to \mathbb{R}$'dir. Burada $\mathbb{R}$ gerçek sayılar kümesidir. Yazım "profilden gerçek sayıya" diye okunur.

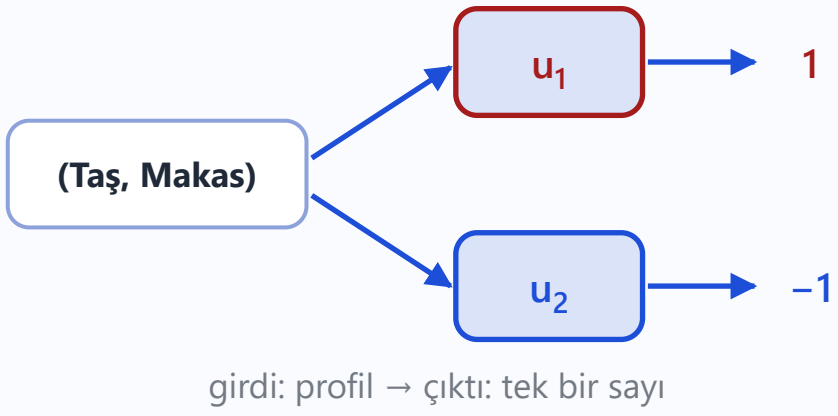
Gösterimdeki alt indis, kimin kazancına bakıldığını söyler. Aynı profil için u_1 Ayşe'nin, u_2 Bora'nın kazancını verir. Bu iki sayı genellikle farklıdır. Parantez içindeki iki strateji ise profili, yani "kim ne oynadı?" sorusunun cevabını taşır; sırası Kartezyen çarpımdaki sırayla aynıdır.

Örnek 0.2

Taş-Kağıt-Makas'ta birkaç kazanç değeri tek tek okunur.

1. $u_1(Taş, Makas)$: Ayşe Taş, Bora Makas oynamıştır. Taş Makas'ı yener, Ayşe 1 puan alır. Değer 1'dir.
2. $u_2(Taş, Makas)$: Aynı profil, ama bu kez Bora'nın kazancı sorulur. Bora kaybetmiştir ve 1 puan verir. Değer -1'dir.
3. $u_1(Kağıt, Kağıt)$: Berabere. Değer 0'dır.
4. $u_1(Taş, Kağıt)$: Kağıt Taş'ı yener; kaybeden Ayşe'dir. Değer -1'dir.

Bu oyunda her profilde $u_1 + u_2 = 0$'dır: birinin kazandığını öteki kaybeder. Bu özellikteki oyunlara **sıfır toplamlı oyun** (*zero-sum game*) denir.



Şekil 0.3: Kazanç fonksiyonu bir makine gibidir: girdi bir profil (Taş, Makas), çıktı tek bir sayıdır. u_1 makinesi 1, u_2 makinesi −1 verir.

Şekil 0.3 aynı girdinin iki ayrı makineye girdiğini gösterir. Bir oyunda her oyuncunun kendi kazanç fonksiyonu vardır. Oyunu tam olarak tanımlamak, her oyuncu için bu fonksiyonun bütün profillerdeki değerini vermek demektir. İki oyunculu Taş-Kağıt-Makas'ta bu, $2 \cdot 9 = 18$ sayı eder.

Fonksiyon gösterimi neden bu kadar önemlidir? Çünkü oyun kuramındaki tanımların hepsi bu gösterimle yazılır. Örneğin "Ayşe, Bora'nın Makas oynadığını bilirse Taş'ı Kağıt'a tercih eder" cümlesi $u_1(Taş, Makas) > u_1(Kağıt, Makas)$ eşitsizliğiyle, yani $1 > -1$ ile söylenir. Gösterim okunabildiğinde tanımlar da okunabilir.

Sık yapılan hata

$u_1(Taş, Kağıt)$ değerini "Kağıt kazanır" diye okumak. Fonksiyonun değeri bir sözcük değil, bir sayıdır: $u_1(Taş, Kağıt) = -1$. Alt indis kimin kazancına bakıldığını söyler; u_1 ile u_2 aynı profilde farklı sayılar verir. Alt indisi atlayan biri, Bora'nın kazancını Ayşe'ninki sanabilir.

0.2.3 Sayma: çarpım kuralı, n! ve C(n, k)

Oyun kuramında sık sorulan bir soru "kaç tane?" sorusudur: kaç strateji profili var, bir oyuncunun kaç saf stratejisi var, bir ağaçta kaç yaprak var? Bu soruların cevabı üç basit sayma kuralıyla verilir. Olasılık hesaplarında da aynı kurallar kullanılır, çünkü eşit olası sonuçlarda olasılık bir sayma oranıdır (0.5.1).

Tanım 0.3

- **Çarpım kuralı** (*product rule*): Art arda yapılan seçimlerde birinci seçimin a, ikincinin b seçeneği varsa iki seçim birlikte $a \cdot b$ farklı biçimde yapılabilir. Daha çok seçimde de seçenek sayıları çarpılır.
- **Faktöriyel**: n farklı nesne $n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$ farklı biçimde sıralanır. Tanım gereği $0! = 1$'dir.
- **Kombinasyon** (*combination, binomial coefficient*): n farklı nesneden sırası önemsiz k tanesini seçme yollarının sayısı

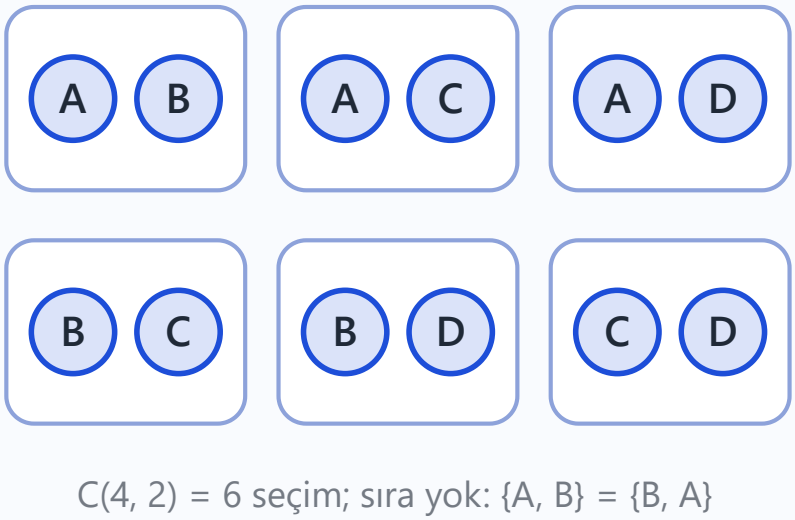
$$C(n, k) = n! / (k! \cdot (n-k)!)$$

Çarpım kuralı, Kartezyen çarpımın eleman sayısı kuralının genel hâlidir. Faktöriyel, çarpım kuralının özel bir uygulamasıdır: bir sıranın birinci yerine n nesneden biri, ikinci yerine kalan $n-1$ nesneden biri konur ve böyle devam edilir. Kombinasyon ise sıranın önemsiz olduğu seçimleri sayar. Formülü şöyle anlaşılır: n nesneden k tanesini *sırayla* seçmenin yolu $n \cdot (n-1) \cdot \ldots \cdot (n-k+1) = n!/(n-k)!$'dir. Bu sayımda aynı k nesne $k!$ farklı sırada ayrı ayrı sayılmıştır. Sıra önemsiz olduğunda her grup bir kez sayılmalıdır, bu yüzden $k!$'e bölünür.

Örnek 0.3

1. **Profil sayısı.** Üç oyuncunun sırasıyla 2, 3 ve 4 stratejisi vardır. Bir profil her oyuncudan bir strateji seçilerek kurulur. Çarpım kuralı: $2 \cdot 3 \cdot 4 = 24$ strateji profili.
2. **Sıralama.** Üç kişi bir sıraya dizilecek. Birinci yere 3, ikinci yere kalan 2, üçüncü yere kalan 1 kişi konabilir: $3! = 3 \cdot 2 \cdot 1 = 6$ farklı sıra.

3. **Bekçinin seçimi.** Bir bekçi 4 hedeften 2'sini koruyabiliyor. Hangi iki hedefi koruyacağı onun bir saf stratejisidir. Sıra önemsizdir; önce A sonra B korumak ile önce B sonra A korumak aynı stratejidir. Sayı *C(4, 2) = 4!/(2! · 2!) = 24/4 = 6*'dır: {A, B}, {A, C}, {A, D}, {B, C}, {B, D}, {C, D} (Şekil 0.4).
4. **Büyük bir örnek.** 20 hedeften 5'i korunacaksa *C(20, 5) = (20 · 19 · 18 · 17 · 16)/(5 · 4 · 3 · 2 · 1) = 1.860.480 / 120 = 15.504* saf strateji vardır.



Şekil 0.4: 4 hedeften 2'sini seçmenin *C(4, 2) = 6* yolu. Sıra yoktur: {A, B} ile {B, A} aynı seçimdir.

Son örnekteki büyüme dikkat çekicidir. Hedef sayısı 4'ten 20'ye, korunan hedef sayısı 2'den 5'e çıkınca saf strateji sayısı 6'dan 15.504'e çıkar. Gerçek güvenlik uygulamalarında (havaalanı, liman, orman devriyesi) hedef sayısı yüzlerle ölçülür. Bu yüzden stratejileri tek tek listeleyen yöntemler çabuk yetersiz kalır. Strateji sayısının bu kadar hızlı büyümesi, algoritmik oyun kuramının temel sorularından birini doğurur: sayısı astronomik olan stratejiler arasından en iyisini bilgisayar hızlıca bulabilir mi?

Sık yapılan hata

Sıranın önemsiz olduğu bir seçimde sırayı saymak. 4 hedeften "önce biri, sonra öteki" biçiminde seçim *4 · 3 = 12* sıralı çift verir. Ama {A, B} ile {B, A} aynı korumadır; her seçim iki kez sayılmıştır. Doğru sayı *12 / 2! = 6*'dır. Ters yöndeki hata da görülür: Kartezyen çarpımda sıra önemlidir ve (Taş, Kağıt) ile (Kağıt, Taş) ayrı sayılmalıdır. Hangi kuralın kullanılacağı tek bir soruyla belirlenir: "sırayı değiştirince farklı bir şey mi elde ediliyor?"

0.3 Toplam ve en büyük

Oyun kuramındaki hesapların çoğu iki işlemden oluşur: bir şeyleri toplamak ve en büyüğünü seçmek. Beklenen kazanç bir toplamdır; en iyi tepki bir en büyük değer seçimidir; tekrarlanan oyunlarda toplam kazanç sonsuz bir toplamdır. Seçenekler sonsuz sayıda olduğunda en büyük değer türevle bulunur.

0.3.1 Toplam sembolü Σ

Σ, Yunan alfabesinin büyük “sigma” harfidir ve “hepsini topla” demektir. Uzun toplamları kısa yazmaya yarar. Sembolün altındaki ifade toplamanın nereden başlayacağını, üstündeki sayı nerede duracağını söyler. Sağındaki ifade ise her adımda neyin ekleneceğidir.

Tanım 0.4

$$\Sigma_{i=1}^{n} x_i = x_1 + x_2 + \ldots + x_n$$

Burada i bir *sayaçtır*: önce 1 olur, sonra 2, …, en son n. Sayacın her değeri için x_i yazılır ve hepsi toplanır.

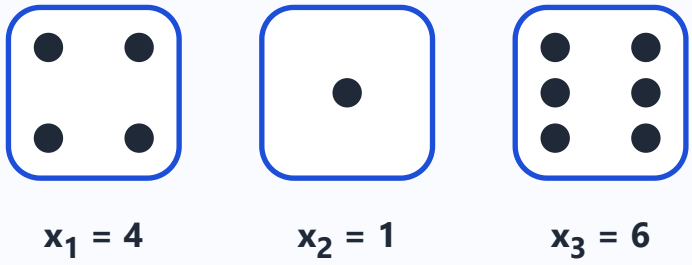
Sayacın adı önemsizdir; i yerine j ya da k yazmak toplamı değiştirmez. Önemli olan sayacın hangi değerlerden geçtiği ve her değer için ne eklendiğidir. Bazı toplamlarda sınırlar yazılmaz ve toplama “bütün olası değerler üzerinden” yapılır. Örneğin $\Sigma_x\, x \cdot P(X = x)$ yazımı, X'in alabileceği her x değeri için bir terim ekler.

Σ ile çalışırken iki kural sık kullanılır. Birincisi, her terimde ortak olan sabit bir çarpan toplamın dışına alınabilir: $\Sigma\, c \cdot x_i = c \cdot \Sigma\, x_i$. İkincisi, iki toplamın toplamı tek toplam olarak yazılabilir: $\Sigma\,(x_i + y_i) = \Sigma\, x_i + \Sigma\, y_i$. İki kural da toplamanın sırasının ve gruplamasının sonucu değiştirmemesinden gelir.

Örnek 0.4

Bir zar üç kez atılmış ve sırasıyla 4, 1 ve 6 gelmiştir: $x_1 = 4$, $x_2 = 1$, $x_3 = 6$ (Şekil 0.5).

1. Toplam açılarak yazılır: $\Sigma_{i=1}^{3} x_i = x_1 + x_2 + x_3$.
2. Değerler yerine konur: $4 + 1 + 6 = 11$.
3. Adil bir zarda her yüzün olasılığı $p_i = 1/6$'dır. Altı yüzün olasılıklarının toplamı $\Sigma_{i=1}^{6} p_i = 6 \cdot 1/6 = 1$'dir. Burada sabit çarpan kuralı kullanılmıştır: altı terim de aynıdır. Olasılıkların toplamının 1 olması her dağılımda geçerlidir.
4. Beklenen değer de bir toplamdır: $\Sigma_{i=1}^{n} x_i \cdot p_i$. Bu toplam 0.5.5'te ayrıntılı işlenir.



Şekil 0.5: Üç zar atışı: $x_1 = 4, x_2 = 1, x_3 = 6$. Toplamları $\Sigma_{i=1}^{3} x_i = 11$.

Σ gösterimi programlamadaki bir döngünün matematik karşılığıdır. $\Sigma_{i=1}^{3} x_i$, "toplam = 0; i = 1'den 3'e kadar her i için toplam'a x[i] ekle" demektir. Bu benzerlik, piyangonun beklenen değerini hesaplayan kodda açıkça görülür (0.5.6).

Sık yapılan hata

Kareyi toplamın dışına taşımak. $(\Sigma\, x_i)^2$ ile $\Sigma\, x_i^2$ farklıdır. İlki önce toplar sonra karesini alır: $(4 + 1 + 6)^2 = 11^2 = 121$. İkincisi önce her terimin karesini alır sonra toplar: $4^2 + 1^2 + 6^2 = 16 + 1 + 36 = 53$. Doğru sıra şudur: önce Σ'nın içindeki ifade her i için ayrı ayrı hesaplanır, sonra sonuçlar toplanır. Sabit çarpan dışarı alınabilir, ama kare ya da başka bir işlem alınamaz.

0.3.2 Geometrik seri ve iskonto faktörü δ

Bazı oyunlar bir kez oynanıp biter; bazıları aynı oyuncular arasında tekrar tekrar oynanır. İki firma her ay fiyat belirler, iki komşu her gün karşılaşır. Tekrarlanan oyunlarda bugünkü kazancı gelecekteki kazançla karşılaştırmak gerekir. İnsanlar genellikle bugünkü bir kazancı gelecekteki aynı kazanca tercih eder; ayrıca oyunun gelecek turda sürüp sürmeyeceği de kesin olmayabilir. Bu iki etki tek bir sayıyla, **iskonto faktörüyle** (*discount factor*) modellenir.

Tanım 0.5

İskonto faktörü $0 \le \delta < 1$ olan bir sayıdır. t tur sonra kazanılacak bir kazanç bugünden bakınca δ^t ile çarpılır: bugünkü 1 kazanç 1, bir tur sonraki δ, iki tur sonraki δ^2 değerindedir.

Her turun kazancı 1 ise bugünden başlayan sonsuz toplam bir **geometrik seridir** (*geometric series*):

$$1 + \delta + \delta^2 + \delta^3 + \ldots = 1/(1 - \delta)$$

Bu formül ilk bakışta şaşırtıcıdır: sonsuz sayıda pozitif sayı toplanıyor ama sonuç sonlu bir sayı çıkıyor. Bunun nedeni terimlerin hızla küçülmesidir. $\delta = 1/2$ için terimler 1, 1/2, 1/4, 1/8, … olarak her adımda yarıya iner (Şekil 0.6). Toplam her adımda 2'ye biraz daha yaklaşır ama onu hiç aşmaz.

Teorem 0.1 (Geometrik seri)

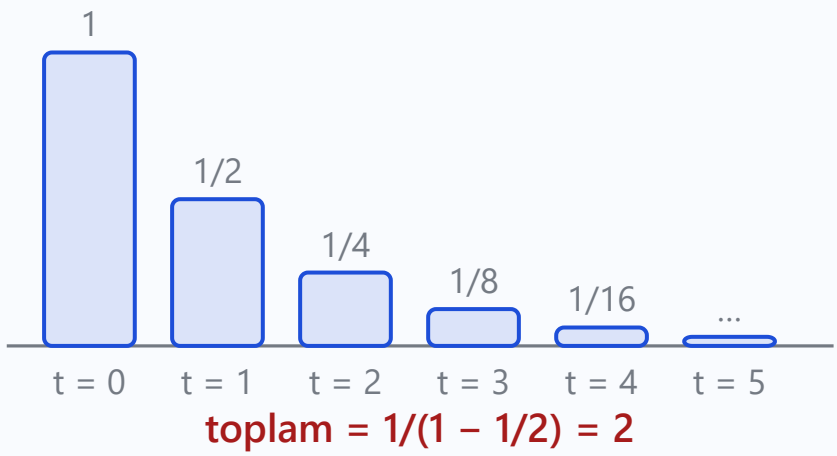
$0 \le \delta < 1$ ise $S = \Sigma_{t=0}^{\infty} \delta^t = 1/(1-\delta)$'dir.

İspat. Toplam S ile gösterilsin: $S = 1 + \delta + \delta^2 + \delta^3 + \dots$. İki tarafı δ ile çarpınca her terimin üssü bir artar: $\delta S = \delta + \delta^2 + \delta^3 + \dots$. İki satır alt alta yazılınca, ikinci satırın bütün terimlerinin birinci satırda da bulunduğu görülür; birinci satırın fazladan tek terimi baştaki 1'dir. Bu yüzden birinci satırdan ikincisi çıkarılınca yalnız o terim kalır: $S - \delta S = 1$. Sol taraf $S(1-\delta)$ biçiminde yazılır. $\delta < 1$ olduğu için $1 - \delta$ sıfır değildir ve iki taraf ona bölünebilir: $S = 1/(1-\delta)$. (Bu türetme, S'nin sonlu bir sayı olduğunu varsayar; $\delta < 1$ iken bu varsayım doğrudur, çünkü terimler hızla küçülür.) □

Her turun kazancı 1 değil de a ise sabit çarpan kuralıyla toplam $a/(1-\delta)$ olur. Toplam bugünden değil de bir tur sonradan başlıyorsa her terim fazladan bir δ taşır: $\delta + \delta^2 + \dots = \delta \cdot (1 + \delta + \dots) = \delta/(1-\delta)$.

Örnek 0.5

1. **$\delta = 0{,}9$ ve her tur 3 kazanç.** Toplam $3 + 3 \cdot 0{,}9 + 3 \cdot 0{,}9^2 + \dots = 3/(1-0{,}9) = 3/0{,}1 = 30$'dur.
2. **$\delta = 1/2$ ve her tur 1.** $1 + 1/2 + 1/4 + \dots = 1/(1-1/2) = 1/(1/2) = 2$. İlk birkaç kısmi toplam 1; 1,5; 1,75; 1,875 olur ve 2'ye yaklaşır.
3. **Yarından başlayan toplam.** $\delta = 0{,}9$ ve her tur 3 kazanç bugün değil de yarından itibaren geliyorsa $3 \cdot 0{,}9/(1-0{,}9) = 2{,}7/0{,}1 = 27$ olur. Fark, bugünkü 3'tür: $30 - 27 = 3$.



Şekil 0.6: $\delta = 1/2$ için turların bugünkü değerleri: 1, 1/2, 1/4, 1/8, 1/16, … Çubuklar her adımda yarıya iner; toplam $1/(1-1/2) = 2$'dir.

Bu formülün oyun kuramındaki önemi şudur: tekrarlanan bir oyunda iki oyuncu sürekli işbirliği yaparsa her tur belli bir kazanç alırlar; biri bir kez hile yaparsa o tur daha fazla kazanır ama sonraki turlarda cezalandırılabilir. Hilenin kârlı olup olmadığı, bir kerelik kazancın ileride kaybedilecek kazançların bugünkü değeriyle karşılaştırılmasıyla anlaşılır. O bugünkü değer tam olarak bir geometrik seridir. δ büyükse (oyuncular sabırlıysa ya da oyunun sürmesi olasıysa) gelecek ağır basar ve işbirliği sürdürülebilir.

Sık yapılan hata

İlk terimi unutmak. "Bugünden sonra" başlayan toplam $\delta + \delta^2 + \ldots = \delta/(1-\delta)$'dır, $1/(1-\delta)$ değil. Hangi formülün kullanılacağı, ilk kazancın hangi turda geldiğine bakarak seçilir. İkinci hata $\delta = 1$ için formülü kullanmaktır: bu durumda payda sıfır olur ve formül geçersizdir. Gerçekten de $1 + 1 + 1 + \ldots$ toplamı sınırsız büyür.

0.3.3 max ve argmax

Bir oyuncu karar verirken seçeneklerinin her birinin ne getireceğine bakar ve en çok getireni seçer. Bu düşüncede iki ayrı soru vardır: "en çok ne kadar kazanılabilir?" ve "en çoğu hangi seçenek getirir?" İlkinin cevabı bir sayıdır, ikincisinin cevabı bir seçenektir. İki soru iki ayrı gösterimle yazılır.

Tanım 0.6

Seçenekler kümesi A ve her seçeneğin değeri $f(a)$ olsun.

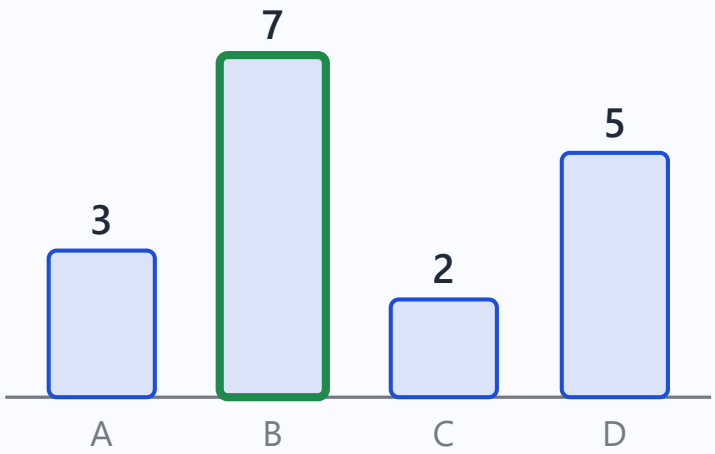
- $\max_{a \in A} f(a)$: değerlerin en büyüğüdür. Bir *sayıdır*.
- $\operatorname{argmax}_{a \in A} f(a)$: en büyük değeri veren seçeneklerin *kümesidir*. Eşitlik yoksa tek elemanlıdır ve çoğu zaman o tek eleman yazılır.

Oyun kuramında oyuncu bir değer değil, bir eylem seçer. Bu yüzden argmax daha sık kullanılır. Bir oyuncunun "en iyi tepkisi", rakibin seçimine karşı kendi kazancının argmax'ıdır (0.8). Max ise o en iyi tepkinin getirdiği kazancı söyler.

Örnek 0.6

Dört kapalı karttan biri seçilecek. Kartların ödülleri A: 3, B: 7, C: 2, D: 5 jetondur.

1. Değerler yan yana yazılır: $f(A) = 3, f(B) = 7, f(C) = 2, f(D) = 5$.
2. En büyük değer bulunur: $\max\{3, 7, 2, 5\} = 7$. Bu bir sayıdır.
3. Bu değeri veren seçenek bulunur: yalnız B 7 verir. $\operatorname{argmax} = B$. Bu bir seçenektir ve kararın kendisidir: "B'yi seç".



Şekil 0.7: Dört kartın ödülleri. En yüksek çubuğun yüksekliği max = 7, onun etiketi argmax = B'dir (yeşil çerçeve).

Şekil 0.7 iki kavramı görsel olarak ayırır: max çubuğun *yüksekliğidir*, argmax çubuğun *altındaki etikettir*. Bu ayrım ilk bakışta önemsiz görünür; oysa oyun kuramının pek çok tanımında max ve argmax aynı satırda birlikte geçer. Birini ötekinin yerine yazmak, tanımı anlamsız kılar.

Sık yapılan hata

argmax yerine değeri yazmak (“argmax = 7”) ya da tersi (“max = B”). İkinci tuzak eşitliktir. Ödüller 3, 7, 2, 7 olsaydı argmax tek bir eleman değil, *{B, D}* kümesi olurdu; ikisi de en iyi seçimdir. Eşitlikte birini “rastgele” seçip ötekini unutmak, oyun kuramında bazı dengelerin gözden kaçmasına yol açar. Çünkü bir oyuncunun iki seçenek arasında kayıtsız kalması, karışık stratejilerin temelidir (0.9.3).

0.3.4 Kod izleme: döngüyle argmax bulma

Bilgisayar argmax'ı bir döngüyle bulur. Aşağıdaki Python kodu, sunumdaki kodun aynısıdır. Ödüller bu kez bilerek eşitlikli seçildi: 3, 7, 2, 7. B ile D eşit en büyüktür; böylece kodun eşitlikte nasıl davrandığı görülür.

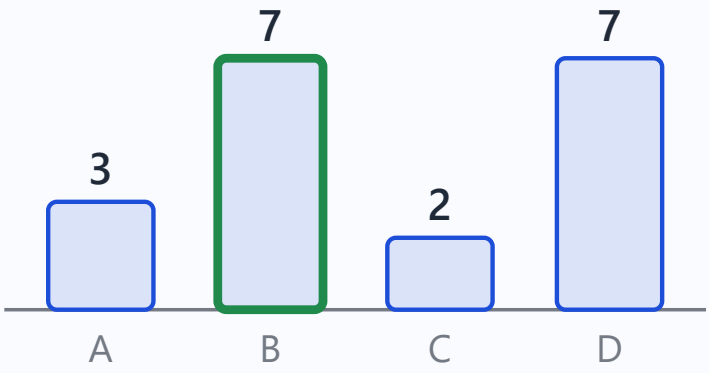
```
eylem  = ["A", "B", "C", "D"]
kazanc = [3, 7, 2, 7]
en_iyi = 0            # en iyinin indisi
for i in range(1, 4):
    if kazanc[i] > kazanc[en_iyi]:
        en_iyi = i
print(kazanc[en_iyi], eylem[en_iyi])
```

Kodun satırları şöyle çalışır:

- `eylem` seçeneklerin adlarını, `kazanc` aynı sıradaki ödülleri tutar. İki liste aynı indisle eşleşir: indis 0 A'ya ve 3'e, indis 1 B'ye ve 7'ye karşılık gelir.
- `en_iyi = 0`: başlangıçta ilk kart (A, 3 jeton) en iyi kabul edilir. Değişken bir *değer* değil, bir *indis* tutar; bu, argmax'ın bir seçenek olmasının koddaki karşılığıdır.
- Döngü `i = 1, 2, 3` için çalışır. `range(1, 4)` 1'den başlar ve 4'ü içermez. 0 atlanır, çünkü ilk kart zaten başlangıç adayıdır.
- Her adımda bakılan kartın ödülü, şimdiye kadarki en iyinin ödülüyle karşılaştırılır. Kesin olarak büyükse en iyi güncellenir.
- Sonunda en iyinin ödülü (max) ve adı (argmax) yazdırılır.

Adım	i	Karşılaştırma	Sonuç	en_iyi
Başlangıç	–	–	A aday	0 (A, 3)
1	1	*7 > 3*?	evet	1 (B, 7)
2	2	*2 > 7*?	hayır	1 (B, 7)
3	3	*7 > 7*?	hayır	1 (B, 7)
Çıktı	–	–	`7 B`	–

Tablo 0.1: Döngüyle argmax izlemesi (ödüller 3, 7, 2, 7).



Şekil 0.8: İzlemenin son durumu. Yeşil çerçeve şimdiye kadarki en iyi kart B'dir. D de 7 verdiği hâlde kod yalnız B'yi döndürür; gerçek argmax kümesi {B, D}'dir.

Tablo 0.1 döngünün her adımını, Şekil 0.8 son durumu gösterir. İzleme tablosunun üçüncü adımında önemli bir ayrıntı vardır. Üçüncü adımda D'nin ödülü 7, en iyinin ödülü de 7'dir. Karşılaştırma kesin "büyüktür" (`>`) olduğu için *7 > 7* yanlıştır ve en iyi değişmez. Kod eşitlikte *ilk bulunanı* tutar. Çıktı `7 B` olur. Oysa matematik açısından argmax kümesi *{B, D}*'dir. Kod, kümenin yalnız bir elemanını vermiştir.

Karşılaştırma `>=` olsaydı son eşit eleman tutulurdu ve çıktı `7 D` olurdu. Bütün en iyileri bulmak için iki geçiş yapılır: önce max bulunur, sonra ödülü max'a eşit olan bütün indisler bir listeye toplanır. Oyun kuramı kodlarında bu ayrıntı önemlidir. Örneğin bir matristeki bütün dengeleri bulan bir program, eşit en büyükleri atlarsa bazı dengeleri kaçırır.

Kodun iş miktarı da kolayca sayılır. n seçenek için döngü $n - 1$ kez çalışır ve her turda bir karşılaştırma yapar. İş, seçenek sayısıyla doğru orantılı büyür; bu, 0.10.3'teki gösterimle $\Theta(n)$'dir. Her seçeneğe en az bir kez bakmadan en büyüğü bulmak mümkün olmadığı için daha hızlı bir yöntem yoktur.

0.3.5 Fonksiyonun en büyüğü: türev sıfır

Seçenekler birkaç kartla sınırlı olduğunda argmax bir tablodan okunur. Ama bazı seçimler süreklidir. Bir firma 4 birim de üretebilir, 4,5 birim de, 4,37 birim de. Seçenek sayısı sonsuz olduğu için tablo yazılamaz. Bu durumda en büyük değer, fonksiyonun grafiğinin şekli kullanılarak bulunur.

Bir dağın tepesini düşünelim. Tepeye çıkarken yol yukarı eğimlidir; tepeden inerken aşağı eğimlidir. Tam tepede yol bir an için düzdür: eğim sıfırdır. Aynı şey düzgün bir kazanç eğrisi için de geçerlidir: tepenin solunda eğim artı, sağında eksidir; tepede sıfırdır. Bir fonksiyonun her noktadaki eğimini veren fonksiyona **türev** (*derivative*) denir ve $f'(q)$ ile gösterilir.

Tanım 0.7

$f'(q)$, f'nin grafiğinin q noktasındaki eğimidir: q çok az artınca f'nin ne kadar hızla değiştiğini söyler. Bu notta gereken kurallar şunlardır:

- $(q^k)' = k \cdot q^{k-1}$; örneğin $(q^2)' = 2q$ ve $(q)' = 1$.
- Sabitin türevi 0'dır.
- $(c \cdot f)' = c \cdot f'$: sabit çarpan türevin dışına çıkar.
- Toplamın türevi türevlerin toplamıdır: $(f + g)' = f' + g'$.

Türevin türevine **ikinci türev** denir ve f'' ile gösterilir. $f'' < 0$ ise eğim azalıyordur; eğimin sıfır olduğu nokta bir tepedir.

Yöntem üç adımdan oluşur: türev alınır, sıfıra eşitlenir ve çıkan denklem çözülür. Bulunan noktaya aday denir, çünkü eğimin sıfır olduğu her nokta tepe değildir. Bir çukurun dibinde de eğim sıfırdır. Bu yüzden dördüncü bir adım gerekir: adayın gerçekten tepe olduğu denetlenir.

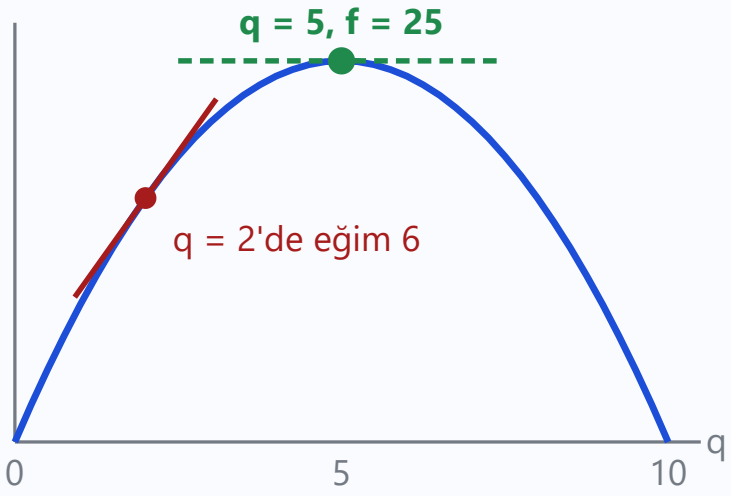
Örnek 0.7

$f(q) = q(10 - q)$ fonksiyonunun en büyük değeri aranıyor.

1. **Açılım.** Çarpım açılır: $f(q) = 10q - q^2$.
2. **Türev.** $10q$'nun türevi 10, q^2'nin türevi $2q$'dur. Toplam kuralıyla $f'(q) = 10 - 2q$.

3. **Sıfıra eşitleme.** $10 - 2q = 0 \Rightarrow 2q = 10 \Rightarrow q = 5$.
4. **Değer.** $f(5) = 5 \cdot (10 - 5) = 25$.
5. **Denetim.** İkinci türev $f''(q) = -2$'dir ve negatiftir; eğim azalıyordur, nokta bir tepedir. Komşu değerler de bunu doğrular: $f(4) = 4 \cdot 6 = 24$ ve $f(6) = 6 \cdot 4 = 24$; ikisi de 25'ten küçüktür.

Sonuç: *argmax* $= 5$, *max* $= 25$.



Şekil 0.9: $f(q) = q(10 - q)$. Tepede $(q = 5, f = 25)$ teğet yataydır, eğim sıfırdır. $q = 2$'de eğim $f'(2) = 10 - 4 = 6$'dır (kırmızı teğet).

Şekil 0.9 türevin geometrik anlamını gösterir. $q = 2$'deki kırmızı doğru eğriye bu noktada teğettir ve eğimi 6'dır: q küçük bir miktar artınca f yaklaşık 6 katı kadar artar. Tepede teğet yataydır. Tepenin sağında, örneğin $q = 8$'de, eğim $10 - 16 = -6$'dır: q arttıkça f azalır.

Sık yapılan hata

$f' = 0$ noktasını hiç denetlememek. Bu nokta bir çukur da olabilir; örneğin $g(q) = q^2$ için $g'(0) = 0$'dır ama 0 noktası bir çukurun dibidir. Tepe olduğunu $f'' < 0$ ya da komşu değerler gösterir ($f(4) = f(6) = 24 < 25$). Seçim bir aralıkla sınırlıysa uç noktalara da bakılır: en büyük değer bazen eğimin sıfır olduğu yerde değil, izin verilen aralığın ucundadır.

0.3.6 Türevle en iyi miktar

Türev yönteminin oyun kuramındaki ilk büyük uygulaması üretim miktarı seçen firmalardır. İki firma aynı malı satar. Piyasa fiyatı toplam üretime bağlıdır: ne kadar çok üretilirse fiyat o kadar düşer. Bir firmanın kârı yalnız kendi miktarına değil, rakibin miktarına da bağlıdır. Bu, bir oyundur.

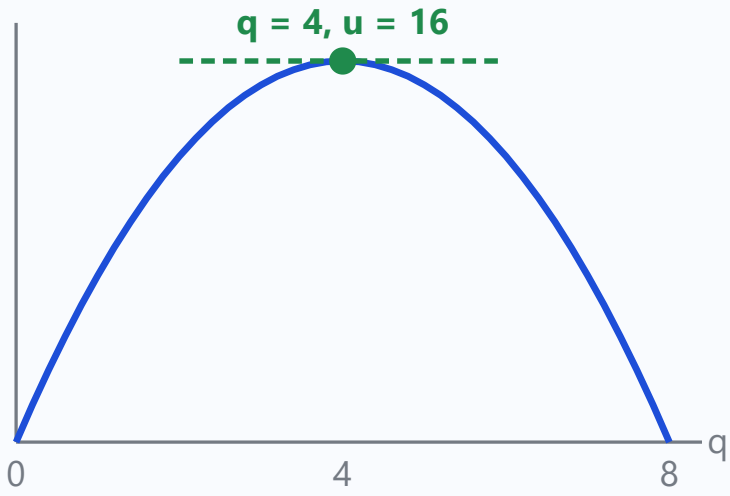
Modelin sayıları şunlardır: fiyat $12 - (q + r)$'dir; burada q firmanın, r rakibin üretim miktarıdır. Üretim maliyeti yoktur. Kâr, miktar çarpı fiyattır:

$$u(q) = q \cdot (12 - q - r)$$

Örnek 0.8

Rakip $r = 4$ birim üretiyor. Firmanın en iyi miktarı aranıyor.

1. **Kâr ifadesi.** $u(q) = q \cdot (12 - q - 4) = q(8 - q) = 8q - q^2$.
2. **Türev.** $u'(q) = 8 - 2q$.
3. **Sıfıra eşitleme.** $8 - 2q = 0 \Rightarrow q = 4$.
4. **Kâr.** $u(4) = 4 \cdot (8 - 4) = 16$. Fiyat $12 - 8 = 4$'tür; 4 birim 4'ten satılır. Kâr eğrisi ve tepesi Şekil 0.10'da görülür.
5. **Denetim.** $u'' = -2 < 0$. Komşular: $u(3) = 3 \cdot 5 = 15$ ve $u(5) = 5 \cdot 3 = 15$; ikisi de 16'dan küçüktür ✓.



Şekil 0.10: Rakip $r = 4$ üretirken firmanın kârı $u(q) = q(8 - q)$. Tepe $q = 4$'te, kâr 16'dır; $q = 8$'de fiyat sıfıra düşer.

Aynı hesap rakibin her miktarı için tekrarlanabilir. Bunun yerine r'yi bir sabit gibi tutarak hesap bir kez ve genel olarak yapılır. Kâr açılınca $u(q) = 12q - q^2 - qr$ olur. r sabit olduğu için qr'nin q'ya göre türevi r'dir (sabit çarpan kuralı). Türev $u'(q) = 12 - 2q - r$'dir ve sıfıra eşitlenince

$$12 - 2q - r = 0 \Rightarrow q = (12 - r)/2$$

bulunur. $r = 4$ konunca $q = 8/2 = 4$ çıkar; bu, örnekteki sonuçtur. $r = 2$ için $q = 5$, $r = 6$ için $q = 3$'tür: rakip ne kadar çok üretirse firma o kadar az üretir.

Formülün bir sınırı vardır: miktar negatif olamaz. $r \geq 12$ ise formül sıfır ya da eksi bir sayı verir. Bu durumda fiyat zaten sıfır ya da altındadır ve firmanın en iyisi hiç üretmemektir. Tam formül şöyledir:

$$q^*(r) = \max\{0, (12 - r)/2\}$$

Rakibin her miktarına en iyi cevabı veren bu formül, 0.8'de matriste işaretlenen **en iyi tepkinin** sürekli hâlidir. Matriste en iyi tepki her sütun için ayrı ayrı argmax alınarak bulunur; burada türev, sonsuz sayıdaki "sütunun" hepsi için argmax'ı tek bir formülde verir. İki firmanın birbirine aynı anda en iyi tepki verdiği miktarlar, 0.4.4'te kısmi türevle bulunur.

Sık yapılan hata

Rakibin miktarının kendi miktarıyla birlikte değiştiğini sanmak. Firma q'yu seçerken r'yi değiştiremez; r onun için bir sabittir. Bu yüzden qr teriminin türevi r'dir, $q + r$ ya da $2qr$ değildir. İkinci hata sınırı unutmaktır: $r = 14$ için formül $q = -1$ verir, ama negatif üretim anlamsızdır ve doğru cevap $q^* = 0$'dır.

0.4 Üs, logaritma ve türev kuralları

Oyun kuramında üslü ifadeler üç yerde karşımıza çıkar: tekrarlanan oyunlarda iskonto (δ^t), riskten kaçınan bir kişinin faydası ($x^{0,88}$ ya da $\ln x$) ve olasılıklı seçim kuralları ($e^{\lambda v}$). Bu ifadelerle çalışmak için üs ve logaritma kurallarına, bu ifadelerin en büyüğünü bulmak için de iki yeni türev kuralına gerek vardır: çarpım kuralı ve zincir kuralı. Birden çok değişkenli fonksiyonlarda ise kısmi türev kullanılır.

0.4.1 Üs kuralları ve kesirli üs

x^a ifadesinde x'e *taban*, a'ya *üs* denir. a pozitif bir tam sayıysa x^a, x'in a kez kendisiyle çarpımıdır: $x^3 = x \cdot x \cdot x$. Üs kuralları bu tanımdan doğrudan çıkar ve sonra kesirli ve negatif üslere aynı biçimde genişletilir.

Tanım 0.8

$x > 0$ ve a, b herhangi gerçek sayılar için:

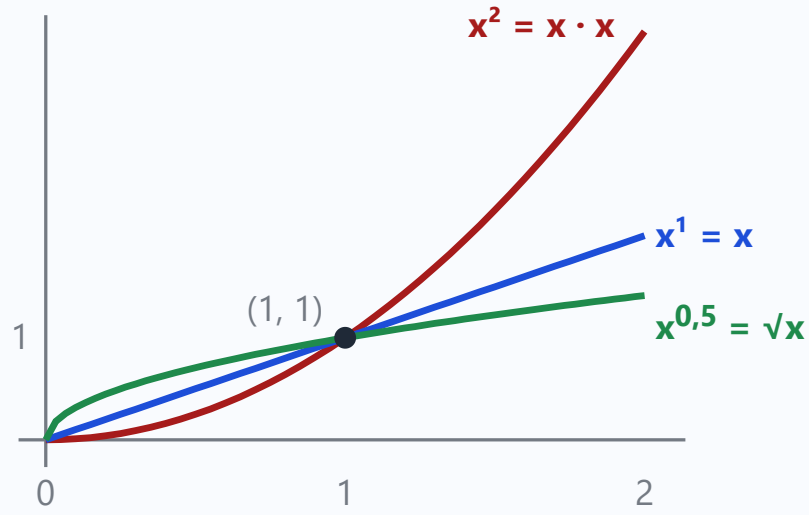
- $x^a \cdot x^b = x^{a+b}$ (aynı tabanlı çarpımda üsler toplanır),
- $(x^a)^b = x^{a \cdot b}$ (üssün üssünde üsler çarpılır),
- $x^0 = 1$,
- $x^{-a} = 1/x^a$,
- $x^{1/2} = \sqrt{x}$.

Kuralların nedeni sayarak görülür. $x^2 \cdot x^3$, iki tane x ile üç tane x'in çarpımıdır; toplam beş tane x çarpılır, yani x^5. $x^0 = 1$ kuralı da birinci kuraldan gelir: $x^a \cdot x^0 = x^{a+0} = x^a$ olmalıdır, bu yüzden x^0 çarpımı değiştirmeyen sayı, yani 1 olmalıdır. Negatif üs aynı mantıkla gelir: $x^a \cdot x^{-a} = x^0 = 1$ olduğuna göre x^{-a}, x^a'nın tersidir. Son olarak $x^{1/2} \cdot x^{1/2} = x^1 = x$'tir; kendisiyle çarpımı x veren pozitif sayı $\sqrt{x}$'tir.

Kesirli üsler ($x^{0,88}$ gibi) aynı kurallara uyar. Bunların anlamı tek bir cümleyle söylenebilir: üs 1'den küçük ve pozitifse x^a, x büyüdükçe gittikçe yavaşlayarak artar. Üs 1'den büyükse gittikçe hızlanarak artar. Şekil 0.11 bu iki durumu, aradaki üs 1 hâliyle birlikte yan yana gösterir.

Örnek 0.9

1. $\delta^2 \cdot \delta^3 = \delta^{2+3} = \delta^5$. İki tur sonraki iskonto ile üç tur daha sonraki iskontonun birleşimi beş turluk iskontodur.
2. $2^{-1} = 1/2$; $9^{1/2} = \sqrt{9} = 3$; $5^0 = 1$.
3. **Fayda.** Bir kişinin x lira paradan aldığı fayda $x^{0,88}$ ile ölçülsün. Hesap makinesiyle $100^{0,88} \approx 57,5$ ve $200^{0,88} \approx 105,9$ bulunur.
4. Para iki katına çıkınca fayda kaç katına çıkar? Üs kuralıyla $(2x)^{0,88} = 2^{0,88} \cdot x^{0,88}$'dir. Oran $2^{0,88} \approx 1,84$'tür: $105,9 / 57,5 \approx 1,84$. Fayda iki katına değil, yaklaşık 1,84 katına çıkar.



Şekil 0.11: x^2 (kırmızı), x^1 (mavi) ve $x^{0,5} = \sqrt{x}$ (yeşil). Üçü de (1, 1)'den geçer. Üs 1'den küçükse eğri yavaşlayarak, büyükse hızlanarak artar.

Son örneğin anlamı karar kuramında önemlidir. Üssü 1'den küçük bir fayda fonksiyonu, "ikinci yüz lira ilk yüz lira kadar sevindirmez" düşüncesini sayıyla ifade eder. Böyle bir kişi, kesin 100 lira ile yarı yarıya 0 ya da 200 lira arasında kesin olanı seçer: kesin seçenek *57,5* fayda verir, piyango ise ortalama *(0 + 105,9)/2 ≈ 53* fayda verir. Bu davranışa riskten kaçınma denir.

Sık yapılan hata

Üsleri toplamda da toplamak: $2^2 + 2^3 = 4 + 8 = 12$'dir, ama $2^5 = 32$'dir. Üsler yalnız *aynı tabanlı çarpımda* toplanır; toplamda hiçbir üs kuralı yoktur. İkinci yaygın hata $x^0 = 0$ sanmaktır. Doğrusu $x^0 = 1$'dir ($x \neq 0$); bu yüzden bugünkü kazancın iskontosu $\delta^0 = 1$'dir ve bugünkü kazanç olduğu gibi sayılır.

0.4.2 e sayısı ve doğal logaritma ln

Bazı tabanlar ötekilerden daha kullanışlıdır. Bunların en önemlisi $e \approx 2{,}718$ sayısıdır. e'nin özel olmasının nedeni türevle ilgilidir: e^x fonksiyonunun her noktadaki eğimi kendi değerine eşittir. Bu özellik hesapları çok sadeleştirir. Oyun kuramında e, olasılıklı seçim kurallarında (0.6.3) ve öğrenme algoritmalarında karşımıza çıkar.

Tanım 0.9

e^x her x için pozitiftir. **Doğal logaritma** $\ln$, e^x'in ters işlemidir: $\ln(e^x) = x$ ve $y > 0$ için $e^{\ln y} = y$. Yani $\ln y$, "e'nin hangi kuvveti y'yi verir?" sorusunun cevabıdır. Kurallar:

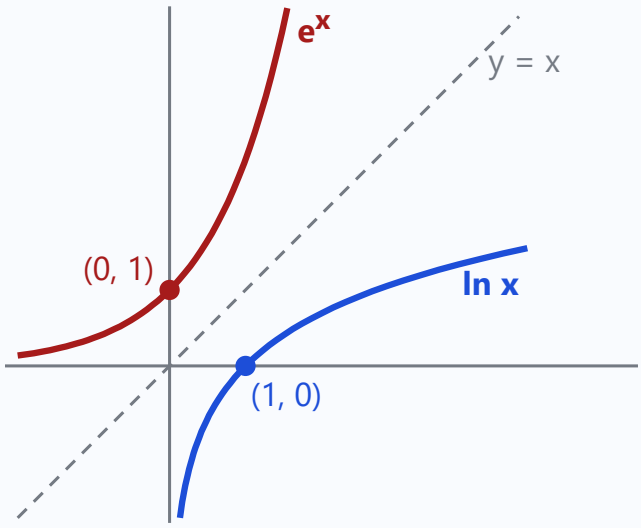
- $\ln(a \cdot b) = \ln a + \ln b$,
- $\ln(a^k) = k \cdot \ln a$,
- $\ln 1 = 0$.

Türevler: $(e^x)' = e^x$ ve $(\ln x)' = 1/x$.

Logaritma kuralları üs kurallarının tersten okunmasıdır. $a = e^m$ ve $b = e^n$ yazılırsa $a \cdot b = e^{m+n}$'dir; iki tarafın ln'i alınınca $\ln(a \cdot b) = m + n = \ln a + \ln b$ çıkar. Logaritma çarpımı toplama çevirir. Bu özellik, çok sayıda olasılığın çarpımıyla çalışan algoritmaların neden logaritma kullandığını da açıklar: küçük sayıların çarpımı yerine logaritmalarının toplamıyla çalışmak hem kolay hem de sayısal olarak güvenlidir.

Örnek 0.10

1. **$\ln 8$.** $8 = 2^3$'tür. Kural $\ln(a^k) = k \cdot \ln a$ ile $\ln 8 = 3 \cdot \ln 2 \approx 3 \cdot 0{,}693 = 2{,}079$. Geri dönüş denetimi: $e^{2{,}079} \approx 8$.
2. **Logaritmik fayda.** Fayda $u(x) = \ln x$ olsun. Servet 10'dan 20'ye çıkınca fayda artışı $\ln 20 - \ln 10 = \ln(20/10) = \ln 2 \approx 0{,}69$'dur. Servet 20'den 40'a çıkınca artış $\ln 40 - \ln 20 = \ln 2 \approx 0{,}69$'dur. Her iki katına çıkış aynı fayda artışını verir; ilkinde 10 lira, ikincisinde 20 lira eklenmiştir.
3. **Marjinal fayda.** $u'(x) = 1/x$: servet 10 iken bir liranın ek faydası yaklaşık $0{,}1$, servet 100 iken yaklaşık $0{,}01$'dir. Zenginleştikçe ek liranın faydası azalır.



Şekil 0.12: e^x (kırmızı) ve $\ln x$ (mavi), $y = x$ doğrusuna göre birbirinin aynadaki görüntüsüdür. e^x (0, 1)'den, $\ln x$ (1, 0)'dan geçer.

Şekil 0.12 ters fonksiyon ilişkisini gösterir. Bir fonksiyon ile tersi, $y = x$ doğrusuna göre simetriktir: $e^0 = 1$ olduğu için e^x eğrisi (0, 1) noktasından, $\ln 1 = 0$ olduğu için $\ln x$ eğrisi (1, 0) noktasından geçer. e^x eğrisi her zaman x ekseninin üstündedir; $\ln x$ ise yalnız pozitif x için tanımlıdır.

Sık yapılan hata

$\ln(a + b) = \ln a + \ln b$ sanmak. Sayıyla denetlenince yanlışlık görülür: $\ln(2 + 6) = \ln 8 \approx 2{,}08$'dir, ama $\ln 2 + \ln 6 = \ln 12 \approx 2{,}48$'dir. Çarpımın logaritması toplanır, toplamınki değil. Ayrıca $\ln 0$ ve negatif sayıların ln'i tanımsızdır: e^x hiçbir zaman sıfır ya da negatif olmaz, dolayısıyla "e'nin hangi kuvveti 0'ı verir?" sorusunun cevabı yoktur.

0.4.3 Çarpımın türevi ve zincir kuralı

0.3.5'teki kurallar, çarpımı önce açıp terimlerin toplamı olarak yazabildiğimiz fonksiyonlara yeter. Ama her fonksiyon böyle açılmaz. $\ln(1 + 2x)$ ya da e^{-x} gibi iç içe fonksiyonlarda ve uzun çarpımlarda iki kural daha gerekir.

Tanım 0.10

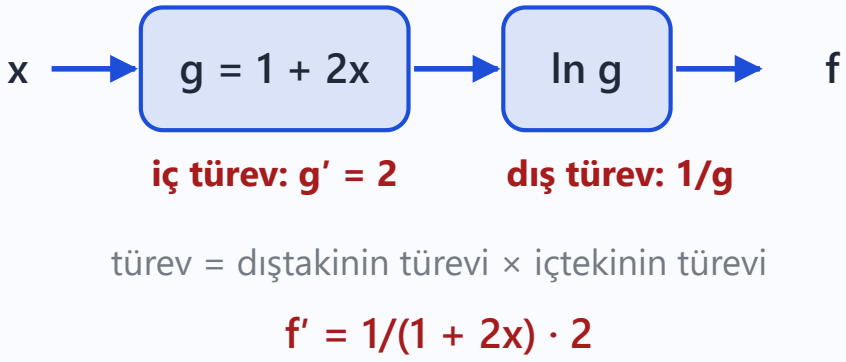
- **Çarpım kuralı:** $(f \cdot g)' = f' \cdot g + f \cdot g'$.
- **Zincir kuralı** (*chain rule*): $[f(g(x))]' = f'(g(x)) \cdot g'(x)$. Önce dıştaki fonksiyonun türevi alınır ve içteki olduğu gibi bırakılır; sonra içtekinin türeviyle çarpılır.

Çarpım kuralı şöyle düşünülebilir: $f \cdot g$ bir dikdörtgenin alanıdır; kenarlardan biri f, öteki g'dir. x biraz artınca iki kenar da biraz uzar. Alanın artışı iki şeritten gelir: f kenarının uzamasıyla eklenen şerit ($f' \cdot g$) ve g kenarının uzamasıyla eklenen şerit ($f \cdot g'$). Köşedeki küçücük kare iki küçük artışın çarpımıdır ve ihmal edilir.

Zincir kuralı ise bir hız aktarımıdır. İçteki fonksiyon x ile g' hızında değişir; dıştaki fonksiyon da kendi girdisiyle f' hızında değişir. x değişince sonuç bu iki hızın çarpımıyla değişir. Şekil 0.13 bu aktarımı bir boru hattı olarak çizer.

Örnek 0.11

1. **Çarpım kuralı.** Kâr $u(q) = q \cdot (12 - q - r)$'de $f = q$ ve $g = 12 - q - r$'dir. $f' = 1$, $g' = -1$ (r sabittir). Kural: $u' = 1 \cdot (12 - q - r) + q \cdot (-1) = 12 - 2q - r$. Çarpımı açıp türev almak (0.3.6) aynı sonucu verir.
2. **Zincir, logaritma.** $[\ln(1 + 2x)]'$: dış fonksiyon $\ln$, iç fonksiyon $g = 1 + 2x$. Dış türev $1/g = 1/(1 + 2x)$, iç türev $g' = 2$. Sonuç $2/(1 + 2x)$'tir.
3. **Zincir, üstel.** $(e^{-x})'$: dış türev e^{-x} (üstel fonksiyon kendi türevidir), iç türev $(-x)' = -1$. Sonuç $-e^{-x}$'tir.
4. **Zincir, kare.** $[(1 - p)^2]'$: dış fonksiyon kare almadır, türevi $2(1 - p)$; iç türev $(1 - p)' = -1$. Sonuç $2(1 - p) \cdot (-1) = -2(1 - p)$'dir. Denetim: $(1 - p)^2 = 1 - 2p + p^2$ açılırsa türev $-2 + 2p = -2(1 - p)$ çıkar ✓.



Şekil 0.13: Zincir kuralı bir boru hattıdır: x önce $g = 1 + 2x$'e (iç türev 2), sonra $\ln g$'ye (dış türev $1/g$) girer. Türev iki hızın çarpımıdır: $f' = 1/(1 + 2x) \cdot 2$.

Zincir kuralının denetlenmesi kolaydır: sonuç, fonksiyonu açıp bilinen kurallarla türev alarak bulunan sonuçla karşılaştırılır. Son örnekte bu denetim yapılmıştır. Açılamayan fonksiyonlarda (logaritma, üstel) ise sayısal bir denetim yapılabilir: $x = 0$'da $\ln(1 + 2x)$'in türevi formüle göre 2'dir; x 0'dan 0,001'e çıkınca fonksiyon $\ln 1{,}002 \approx 0{,}002$ kadar artar, yani artış hızı yaklaşık 2'dir ✓.

Sık yapılan hata

İçtekinin türevini unutmak: $(e^{-x})' = e^{-x}$ yazmak. Doğrusu $-e^{-x}$'tir; eksi işareti iç türevden gelir. Çarpımda da sık bir hata vardır: $(f \cdot g)' = f' \cdot g'$ yazmak yanlıştır. Örneğin $(q \cdot q)' = (q^2)' = 2q$'dur, ama $f' \cdot g' = 1 \cdot 1 = 1$ verir.

0.4.4 Kısmi türev

Bir oyunda kazanç yalnız bir oyuncunun seçimine değil, herkesin seçimine bağlıdır. Firmanın kârı $u(q, r)$ hem kendi miktarı q'ya hem rakibin miktarı r'ye bağlıdır. Böyle iki değişkenli bir fonksiyonda "türev" sorusu belirsizdir: hangi değişken değişiyor? Oyunda her oyuncu yalnız kendi değişkenini seçebildiği için doğru soru şudur: "öteki değişken yerinde dururken, benim değişkenim değişince kazancım nasıl değişir?"

Tanım 0.11

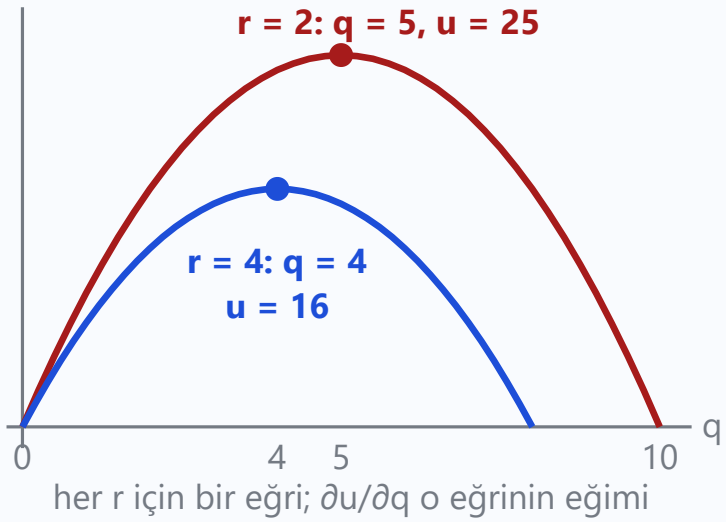
$u(q, r)$'nin q'ya göre **kısmi türevi** (*partial derivative*) $\partial u/\partial q$, r sabit bir sayı gibi tutularak yalnız q'ya göre alınan türevdir. $\partial u/\partial r$ de aynı biçimde q sabit tutularak alınır. ∂ sembolü "kısmi" diye okunur.

Kısmi türev yeni bir kural gerektirmez. $\partial u/\partial q$ alınırken r'nin bulunduğu her yere 4, 7 ya da başka bir sabit yazılmış gibi davranılır; bütün bildiğimiz kurallar aynen uygulanır. Oyun kuramında her oyuncu kendi değişkenine göre kısmi türevi sıfıra eşitler. Böylece her oyuncu için bir denklem, yani en iyi tepki formülü elde edilir. Bu denklemler birlikte çözülünce herkesin aynı anda en iyi tepki verdiği nokta bulunur.

Örnek 0.12

İki firma: birincinin kârı $u_1(q, r) = q(12 - q - r)$, ikincininki $u_2(q, r) = r(12 - q - r)$.

1. Birinci kâr açılır: $u_1 = 12q - q^2 - qr$.
2. r sabit tutulur: $\partial u_1/\partial q = 12 - 2q - r$. ($12q$'dan 12, q^2'den $2q$, qr'den r gelir.)
3. Karşılaştırma için öteki kısmi türev de alınabilir: q sabitken $\partial u_1/\partial r = -q$. Rakibin bir birim fazla üretmesi, fiyatı bir birim düşürür ve birinci firmanın kârını q kadar azaltır.
4. İkinci firma kendi değişkeni r'yi seçer: $u_2 = 12r - qr - r^2$ ve $\partial u_2/\partial r = 12 - q - 2r$.
5. İki firma da kendi kısmi türevini sıfıra eşitler: $12 - 2q - r = 0$ ve $12 - q - 2r = 0$.
6. Birinci denklemden ikincisi çıkarılır: $(12 - 2q - r) - (12 - q - 2r) = -q + r = 0$, yani $q = r$. Bu birinci denkleme konur: $12 - 3q = 0 \Rightarrow q = 4$. Sonuç $q = r = 4$'tür.
7. Denetim: $r = 4$'e karşı birinci firmanın en iyi miktarı $(12 - 4)/2 = 4$'tür (0.3.6); simetriden aynısı ikinci firma için de geçerlidir. Her firmanın kârı $4 \cdot (12 - 8) = 16$'dır.



Şekil 0.14: Rakibin her miktarı *r* için ayrı bir kâr eğrisi vardır. *r = 2*'de tepe *q = 5*, *u = 25*; *r = 4*'te tepe *q = 4*, *u = 16*'dır. *∂u/∂q*, o eğrinin eğimidir.

Şekil 0.14 kısmi türevin geometrik anlamını gösterir. *r*'nin her değeri, *q*'ya bağlı ayrı bir eğri verir. *∂u/∂q*, *r*'yi sabitleyerek seçilen eğrinin eğimidir. Rakip daha çok ürettikçe eğri hem aşağı iner hem de tepesi sola kayar: firma hem daha az kazanır hem de daha az üretmek ister. Örnekte bulunan *q = r = 4* noktasında iki firmanın da kendi eğrisinin tepesinde olması, hiçbirinin miktarını tek başına değiştirmek istememesi demektir. Oyun kuramında bu özelliğe denge denir.

Sık yapılan hata

Sabit tutulan değişkeni sıfır almak ya da onu da türevlemek. *∂(q · r)/∂q = r*'dir; 0 değildir (r sabittir ama sıfır değildir) ve *1 + r* değildir (r'nin türevi alınmaz). İkinci bir hata da yanlış değişkene göre türev almaktır: birinci firma *r*'yi seçemez, bu yüzden onun koşulu $\partial u_1/\partial r = 0$ değil, $\partial u_1/\partial q = 0$'dır.

0.5 Olasılık ve beklenen değer

Oyunlarda belirsizlik iki kaynaktan gelir. Birincisi doğadır: zar, kart, hava durumu. İkincisi rakiptir: rakibin ne oynayacağı kesin bilinmez ve bazen rakip bilerek rastgele davranır. İki durumda da kararı belirleyen soru aynıdır: “ortalamada ne kazanırım?” Bu soruyu cevaplamak için olasılığı sayarak bulmak, bilgi gelince güncellemek ve ortalama kazancı, yani beklenen değeri hesaplamak gerekir. Değerler sürekli olduğunda (0 ile 1 arasındaki her sayı gibi) toplamın yerini integral alır.

0.5.1 Sayarak olasılık hesaplamak

Bir deneyin (zar atma, kart çekme) olası sonuçlarının hepsi eşit şanslıysa bir olayın olasılığı bir sayma oranıdır: istenen sonuçların sayısı, bütün sonuçların sayısına bölünür. Olasılık her zaman 0 ile 1 arasında bir sayıdır; 0 "asla", 1 "kesin" demektir.

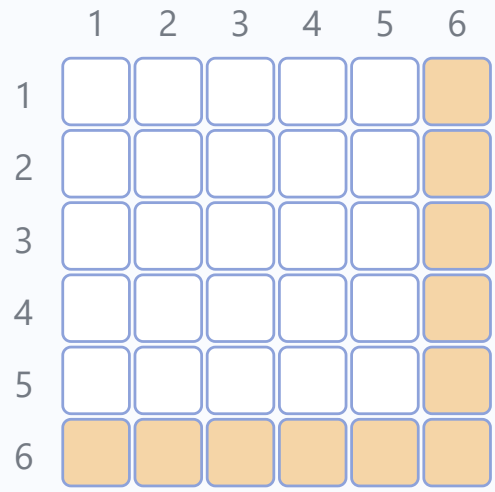
Tanım 0.12

- Sonuçlar eşit olasıysa: *P(olay) = istenen sonuç sayısı / tüm sonuç sayısı*.
- İki olay **bağımsızsa** (biri ötekinin olasılığını etkilemiyorsa) ikisinin birlikte olma olasılığı çarpımdır: *P(A ve B) = P(A) · P(B)*.
- İki olay aynı anda olamıyorsa, birinin ya da ötekinin olma olasılığı toplamdır: *P(A ya da B) = P(A) + P(B)*.
- **Tümleyen**: bir olayın olmama olasılığı *P(A değil) = 1 − P(A)*'dır.

Bağımsızlık için çarpım kuralı, 0.2.3'teki sayma kuralının olasılık hâlidir. İki zar atıldığında birinci zarın 6 sonucu, ikinci zarın 6 sonucuyla eşleşerek 36 sonuçtan birini oluşturur. Bu yüzden iki zarın da 6 gelme olasılığı *1/36 = 1/6 · 1/6*'dır. Tümleyen kuralı ise "en az bir" sorularında çok işe yarar: "en az bir 6" olayını doğrudan saymak yerine "hiç 6 gelmemesi" sayılır ve 1'den çıkarılır.

Örnek 0.13

1. **Zarda çift gelme.** İstenen sonuçlar *{2, 4, 6}*: üç sonuç. Tüm sonuçlar altı tanedir. *P = 3/6 = 1/2*.
2. **İki yazı-turada iki yazı.** Atışlar bağımsızdır: *1/2 · 1/2 = 1/4*. Sayarak denetim: dört eşit olası sonuç vardır (YY, YT, TY, TT) ve yalnız biri iki yazıdır.
3. **İki zarda en az bir 6.** Tümleyen kullanılır. Bir zarda 6 gelmeme olasılığı *5/6*'dır. İki zar bağımsızdır, hiç 6 gelmemesi *(5/6) · (5/6) = 25/36*'dır. O hâlde *P(en az bir 6) = 1 − 25/36 = 11/36*.
4. **Sayarak denetim.** Şekil 0.15'te 36 sonuç bir tabloya dizilmiştir. Altıncı satırda 6 hücre, altıncı sütunda 6 hücre vardır; (6, 6) hücresi ikisinde de bulunduğu için bir kez çıkarılır: *6 + 6 − 1 = 11* turuncu hücre.



Şekil 0.15: İki zarın 36 eşit olası sonucu. En az bir 6 içeren 11 hücre turuncudur; (6, 6) köşesi bir kez sayılır.

Sık yapılan hata

"İki zarda en az bir 6" için *1/6 + 1/6 = 12/36* demek. Bu toplam (6, 6) sonucunu iki kez sayar; doğru cevap *11/36*'dır. Olasılıklar ancak olaylar aynı anda olamıyorsa toplanır. "Birinci zar 6" ile "ikinci zar 6" aynı atışta birlikte olabilir. Aynı hata daha çok zarda daha da büyür: altı zarda toplama yöntemi $6 \cdot 1/6 = 1$ verir, yani "kesin en az bir 6" der; oysa doğru cevap $1 - (5/6)^6 \approx 0{,}665$'tir.

0.5.2 Olasılık dağılımı

Bir deneyin sonucu bir sayıysa (zarın yüzü, iki zarın toplamı, bir piyangonun ödülü) bu sayıya **rastgele değişken** denir ve genellikle büyük harfle yazılır: X. X'in her olası değerine bir olasılık yazan tabloya **olasılık dağılımı** denir.

Tanım 0.13

X'in alabileceği değerler $x_1, \ldots, x_k$ ise dağılımı, $P(X = x_j)$ olasılıklarının listesidir. Bir listenin dağılım olabilmesi için iki koşul gerekir:

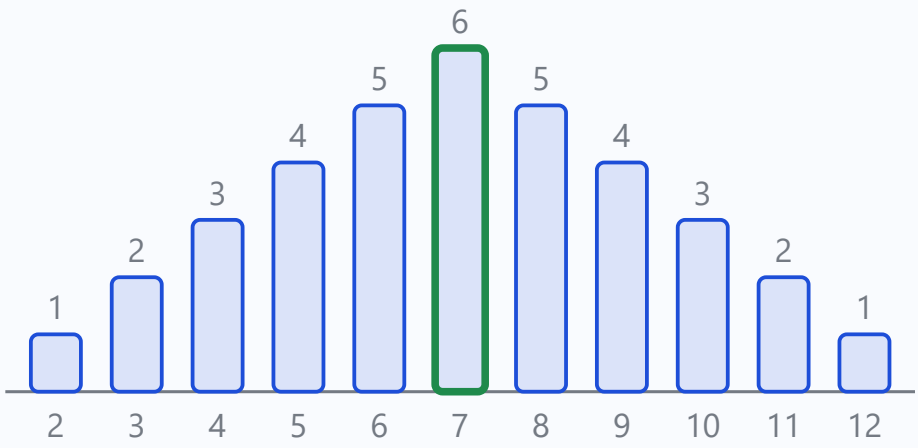
- her olasılık 0 ile 1 arasındadır: $0 \le P(X = x_j) \le 1$,
- olasılıkların toplamı 1'dir: $\Sigma_j P(X = x_j) = 1$.

Bu iki koşul, karışık stratejilerin tanımında da aynen kullanılır (0.6.1). Bir oyuncunun hamlelerine verdiği olasılıklar bir dağılım oluşturmalıdır. Hesaplarda yapılan hatalar çoğu zaman ikinci koşul denetlenerek yakalanır: olasılıklar toplanır ve 1 çıkmazsa bir yerde yanlışlık vardır.

Örnek 0.14

X, iki adil zarın toplamı olsun. *X*'in değerleri 2'den 12'ye kadardır.

1. 36 eşit olası sonuç vardır (0.5.1).
2. Toplam 7'yi veren çiftler sayılır: (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1). Altı çift vardır: $P(X = 7) = 6/36 = 1/6$.
3. Toplam 2 yalnız (1, 1) ile olur: $P(X = 2) = 1/36$. Toplam 12 de yalnız (6, 6) ile olur.
4. Genel olarak toplam 2'den 7'ye çıkarken çift sayısı 1, 2, 3, 4, 5, 6; 7'den 12'ye çıkarken 5, 4, 3, 2, 1 olur.
5. Denetim: $1 + 2 + 3 + 4 + 5 + 6 + 5 + 4 + 3 + 2 + 1 = 36$; toplam olasılık $36/36 = 1$'dir ✓.



Şekil 0.16: İki zarın toplamının dağılımı. Çubukların üstündeki sayılar, 36 sonuçtan kaçının o toplamı verdiğini gösterir. En olası toplam 7'dir (yeşil, 6/36).

Sık yapılan hata

Her toplamı eşit olası sanmak: "11 farklı toplam var, her biri 1/11." Eşit olası olan *sonuçlardır* (zar çiftleri), toplamlar değil. Toplam 7, altı farklı çiftle; toplam 2 ise tek bir çiftle elde edilir. Şekil 0.16'daki çubukların farklı yükseklikte olması bunun görüntüsüdür. Sayarak olasılık kuralı (0.5.1) yalnız eşit olası sonuçlara uygulanır; önce eşit olası sonuçların neler olduğu belirlenmelidir.

0.5.3 Koşullu olasılık

Bir oyuncu oyun sırasında bilgi edinir: rakibin bir hamlesini görür, bir işaret alır, bir kartını açar. Bu bilgi olasılıkları değiştirir. Örneğin desteden çekilen kartın kırmızı olduğu öğrenilince onun kupa olma olasılığı 1/4'ten 1/2'ye çıkar. Yeni bilgiyle güncellenmiş olasılığa **koşullu olasılık** (*conditional*

probability) denir.

Tanım 0.14

P(B) > 0 olmak üzere, *B* olayının gerçekleştiği bilinirken *A*'nın olasılığı

$$P(A \mid B) = P(A \cap B) / P(B)$$

ile tanımlanır. $A \cap B$, *A* ile *B*'nin ikisinin birden gerçekleşmesidir. $P(A \mid B)$ "B verildiğinde A'nın olasılığı" diye okunur.

Formülün anlamı şudur: *B*'nin gerçekleştiği öğrenilince *B*'nin dışındaki sonuçlar artık imkânsızdır. Geriye yalnız *B*'nin içindeki sonuçlar kalır ve bu daralmış dünyada *A*'nın payı aranır. Payda $P(B)$, yeni dünyanın büyüklüğüdür; pay $P(A \cap B)$, bu dünyada *A*'nın kapladığı yerdir. Kart örneğinde kırmızı kartlar 26 tanedir ve bunların 13'ü kupadır: $P(\text{kupa} \mid \text{kırmızı}) = (13/52)/(26/52) = 1/2$.

Örnek 0.15

Bir kavşaktaki ışık üç durumdan birini 1/3 olasılıkla seçer: (Ayşe Dur, Bora Dur), (Ayşe Dur, Bora Geç), (Ayşe Geç, Bora Dur). Her sürücüye yalnız kendi işaretini gösterir. Ayşe "Dur" görüyor. Bora'nın "Geç" görmüş olma olasılığı nedir?

1. *B* = "Ayşe Dur görür". Bu olay ilk iki durumda gerçekleşir: $P(B) = 1/3 + 1/3 = 2/3$.
2. *A* = "Bora Geç görür". $A \cap B$ yalnız ikinci durumdur: $P(A \cap B) = 1/3$.
3. Tanım uygulanır: $P(\text{Bora Geç} \mid \text{Ayşe Dur}) = (1/3)/(2/3) = 1/2$.
4. Yorum: Ayşe "Dur" gördüğünde kalan iki eşit olası durumdan birinde Bora geçiyor, ötekinde duruyor. Ayşe için Bora'nın geçme olasılığı 1/3'ten 1/2'ye çıkmıştır. Şekil 0.17 bu daralmayı kutularla gösterir.



Şekil 0.17: Işık üç durumdan birini 1/3 olasılıkla seçer. Ayşe "Dur" görünce yalnız ilk iki kutu kalır; bunlardan birinde Bora geçer: *P(Bora Geç | Ayşe Dur) = (1/3)/(2/3) = 1/2*.

Bu örnek oyun kuramında önemli bir fikrin temelidir. Işık, sürücülerin çarpışmasını önleyen bir koordinasyon aracıdır. Her sürücü ışığın önerisine uyup uymamaya kendisi karar verir. Ayşe "Dur" gördüğünde Bora'nın yarı yarıya geçtiğini bilir; bu bilgiyle durmanın onun için iyi bir seçim olup olmadığını hesaplayabilir. Önerilerin herkesin işine geldiği bu tür düzenlere korelasyonlu denge denir. Hesabın tamamı koşullu olasılığa dayanır.

Sık yapılan hata

Bilgiyi yok saymak ("Bora Geç görür: 1/3") ya da koşulun yönünü karıştırmak. *P(A | B)* ile *P(B | A)* genellikle farklıdır. Örnekte *P(Ayşe Dur | Bora Geç) = 1*'dir: Bora geçiyorsa Ayşe kesin durur. Ama *P(Bora Geç | Ayşe Dur) = 1/2*'dir. Hangi olayın bilindiği (çizginin sağı) ve hangisinin sorulduğu (çizginin solu) her zaman ayrı ayrı yazılmalıdır.

0.5.4 Bayes kuralı

Poker gibi oyunlarda oyuncular rakibin "tipini" bilmez: rakip agresif mi, sakin mi? Ama rakibin hamlelerini görür. Büyük bir bahis, rakibin agresif olduğuna dair bir işarettir; ne kadar güçlü bir işaret olduğu ise hesapla bulunur. Gözlemden sonra inancı güncellemenin kuralı **Bayes kuralıdır** (*Bayes' rule*).

Teorem 0.2 (Bayes kuralı)

Gözlemden önceki inanç $P(A)$'ya **ön olasılık** (*prior*), gözlem B yapıldıktan sonraki inanç $P(A \mid B)$'ye **son olasılık** (*posterior*) denir. $P(B) > 0$ ise

$$P(A \mid B) = P(B \mid A) \cdot P(A) \,/\, P(B)$$

Payda çoğu zaman iki parçadan hesaplanır: $P(B) = P(B \mid A) \cdot P(A) + P(B \mid A \text{ değil}) \cdot P(A \text{ değil})$.

İspat. Koşullu olasılığın tanımı iki yönde yazılır. $P(A \mid B) = P(A \cap B)/P(B)$ ve $P(B \mid A) = P(A \cap B)/P(A)$'dır. İkinci eşitlik $P(A)$ ile çarpılınca $P(A \cap B) = P(B \mid A) \cdot P(A)$ olur. Bu, birinci eşitliğin payına konunca kural elde edilir. Payda için şu gözlem yeterlidir: B, ya A ile birlikte ya da A olmadan gerçekleşir; bu iki durum aynı anda olamaz. Bu yüzden $P(B) = P(A \cap B) + P(A \text{ değil} \cap B)$'dir ve her parça aynı çarpım kuralıyla yazılır. □

Örnek 0.16

Oyuncuların %20'si agresiftir. Agresif bir oyuncu büyük bahsi 0,8, sakin bir oyuncu 0,2 olasılıkla yapar. Bir oyuncunun büyük bahis yaptığı görüldü. Bu oyuncunun agresif olma olasılığı nedir?

1. Ön olasılık: $P(\text{agresif}) = 0{,}2$; dolayısıyla $P(\text{sakin}) = 0{,}8$.
2. Gözlemin olasılıkları: $P(\text{büyük} \mid \text{agresif}) = 0{,}8$, $P(\text{büyük} \mid \text{sakin}) = 0{,}2$.
3. Payda: $P(\text{büyük}) = 0{,}8 \cdot 0{,}2 + 0{,}2 \cdot 0{,}8 = 0{,}16 + 0{,}16 = 0{,}32$.
4. Pay: $P(\text{büyük} \mid \text{agresif}) \cdot P(\text{agresif}) = 0{,}8 \cdot 0{,}2 = 0{,}16$.
5. Son olasılık: $P(\text{agresif} \mid \text{büyük}) = 0{,}16 \,/\, 0{,}32 = 1/2$.
6. **Sayarak denetim.** 100 oyuncu düşünülür (Şekil 0.18). 20'si agresiftir ve bunların $0{,}8 \cdot 20 = 16$'sı büyük bahis yapar. 80'i sakindir ve bunların $0{,}2 \cdot 80 = 16$'sı büyük bahis yapar. Büyük bahis yapan $16 + 16 = 32$ kişiden 16'sı agresiftir: $16/32 = 1/2$ ✓.



Şekil 0.18: Bayes kuralının sayı ağacı. 100 oyuncudan 20'si agresif, 80'i sakindir. Büyük bahis yapan 16 + 16 = 32 kişinin 16'sı agresiftir.

Sonuç ilk bakışta şaşırtıcıdır: agresif oyuncular büyük bahsi %80 oranında yaptığı hâlde, büyük bahis yapan birinin agresif olma olasılığı yalnız %50'dir. Bunun nedeni ön olasılıktır. Sakin oyuncular dört kat kalabalıktır. Her biri seyrek büyük bahis yapsa da sayıları çok olduğu için büyük bahislerin yarısı onlardan gelir. Gözlem inancı 0,2'den 0,5'e çıkarmıştır; bu güçlü bir güncellemedir ama kesinlik vermez.

Sayı ağacı yöntemi Bayes hesaplarının en güvenli yoludur. Olasılıklar yerine 100 ya da 1000 kişilik bir grup düşünülür, her daldaki kişi sayısı çarpılarak bulunur ve sonunda yalnız gözlemle uyumlu dallar sayılır. Formüldeki her terim ağaçta bir dala karşılık gelir: pay, gözlemle uyumlu "agresif" dalı; payda, gözlemle uyumlu bütün dallardır.

Sık yapılan hata

Ön olasılığı unutmak: "Agresifler %80 büyük bahis yapar, o hâlde bu oyuncu %80 agresiftir." Bu cümle *P(büyük | agresif)* ile *P(agresif | büyük)*'ü karıştırır (0.5.3'teki yön hatası). Sakin oyuncular dört kat kalabalık olduğu için onların seyrek büyük bahsi de toplamın yarısını oluşturur. Ön olasılık atlanırsa nadir tipler her zaman olduğundan sık görünür.

0.5.5 Beklenen değer

Belirsiz bir durumda karar vermenin en temel aracı beklenen değerdir. Bir oyun çok kez oynansaydı tur başına ortalama ne kazanılırdı? Bu sorunun cevabı, her olası sonucun kendi olasılığı oranında hesaba katıldığı ağırlıklı bir ortalamadır.

Tanım 0.15

X, değeri şansa bağlı bir sayı olsun. *X*'in **beklenen değeri** (*expected value*), alabileceği her *x* değerinin olasılığıyla çarpılıp toplanmasıdır:

$$E[X] = \Sigma_x \, x \cdot P(X = x)$$

Beklenen değerin neden "uzun vadeli ortalama" olduğu şöyle görülür. Adil bir zar 600 kez atılsın. Her yüz yaklaşık 100 kez gelir. Bütün atışların toplamı yaklaşık $1 \cdot 100 + 2 \cdot 100 + \ldots + 6 \cdot 100 = 2100$'dür; atış başına ortalama $2100/600 = 3{,}5$'tir. Bu hesap, her değeri sıklığıyla, yani olasılığıyla çarpıp toplamaktan başka bir şey değildir.

Beklenen değerin iki kullanışlı özelliği vardır. Birincisi, bir sabitle çarpmak ve bir sabit eklemek beklenen değere aynen geçer: $E[aX + b] = a \cdot E[X] + b$. Bir oyunun ödülleri iki katına çıkarsa beklenen ödül de iki katına çıkar; giriş ücreti düşülürse beklenen net kazanç da o kadar düşer. İkincisi, toplamın beklenen değeri beklenen değerlerin toplamıdır: iki zarın toplamının beklenen değeri $3{,}5 + 3{,}5 = 7$'dir. Bu ikinci özellik, zarlar bağımsız olmasa bile geçerlidir.

Örnek 0.17

X adil bir zarın gelen yüzü olsun.

1. Değerler 1, 2, 3, 4, 5, 6; her birinin olasılığı 1/6'dır (Şekil 0.19).
2. Her değer olasılığıyla çarpılır ve toplanır: $E[X] = 1 \cdot (1/6) + 2 \cdot (1/6) + 3 \cdot (1/6) + 4 \cdot (1/6) + 5 \cdot (1/6) + 6 \cdot (1/6)$.
3. Ortak çarpan 1/6 dışarı alınır: $E[X] = (1 + 2 + 3 + 4 + 5 + 6)/6 = 21/6 = 3{,}5$.



Şekil 0.19: Adil zarın dağılımı: her yüz 1/6. Beklenen değer $E[X] = 21/6 = 3{,}5$, hiçbir çubuğun üstünde değildir.

Oyun kuramında beklenen değer her yerdedir. Bir oyuncu, rakibin olasılıklı davranışına karşı her hamlesinin beklenen kazancını hesaplar ve en büyüğünü seçer (0.6.2). Bir mekanizma tasarımcısı, katılımcıların beklenen ödemelerini hesaplar. Bu hesapların hepsi aynı formülün uygulamasıdır: değer çarpı olasılık, sonra topla.

> **Sık yapılan hata**
>
> Beklenen değeri "en olası sonuç" ya da "kesin gelecek sonuç" sanmak. Zarda 3,5 hiç gelmez; 3,5 yalnızca çok sayıda atışın ortalamasıdır. Kazancı 0 ya da 100 olan bir piyangonun beklenen değeri 30 olabilir (100'ü 0,3 olasılıkla kazandırıyorsa), ama piyango hiçbir zaman 30 ödemez. Beklenen değer tek bir oyunun sonucunu değil, tekrarın ortalamasını söyler.

0.5.6 Kod izleme: piyangonun beklenen değeri

Bir piyango bileti 0,05 olasılıkla 100 TL, 0,25 olasılıkla 20 TL kazandırıyor; 0,70 olasılıkla hiçbir şey kazandırmıyor. Bilet 12 TL'dir. Aşağıdaki kod, sunumdaki kodun aynısıdır ve beklenen değer formülünü bir döngüyle uygular.

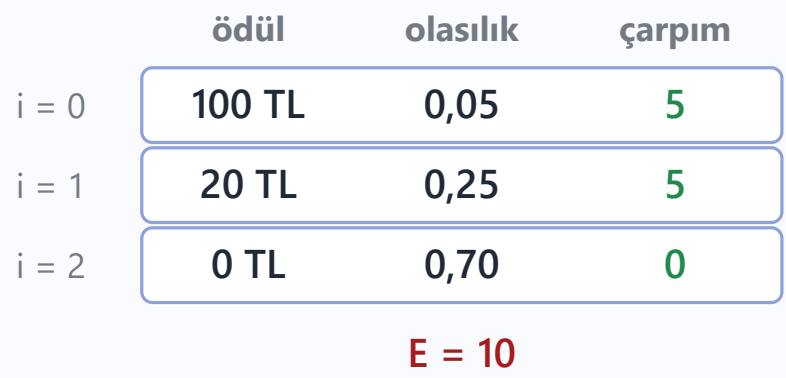
```
odul = [100, 20, 0]          # TL
olas = [0.05, 0.25, 0.70]
E = 0
for i in range(3):
    E = E + odul[i] * olas[i]
print("E =", E)
print("net =", E - 12)  # bilet 12 TL
```

Kodun yapısı Σ toplamının birebir karşılığıdır:

- `odul` ve `olas` listeleri aynı indisle eşleşir: indis 0 en büyük ödüldür (100 TL, 0,05).
- `E = 0`: toplayıcı sıfırdan başlar. Σ'nın "henüz hiçbir şey toplanmadı" durumudur.
- Döngünün her adımında bir terim, `odul[i] * olas[i]`, toplayıcıya eklenir.
- Son iki satır beklenen ödülü ve bilet fiyatı düşülmüş net beklenen kazancı yazar.

i	odul[i]	olas[i]	Terim	E (adımdan sonra)
–	–	–	–	0
0	100	0,05	5	5
1	20	0,25	5	10
2	0	0,70	0	10

Tablo 0.2: Piyango kodunun izlemesi.



Şekil 0.20: İzlemenin son durumu: üç terim 5, 5 ve 0'dır; toplayıcı *E = 10*'dur. Olasılıkların toplamı *0,05 + 0,25 + 0,70 = 1*'dir.

Tablo 0.2 her adımda eklenen terimi ve toplayıcının yeni değerini, Şekil 0.20 son durumu gösterir. İzlemenin dikkat çeken adımı üçüncü terimdir. Boş biletin ödülü 0'dır ve *0 · 0,70 = 0* toplayıcıyı değiştirmez. Ama bu satır gereksiz değildir: olasılığı 0,70'tir ve dağılımın toplamının 1 olması için sayılması gerekir. Olasılıkların toplamını denetlemek, eksik bir sonucun unutulmadığını gösterir.

Bir biletin beklenen ödülü 10 TL'dir. Bilet 12 TL olduğuna göre net beklenen kazanç *10 − 12 = −2* TL'dir. Çok kez oynayan biri bilet başına ortalama 2 TL kaybeder. Aynı sonuç beklenen değerin doğrusallığından da çıkar: net kazanç *X − 12*'dir ve *E[X − 12] = E[X] − 12 = −2*'dir. Kod çalıştırılınca iki satır yazar: `E = 10.0` ve `net = -2.0`. Python ondalık sayıları noktayla yazar (`10.0`). Bu örnekte sonuç tam çıkar; başka olasılıklarla kayan nokta yuvarlaması nedeniyle `9.999999999999998` gibi çok yakın bir sayı da görülebilir. Bu, hesabın yanlış olduğunu değil, bazı ondalık sayıların ikili sistemde tam gösterilemediğini gösterir.

0.5.7 Bu zar oyunu oynanır mı?

Beklenen değerin en doğrudan kullanımı, bir oyuna girmeye değip değmeyeceğine karar vermektir. Hesap bir tabloyla düzenli biçimde yapılabilir; “adil fiyat” sorusu ise tek bilinmeyenli bir denklemle cevaplanır.

Örnek 0.18

Panayırda bir oyun: 10 TL ödeyip bir zar atılıyor. 6 gelirse 40 TL, 5 gelirse 10 TL alınıyor; başka bir sayıda hiçbir şey alınmıyor.

1. Olası ödüller ve olasılıkları bir tabloya yazılır (Tablo 0.3). 1, 2, 3 ve 4 aynı ödülü (0) verdiği için tek satırda birleştirilir; olasılıkları toplanır: *4/6*.
2. Her satır için *x · P(x)* hesaplanır: *40/6*, *10/6* ve 0.
3. Beklenen ödül *50/6 ≈ 8,33* TL'dir.

4. Giriş 10 TL olduğu için net beklenen kazanç $50/6 - 10 = -10/6 \approx -1{,}67$ TL'dir. Risk umursanmıyorsa oynamak kârlı değildir.

Zar	Ödül x	P(x)	x · P(x)
6	40	1/6	40/6
5	10	1/6	10/6
1, 2, 3, 4	0	4/6	0
Toplam		6/6 = 1	50/6 ≈ 8,33

Tablo 0.3: Zar oyununun beklenen değer tablosu.

Oyun ne olsaydı adil olurdu? Bir oyuna, net beklenen kazancı sıfırsa *adil* denir. 6 için ödül bilinmeyen bir y olsun; öteki ödüller aynı kalsın. Beklenen ödül $y/6 + 10/6 = (y + 10)/6$'dır. Adillik için bu, giriş ücretine eşit olmalıdır:

$$(y + 10)/6 = 10 \Rightarrow y + 10 = 60 \Rightarrow y = 50$$

6 için ödül 50 TL olsaydı oyun adil olurdu. Aynı soru başka biçimde de sorulabilir: ödüller aynı kalırsa giriş ücreti ne olmalıydı? Cevap beklenen ödülün kendisidir, yani yaklaşık 8,33 TL. Bu, tek bilinmeyenli bir denklemdir; iki bilinmeyenli denklemler 0.9.2'de işlenir.

Sık yapılan hata

"Kazanma şansım 2/6, fena değil" diye karar vermek. Kazanma olasılığı tek başına yetmez; ödüllerin büyüklüğü ve giriş ücreti de hesaba girer. 6 için ödül 1000 TL olsaydı aynı 2/6 şansla oyun çok kârlı olurdu; 11 TL olsaydı çok zararlı olurdu. Karar, net beklenen değere göre verilir. (Riskten kaçınan biri net beklenen değeri pozitif bir oyunu da reddedebilir; bu tercih fayda fonksiyonlarıyla modellenir.)

0.5.8 Sürekli düzgün dağılım U[0, 1]

Şimdiye kadar rastgele değişkenler birkaç değerden birini alıyordu. Bazı büyüklükler ise bir aralıktaki herhangi bir sayı olabilir: bir alıcının bir mala biçtiği değer, bir teklif, bir bekleme süresi. Açık artırma modellerinde en sık kullanılan varsayım, rakiplerin değerlerinin 0 ile 1 arasında "eşit şanslı" dağıldığıdır. Bu dağılıma **düzgün dağılım** (*uniform distribution*) denir.

Tanım 0.16

X ~ U[0, 1] yazımı "*X*, 0 ile 1 arasında düzgün dağılır" diye okunur: bu aralıktaki her değer eşit şanslıdır. Grafiği yüksekliği 1 olan bir dikdörtgendir; bu yüksekliğe **olasılık yoğunluğu** denir: *f(x) = 1* (*0 ≤ x ≤ 1*). Olasılık, yoğunluğun altındaki *alandır*:

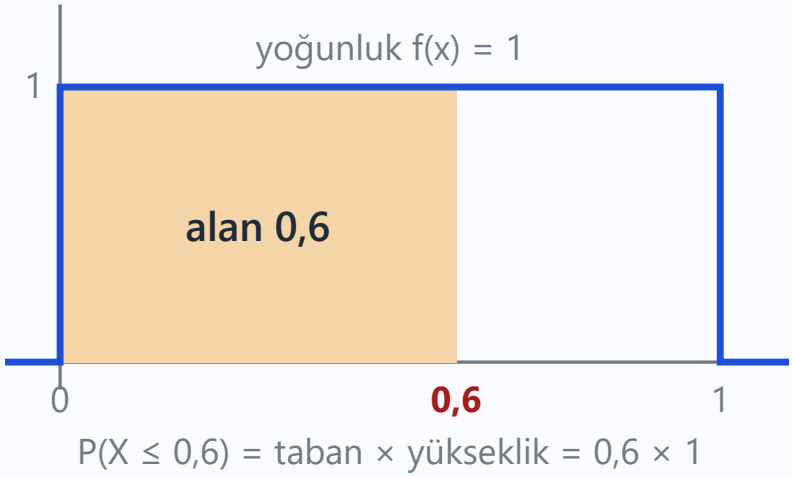
> $P(a \le X \le b) = b - a$ ve özellikle $P(X \le x) = x$.

Sürekli dağılımda tek bir noktanın olasılığı sıfırdır. Bu ilk bakışta tuhaftır: 0,5 değeri pekâlâ gelebilir, olasılığı nasıl sıfır olur? Açıklama şudur: aralıkta sonsuz sayıda değer vardır. Her birine pozitif bir olasılık verilseydi toplam sonsuz olurdu. Bu yüzden olasılık noktalara değil aralıklara verilir; bir aralığın olasılığı onun uzunluğudur. Uzunluğu sıfır olan bir noktanın olasılığı da sıfırdır. Bunun bir sonucu, *P(X ≤ 0,6)* ile *P(X < 0,6)*'nın aynı olmasıdır.

Örnek 0.19

Bir açık artırmada rakibin değeri *X ~ U[0, 1]*'dir. Rakip değeri kadar teklif veriyor.

1. **Tek rakip.** 0,6 teklif, rakibin değeri 0,6'dan küçükse kazanır: *P(X ≤ 0,6) = 0,6* (Şekil 0.21). Alan: taban 0,6 × yükseklik 1.
2. **İki bağımsız rakip.** İkisini birden geçmek için iki olay birlikte gerçekleşmelidir. Bağımsızlıktan *0,6 · 0,6 = 0,36*.
3. **Bir aralık.** *P(0,2 ≤ X ≤ 0,5) = 0,5 − 0,2 = 0,3*.
4. **Ortalama.** Simetriden ortalama aralığın ortasıdır: *E[X] = 1/2*. Bu sonuç 0.5.9'da integralle de bulunur.



Şekil 0.21: *U[0, 1]*'in yoğunluğu: yüksekliği 1 olan dikdörtgen. *P(X ≤ 0,6)*, turuncu alandır: taban × yükseklik =

0,6 × 1 = 0,6.

İkinci adımdaki çarpım kuralının etkisi, rakip sayısı arttıkça hızla büyür. n bağımsız rakibin hepsini 0,6 teklifle geçme olasılığı $0{,}6^n$'dir: 1 rakipte 0,6; 2'de 0,36; 5'te yaklaşık 0,078. Rakip sayısı arttıkça kazanmak için daha yüksek teklif gerekir. Açık artırma kuramındaki birçok hesap bu basit gözleme dayanır.

Sık yapılan hata

Yoğunluğun yüksekliğini (1) olasılık sanmak ya da $P(X = 0{,}5)$'e pozitif bir değer vermek. Sürekli dağılımda olasılık bir alandır: taban (aralık) × yükseklik. Yoğunluk 1'den büyük de olabilir: $U[0; 0{,}5]$'in yoğunluğu 2'dir, çünkü 0,5 uzunluklu bir tabanın üstündeki alanın 1 olması gerekir. Yoğunluk bir olasılık değil, "birim uzunluk başına olasılıktır".

0.5.9 İntegral: eğri altındaki alan

Düzgün dağılımda alanlar dikdörtgen olduğu için kolay hesaplanır. Ama beklenen değer gibi hesaplarda eğik ya da eğri çizgilerin altındaki alanlar gerekir. Bir eğrinin altındaki alanı bulma işlemine **integral** denir. İntegral, Σ toplamının sürekli hâlidir.

Tanım 0.17

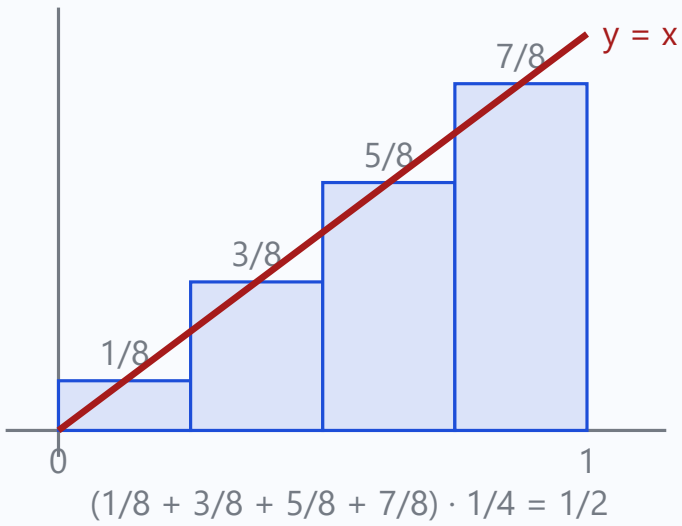
$\int_a^b f(x)\,dx$, f'nin grafiği altında $x = a$ ile $x = b$ arasında kalan alandır. Alan ince dikdörtgenlere bölünüp toplanır; dikdörtgenler inceldikçe toplam integrale yaklaşır. dx, dikdörtgenlerin çok küçük genişliğini simgeler.

Tanımdaki fikir Σ ile doğrudan bağlantılıdır. Aralık n eşit parçaya bölünürse her parçanın genişliği $(b - a)/n$ olur. Her parçanın üstüne, yüksekliği o parçadaki f değeri olan bir dikdörtgen konur. Dikdörtgenlerin alanlarının toplamı bir Σ'dır. n büyüdükçe dikdörtgenler eğriye daha iyi uyar ve toplam gerçek alana yaklaşır. Σ'daki toplama işareti integralde uzatılmış bir S'ye (∫) dönüşmüştür; bu S, Latince "summa" (toplam) sözcüğünün baş harfidir.

Örnek 0.20

1. **Geometriyle.** $\int_0^1 x\,dx$, $y = x$ doğrusunun 0 ile 1 arasında altında kalan alandır. Bu alan, tabanı 1 ve yüksekliği 1 olan bir üçgendir: $1 \cdot 1 / 2 = 1/2$.
2. **Dikdörtgenlerle.** Aralık dört parçaya bölünür, her parçanın genişliği 1/4'tür. Her parçanın ortasındaki yükseklik alınır: 1/8, 3/8, 5/8, 7/8 (Şekil 0.22). Toplam $(1/8 + 3/8 + 5/8 + 7/8) \cdot 1/4 = (16/8) \cdot 1/4 = 1/2$. Doğru bir çizgi olduğu için orta nokta yöntemi burada tam sonucu verir.
3. **Genel kural.** Aynı üçgen düşüncesiyle $\int_0^b x\,dx = b \cdot b / 2 = b^2/2$.

4. **Olasılık bir alandır.** *U[0, 1]* için $P(X \le b) = \int_0^b 1\,dx = b$: yüksekliği 1, tabanı b olan dikdörtgen.
5. **Beklenen değer bir alandır.** $E[X] = \int_0^1 x \cdot 1\,dx = 1/2$. Bu, $E[X] = \Sigma\, x \cdot P(x)$ formülünün sürekli karşılığıdır: toplama yerine integral, olasılık yerine yoğunluk çarpı küçük genişlik.



Şekil 0.22: $y = x$ altındaki alan, genişliği 1/4 olan dört dikdörtgenle: $(1/8 + 3/8 + 5/8 + 7/8) \cdot 1/4 = 1/2$. Bu, $\int_0^1 x\,dx$'in değeridir.

Bu notta integral yalnız alan fikri olarak gereklidir; integral hesaplama teknikleri gerekmez. Karşılaşılan alanlar dikdörtgen ve üçgendir. Önemli olan, olasılığın ve beklenen değerin sürekli durumda birer alan olduğunu görmektir.

Sık yapılan hata

İntegrali eğrinin bir noktadaki yüksekliği sanmak. $\int_0^1 x\,dx$'in sonucu $x = 1$'deki yükseklik (1) değil, birikmiş alan olan 1/2'dir. Benzer bir hata, üçgen alanında 2'ye bölmeyi unutmaktır: $\int_0^2 x\,dx$ için $2 \cdot 2 = 4$ değil, $2 \cdot 2 / 2 = 2$ doğrudur. $E[X] = \int x \cdot f(x)\,dx$, $E[X] = \Sigma\, x \cdot P(x)$'in sürekli karşılığıdır; yoğunluk $f(x)$ çarpanı atlanmamalıdır.

0.5.10 Beklenen değer ve kuyruk olasılığı

Beklenen değeri hesaplamanın ikinci bir yolu vardır. Bu yol, açık artırma kuramındaki gelir ve ödeme hesaplarında sık kullanılır. Fikir şudur: X'in değerine doğrudan bakmak yerine, X'in her eşiği aşma olasılığına bakılır.

Tanım 0.18

$P(X > t)$, X'in t eşiğini aşma olasılığıdır. Buna **kuyruk olasılığı** (*tail probability*) denir. $X \geq 0$ ise beklenen değer, kuyruk olasılığı eğrisinin altındaki alandır:

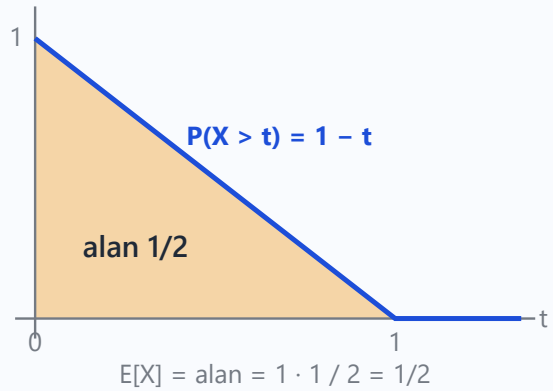
$$E[X] = \int_0^\infty P(X > t)\, dt$$

Değerleri 0, 1, 2, … olan X için toplam hâli: $E[X] = P(X \geq 1) + P(X \geq 2) + P(X \geq 3) + \ldots$

Toplam hâlinin neden doğru olduğu sayarak görülür. $X = 3$ olan bir sonuç düşünelim. Bu sonuç "$X \geq 1$", "$X \geq 2$" ve "$X \geq 3$" olaylarının üçünde de vardır, "$X \geq 4$"te yoktur. Kuyrukların toplamında bu sonuç tam üç kez sayılır, yani kendi değeri kadar. Genel olarak değeri x olan her sonuç kuyruk toplamında x kez sayılır. Bu yüzden kuyrukların toplamı $\Sigma\, x \cdot P(X = x)$'e, yani beklenen değere eşittir. İntegral hâli aynı fikrin sürekli sürümüdür.

Örnek 0.21

1. **Düzgün dağılım.** $X \sim U[0, 1]$ için $0 \leq t \leq 1$ iken $P(X > t) = 1 - t$; $t > 1$ için 0'dır.
2. Kuyruk eğrisi, (0, 1)'den (1, 0)'a inen bir doğrudur (Şekil 0.23). Altındaki alan taban 1, yükseklik 1 olan üçgendir: $E[X] = \int_0^1 (1 - t)\, dt = 1/2$ ✓. Bu, 0.5.9'daki sonuçla aynıdır.
3. **Zar.** $P(X \geq 1) = 6/6$, $P(X \geq 2) = 5/6$, …, $P(X \geq 6) = 1/6$.
4. Toplam: $(6 + 5 + 4 + 3 + 2 + 1)/6 = 21/6 = 3{,}5$ ✓. Bu da 0.5.5'teki sonuçla aynıdır.



Şekil 0.23: $U[0, 1]$ için kuyruk olasılığı $P(X > t) = 1 - t$. Altındaki turuncu üçgenin alanı $1 \cdot 1 / 2 = 1/2 = E[X]$'tir.

Kuyruk formülü neden gereklidir? Bazı problemlerde X'in dağılımı doğrudan değil, eşik olasılıkları biçiminde bilinir. Örneğin iki rakibin en yüksek değerini aşma olasılığı çarpım kuralıyla kolayca bulunur (0.5.8); bu olasılıktan beklenen değere geçmenin en kısa yolu kuyruk formülüdür.

Sık yapılan hata

Yoğunluğun altındaki alanı beklenen değer sanmak. *U[0, 1]*'de yoğunluğun altındaki alan 1'dir (toplam olasılık); *E[X] = 1/2* ise *P(X > t)* eğrisinin altındaki alandır. İki eğri farklıdır. Ayrıca formül yalnız *X ≥ 0* için geçerlidir; negatif değer alabilen bir değişkende ek bir terim gerekir.

0.6 Karışık strateji

Taş-Kağıt-Makas'ta tahmin edilebilir olmak kaybettirir. Hep Taş oynayan biri, rakibi bunu fark edince her turda kaybeder. Bunun çaresi, hangi hamlenin oynanacağını şansa bırakmaktır. Bu fikir, olasılık ile oyun kuramını birleştirir: oyuncu artık bir hamle değil, hamleleri üzerinde bir olasılık dağılımı seçer. Rakip böyle oynadığında her hamle beklenen kazancıyla değerlendirilir. Değerleri olasılıklara çeviren yumuşak bir kural (softmax) ise keskin argmax seçiminin yerini alabilir.

0.6.1 Karışık strateji fikri

Bir oyuncu her turda tek bir hamle yapar. Ama bu hamleyi seçmeden önce bir zar atabilir ya da para fırlatabilir ve hamlesini sonuca göre belirleyebilir. Böylece rakip, oyuncunun hangi hamleyi yapacağını önceden bilemez; en fazla hangi olasılıklarla oynadığını öğrenebilir.

Tanım 0.19

Karışık strateji (*mixed strategy*), bir oyuncunun kendi eylemleri üzerine koyduğu bir olasılık dağılımıdır. Taş-Kağıt-Makas'ta bir karışık strateji üç sayıyla yazılır: (p_T, p_K, p_M); her sayı 0 ile 1 arasındadır ve toplamları 1'dir.

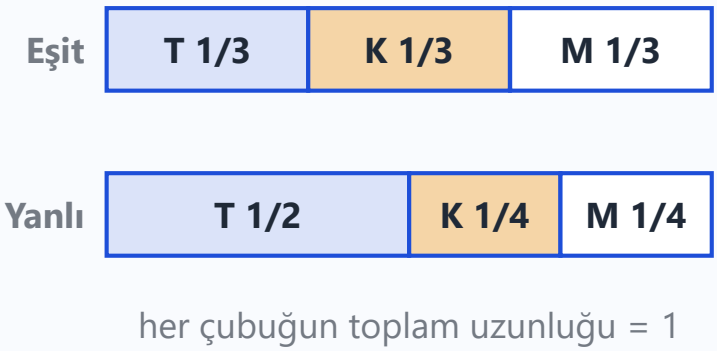
Tek bir eylemi kesin oynamaya (bir olasılığı 1, ötekileri 0 olan dağılım) **saf strateji** (*pure strategy*) denir. Pozitif olasılık verilen eylemlerin kümesine stratejinin **desteği** denir.

Karışık strateji yeni bir hamle eklemez; var olan hamleleri bir rastgelelik düzeneğiyle birleştirir. Her saf strateji, karışık stratejinin özel bir hâlidir: *(1, 0, 0)* "hep Taş" demektir. Böylece karışık stratejiler saf stratejileri de içeren daha geniş bir küme oluşturur.

Örnek 0.22

1. **Zarla eşit karışım.** Zar atılır: 1–2 gelirse Taş, 3–4 gelirse Kağıt, 5–6 gelirse Makas. Her hamle altı sonuçtan ikisine karşılık gelir: *(2/6, 2/6, 2/6) = (1/3, 1/3, 1/3)*.
2. **İki parayla yanlı karışım.** Bir para atılır; yazı gelirse Taş oynanır. Tura gelirse ikinci bir para atılır; yazıda Kağıt, turada Makas. Taş'ın olasılığı 1/2'dir. Kağıt için iki bağımsız olay gerekir (tura, sonra yazı): *1/2 · 1/2 = 1/4*. Makas da aynı biçimde 1/4'tür. Sonuç *(1/2, 1/4, 1/4)*; toplam *1/2 + 1/4 + 1/4 = 1* ✓.

3. **Saf strateji.** “Hep Taş” *(1, 0, 0)* karışık stratejisidir. Desteği yalnız {Taş}'tır.



Şekil 0.24: İki karışık strateji birer şerit olarak: eşit karışım *(1/3, 1/3, 1/3)* ve yanlı karışım *(1/2, 1/4, 1/4)*. Her şeridin toplam uzunluğu 1'dir; parçaların uzunlukları olasılıklardır.

Şekil 0.24'teki şerit gösterimi, karışık stratejiyi uygulamanın bir yolunu da verir. 0 ile 1 arasında düzgün rastgele bir sayı (0.5.8) çekilir ve sayının şeridin hangi parçasına düştüğüne bakılır. Yanlı karışımda sayı 0,5'ten küçükse Taş, 0,5 ile 0,75 arasındaysa Kağıt, 0,75'ten büyükse Makas oynanır. Parçaların uzunlukları olasılıklara eşit olduğundan her hamle doğru olasılıkla seçilir. Bilgisayar programları karışık stratejileri tam olarak bu yolla uygular.

Sık yapılan hata

Karışık stratejiyi “sırayla değiştirmek” sanmak (Taş, Kağıt, Makas, Taş, …). Bu düzende her hamle uzun vadede 1/3 sıklıkla görülür, ama düzen tahmin edilebilir: Taş'tan sonra Kağıt geleceğini bilen rakip Makas oynar ve her turda kazanır. Karışık stratejide her turda tek bir saf hamle oynanır; hangisinin oynanacağı her turda yeniden ve öncekilerden bağımsız olarak rastgele seçilir.

0.6.2 Karışık stratejiye karşı beklenen kazanç

Rakip karışık bir strateji oynuyorsa bir hamlenin getirisi kesin değildir; şansa bağlıdır. Bu yüzden her hamle, rakibin olasılıklarına göre beklenen kazancıyla değerlendirilir. Hesap, 0.5.5'teki beklenen değer formülünün doğrudan uygulamasıdır: kazanç matrisinde oyuncunun satırındaki her sayı, rakibin ilgili sütunu oynama olasılığıyla çarpılır ve toplanır.

Örnek 0.23

Bora *(1/2 Taş, 1/4 Kağıt, 1/4 Makas)* oynuyor. Ayşe'nin her saf hamlesinin beklenen kazancı aranıyor.

1. **Ayşe Taş.** Ayşe, Bora Taş oynarsa 0, Kağıt oynarsa −1, Makas oynarsa +1 alır. Beklenen: $0 \cdot 1/2 + (-1) \cdot 1/4 + 1 \cdot 1/4 = 0$.
2. **Ayşe Kağıt.** Taş'a karşı +1, Kağıt'a karşı 0, Makas'a karşı −1. Beklenen: $1/2 + 0 - 1/4 = 1/4$.
3. **Ayşe Makas.** Taş'a karşı −1, Kağıt'a karşı +1, Makas'a karşı 0. Beklenen: $-1/2 + 1/4 + 0 = -1/4$.
4. **Karar.** $\text{argmax}\{0, 1/4, -1/4\} = \text{Kağıt}$. Taş ağırlıklı bir rakibe karşı Kağıt tur başına ortalama 1/4 puan kazandırır.

Ayşe ↓ / Bora →	Taş (1/2)	Kağıt (1/4)	Makas (1/4)	Beklenen kazanç
Taş	0	−1	+1	0 − 1/4 + 1/4 = 0
Kağıt	+1	0	−1	**1/2 + 0 − 1/4 = 1/4**
Makas	−1	+1	0	−1/2 + 1/4 + 0 = −1/4

Tablo 0.4: Bora (1/2, 1/4, 1/4) oynarken Ayşe'nin kazançları ve her satırın beklenen değeri.

Bu iki adımlı hesap, yani önce her satırın beklenen değeri, sonra bunların argmax'ı, oyun kuramının en temel hesabıdır. Sonuca **en iyi tepki** denir (0.8). Tablo 0.4 bu hesabın düzenli bir yazımıdır: son sütun her satırın beklenen değerini, yeşil hücre de en iyi tepkiyi gösterir.

Ayşe de karışık oynayabilir. Ayşe (x_T, x_K, x_M) oynarsa onun beklenen kazancı, satır beklentilerinin bu olasılıklarla tartılmış ortalamasıdır. Örneğin Ayşe $(1/3, 1/3, 1/3)$ oynarsa kazancı $1/3 \cdot 0 + 1/3 \cdot 1/4 + 1/3 \cdot (-1/4) = 0$'dır; saf Kağıt'tan kötüdür. Bir karışımın getirisi, karıştırılan hamlelerin getirilerinin ortalamasıdır ve bu yüzden en iyi saf hamlenin getirisini aşamaz. Bu gözlem 0.7.3'te matris diliyle yeniden ele alınır.

Sık yapılan hata

Yalnızca rakibin en sık hamlesine bakmak ("Bora en çok Taş oynuyor, o hâlde Kağıt"). Bora $(0{,}4\ \text{Taş}, 0{,}25\ \text{Kağıt}, 0{,}35\ \text{Makas})$ oynasaydı Taş'ı yenen Kağıt $0{,}4 - 0{,}35 = 0{,}05$ getirirdi. Taş ise $-0{,}25 + 0{,}35 = 0{,}10$ getirir; Makas $-0{,}4 + 0{,}25 = -0{,}15$'tir. En iyi hamle Taş'tır, çünkü Makas oynanma olasılığı da yüksektir ve Taş Makas'ı yener. Her satırın beklenen değeri hesaplanmadan karar verilmemelidir.

0.6.3 Üstel oranlar: logit ve softmax

Beklenen kazançlar hesaplandıktan sonra en iyiyi seçmek (argmax) keskin bir kuraldır: küçük bir fark bile seçimi tamamen değiştirir. Gerçek insanlar ve öğrenen algoritmalar çoğu zaman daha yumuşak davranır: iyi seçeneği daha sık, kötü seçeneği daha seyrek seçer, ama kötüyü hiç dışlamaz. Bu davranışı modelleyen kurala **softmax** denir.

Tanım 0.20

Değerleri $v_1, \ldots, v_k$ olan seçeneklere softmax kuralı şu olasılıkları verir:

$$P(i) = e^{\lambda v_i} / \Sigma_j e^{\lambda v_j}$$

$\lambda \geq 0$ **duyarlılıktır**. İki seçenekte kurala **logit** (*logit*) denir ve şu biçimde yazılır:

$$P(1) = 1 / (1 + e^{-\lambda(v_1 - v_2)})$$

Kuralın yapısı basittir. Her seçeneğin değeri önce λ ile çarpılır, sonra e'nin üssüne konur. Bu üslü sayılar her zaman pozitiftir. Son olarak her biri toplama bölünür; böylece sonuçlar 0 ile 1 arasında kalır ve toplamları 1 olur. Değeri büyük olan seçenek daha büyük bir üslü sayı, dolayısıyla daha büyük bir olasılık alır.

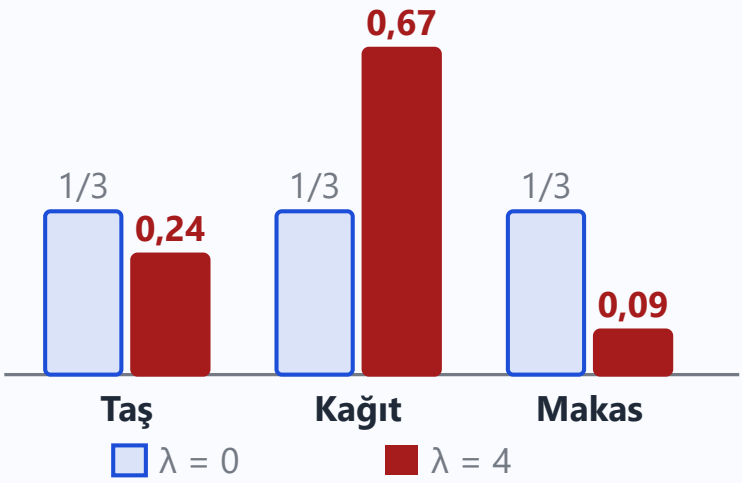
İki seçenekli hâlin softmax'tan nasıl çıktığı şöyle görülür: $P(1) = e^{\lambda v_1}/(e^{\lambda v_1} + e^{\lambda v_2})$'dir. Pay ve payda $e^{\lambda v_1}$'e bölünür. Pay 1 olur; paydada $1 + e^{\lambda v_2}/e^{\lambda v_1} = 1 + e^{-\lambda(v_1 - v_2)}$ kalır (üs kuralı: $e^a/e^b = e^{a-b}$). Bu formül, olasılığın yalnız *değer farkına* bağlı olduğunu açıkça gösterir.

λ'nın iki uç durumu kuralın anlamını verir. $\lambda = 0$ iken bütün üsler $e^0 = 1$ olur ve her seçenek eşit olasılık alır: değerler hiç önemsenmez. λ büyüdükçe en büyük değerin üslü sayısı ötekilere göre çok büyür ve olasılık argmax'ta toplanır. Softmax, rastgele seçim ile argmax arasında ayarlanabilir bir köprüdür.

Örnek 0.24

Beklenen kazançlar 0.6.2'deki gibi Taş 0, Kağıt 1/4, Makas −1/4'tür ve $\lambda = 4$'tür.

1. Üsler $\lambda \cdot v$: $4 \cdot 0 = 0$, $4 \cdot 1/4 = 1$, $4 \cdot (-1/4) = -1$.
2. Üslü sayılar: $e^0 = 1$, $e^1 \approx 2{,}718$, $e^{-1} \approx 0{,}368$.
3. Toplam: $1 + 2{,}718 + 0{,}368 \approx 4{,}086$.
4. Olasılıklar: $1/4{,}086 \approx 0{,}24$, $2{,}718/4{,}086 \approx 0{,}67$, $0{,}368/4{,}086 \approx 0{,}09$. Sonuç $P \approx (0{,}24; 0{,}67; 0{,}09)$.
5. Denetim: $0{,}24 + 0{,}67 + 0{,}09 = 1$ ✓. En iyi hamle Kağıt en sık, en kötü hamle Makas en seyrek seçilir; hiçbiri sıfır değildir.



Şekil 0.25: Softmax olasılıkları. $\lambda = 0$'da (açık çubuklar) her hamle 1/3; $\lambda = 4$'te (koyu çubuklar) Taş 0,24, Kağıt 0,67, Makas 0,09.

Şekil 0.25 duyarlılığın etkisini gösterir. λ 4'ten 40'a çıkarılsaydı üsler 0, 10 ve −10 olurdu ve Kağıt'ın olasılığı yaklaşık 0,99995'e çıkardı: kural neredeyse argmax'a dönüşürdü. Softmax, davranışsal oyun kuramında "küçük hatalar yapan oyuncuları", makine öğrenmesinde ise keşif ile sömürü arasındaki dengeyi modellemek için kullanılır. Oyuncuların birbirine softmax ile tepki verdiği durağan durumlara kuantal tepki dengesi (*quantal response equilibrium*) denir (McKelvey ve Palfrey, 1995).

Sık yapılan hata

Değerleri doğrudan toplamlarına bölmek: $v_i / \Sigma\, v_j$. Değerler sıfır ya da negatif olabilir. Örnekte $0 + 1/4 - 1/4 = 0$'dır ve sıfıra bölünemez; Makas'ın olasılığı da negatif çıkardı. e^x her zaman pozitif olduğu için kullanılır. Ayrıca $e^a / e^b = e^{a-b}$ olduğundan bütün değerlere aynı sayıyı eklemek olasılıkları değiştirmez: değerler 0, 1/4, −1/4 yerine 10, 10,25, 9,75 olsaydı softmax aynı sonucu verirdi.

0.7 Kazanç matrisi

İki oyunculu bir oyunun bütün bilgisi tek bir tabloya sığar: satırlar bir oyuncunun, sütunlar ötekinin seçenekleridir; her hücreye o profilde iki oyuncunun kazancı yazılır. Bu tabloya **kazanç matrisi** (*payoff matrix*) denir. Matrisi doğru okumak, oyun kuramının en sık kullanılan becerisidir. Karışık stratejilerde beklenen kazanç da matris çarpımıyla yazılır.

0.7.1 Kazanç matrisini okumak

Tanım 0.21

İki oyunculu bir oyunun kazanç matrisinde satırları **satır oyuncusu**, sütunları **sütun oyuncusu** seçer. İki seçimin kesiştiği hücre oyunun sonucudur. Hücredeki *(a, b)* çiftinde *a* satır oyuncusunun, *b* sütun oyuncusunun kazancıdır. Her hücrede $a + b = 0$ ise oyun sıfır toplamlıdır.

Tanımda iki ayrı bilgi vardır. Birincisi *kimin neyi seçtiğidir*: satır oyuncusu yalnız satırı, sütun oyuncusu yalnız sütunu seçer. Hiçbir oyuncu hücreyi tek başına seçemez. İkincisi *hücredeki sayıların kime ait olduğudur*: ilk sayı satır oyuncusunun, ikinci sayı sütun oyuncusunun. Bu sıra Kartezyen çarpımdaki sırayla (0.2.1) aynıdır: profil (satır seçimi, sütun seçimi), kazanç (satır kazancı, sütun kazancı).

Örnek 0.25

Kart Gösterme. Ayşe elinde kırmızı ya da siyah bir kart tutar; Bora kartın rengini tahmin eder. Bora kırmızıyı bilirse Ayşe'den 3 jeton, siyahı bilirse 1 jeton alır; yanılırsa Ayşe'ye 1 jeton verir.

1. Ayşe satır oyuncusudur (Kırmızı, Siyah); Bora sütun oyuncusudur (“Kırmızı”, “Siyah” tahmini).
2. (Kırmızı, Kırmızı): Ayşe kırmızı tuttu, Bora bildi. Ayşe 3 verir, Bora 3 alır: *(−3, 3)*.
3. (Siyah, Siyah): Bora siyahı bildi; Ayşe 1 verir: *(−1, 1)*.
4. (Kırmızı, Siyah) ve (Siyah, Kırmızı): Bora yanıldı; Ayşe 1 alır: *(1, −1)*.
5. Her hücrede $a + b = 0$'dır: oyun sıfır toplamlıdır.



Şekil 0.26: Kart Gösterme'nin kazanç matrisi. Satırlar Ayşe'nin kartı, sütunlar Bora'nın tahminidir. Turuncu hücre (Kırmızı, Kırmızı) = (−3, 3): Ayşe 3 verir, Bora 3 alır.

Sıfır toplamlı oyunlarda ikinci sayı her zaman birincinin eksi işaretlisi olduğu için çoğu zaman yalnız satır oyuncusunun kazançları yazılır. Kart Gösterme'de bu, $A = [[-3, 1], [1, -1]]$ matrisidir: ilk satır Kırmızı, ikinci satır Siyah; ilk sütun "Kırmızı", ikinci sütun "Siyah" tahminidir. Bu kısa yazım 0.7.3'te kullanılır.

Sık yapılan hata

b sayısını satır oyuncusunun kazancı sanmak ya da Ayşe'nin sonucu tek başına seçtiğini düşünmek. Ayşe yalnızca satırı seçer; hangi hücreye düşüleceğini Bora'nın sütunu belirler. "Ayşe en çok 1 alabilir, o hâlde 1'i seçer" cümlesi yanlıştır: 1'i almak için Bora'nın yanılması gerekir ve bu Bora'nın elindedir.

0.7.2 Matrisi okuma alıştırması

Okuma kuralları birkaç soruyla pekiştirilir. Sorular Kart Gösterme matrisi üzerindedir (Şekil 0.26). Her sorunun cevabında hangi satıra ya da sütuna, hangi sayıya bakıldığı ayrıca belirtilmiştir.

Örnek 0.26

1. **Ayşe Siyah tutar, Bora "Kırmızı" derse kim kaç jeton alır?** Siyah satırı ile "Kırmızı" sütununun kesiştiği hücreye bakılır: *(1, −1)*. Bora yanılmıştır; Ayşe 1 jeton alır, Bora 1 verir.
2. **Ayşe Siyah tutarken Bora'nın alabileceği kazançlar hangileri?** Siyah satırındaki ikinci sayılara bakılır: −1 ("Kırmızı" derse) ve 1 ("Siyah" derse).
3. **Bora her turda "Kırmızı" diyorsa Ayşe hangi kartı tutmalı?** "Kırmızı" sütunundaki ilk sayılar karşılaştırılır: Kırmızı −3, Siyah 1. $1 > -3$ olduğu için Siyah.
4. **Oyunda Ayşe'nin alabileceği en büyük ve en küçük kazanç kaç?** Bütün ilk sayılar: −3, 1, 1, −1. En büyük 1, en küçük −3.

Üçüncü sorunun yöntemi özellikle önemlidir: rakibin seçimi sabitlenir (bir sütun seçilir), oyuncu kendi sayılarını (ilk sayıları) karşılaştırır ve en büyüğünü seçer. Bu, en iyi tepkinin tanımıdır (0.8.1). Dördüncü soru ise Ayşe'nin oyunda neler yaşayabileceğinin sınırlarını verir; Ayşe'nin bu sınırlar arasında hangi sonucu alacağı Bora'nın seçimine bağlıdır.

0.7.3 Beklenen kazanç bir matris–vektör çarpımıdır

İki oyuncu da karışık strateji oynadığında beklenen kazancı hesaplamak için her hücrenin kazancı, o hücreye düşülme olasılığıyla çarpılıp toplanır. Oyunlar büyüdükçe bu hesap kalabalıklaşır. Matris gösterimi hesabı iki düzenli adıma ayırır ve bilgisayarda tek bir işlemle yapılmasını sağlar.

Tanım 0.22

Karışık strateji bir **olasılık vektörüdür**: $x = (x_1, \ldots, x_m)$, her $x_i \geq 0$ ve $\Sigma\, x_i = 1$. A satır oyuncusunun $m \times n$ kazanç matrisi, y sütun oyuncusunun olasılık vektörü olsun.

- Ay: A'nın her satırı y ile çarpılıp toplanır; çıkan m sayının her biri o satırın y'ye karşı beklenen kazancıdır: $(Ay)_i = \Sigma_j\, A_{ij}\, y_j$.
- $x^{\mathrm{T}}Ay$: bu satır beklentileri x'in olasılıklarıyla tartılıp toplanır: $x^{\mathrm{T}}Ay = \Sigma_i\, x_i\, (Ay)_i$. Bu, satır oyuncusunun beklenen kazancıdır.

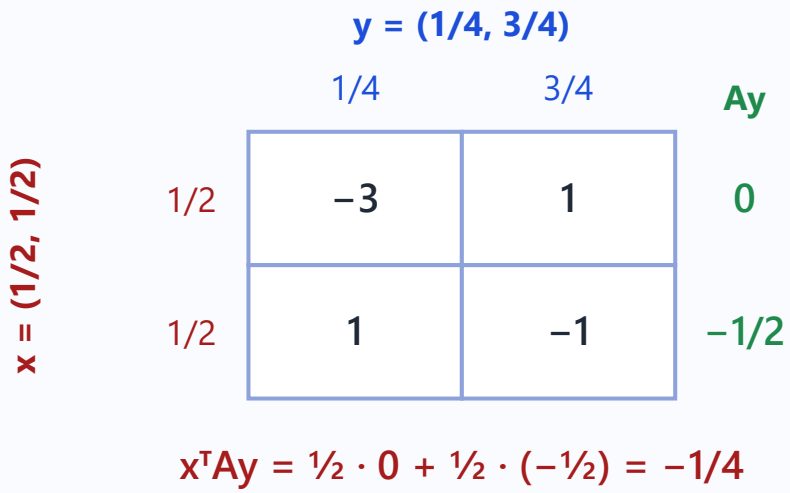
$^{\mathrm{T}}$ işareti, x'in satır olarak yazıldığını gösterir.

Bu iki adım, 0.6.2'deki hesabın aynısıdır. Orada önce her satırın beklenen değeri bulunmuştu (Tablo 0.4'ün son sütunu); bu Ay'dir. Sonra Ayşe'nin karışımı bu sayıları tartmıştı; bu $x^{\mathrm{T}}Ay$'dir. Matris gösterimi yeni bir şey öğretmez, yalnız aynı hesabı her boyuttaki oyun için aynı kısa yazımla ifade eder.

Örnek 0.27

Kart Gösterme'de Ayşe'nin kazanç matrisi $A = [[-3, 1], [1, -1]]$'dir. Ayşe $x = (1/2, 1/2)$, Bora $y = (1/4, 3/4)$ oynuyor.

1. **Birinci satır (Kırmızı).** $-3 \cdot 1/4 + 1 \cdot 3/4 = -3/4 + 3/4 = 0$.
2. **İkinci satır (Siyah).** $1 \cdot 1/4 + (-1) \cdot 3/4 = 1/4 - 3/4 = -1/2$.
3. Böylece $Ay = (0, -1/2)$.
4. **Tartma.** $x^{\mathrm{T}}Ay = 1/2 \cdot 0 + 1/2 \cdot (-1/2) = -1/4$. Hesabın düzeni Şekil 0.27'de çizilmiştir.
5. **Yorum.** Ayşe tur başına ortalama 1/4 jeton kaybeder; Bora (sıfır toplamlı oyun) 1/4 kazanır. Ay'nin en büyük elemanı 0'dır ve Kırmızı satırına aittir: Ayşe bu y'ye karşı hep Kırmızı tutsaydı kaybı 0'a inerdi.



Şekil 0.27: x^TAy hesabı. Sütun olasılıkları $y = (1/4, 3/4)$ üstte, satır olasılıkları $x = (1/2, 1/2)$ soldadır. Sağdaki sütun $Ay = (0, -1/2)$; sonuç $\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot (-\frac{1}{2}) = -1/4$.

Örneğin son adımı genel bir kuralı gösterir: x^TAy, Ay'nin elemanlarının ağırlıklı ortalamasıdır; ağırlıklı ortalama da en büyük elemanı aşamaz. Bu yüzden bir oyuncu, rakibin stratejisi biliniyorsa saf bir en iyi tepkiyle her karışım kadar iyi sonuç alabilir. Karışık oynamanın amacı rakibe karşı daha çok kazanmak değil, rakibin bizi sömürmesini önlemektir. Bu fark kayıtsızlık fikrinin özüdür (0.9.3).

Sık yapılan hata

Ay'yi bulup orada durmak ya da x ile y'yi ters yerde kullanmak. y sütunları tartar (Ay: her satırın beklentisi); x de bu satır beklentilerini tartar. Ters yapılırsa (satırlar y ile tartılırsa) boyutlar uyuşmaz ya da yanlış sayılar çıkar. Ay'nin en büyük elemanı, satır oyuncusunun y'ye karşı en iyi saf tepkisini verir.

0.8 En iyi tepki

Önceki kesimlerin bütün araçları tek bir kavramda birleşir: en iyi tepki. Kazanç fonksiyonu, argmax, beklenen değer ve matris okuma bu kavramın parçalarıdır. En iyi tepki, oyun kuramının en temel kavramlarından biri olan dengenin de yapı taşıdır.

0.8.1 En iyi tepki

“Rakibin ne yapacağını bilseydim ne yapardım?” sorusunun cevabı en iyi tepkidir. Soru bir varsayım içerir: rakibin seçimi bilinir ve değişmez. Bu varsayım altında oyun, tek kişilik bir karar problemine dönüşür ve cevap bir argmax'tır.

Tanım 0.23

Rakip s_2 oynarken oyuncu 1'in **en iyi tepkisi** (*best response*), kendi kazancını en büyük yapan stratejilerin kümesidir:

$$BR_1(s_2) = \operatorname{argmax}_{s_1 \in S_1} u_1(s_1, s_2)$$

Rakip karışık bir y oynuyorsa u_1 yerine beklenen kazanç kullanılır; matris diliyle en iyi saf tepki, Ay'nin en büyük elemanının satırıdır.

Tanımın üç parçası vardır. *Rakibin seçimi sabitlenir*: matriste bir sütun seçilir. *Yalnız kendi kazançlarına bakılır*: o sütunda ilk sayılar okunur. *Argmax alınır*: en büyük sayının satırı seçilir. Eşitlik varsa birden çok en iyi tepki vardır ve hepsi kümeye girer.

Örnek 0.28

Kart Gösterme'de Ayşe'nin en iyi tepkileri bulunur.

1. **Bora hep "Kırmızı" der.** "Kırmızı" sütunundaki ilk sayılar: Kırmızı −3, Siyah 1. $BR_1(\text{"Kırmızı"}) = \text{Siyah}$ (Şekil 0.28).
2. **Bora hep "Siyah" der.** "Siyah" sütunundaki ilk sayılar: Kırmızı 1, Siyah −1. $BR_1(\text{"Siyah"}) = \text{Kırmızı}$.
3. **Bora karıştırıyor.** Bora "Kırmızı"yı $q = 1/2$ olasılıkla desin. Kırmızı satırı: $-3 \cdot 1/2 + 1 \cdot 1/2 = -1$. Siyah satırı: $1 \cdot 1/2 + (-1) \cdot 1/2 = 0$. $0 > -1$, en iyi tepki Siyah'tır.
4. **Başka bir karışım.** $y = (1/4, 3/4)$ için 0.7.3'te $Ay = (0, -1/2)$ bulunmuştu: en iyi tepki Kırmızı'dır.



Şekil 0.28: Bora "Kırmızı" derken Ayşe'nin en iyi tepkisi. Ayşe "Kırmızı" sütunundaki ilk sayıları karşılaştırır: −3 ve 1. Daire içindeki 1, Siyah'ın en iyi tepki olduğunu gösterir.

Son iki adım ilginç bir gözlem verir: Bora'nın karışımı değişince Ayşe'nin en iyi tepkisi de değişti. $q = 1/2$'de Siyah, $q = 1/4$'te Kırmızı en iyidir. Arada bir yerde iki hamlenin eşit getirdiği bir q olmalıdır. Bu değer 0.9.3'te bulunur ve karışık stratejilerin nasıl hesaplandığının anahtarıdır.

Sık yapılan hata

En iyi tepkiyi seçerken rakibin kazancına (ikinci sayıya) bakmak. Ayşe yalnızca kendi sayılarını karşılaştırır; Bora'nın ne kazandığı Ayşe'nin kararını doğrudan etkilemez. Ayrıca en iyi tepki sabit değildir: rakip değişince değişebilir. "Ayşe'nin en iyi hamlesi Siyah'tır" cümlesi eksiktir; doğrusu "Bora 'Kırmızı' derken Ayşe'nin en iyi tepkisi Siyah'tır" biçimindedir.

0.8.2 İzleme: 2×2 matriste en iyi tepkiler

En iyi tepkiler bütün matris için bir kerede işaretlenebilir. Yöntem iki turdan oluşur: önce satır oyuncusunun her sütundaki en iyi tepkisi, sonra sütun oyuncusunun her satırdaki en iyi tepkisi işaretlenir. Tablo 0.5'teki oyunda Ayşe satırı (Yukarı/Aşağı), Bora sütunu (Sol/Sağ) seçer.

Ayşe \ Bora	Sol	Sağ
Yukarı	(1, 2)	(3, 1)
Aşağı	**(2, 3)**	(0, 0)

Tablo 0.5: İzleme oyunu. Hücrede (Ayşe, Bora) kazançları. Yeşil hücrede iki oyuncu da en iyi tepkisini oynar.

Örnek 0.29

1. **Bora Sol oynarsa** Ayşe Sol sütunundaki ilk sayıları karşılaştırır: Yukarı 1, Aşağı 2. $2 > 1 \Rightarrow$ Aşağı. (Aşağı, Sol) hücresindeki 2 işaretlenir.
2. **Bora Sağ oynarsa:** Yukarı 3, Aşağı 0 ⇒ Yukarı. (Yukarı, Sağ) hücresindeki 3 işaretlenir.
3. **Ayşe Yukarı oynarsa** Bora Yukarı satırındaki ikinci sayıları karşılaştırır: Sol 2, Sağ 1 ⇒ Sol. (Yukarı, Sol)'daki 2 işaretlenir.
4. **Ayşe Aşağı oynarsa:** Sol 3, Sağ 0 ⇒ Sol. (Aşağı, Sol)'daki 3 işaretlenir. Bora için Sol her durumda daha iyidir.
5. **Sonuç.** İki sayısı da işaretli tek hücre (Aşağı, Sol) = (2, 3)'tür. Bu hücrede iki oyuncu da en iyi tepkisini oynar; kimse tek başına değiştirmek istemez. İşaretlemenin son hâli Şekil 0.29'da görülür.



Şekil 0.29: İşaretlemenin son hâli. Daire içindeki sayılar en iyi tepkilerdir. İki sayısı da işaretli tek hücre (Aşağı, Sol) = (2, 3)'tür (yeşil çerçeve).

İzlemede bulunan özelliğe, yani her oyuncunun ötekinin seçimine karşı en iyi tepkisini oynamasına, **Nash dengesi** denir. Denge, kimsenin tek başına sapmak istemediği bir durumdur. İşaretleme yöntemi, iki oyunculu matris oyunlarında bütün saf dengeleri bulmanın en kısa yoludur: her sütunda satır oyuncusunun, her satırda sütun oyuncusunun en büyük sayısı işaretlenir; iki işaretli hücreler dengedir.

Yöntemi Kart Gösterme'ye uygulamak öğreticidir. Bora “Kırmızı” derken Ayşe'nin en iyisi Siyah'tır; Ayşe Siyah tutarken Bora'nın en iyisi “Siyah” demesidir; Bora “Siyah” derken Ayşe'nin en iyisi Kırmızı'dır; Ayşe Kırmızı tutarken Bora'nın en iyisi “Kırmızı”dır. En iyi tepkiler bir döngü kurar ve hiçbir hücrede iki işaret buluşmaz. Böyle oyunlarda saf strateji dengesi yoktur; oyuncular karıştırmalıdır. Karışımın olasılıkları kayıtsızlık denklemiyle bulunur.

Sık yapılan hata

Satır oyuncusunun en iyi tepkisini satır boyunca aramak. Satır oyuncusu yalnız satırını değiştirebilir; bu yüzden onun karşılaştırmaları *bir sütunun içinde*, yukarıdan aşağıya yapılır. Sütun oyuncusunun karşılaştırmaları da *bir satırın içinde*, soldan sağa yapılır. İkinci hata, tek işaretli bir hücreyi denge saymaktır: (Yukarı, Sağ)'da Ayşe'nin 3'ü işaretlidir, ama Bora Sol'a geçmek ister.

0.9 Denklemler ve kayıtsızlık

Oyun kuramında bilinmeyenler çoğu zaman olasılıklardır: bir oyuncu bir hamleyi hangi olasılıkla oynamalıdır? Bu sorular denklemlere dönüşür. Çoğu doğrusaldır (birinci derece), bazıları ikinci derecedir. Bu iki denklem türünün çözümü, oyun kuramının en çok kullanılan denklemine, kayıtsızlık denklemine zemin hazırlar. Girişteki iki sorunun cevabı da bu denkleme dayanır.

0.9.1 İkinci derece denklem ve diskriminant

Bilinmeyenin karesinin geçtiği denkleme ikinci derece denklem denir. Grafiği bir paraboldür ve kökler, parabolün yatay ekseni kestiği noktalardır. Parabol ekseni iki noktada kesebilir, ekseni yalnız bir noktada teğet geçebilir ya da eksene hiç değmeyebilir. Hangisinin olduğunu tek bir sayı, diskriminant söyler.

Tanım 0.24

$ax^2 + bx + c = 0$ ($a \neq 0$) denkleminin **diskriminantı** (*discriminant*) $\Delta = b^2 - 4ac$'dir.

- $\Delta > 0$ ise iki farklı gerçek kök,
- $\Delta = 0$ ise tek (çakışık) kök,
- $\Delta < 0$ ise gerçek kök yoktur.

Kökler şu formülle bulunur:

$$x = (-b \pm \sqrt{\Delta}) / (2a)$$

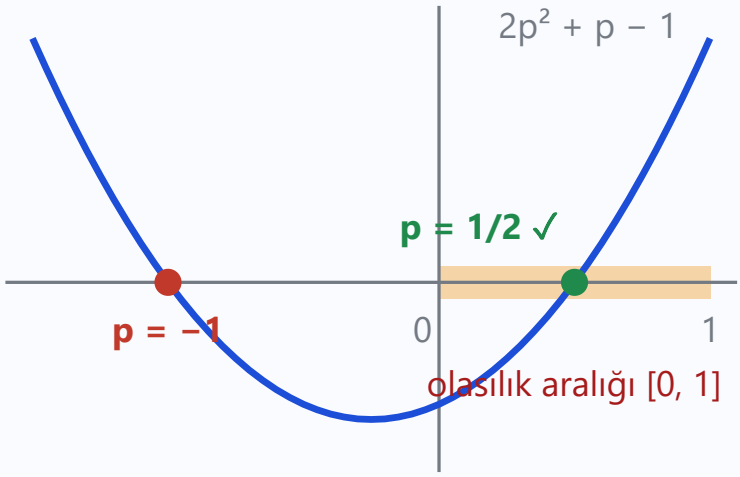
Formüldeki "±" işareti iki ayrı hesabı kısaltır: biri $+\sqrt{\Delta}$ ile, öteki $-\sqrt{\Delta}$ ile yapılır. $\Delta = 0$ iken iki hesap aynı sonucu verir; $\Delta < 0$ iken negatif bir sayının karekökü gerçek sayı olmadığı için formül uygulanamaz.

Oyun kuramında bilinmeyen çoğu zaman bir olasılıktır. Bu durumda denklemin iki kökünden yalnız 0 ile 1 arasında olanı anlamlıdır. Öteki kök matematik olarak doğrudur ama olasılık olamaz. Her çözümün sonunda bu denetim yapılmalıdır.

Örnek 0.30

Bir olasılık p için $2p^2 + p - 1 = 0$ denklemi bulunmuş olsun.

1. Katsayılar okunur: $a = 2$, $b = 1$, $c = -1$.
2. Diskriminant: $\Delta = 1^2 - 4 \cdot 2 \cdot (-1) = 1 + 8 = 9$. $\Delta > 0$ olduğu için iki kök vardır; $\sqrt{\Delta} = 3$.
3. Kökler: $p = (-1 \pm 3)/(2 \cdot 2) = (-1 \pm 3)/4$. Artı işaretiyle $2/4 = 1/2$, eksi işaretiyle $-4/4 = -1$.
4. Olasılık 0 ile 1 arasında olmalıdır: $p = -1$ atılır, $p = 1/2$ kalır (Şekil 0.30).
5. Denetim: $2 \cdot (1/2)^2 + 1/2 - 1 = 2 \cdot 1/4 + 1/2 - 1 = 1/2 + 1/2 - 1 = 0$ ✓.



Şekil 0.30: $2p^2 + p - 1$ parabolü ekseni $p = -1$ ve $p = 1/2$'de keser. Turuncu şerit olasılık aralığı [0, 1]'dir; bu aralıkta yalnız $p = 1/2$ vardır.

Diskriminantın öteki iki durumu da birer örnekle görülebilir. $p^2 - p + 1/4 = 0$ için $\Delta = 1 - 4 \cdot 1 \cdot 1/4 = 0$'dır; tek kök $p = 1/(2 \cdot 1) = 1/2$'dir ve parabol ekseni bu noktada yalnız teğet geçer. $p^2 + p + 1 = 0$ için $\Delta = 1 - 4 = -3 < 0$'dır; parabol eksene hiç değmez ve denklemin gerçek kökü yoktur.

Sık yapılan hata

$-b$ yerine b yazmak ya da yalnız $\sqrt{\Delta}$'yı $2a$'ya bölmek. Kesir çizgisi $(-b \pm \sqrt{\Delta})$'nın tamamının altındadır. c negatifken $-4ac$'nin işaretini kaçırmak da sık görülür: örnekte $-4 \cdot 2 \cdot (-1) = +8$'dir; işaret kaçırılırsa $\Delta = 1 - 8 = -7$ bulunur ve "kök yok" gibi yanlış bir sonuca varılır.

0.9.2 İki bilinmeyenli doğrusal denklem

İki bilinmeyeni bulmak için iki denklem gerekir. Tek bir denklem, sonsuz sayıda çözüm çifti verir; ikinci denklem bunlardan birini seçer. Çözümün fikri, bilinmeyenlerden birini yok etmektir: ya iki denklem birbirinden çıkarılır ya da birinden bir bilinmeyen çekilip ötekine konur. Kalan tek bilinmeyenli denklem çözülür ve bulunan değer geri yerine konur.

Örnek 0.31

Bir oyun dükkânında 2 zar ve 1 deste kart 50 TL, 1 zar ve 1 deste 35 TL'dir. Bir zar ve bir deste kaçar TL'dir? Zarın fiyatı z, destenin fiyatı d olsun.

1. Denklemler: (1) $2z + d = 50$ ve (2) $z + d = 35$.
2. (1)'den (2) çıkarılır: $(2z + d) - (z + d) = 50 - 35$. Sol tarafta $d - d = 0$ olur ve $z = 15$ kalır.
3. $z = 15$, (2)'de yerine konur: $15 + d = 35 \Rightarrow d = 20$.

4. Denetim (1): $2 \cdot 15 + 20 = 50$ ✓. Denetim (2): $15 + 20 = 35$ ✓.

Oyun kuramında bu denklemlerin tipik biçimi şudur: iki olasılık x ve y için biri "olasılıkların toplamı 1'dir", öteki de oyundan gelen bir koşuldur. Örneğin $x + y = 1$ ve $x = 2y$ ise ikinci denklem birinciye konur (yerine koyma yöntemi): $2y + y = 1 \Rightarrow 3y = 1 \Rightarrow y = 1/3$ ve $x = 2/3$. Denetim: $2/3 + 1/3 = 1$ ✓ ve $2/3 = 2 \cdot 1/3$ ✓.

Örnek 0.32

Bora Sol'u q, Sağ'ı r olasılığıyla oynuyor. Ayşe'nin kazançları Yukarı'da 3 (Sol) ve 0 (Sağ), Aşağı'da 1 (Sol) ve 2 (Sağ)'dir. Ayşe'nin iki satırının eşit beklenen kazanç vermesi isteniyor.

1. Denklemler: $q + r = 1$ ve $3q + 0 \cdot r = 1 \cdot q + 2r$.
2. İkinci denklem sadeleşir: $3q - q = 2r \Rightarrow 2q = 2r \Rightarrow q = r$.
3. Birinciye konur: $q + q = 1 \Rightarrow q = 1/2$, $r = 1/2$.
4. Denetim: Yukarı $3 \cdot 1/2 = 3/2$; Aşağı $1 \cdot 1/2 + 2 \cdot 1/2 = 3/2$ ✓.

Sık yapılan hata

Çıkarırken bir tarafı unutmak ya da işareti kaybetmek. $(2z + d) - (z + d)$ işleminde parantezin önündeki eksi, içindeki iki terime de uygulanır: $2z + d - z - d$. Sağ taraf da $50 - 35$ olmalıdır. Bulunan değerler iki denklemde de denetlenir; yalnız birinde denetlemek, öteki denklemdeki bir hatayı gizleyebilir. Olasılık sorularında $r = 1 - q$ denklemini yazmayı unutmak, tek denklemle iki bilinmeyen bulmaya çalışmak demektir.

0.9.3 Kayıtsızlık fikri

Karışık strateji oynayan bir oyuncunun karıştırdığı hamleler hakkında ne söylenebilir? Cevap basit bir akıl yürütmeden gelir. Oyuncu iki hamlesini gerçekten karıştırıyorsa, yani ikisine de pozitif olasılık veriyorsa, iki hamle ona eşit beklenen kazanç vermelidir. Biri daha iyi olsaydı oyuncu hep onu oynardı; daha kötü olana olasılık vermesi kazancını boşuna düşürürdü.

Tanım 0.25

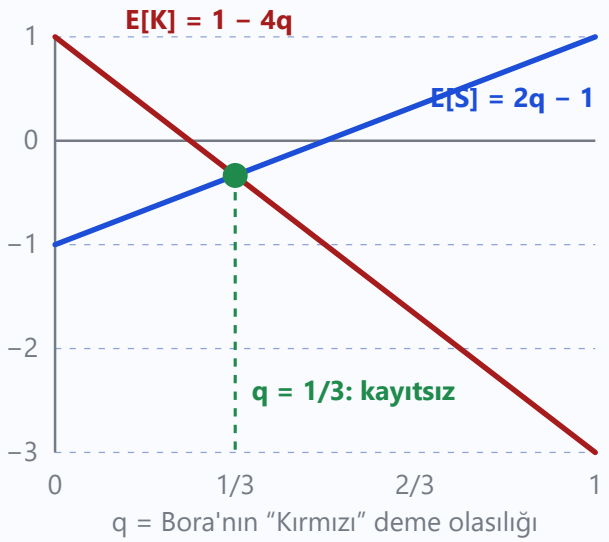
Bir oyuncunun iki eylemi, rakibin stratejisine karşı eşit beklenen kazanç veriyorsa oyuncu bu iki eylem arasında **kayıtsızdır**. **Kayıtsızlık ilkesi** (*indifference principle*): bir oyuncu birden çok eylemi pozitif olasılıkla karıştırıyor ve bu en iyi seçimiyse, karıştırdığı eylemlerin hepsi ona eşit beklenen kazanç verir.

İlkenin şaşırtıcı bir sonucu vardır. Bir oyuncunun kayıtsız olması, *rakibin* olasılıklarına bağlıdır, kendi olasılıklarına değil. Ayşe'nin iki hamlesinin beklenen kazancı, Bora'nın ne sıklıkla ne oynadığıyla hesaplanır. Dolayısıyla "Ayşe kayıtsızdır" koşulu, Bora'nın olasılığı hakkında bir denklemdir. Ayşe'nin kendi olasılığı ise Bora'nın kayıtsızlık koşulundan bulunur.

Örnek 0.33

Kart Gösterme'de Bora "Kırmızı"yı q, "Siyah"ı $1 - q$ olasılıkla desin.

1. Ayşe Kırmızı tutarsa: $E[K] = -3 \cdot q + 1 \cdot (1 - q) = 1 - 4q$.
2. Ayşe Siyah tutarsa: $E[S] = 1 \cdot q + (-1) \cdot (1 - q) = 2q - 1$.
3. $q < 1/3$ iken, örneğin $q = 0$'da, $E[K] = 1 > E[S] = -1$: Kırmızı daha iyidir.
4. $q > 1/3$ iken, örneğin $q = 1$'de, $E[K] = -3 < E[S] = 1$: Siyah daha iyidir.
5. Yalnız $q = 1/3$'te iki doğru kesişir ve Ayşe kayıtsızdır.



Şekil 0.31: Ayşe'nin iki kartının beklenen kazancı, Bora'nın "Kırmızı" deme olasılığı q'ya göre. Kırmızı doğru $1 - 4q$, mavi doğru $2q - 1$'dir; $q = 1/3$'te kesişirler ve Ayşe kayıtsızdır.

Şekil 0.31 fikri görsel olarak özetler. Ayşe'nin en iyi tepkisi her q için üstteki doğrudur: q küçükken Kırmızı, büyükken Siyah. İki doğrunun kesişme noktası, Ayşe'nin karıştırmaya razı olabileceği tek q değeridir. 0.8.1'deki gözlem de burada açıklanır: $q = 1/4$ kesişmenin solundadır ve Kırmızı en iyidir; $q = 1/2$ sağındadır ve Siyah en iyidir.

Sık yapılan hata

Ayşe'nin kendi olasılığını kendi kayıtsızlık denkleminden bulmaya çalışmak. Ayşe'nin kayıtsızlığı Bora'nın q değerini belirler; çünkü $E[K]$ ve $E[S]$'de yalnız q geçer. Ayşe'nin Kırmızı tutma olasılığı p ise p, Bora'yı kayıtsız bırakan denklemden çıkar.

0.9.4 İzleme: kayıtsızlık denklemi

Kayıtsızlık denklemi her zaman aynı dört adımla kurulur ve çözülür. Aşağıdaki izlemede p, Ayşe'nin Kırmızı tutma olasılığı; q, Bora'nın "Kırmızı" deme olasılığıdır. Kullanılan matris Şekil 0.32'de yeniden verilmiştir.



Şekil 0.32: İzlemede kullanılan matris. K ve S Ayşe'nin kartları, "K" ve "S" Bora'nın tahminleridir.

Örnek 0.34

1. **Bilinmeyen.** Bora "Kırmızı"yı q, "Siyah"ı $1 - q$ olasılıkla der. Toplam 1'dir; ikinci bir bilinmeyene gerek yoktur.
2. **Her satırın beklenen kazancı.** Kırmızı: $E[K] = -3 \cdot q + 1 \cdot (1 - q) = 1 - 4q$. Siyah: $E[S] = 1 \cdot q + (-1) \cdot (1 - q) = 2q - 1$.
3. **Eşitleme ve çözüm.** $1 - 4q = 2q - 1 \Rightarrow 1 + 1 = 2q + 4q \Rightarrow 2 = 6q \Rightarrow q = 1/3$.
4. **Denetim.** $E[K] = 1 - 4/3 = -1/3$ ve $E[S] = 2/3 - 1 = -1/3$ ✓. $q = 1/3$ 0 ile 1 arasındadır ✓.

Bora $q = 1/3$ seçince Ayşe hangi kartı tutarsa tutsun aynı beklenen kazancı, $-1/3$'ü alır.

Aynı dört adım Bora için tekrarlanınca Ayşe'nin olasılığı bulunur. Ayşe Kırmızı'yı p olasılıkla tutsun. Bora'nın kazançları Ayşe'ninkilerin eksi işaretlisidir. Bora "Kırmızı" derse $3 \cdot p + (-1) \cdot (1 - p) = 4p - 1$, "Siyah" derse $(-1) \cdot p + 1 \cdot (1 - p) = 1 - 2p$ kazanır. Eşitlik: $4p - 1 = 1 - 2p \Rightarrow 6p = 2 \Rightarrow p = 1/3$. Denetim: iki taraf da $1/3$'tür. Bu oyunda iki oyuncunun olasılığı rastlantıyla aynı çıkmıştır; genel olarak farklıdır.

Sonuçları bir araya getirelim. Ayşe $p = 1/3$ ile Kırmızı, Bora $q = 1/3$ ile "Kırmızı" oynarsa ikisi de ötekinin stratejisine karşı kayıtsızdır; hiçbiri stratejisini değiştirerek kazancını artıramaz. Ayşe tur başına $-1/3$, Bora $+1/3$ alır. Oyun Bora'nın lehinedir. Ayşe'nin Kırmızı'yı yalnız 1/3 olasılıkla tutması da anlamlıdır: Kırmızı yakalanırsa 3 jeton kaybettirir, bu yüzden Ayşe onu seyrek tutar. Bu, 0.8.2'de saf dengesi olmadığı görülen oyunun karışık strateji dengesidir.

0.9.5 Oyuna dönüş: Taş-Kağıt-Makas

Girişteki iki soru artık cevaplanabilir. İkisi de beklenen kazanç ve en iyi tepki hesabıdır.

Yenilmeyen bir oynayış var mı? Evet: $(1/3, 1/3, 1/3)$. Rakip ne oynarsa oynasın beklenen kazanç 0'dır. Örneğin rakip Kağıt oynarsa: Taş −1, Kağıt 0, Makas +1 getirir ve beklenen kazanç $1/3 \cdot (-1) + 1/3 \cdot 0 + 1/3 \cdot (+1) = 0$'dır. Rakip Taş ya da Makas oynadığında da aynı hesap 0 verir; rakip karıştırırsa sonuç bu sıfırların ortalaması, yine 0'dır. Kayıtsızlık açısından bakıldığında, eşit karışım rakibin üç hamlesinin getirisini eşitler. Rakibin sömürebileceği bir taraf kalmaz.

Hep Taş oynayan ne kaybeder? Rakibin en iyi tepkisi hep Kağıt'tır ve her tur kazanır: tur başına -1. Daha hafif bir yanlılık da kaybettirir. $(1/2, 1/4, 1/4)$ oynayana karşı rakibin her hamlesinin getirisi hesaplanır: Kağıt $1/2 \cdot 1 + 1/4 \cdot 0 + 1/4 \cdot (-1) = 1/4$, Taş $0 + 1/4 \cdot (-1) + 1/4 \cdot 1 = 0$, Makas $1/2 \cdot (-1) + 1/4 \cdot 1 + 0 = -1/4$. Rakip Kağıt oynar ve yanlı oyuncu tur başına $-1/4$ alır.

Oyuncunun karışımı (T, K, M)	Rakibin en iyi tepkisi	Oyuncunun beklenen kazancı
(1, 0, 0): hep Taş	Kağıt	−1
(1/2, 1/4, 1/4)	Kağıt	−1/4
(1/3, 1/3, 1/3)	hepsi eşit	0

Tablo 0.6: Karışıma göre rakibin en iyi tepkisi ve karışımı oynayanın beklenen kazancı.

Tablo 0.6 tek bir ilkeyi gösterir: eşit karışımdan her sapma, onu fark eden bir rakibe karşı kaybettirir. Genel olarak (p_T, p_K, p_M) oynayana karşı rakibin Kağıt'ı $p_T - p_M$, Makas'ı $p_K - p_T$, Taş'ı $p_M - p_K$ getirir. Bu üç sayının toplamı sıfırdır; bu yüzden ya hepsi sıfırdır (eşit karışım) ya da en az biri pozitiftir ve rakip onu seçer.

İnsanlar nasıl oynuyor? Laboratuvar deneylerinde her hamlenin sıklığı 1/3'e yakın çıksa da ardışık hamleler bağımsız değildir: kazanan aynı hamleyi tekrarlamaya, kaybeden hamlesini değiştirmeye eğilimlidir (Wang, Xu ve Zhou, 2014). Uzun vadeli sıklıklar eşit karışıma benzese de her tur öncekinden bağımsız seçilmediği için oynayış tahmin edilebilir hâle gelir. Örüntü öğrenen bir rakip tam olarak bunu sömürür: sıklıklara değil, "bir önceki turdan sonra ne geliyor?" sorusuna bakar. Kuramın önerdiği $(1/3, 1/3, 1/3)$ karışımı, ancak her tur bağımsız bir rastgele seçimle uygulanırsa yenilmezdir.

0.10 Oyun ağacı ve büyüme hızı

Matris, oyuncuların aynı anda seçim yaptığı oyunlar içindir. Satrançta, pazarlıkta ya da bir firmanın pazara girip girmemesinde ise oyuncular sırayla hamle yapar ve sonraki oyuncu öncekinin hamlesini görür. Böyle oyunlar bir ağaçla çizilir ve geriye doğru çıkarımla çözülür. Ağaçlar hızla büyüdüğü için büyüme hızını anlatan O ve Θ gösterimi de gerekir.

0.10.1 Oyun ağacını okumak

Tanım 0.26

Sıralı oyunlarda (*sequential game*) **oyun ağacı** (*game tree*) kullanılır:

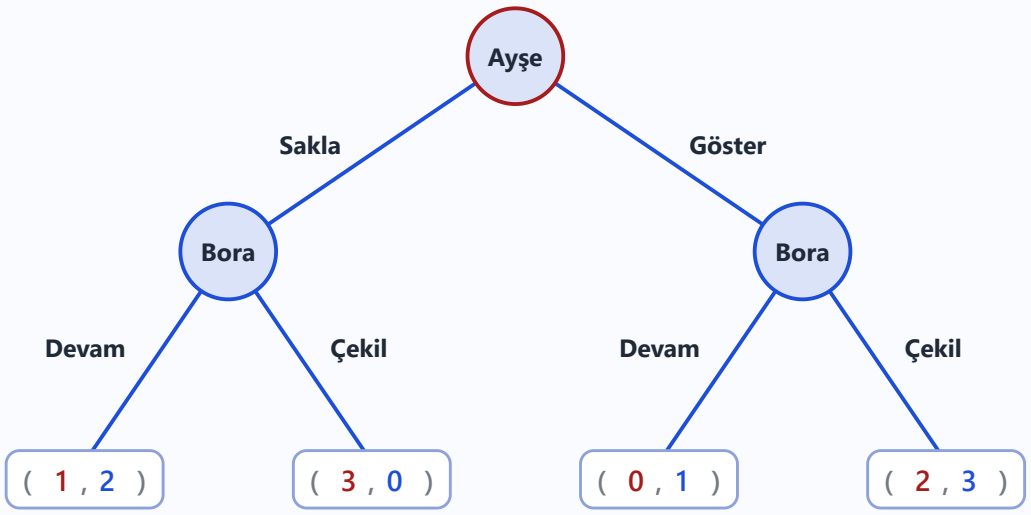
- Daire içindeki **düğüm**, sıranın kimde olduğunu gösterir. En üstteki düğüme **kök** denir; oyun orada başlar.
- Bir düğümden çıkan **dallar**, o oyuncunun o anda seçebileceği eylemlerdir.
- En alttaki **yapraklar**, oyun bittiğinde oyuncuların kazançlarını gösterir; sayılar hamle sırasıyla yazılır.

Ağaçta kökten bir yaprağa giden her yol, oyunun bir oynanışıdır. Yaprak sayısı, oyunun olası sonuçlarının sayısıdır. Matris ile ağaç arasındaki temel fark bilgidir: ağaçta sonraki oyuncu, önceki oyuncunun hangi dalı seçtiğini bilir ve buna göre karar verir. Bu yüzden aynı oyuncunun ağaçta birden çok düğümü olabilir; her düğüm farklı bir geçmişe karşılık gelir ve her birinde ayrı karar verilir.

Örnek 0.35

Ayşe kartını saklar ya da gösterir. Bora, Ayşe'nin ne yaptığını görerek devam eder ya da çekilir (Şekil 0.33).

1. Kök Ayşe'nindir; iki dalı vardır: Sakla ve Göster.
2. Her dalın ucunda Bora'nın bir düğümü vardır. Bora'nın iki düğümü vardır ve her birinde ayrı karar verir: Sakla'dan sonra bir kez, Göster'den sonra bir kez.
3. Dört yaprak vardır. Sakla → Devam: (1, 2); Sakla → Çekil: (3, 0); Göster → Devam: (0, 1); Göster → Çekil: (2, 3).
4. Örneğin Sakla → Çekil yolu (3, 0) yaprağına varır: Ayşe 3, Bora 0 alır.



Şekil 0.33: Kart oyununun ağacı. Kökte Ayşe (kırmızı) Sakla ya da Göster der; iki düğümde Bora (mavi) Devam ya da Çekil der. Yapraklarda (Ayşe, Bora) kazançları yazılıdır.

Sık yapılan hata

Yapraktaki sırayı karıştırmak: ilk sayı ilk hamleyi yapan oyuncunun (Ayşe), ikinci sayı Bora'nın kazancıdır. Bir de matristen farkı unutmak: burada Bora, Ayşe'nin hamlesini görür. “Bora Devam der” cümlesi eksiktir; Bora Sakla'dan sonra ve Göster'den sonra farklı şeyler söyleyebilir.

0.10.2 Geriye doğru çıkarım

Sıralı bir oyunu çözmek için kökten değil, yapraklardan başlanır. Fikir şudur: son hamleyi yapan oyuncunun kararı kolaydır, çünkü ondan sonra kimse hamle yapmaz; o yalnız kendi kazancını karşılaştırır. Onun kararı bilinince bir önceki oyuncu, her dalın sonunda ne olacağını bilerek karar verebilir. Bu, köke ulaşılana kadar tekrarlanır.

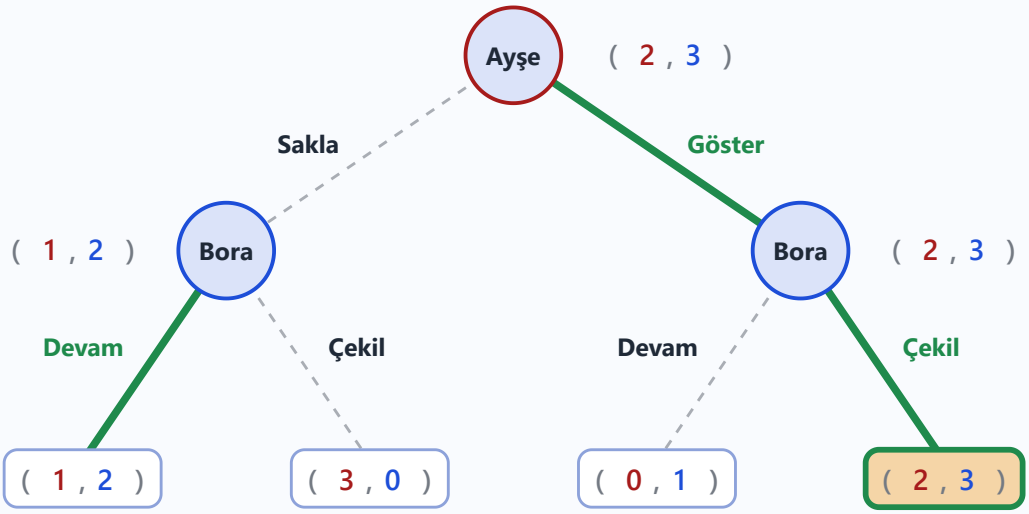
Tanım 0.27

Geriye doğru çıkarım (*backward induction*): yapraklara en yakın düğümlerden başlanır. Her düğümde sırası gelen oyuncu, dallarının ucundaki kazanç çiftlerinden *kendi* kazancı en büyük olanı seçer; düğüm o çiftle değiştirilir. İşlem köke ulaşılınca biter. Kökte bulunan çift oyunun sonucudur; her düğümdeki seçimler birlikte oyunun çözümünü verir.

Örnek 0.36

1. **Başlangıç: son hamle.** Bora'nın iki düğümü yapraklara en yakın düğümlerdir.

2. **Sakla'dan sonra** Bora ikinci sayıları karşılaştırır: Devam 2, Çekil 0 ⇒ Devam. Bu düğümün değeri (1, 2) olur.
3. **Göster'den sonra** Bora: Devam 1, Çekil 3 ⇒ Çekil. Bu düğümün değeri (2, 3) olur.
4. **Kök.** Ayşe, Bora'nın cevaplarını bilerek ilk sayıları karşılaştırır: Sakla → 1, Göster → 2 ⇒ Göster.
5. **Sonuç.** Oynanan yol Göster → Çekil'dir; kazançlar (2, 3). Tam çözüm: Ayşe Göster; Bora Sakla'dan sonra Devam, Göster'den sonra Çekil (Şekil 0.34).



Şekil 0.34: Geriye doğru çıkarımın sonucu. Yeşil dallar her düğümdeki seçimdir; kesikli dallar seçilmeyenlerdir. Sonuç yolu Göster → Çekil, kazanç (2, 3).

Sonuç ilk bakışta şaşırtıcı olabilir: (3, 0) yaprağı Ayşe için en büyük sayıdır, ama oyun oraya hiç varmaz. Ayşe Sakla derse Bora Çekil'i değil Devam'ı seçer, çünkü Devam ona 2, Çekil 0 verir. Ayşe bunu önceden hesaplar ve 1 yerine 2 aldığı Göster'i seçer. Geriye doğru çıkarımın özü budur: bir oyuncu kendi en iyi yaprağına değil, ötekilerin en iyi tepkileri altında ulaşılabilen en iyi sonuca bakar.

Çözümün tam yazımında Bora'nın *her* düğümdeki seçimi verilir, oynanmayan düğümünkü de. Bunun nedeni, Ayşe'nin kararının Bora'nın oynanmayan düğümdeki tepkisine dayanmasıdır: Ayşe'nin Sakla'yı reddetmesinin gerekçesi, Bora'nın Sakla'dan sonra Devam diyeceğidir. Bir stratejinin her durum için ne yapılacağını söyleyen eksiksiz bir plan olması, sıralı oyunlarda bu yüzden önemlidir.

Sık yapılan hata

Kökten başlamak: "Ayşe en büyük sayıyı, 3'ü görür ve Sakla der." Bu akıl yürütme, Bora'nın Ayşe'ye yardım edeceğini varsayar. Doğru yöntem yapraktan köke ilerler: önce son hamle çözülür. İkinci hata, her düğümde yanlış oyuncunun sayısına bakmaktır: Bora'nın düğümünde ikinci sayılar, Ayşe'nin düğümünde ilk sayılar karşılaştırılır.

0.10.3 Büyüme hızı: O ve Θ

Geriye doğru çıkarım her düğümü bir kez ziyaret eder. Küçük ağaçlarda bu hızlıdır; ama satranç gibi bir oyunun ağacında düğüm sayısı evrendeki atom sayısını aşar. Bir yöntemin pratikte kullanılıp kullanılamayacağını anlamak için adım sayısının girdi büyüklüğüyle *nasıl büyüdüğüne* bakılır. Bunun dili O ve Θ gösterimidir.

Tanım 0.28

- $f(n) = O(g(n))$: büyük n için f, g'nin bir sabit katını aşmaz. Biçimsel olarak öyle bir $c > 0$ ve n_0 vardır ki her $n \geq n_0$ için $f(n) \leq c \cdot g(n)$'dir. Bu bir *üst sınırdır*.
- $f(n) = \Theta(g(n))$: f, g ile aynı hızda büyür; büyük n için alttan ve üstten g'nin sabit katları arasında kalır: $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$.

Sabit çarpanlar ve daha yavaş büyüyen terimler atılır: $3n^2 + 5n = \Theta(n^2)$.

Son örnek tanımla denetlenebilir. Alt sınır kolaydır: $3n^2 + 5n \geq 3n^2$, yani $c_1 = 3$. Üst sınır için $n \geq 5$ iken $5n \leq n \cdot n = n^2$ olduğu kullanılır: $3n^2 + 5n \leq 3n^2 + n^2 = 4n^2$, yani $c_2 = 4$ ve $n_0 = 5$. Böylece $3n^2 \leq 3n^2 + 5n \leq 4n^2$ ve fonksiyon $\Theta(n^2)$'dir. $5n$ terimi büyük n'de n^2'nin yanında önemsizleşir.

Örnek 0.37

1. **Sıralama.** Sık görülen fonksiyonlar yavaştan hızlıya: $1 < \log\log n < \log n < n < n \log n < n^2 < 2^n$ (büyük n için).
2. **Sayılarla.** $n = 16$'da $\log_2 \log_2 n = 2$, $\log_2 n = 4$, $n^2 = 256$, $2^n = 65.536$'dır. n 256'ya çıkınca $\log_2 n$ yalnız 8 olur, ama 2^n yaklaşık 10^{77}'ye çıkar (Tablo 0.7).
3. **Oyun ağacı.** Her düğümde 2 dal varsa ve ağacın derinliği d ise yaprak sayısı 2^d'dir: her düzeyde düğüm sayısı ikiye katlanır. Geriye doğru çıkarım her düğüme bir kez baktığı için $\Theta(2^d)$ adım sürer. $d = 30$'da yaklaşık $2^{30} \approx 10^9$ yaprak vardır.

n	$\log_2 \log_2 n$	$\log_2 n$	n^2	2^n
16	2	4	256	65.536
256	3	8	65.536	$\approx 10^{77}$
65.536	4	16	$\approx 4{,}3 \cdot 10^9$	çok büyük

Tablo 0.7: Büyüme hızlarının karşılaştırması.

Tablo 0.7 üstel büyümenin neden bir engel olduğunu gösterir. Saniyede 10^9 işlem yapan bir bilgisayar, $d = 30$ derinlikli ikili bir ağacı yaklaşık bir saniyede çözer; $d = 60$'ta ise yaklaşık $2^{60}/10^9 \approx 1{,}15 \cdot 10^9$ saniye, yani 36 yıldan fazla gerekir. Derinliği iki katına çıkarmak süreyi iki katına değil, milyar katına çıkarır. Bu yüzden algoritmik oyun kuramında "polinom zamanda" (n^2, n^3 gibi) çözülebilen problemler kolay, yalnız üstel zamanda çözülebildiği düşünülen problemler zor sayılır.

Sık yapılan hata

O'yu "tam olarak" sanmak: $n = O(n^2)$ doğrudur (yalnız üst sınırdır), ama $n = \Theta(n^2)$ yanlıştır, çünkü n alttan n^2'nin hiçbir sabit katıyla sınırlanamaz. O ve Θ büyük n'deki eğilimi söyler; küçük n'de $\Theta(n^2)$ bir yöntem $\Theta(n)$ bir yöntemden hızlı olabilir. Örneğin n^2 adımlı bir yöntem, $100n$ adımlı bir yöntemden $n < 100$ için daha hızlıdır.

0.11 Sık yapılan hatalar

Bölümdeki hataların çoğu birkaç ortak kaynaktan gelir: sırayı karıştırmak (profilde, matriste, ağaçta), bir sayıyı bir seçenekle karıştırmak (max ve argmax), olasılıkları gerekmeyen yerde toplamak ve bir oyuncunun neyi seçebildiğini unutmak. Tablo 0.8 her konunun en sık hatasını ve doğrusunu bir arada verir.

Konu	Hata	Doğrusu
Profil	(Taş, Kağıt) = (Kağıt, Taş)	Sıra önemli: önce Ayşe, sonra Bora
Sayma	Sırasız seçimde sırayı saymak ($4 \cdot 3 = 12$)	$C(4, 2) = 12/2! = 6$
Σ	$(\Sigma x_i)^2 = \Sigma x_i^2$	Önce terimler hesaplanır, sonra toplanır
Seri	Yarından başlayan toplam için $1/(1 - \delta)$	$\delta/(1 - \delta)$; ilk terim hangi turda?
argmax	"argmax = 7"	argmax bir seçenektir (B); max değerdir (7)
Türev	$f' = 0$ noktasını denetlememek	$f'' < 0$ ya da komşu değerler; uç noktalar
Üs ve ln	$2^2 + 2^3 = 2^5$; $\ln(a + b) = \ln a + \ln b$	Üsler çarpımda toplanır; ln çarpımı toplama çevirir
Zincir kuralı	$(e^{-x})' = e^{-x}$	İç türevle çarpılır: $-e^{-x}$
Kısmi türev	$\partial(qr)/\partial q = 0$	r sabittir ama sıfır değildir: r
Olasılık	En az bir 6: $1/6 + 1/6$	$1 - (5/6)^2 = 11/36$
Bayes	$P(A \mid B) = P(B \mid A)$	Ön olasılık da hesaba girer
Beklenen değer	"E = 3,5 gelecek"	Uzun vadeli ortalama; tek atışta gelmeyebilir
U[0, 1]	Yoğunluk = olasılık; $P(X = 0{,}5) > 0$	Olasılık bir alandır; tek noktanınki 0
Karışık strateji	Sırayla değiştirmek	Her tur rastgele tek bir saf hamle
Softmax	$v_i / \Sigma v_j$	$e^{\lambda v_i} / \Sigma e^{\lambda v_j}$
Matris	b'yi satır oyuncusuna yazmak	(a, b): a satır, b sütun oyuncusunun
En iyi tepki	Rakibin sayısına bakmak	Rakibin seçimi sabitlenir, kendi sayıları karşılaştırılır
İkinci derece	Yalnız $\sqrt{\Delta}$'yı $2a$'ya bölmek	$(-b \pm \sqrt{\Delta})/(2a)$; olasılık [0, 1]'de
Kayıtsızlık	p'yi Ayşe'nin denkleminden bulmak	p, Bora'yı kayıtsız bırakan denklemden çıkar
Oyun ağacı	Kökten başlamak	Yapraktan köke: önce son hamle
O ve Θ	$n = \Theta(n^2)$	$n = O(n^2)$; O yalnız üst sınırdır

Tablo 0.8: Konulara göre sık yapılan hatalar.

0.12 Özet

- **Gösterim.** $S_1 \times S_2$ bütün strateji profillerinin kümesidir; profilde sıra önemlidir. u_i, bir profile oyuncu i'nin kazancını (bir sayı) karşılık getirir. Profil sayısı çarpım kuralıyla, sırasız seçimler $C(n, k)$ ile sayılır.
- **Σ, seri, max ve argmax.** Σ ile önce her terim hesaplanır, sonra toplanır. $0 \le \delta < 1$ için $1 + \delta + \delta^2 + \ldots = 1/(1 - \delta)$. max bir değerdir, argmax onu veren seçenek(ler)dir; eşitlikte argmax bir kümedir.
- **Türev.** Sürekli seçimde en büyük değer $f' = 0$ ile bulunur ve $f'' < 0$ ile denetlenir. $u(q) = q(12 - q - r)$ için en iyi miktar $q^*(r) = \max\{0, (12 - r)/2\}$'dir.
- **Üs, e ve ln.** $x^a \cdot x^b = x^{a+b}$, $x^0 = 1$. ln, e^x'in tersidir ve çarpımı toplamaya çevirir; $(\ln x)' = 1/x$, $(e^x)' = e^x$.
- **Türev kuralları.** $(fg)' = f'g + fg'$; zincir: dış türev × iç türev; $\partial u/\partial q$: öteki değişkenler sabit tutulur. Her oyuncu kendi değişkenine göre kısmi türevi sıfıra eşitler; iki firma örneğinde $q = r = 4$.
- **Olasılık.** Eşit olası sonuçlarda sayılır ve bölünür; bağımsız olayların olasılıkları çarpılır, aynı anda olamayan olaylarınki toplanır; "en az bir" için tümleyen kullanılır. $P(A \mid B) = P(A \cap B)/P(B)$; Bayes: $P(A \mid B) = P(B \mid A) \cdot P(A)/P(B)$.
- **Beklenen değer.** $E[X] = \Sigma\, x \cdot P(x)$: uzun vadeli ortalamadır, tek denemenin sonucu değildir. Karar net beklenen değere göre verilir. $X \ge 0$ ise $E[X] = \int P(X > t)\, dt$.
- **Sürekli dağılım.** $U[0, 1]$'de olasılık bir alandır: $P(X \le x) = x$, tek noktanın olasılığı 0, $E[X] = 1/2$. İntegral eğri altındaki alandır, Σ'nın sürekli hâlidir.
- **Karışık strateji.** Eylemler üzerinde bir olasılık dağılımıdır; her tur tek bir saf hamle rastgele seçilir. Softmax: $P(i) \propto e^{\lambda v_i}$; $\lambda = 0$'da eşit, λ büyüdükçe argmax.
- **Kazanç matrisi.** (a, b): a satır, b sütun oyuncusunun kazancı. Beklenen kazanç $x^T A y$'dir; Ay her satırın beklentisidir.
- **En iyi tepki.** Rakibin seçimi sabitlenir; kendi kazancının argmax'ı alınır. İki sayısı da işaretli hücrede iki oyuncu da en iyi tepkisini oynar (Nash dengesi).
- **Denklemler.** $x = (-b \pm \sqrt{\Delta})/(2a)$; olasılık olan kök [0, 1]'dedir. Kayıtsızlık: karıştırılan eylemler eşit beklenen kazanç verir; bir oyuncunun kayıtsızlığı rakibin olasılığını belirler. Kart Gösterme'de $p = q = 1/3$, Ayşe'nin kazancı $-1/3$.
- **Taş-Kağıt-Makas.** $(1/3, 1/3, 1/3)$ yenilmez (beklenen kazanç 0); hep Taş tur başına −1, $(1/2, 1/4, 1/4)$ tur başına −1/4 kaybettirir.
- **Geriye doğru çıkarım.** Yapraktan köke; sırası gelen kendi kazancının en büyüğünü seçer. Kart oyununda sonuç Göster → Çekil, (2, 3).
- **O ve Θ.** $\log\log n < \log n < n < n \log n < n^2 < 2^n$; sabitler ve küçük terimler atılır. İkili ağaçta geriye doğru çıkarım $\Theta(2^d)$ adımdır.

Tablo 0.9 her aracı, oyun kuramında en çok kullanıldığı yerle eşleştirir.

Araç	Ne yapar?	Oyun kuramında nerede?
Kartezyen çarpım	Bütün profilleri listeler	Kazanç matrisinin hücreleri
Kazanç fonksiyonu	Profile sayı karşılık getirir	Her tanımın yazımı
Geometrik seri	Sonsuz iskontolu toplam	Tekrarlanan oyunlar
argmax / türev	En iyi seçeneği bulur	En iyi tepki
Kısmi türev	Öteki değişkenler sabitken eğim	Sürekli stratejili oyunlarda denge
Bayes kuralı	Gözlemden sonra inancı günceller	Eksik bilgili oyunlar
Beklenen değer	Uzun vadeli ortalama	Karışık stratejilere karşı kazanç
U[0, 1], integral, kuyruk	Sürekli belirsizlik	Açık artırmalar
Softmax	Değerleri yumuşak olasılıklara çevirir	Davranışsal modeller, öğrenme
Kayıtsızlık denklemi	Karışımın olasılıklarını bulur	Karışık strateji dengesi
Geriye doğru çıkarım	Ağacı yapraktan köke çözer	Sıralı oyunlar
O ve Θ	Büyüme hızını ölçer	Dengeyi hesaplamanın zorluğu

Tablo 0.9: Bölümün araçları ve oyun kuramındaki kullanımları.

0.13 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

Kartezyen çarpım *Cartesian product* · strateji profili *strategy profile* · kazanç fonksiyonu *payoff function* · çarpım kuralı *product rule* · kombinasyon C(n, k) *binomial coefficient* · geometrik seri *geometric series* · iskonto faktörü *discount factor* · argmax *argmax* · türev *derivative* · zincir kuralı *chain rule* · kısmi türev *partial derivative* · koşullu olasılık *conditional probability* · Bayes kuralı *Bayes' rule* · ön olasılık *prior* · son olasılık *posterior* · beklenen değer *expected value* · beklenen kazanç *expected payoff* · düzgün dağılım *uniform distribution* · olasılık yoğunluğu *probability density* · integral *integral* · kuyruk olasılığı *tail probability* · karışık strateji *mixed strategy* · saf strateji *pure strategy* · destek *support* · olasılık vektörü *probability vector* · logit fonksiyonu *logit function* · softmax seçimi *softmax choice* · kazanç matrisi *payoff matrix* · sıfır toplamlı oyun *zero-sum game* · en iyi tepki *best response* · diskriminant *discriminant* · kayıtsızlık ilkesi *indifference principle* · sıralı oyun *sequential game* · oyun ağacı *game tree* · geriye doğru çıkarım *backward induction* · büyük-O ve Θ gösterimi *big-O and Θ notation*

0.14 Kendini Dene

1. Üç oyuncunun sırasıyla 3, 3 ve 2 stratejisi vardır. Kaç strateji profili vardır? Bir bekçi 5 hedeften 3'ünü koruyabiliyorsa kaç saf stratejisi vardır; hedefler "sırayla" seçilseydi kaç sıralı seçim sayılırdı?

   Cevap

   Çarpım kuralı: $3 \cdot 3 \cdot 2 = 18$ profil. Bekçi: $C(5, 3) = 5!/(3! \cdot 2!) = 120/12 = 10$ saf strateji. Sıralı seçim $5 \cdot 4 \cdot 3 = 60$'tır; her üçlü $3! = 6$ farklı sırada sayıldığı için $60/6 = 10$.

2. Bir işbirliği her tur 4 kazandırıyor ve gelecek turlar $\delta = 0{,}75$ ile tartılıyor. Bu turdan başlayan toplam iskontolu kazanç kaçtır? Kazanç ancak bir sonraki turdan itibaren gelseydi ne olurdu?

   Cevap

   Bugünden başlayan: $4/(1 - 0{,}75) = 4/0{,}25 = 16$. Yarından başlayan: $4 \cdot 0{,}75/(1 - 0{,}75) = 3/0{,}25 = 12$. Fark bugünkü 4'tür: $16 - 12 = 4$.

3. Bir torbada iki para var: biri adil, öteki iki yüzü de yazı olan hileli bir para. Torbadan rastgele bir para çekilip atılıyor ve yazı geliyor. Paranın hileli olma olasılığı nedir? Aynı para ikinci kez atılıp yine yazı gelirse bu olasılık ne olur?

   Cevap

   Ön olasılık $P(\text{hileli}) = 1/2$. $P(\text{yazı}) = 1/2 \cdot 1 + 1/2 \cdot 1/2 = 3/4$. Bayes: $P(\text{hileli} \mid \text{yazı}) = (1 \cdot 1/2)/(3/4) = 2/3$. İki yazı için adil paranın olasılığı $1/4$'tür: $P(\text{iki yazı}) = 1/2 \cdot 1 + 1/2 \cdot 1/4 = 5/8$ ve $P(\text{hileli} \mid \text{iki yazı}) = (1/2)/(5/8) = 4/5$. 100 deneyle sayarak: 50 hileli denemenin hepsi, 50 adil denemenin 12,5'i iki yazı verir; $50/62{,}5 = 0{,}8$.

4. Bir çark 1/2 olasılıkla 0 TL, 1/3 olasılıkla 6 TL, 1/6 olasılıkla 18 TL ödüyor; çevirmek 6 TL. Net beklenen kazanç kaçtır? Diğer ödüller aynı kalırken 18 TL'lik ödül kaç TL olsaydı oyun adil olurdu?

   Cevap

   Olasılıklar toplamı $1/2 + 1/3 + 1/6 = 1$ ✓. $E = 0 \cdot 1/2 + 6 \cdot 1/3 + 18 \cdot 1/6 = 0 + 2 + 3 = 5$ TL; net $5 - 6 = -1$ TL. Adil ödül y: $2 + y/6 = 6 \Rightarrow y/6 = 4 \Rightarrow y = 24$ TL.

5. Sıfır toplamlı bir oyunda satır oyuncusunun (Ayşe) kazanç matrisi $A = [[2, -1], [-1, 1]]$'dir (satırlar Yukarı, Aşağı; sütunlar Sol, Sağ). Bora $y = (1/2, 1/2)$ oynarsa Ay nedir ve Ayşe'nin en iyi tepkisi hangisidir? Bora Sol'u hangi q ile, Ayşe Yukarı'yı hangi p ile oynarsa iki oyuncu da kayıtsız kalır?

Ayşe'nin beklenen kazancı kaçtır?

Cevap

$Ay = (2 \cdot 1/2 - 1 \cdot 1/2, -1 \cdot 1/2 + 1 \cdot 1/2) = (1/2, 0)$; en iyi tepki Yukarı. Ayşe'nin kayıtsızlığı: $2q - (1 - q) = -q + (1 - q) \Rightarrow 3q - 1 = 1 - 2q \Rightarrow q = 2/5$. Bora'nın kazançları $-A$'dır; kayıtsızlığı: Sol $-2p + (1 - p) = 1 - 3p$, Sağ $p - (1 - p) = 2p - 1 \Rightarrow 1 - 3p = 2p - 1 \Rightarrow p = 2/5$. Ayşe'nin kazancı $3 \cdot 2/5 - 1 = 1/5$ (denetim: $1 - 2 \cdot 2/5 = 1/5$ ✓).

6. Fiyat $20 - (q + r)$, maliyet sıfır; firmanın kârı $u(q) = q(20 - q - r)$'dir. Rakip $r = 6$ üretirse en iyi miktar ve kâr nedir? Bu noktanın tepe olduğu iki yolla nasıl gösterilir? Genel en iyi tepki formülü ve $\partial u/\partial r$ nedir?

Cevap

$u(q) = q(14 - q) = 14q - q^2$; $u'(q) = 14 - 2q = 0 \Rightarrow q = 7$, $u(7) = 49$. $u'' = -2 < 0$; komşular $u(6) = u(8) = 48 < 49$. Genel: $\partial u/\partial q = 20 - 2q - r = 0 \Rightarrow q^*(r) = \max\{0, (20 - r)/2\}$. $\partial u/\partial r = -q$.

7. Ayşe önce Sol ya da Sağ der; Bora bunu görüp cevap verir. Sol'dan sonra Bora a derse (2, 1), b derse (0, 0); Sağ'dan sonra c derse (3, 0), d derse (1, 2) olur (Ayşe, Bora). Geriye doğru çıkarımın sonucu nedir? Her düğümde 3 dal olan d derinlikli bir ağaçta kaç yaprak vardır; $d = 10$ için kaç?

Cevap

Sol'dan sonra Bora a'yı seçer ($1 > 0$) ⇒ (2, 1). Sağ'dan sonra Bora d'yi seçer ($2 > 0$) ⇒ (1, 2). Ayşe kökte 2 ile 1'i karşılaştırır ve Sol der: sonuç (2, 1). (3, 0) Ayşe'nin en büyük sayısıdır ama Bora c'yi seçmez. Yaprak sayısı 3^d; $d = 10$ için $3^{10} = 59.049$. Geriye doğru çıkarım $\Theta(3^d)$ adım sürer.

8. Üç hamlenin değerleri $v = (\ln 2, 0, 0)$ ve $\lambda = 1$'dir. Softmax olasılıkları nedir? İki seçenekte $v_1 - v_2 = 1$ ve $\lambda = \ln 4$ ise logit olasılığı $P(1)$ kaçtır? Bütün değerlere 5 eklenirse bu olasılıklar değişir mi?

Cevap

Üslü sayılar $e^{\ln 2} = 2$, $e^0 = 1$, $e^0 = 1$; toplam 4. Olasılıklar $(2/4, 1/4, 1/4) = (1/2, 1/4, 1/4)$: iki paralı yanlı karışımın aynısı. Logit: $e^{-\lambda \cdot 1} = e^{-\ln 4} = 1/4$, $P(1) = 1/(1 + 1/4) = 4/5$. Değerlere 5 eklemek her üssü $e^{5\lambda}$ ile çarpar; pay ve paydada sadeleşir ve olasılıklar değişmez.

0.15 İleri okuma

- M. J. Osborne, *An Introduction to Game Theory* (2004), Bölüm 17 (Matematik eki): sayılar, kümeler, fonksiyonlar, profiller, diziler ve olasılık konularını oyun kuramında kullanıldığı ölçüde, kısa ve örnekli bir özetle verir.

- J. K. Blitzstein ve J. Hwang, *Introduction to Probability* (2019), Bölüm 1–5: sayma, koşullu olasılık, Bayes kuralı, beklenen değer ve düzgün dağılımı çok sayıda örnek ve simülasyonla anlatır.
- E. Lehman, F. T. Leighton ve A. R. Meyer, *Mathematics for Computer Science* (2017): sayma, toplamlar ve asimptotik gösterim (O, Θ) için bilgisayar mühendisliği öğrencilerine yönelik ücretsiz bir ders kitabıdır.
- Z. Wang, B. Xu ve H.-J. Zhou, "Social cycling and conditional responses in the Rock-Paper-Scissors game" (2014): Taş-Kağıt-Makas'ta insanların kazanınca tekrarlayıp kaybedince değiştirdiğini gösteren laboratuvar deneyidir; kısa ve okunabilirdir.

0.16 Kaynaklar

1. J. K. Blitzstein, J. Hwang, *Introduction to Probability*, 2. baskı, CRC Press (Chapman & Hall), 2019.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 4. baskı, MIT Press, 2022 (Bölüm 3: asimptotik gösterim).
3. E. Lehman, F. T. Leighton, A. R. Meyer, *Mathematics for Computer Science*, MIT, 2017.
4. R. D. McKelvey, T. R. Palfrey, "Quantal response equilibria for normal form games", *Games and Economic Behavior* 10(1), 6–38, 1995.
5. M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2004.
6. Z. Wang, B. Xu, H.-J. Zhou, "Social cycling and conditional responses in the Rock-Paper-Scissors game", *Scientific Reports* 4, 5830, 2014.

KISIM I · BÖLÜM 1

Karar Kuramı

Oyun kuramı, birden çok karar vericinin birbirini etkilediği durumları inceler. Ama önce tek bir karar vericinin, sonucu şansa bağlı seçenekler arasında nasıl seçim yaptığını anlamak gerekir. Bu not, tek kişilik karar kuramını anlatır: riskli seçenekleri beklenen faydayla karşılaştırmayı, fayda sayılarının ne anlama geldiğini, risk tutumlarını, olasılıklar bilinmediğinde kullanılan ölçütleri, yeni bilgiyle inancı Bayes kuralıyla güncellemeyi ve karar ağaçlarıyla bilginin parasal değerini hesaplamayı. Bu araçların hepsi, sonraki konulardaki stratejik ve kolektif kararların da yapı taşlarıdır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum01`

ÖĞRENME ÇIKTILARI

Beklenen fayda ve risk
Bir piyangonun beklenen değerini ve faydasını hesaplamak; kesinlik eşdeğeri ve risk primiyle risk tutumunu yorumlamak; olasılıklar bilinmezken maksimin, Laplace ve Savage ölçütlerini uygulamak.

Fayda ölçeği
vNM aksiyomlarını, Allais ve Ellsberg seçimlerini açıklamak; fayda sayılarının hangi dönüşümlerde kararı koruduğunu göstermek.

Bayesçi karar
Ön olasılığı test sonucuyla güncellemek; taban oranı ihmalini sayısal örnekte göstermek; beklenen kaybı en küçük eylemi seçmek.

Karar ağaçları
Geriye doğru çıkarımla en iyi stratejiyi bulmak; mükemmel ve örnek bilginin değerini hesaplamak.

1.1 Beklenen fayda

Her gün sonucunu önceden bilmediğimiz kararlar veririz: şemsiye almak, bir derse çalışmak için hangi konuya öncelik vermek, bir ürünü almak ya da beklemek. Bu kararların ortak yanı, sonucun yalnız bizim seçimimize değil, bizim denetleyemediğimiz bir şeye de bağlı olmasıdır. Karar kuramı bu "denetlenemeyen şeyi" **doğa** ya da **şans** diye adlandırır. Şansa bağlı seçenekleri karşılaştırmanın temel aracı beklenen faydadır.

1.1.1 Bir oyunla başlangıç: Bayesçi Karar Laboratuvarı

Konuya küçük bir oyunla girelim. Bayesçi Karar Laboratuvarı'nda bir etkinlik planlanır ve iki seçenek vardır: etkinliği *açık havada* ya da *kapalı mekânda* yapmak. Havanın yağmurlu mu kuru mu olacağı gizlidir. Oyuncu yalnız yağmurun ne kadar olası olduğunu bilir; buna **ön olasılık** (*prior*) denir. Ayrıca gürültülü bir sinyal görür: bir hava tahmini. Sonra seçimini yapar, durum açılır ve tablodaki kazancı alır.



Şekil 1.1: Bayesçi Karar Laboratuvarı'nın birinci düzeyi. Açık hava yağmurda −6, kuruda 8 kazandırır; kapalı mekân yağmurda 4, kuruda 3 kazandırır. Yağmurun ön olasılığı 0,35'tir. Doğruluğu %90 olan tahmin “yağmur” demiştir.

Şekil 1.1 oyunun ilk düzeyini gösterir. Tabloyu okumak kolaydır. Açık havayı seçen biri, yağmur yağarsa −6 alır; hava kuru giderse 8 alır. Kapalı mekânı seçen biri, yağmurda 4, kuruda 3 alır. Açık hava büyük bir kazanç ama büyük bir kayıp riski taşır; kapalı mekân ise iki durumda da makul bir sonuç verir.

Oyun iki soruyla başlar. Birincisi: %90 doğru bir tahmin “yağmur” dediyse, yağmur olasılığı kaçtır? Çoğu kişinin ilk cevabı %90'dır. Doğru cevap %82,9'dur ve bu farkın nedeni ön olasılıktır: yağmur zaten pek olası değildir (0,35). Bu sayının nasıl bulunduğu 1.3.6'da adım adım hesaplanır. İkinci soru daha felsefidir: bir karardan sonra kötü bir sonuç çıktıysa karar da kötü müydü? Cevap “hayır” olabilir. İyi karar, eldeki bilgiyle en iyi beklentiyi veren karardır. Şans kötü giderse iyi bir karar da kötü sonuç verebilir. Karar kuramı kararları sonuçlarına göre değil, verildikleri andaki bilgiye göre değerlendirir.

Bu iki soru bölümün iki ana konusunu özetler. İlki, yeni bilgiyle inancın nasıl güncelleneceğidir (Bayes kuralı). İkincisi, inanç belliyken hangi seçeneğin seçileceğidir (beklenen fayda). Konu, ikincisinden başlayarak kurulur.

1.1.2 Karar almanın üç türü

Karar alma (*decision making*), seçenekler arasından sonuçlarını değerlendirerek seçim yapma sürecidir. Kararlar, sonucu neyin belirlediğine göre üç türe ayrılır. Tablo 1.1 bu ayrımı özetler.

Tür	Sonucu belirleyen	Araç	Örnek
Bireysel, risk altında	Doğa / şans (olasılıklar bilinir ya da tahmin edilir)	Beklenen fayda, Bayes, karar ağacı	Yatırım, sigorta, tıbbi test
Stratejik	Diğer rasyonel aktörlerin seçimleri	En iyi tepki, Nash dengesi	Fiyat rekabeti, pazarlık, spor
Kolektif	Grubun toplu seçimi ve etkileşimi	Oylama, ortak mallar	Politika, şirket stratejisi

Tablo 1.1: Karar almanın üç türü.

Bireysel kararda karar verici yalnız doğaya karşı seçim yapar. Doğa "akıllı bir rakip" değildir: yağmur, etkinliği bozmak için yağmaz. Doğanın davranışı olasılıklarla anlatılır. Stratejik kararda ise sonucu belirleyen başka akıllı karar vericilerdir. Onlar da bizim ne yapacağımızı düşünür. Bu karşılıklı düşünme durumu oyun kuramının asıl konusudur. Kolektif kararda bir grup, oylama ya da ortak kaynak paylaşımı gibi bir kuralla birlikte karar verir.

Bu ayrım neden önemlidir? Çünkü stratejik ve kolektif kararların analizinde de tek kişilik karar kuramının araçları kullanılır. Rakibinin ne yapacağına dair bir inancı olan oyuncu, kendi seçimini beklenen faydayla yapar. Rakibinin hangi "tipte" olduğunu bilmiyorsa, onun hamlelerini gözleyerek inancını Bayes kuralıyla günceller. Sırayla hamle yapılan bir oyun, karar ağacına benzeyen bir oyun ağacıyla çözülür. Bu yüzden bu bölümdeki araçlar dersin geri kalanının temelidir.

Karar kuramında bir de risk ile belirsizlik arasındaki ayrım vardır. **Risk altında karar**da olasılıklar bilinir ya da güvenilir biçimde tahmin edilir: zar atmak, rulet, sigorta istatistikleri. **Belirsizlik altında karar**da (*decision under uncertainty*) olasılıklar bilinmez: yeni bir ürünün pazarda tutup tutmayacağı, yeni bir teknolojinin başarısı. Bölümün büyük kısmı risk altında karar üzerinedir; belirsizlik altında karar 1.2.5 ve 1.2.6'da ele alınır.

1.1.3 Piyango ve beklenen değer

Riskli bir seçeneği matematiksel olarak anlatmanın yolu onu bir piyango olarak yazmaktır. Burada "piyango" sözcüğü yalnız bilet alınan şans oyununu değil, sonucu şansa bağlı her seçeneği anlatır.

Tanım 1.1

Bir **piyango** (*lottery*), olasılıklı sonuçlar kümesidir. Sonuçlar $x_1, \ldots, x_k$, olasılıkları $p_1, \ldots, p_k$ ise piyango $L = (p_1: x_1; \ldots; p_k: x_k)$ biçiminde yazılır. Olasılıklar sıfırdan küçük değildir ve toplamları 1'dir: $p_1 + \ldots + p_k = 1$.

Sonuçlar sayı ise piyangonun **beklenen değeri**

$$E[X] = \Sigma\, p_i\, x_i = p_1 x_1 + p_2 x_2 + \ldots + p_k x_k$$

olarak tanımlanır. Kesin bir tutar da tek sonuçlu bir piyangodur: olasılığı 1'dir.

Σ (sigma) işareti "topla" demektir. $\Sigma\, p_i\, x_i$, her sonucu olasılığıyla çarpıp çarpımları toplamak anlamına gelir. Beklenen değer, piyango çok kez oynandığında bir oyun başına düşen ortalama kazançtır. Örneğin yazı gelirse 10 TL veren, tura gelirse hiçbir şey vermeyen bir oyun 1000 kez oynanırsa yaklaşık 500 kez yazı gelir; toplam kazanç yaklaşık 5000 TL, oyun başına ortalama 5 TL olur. Beklenen değer de $\tfrac{1}{2} \cdot 10 + \tfrac{1}{2} \cdot 0 = 5$'tir.

Bölüm boyunca iki seçenekli bir örnek sık sık kullanılır:

- **Piyango A:** %50 olasılıkla 900 TL, %50 olasılıkla 0 TL.
- **Kesin B:** olasılık 1 ile 400 TL.

Örnek 1.1

A ile B'nin beklenen değerlerini karşılaştıralım.

1. A'nın iki sonucu vardır: 900 TL (olasılık ½) ve 0 TL (olasılık ½). Her sonuç olasılığıyla çarpılır: $\tfrac{1}{2} \cdot 900 = 450$ ve $\tfrac{1}{2} \cdot 0 = 0$.
2. Çarpımlar toplanır: $E[A] = 450 + 0 = 450$ TL.
3. B kesin olduğu için beklenen değeri tutarın kendisidir: $E[B] = 1 \cdot 400 = 400$ TL.
4. $450 > 400$ olduğundan, yalnız beklenen değere bakan bir karar verici A'yı seçer.

Oysa insanlara sorulduğunda çoğu B'yi seçer: kesin 400 TL'yi, yarı yarıya 900 TL ya da hiçbir şey almaya tercih eder. Bu seçim akıl dışı mıdır? Hayır; tutarlı bir ölçütle açıklanabilir.

Beklenen değer, kararları sadeleştiren güçlü bir ölçüttür ama bir şeyi gözden kaçırır: paranın değeri herkes için ve her tutarda aynı değildir. Elinde hiç parası olmayan birine 400 TL'nin getirdiği rahatlık büyüktür. 400 TL'den 900 TL'ye çıkmanın ek rahatlığı ise o kadar büyük değildir. Bu gözlem fayda kavramına götürür.

Sık yapılan hata

- **Olasılıkları toplamamak.** Piyangonun olasılıklarının toplamı 1 olmalıdır. "%50 olasılıkla 900 TL, %40 olasılıkla 0 TL" diye yazılan bir piyangoda %10'luk bir sonuç unutulmuştur.
- **Beklenen değeri "en olası sonuç" sanmak.** A'nın beklenen değeri 450 TL'dir, ama A hiçbir zaman 450 TL vermez: ya 900 ya 0 verir. Beklenen değer bir ortalamadır, gerçekleşen bir sonuç değildir.
- **Yüzdeyi çarpımda yüzde olarak bırakmak.** %50 hesapta 0,5 ya da ½ olarak yazılır; *50 · 900* yazmak sonucu 100 kat büyütür.

1.1.4 Beklenen fayda

Fayda, bir sonucun karar verici için ne kadar "iyi" olduğunu gösteren bir sayıdır. Para tutarı bir sonuçtur; fayda ise bu sonucun kişiye göre değeridir. Bir **fayda fonksiyonu** (*utility function*) u, her sonuca bir sayı atar. Para için sık kullanılan bir örnek $u(x) = \sqrt{x}$'tir: 0 TL'nin faydası 0, 400 TL'nin faydası 20, 900 TL'nin faydası 30'dur.

Tanım 1.2

Fayda fonksiyonu u olan bir karar verici için piyango $L = (p_1: x_1; \ldots; p_k: x_k)$'nın **beklenen faydası** (*expected utility*)

$$E[u(L)] = \Sigma\, p_i\, u(x_i)$$

sayısıdır. **Beklenen fayda kuralı**na göre rasyonel karar verici, seçenekler arasından beklenen faydası en büyük olanı seçer; yani beklenen faydayı **maksimize eder**.

Tanım, beklenen değer tanımına çok benzer. Tek fark, sonucun kendisi (x_i) yerine faydasının ($u(x_i)$) olasılıkla çarpılmasıdır. Kural, kuramı 1944'te sistemli biçimde kuran John von Neumann ile Oskar Morgenstern'in adıyla "von Neumann–Morgenstern beklenen faydası" diye de anılır (von Neumann ve Morgenstern, 1944).



Şekil 1.2: Piyango A ve kesin B. Kırmızı değerler her ölçütün seçtiği seçeneği gösterir: beklenen değer A'yı (450 > 400), beklenen fayda ($\sqrt{x}$) B'yi seçer (20 > 15).

Örnek 1.2

Fayda fonksiyonu $u(x) = \sqrt{x}$ olan bir karar verici, A ile B arasında seçim yapıyor.

1. A'nın sonuçlarının faydaları bulunur: $u(900) = \sqrt{900} = 30$ ve $u(0) = \sqrt{0} = 0$.
2. Faydalar olasılıklarla çarpılıp toplanır: $E[u(A)] = ½ \cdot 30 + ½ \cdot 0 = 15$.
3. B kesindir: $u(B) = \sqrt{400} = 20$.
4. $20 > 15$ olduğundan bu karar verici B'yi seçer.

Şekil 1.2 iki ölçütün kararını yan yana gösterir. Aynı iki seçenek için beklenen değer A'yı, beklenen fayda B'yi seçmiştir. Kesin 400 TL'yi seçen insanlar akıl dışı davranmaz; fayda fonksiyonları içbükeydir, yani paranın her ek lirası onlara biraz daha az fayda katar.

Beklenen parasal değer ile beklenen fayda aynı kararı vermeyebilir. Aradaki farkı fayda fonksiyonunun eğriliği, yani karar vericinin **risk tutumu** belirler. Fonksiyon düz bir doğruysa ($u(x) = x$) iki ölçüt aynı kararı verir. Fonksiyon aşağı doğru bükülüyorsa ($\sqrt{x}$ gibi) karar verici riskten kaçınır. Bu konu 1.2'de ayrıntılı işlenir.

Beklenen fayda kuralı neden “rasyonel” sayılır? Bunun iki cevabı vardır. Birincisi pratiktir: aynı tür karar çok kez verildiğinde kural, ortalamada en iyi sonucu getiren seçimi önerir. Fayda fonksiyonu da karar vericinin riske karşı tutumunu yansıtır. İkincisi kuramsaldır ve daha güçlüdür: tercihleri birkaç makul koşulu sağlayan herkes, sanki bir fayda fonksiyonunun beklenen değerini maksimize ediyormuş gibi davranır. Bu koşullar 1.1.6'da anlatılır.

1.1.5 St. Petersburg paradoksu

Fayda fikrinin tarihsel çıkış noktası 18. yüzyılda tartışılan bir yazı-tura oyunudur. Bir para, ilk kez yazı gelene kadar atılır. Yazı ilk atışta gelirse oyuncu 2 TL alır, ikinci atışta gelirse 4 TL, üçüncüde 8 TL; genel olarak k-ıncı atışta gelirse 2^k TL alır. Bu oyunu oynamak için en fazla ne ödenmelidir?

Beklenen değeri hesaplayalım. Yazının ilk kez k-ıncı atışta gelme olasılığı $(\tfrac{1}{2})^k$'dir, çünkü önceki $k-1$ atışın tura, k-ıncının yazı gelmesi gerekir. O durumda kazanç 2^k'dir. Her terim olasılık çarpı kazançtır:

$$E[X] = \tfrac{1}{2}\cdot 2 + \tfrac{1}{4}\cdot 4 + \tfrac{1}{8}\cdot 8 + \dots = 1 + 1 + 1 + \dots$$

Her terim 1'dir ve sonsuz sayıda terim vardır. Toplam sonsuza gider: oyunun beklenen değeri sonsuzdur. Yalnız beklenen değere bakan biri bu oyuna servetinin tamamını vermeye razı olmalıdır. Oysa gerçekte kimse bu oyuna birkaç on liradan fazla ödemez. Bu çelişkiye **St. Petersburg paradoksu** (*St. Petersburg paradox*) denir.

Daniel Bernoulli 1738'de bu paradoksa şu cevabı verdi: insanlar paranın kendisini değil, paranın faydasını değerlendirir ve servet büyüdükçe her ek liranın faydası azalır (Bernoulli, 1738). Benzer bir fikri Gabriel Cramer de 1728'de önermişti. Bernoulli logaritmik bir fayda kullandı. Logaritma çok yavaş büyüyen bir fonksiyondur: $\log_2 x$, x'in 2'nin kaçıncı kuvveti olduğunu söyler ($\log_2 8 = 3$, $\log_2 1024 = 10$).

Örnek 1.3

Fayda $u(x) = \log_2 x$ olsun. k-ıncı atışta yazı gelirse kazanç 2^k, faydası $\log_2 2^k = k$'dır.

1. Beklenen fayda: $E[u] = \tfrac{1}{2}\cdot 1 + \tfrac{1}{4}\cdot 2 + \tfrac{1}{8}\cdot 3 + 1/16\cdot 4 + \dots$
2. İlk terimler 0,5; 0,5; 0,375; 0,25; 0,156; … biçiminde hızla küçülür. Toplam sonsuza gitmez; 2'ye yaklaşır.
3. Faydası 2 olan kesin tutar $\log_2 x = 2$ denkleminden $x = 4$ TL'dir.

Bu karar verici oyunu kesin 4 TL kadar değerli bulur. Beklenen değer sonsuz olsa da oyuna ödemeye razı olduğu tutar küçüktür. Paradoks, fayda kavramıyla ortadan kalkar.

St. Petersburg örneği iki şey öğretir. Birincisi, çok küçük olasılıklı çok büyük kazançlar beklenen değeri şişirebilir ama insanların kararını pek etkilemez. İkincisi, azalan marjinal fayda (her ek liranın daha az fayda katması) riskten kaçınmanın matematiksel açıklamasıdır.

1.1.6 von Neumann–Morgenstern aksiyomları

Beklenen fayda kuralı keyfî bir formül değildir. von Neumann ve Morgenstern, tercihleri dört makul koşulu sağlayan her karar verici için bir fayda fonksiyonu bulunduğunu ve bu kişinin beklenen faydayı maksimize ediyormuş gibi davrandığını gösterdi. Bu koşullara **von Neumann–Morgenstern aksi-**

yomları (*von Neumann–Morgenstern axioms*) denir. Aksiyom, ispatlanmadan kabul edilen temel varsayım demektir.

Aksiyomları yazmak için bir gösterim gerekir. $A \succ B$, "A, B'ye tercih edilir" demektir. $A \sim B$, "karar verici A ile B arasında kayıtsızdır" demektir; ikisi ona eşit derecede iyidir.

Tanım 1.3

Piyangolar üzerindeki bir tercih ilişkisi şu dört aksiyomu sağlar:

- **Tamlık** (*completeness*): Herhangi iki seçenek karşılaştırılabilir. Her A ve B için ya $A \succ B$, ya $B \succ A$, ya da $A \sim B$'dir.
- **Geçişlilik** (*transitivity*): $A \succ B$ ve $B \succ C$ ise $A \succ C$'dir.
- **Süreklilik** (*continuity*): $A \succ B \succ C$ ise arada kalan B, iki ucun bir karışımına denk getirilebilir: bir p olasılığı için $B \sim (p\text{: A};\ 1-p\text{: C})$.
- **Bağımsızlık** (*independence*): İki piyangoya aynı ortak parçayı aynı olasılıkla eklemek sıralamayı değiştirmez. $A \succ B$ ise her C ve her $0 < p \le 1$ için $(p\text{: A};\ 1-p\text{: C}) \succ (p\text{: B};\ 1-p\text{: C})$'dir.

Teorem 1.1 (von Neumann–Morgenstern)

Bir tercih ilişkisi dört aksiyomu sağlıyorsa bir u fayda fonksiyonu vardır ve her iki piyango için $A \succ B$ ancak ve ancak $E[u(A)] > E[u(B)]$'dir. Yani karar verici beklenen faydayı maksimize eder.

Teoremin ispatı bu notun kapsamı dışındadır. Yine de her aksiyomun neden makul olduğu ve çiğnenince ne olduğu kolayca görülür.

Tamlık, karar vericinin her durumda bir görüşü olduğunu söyler. "Karşılaştıramıyorum" diyen bir karar verici, karar veremez. **Geçişlilik** ihlal edilirse karar verici sömürülebilir. Diyelim ki biri elmayı armuta, armudu kiraza, kirazı da elmaya tercih ediyor. Elinde kiraz varken ona "biraz para ver, kirazını armutla değiştireyim" denir; kabul eder. Sonra armudu elmayla, elmayı yine kirazla, her seferinde biraz para karşılığı değiştirir. Döngünün sonunda yine kirazı vardır ama parası azalmıştır. Bu döngüye *para pompası* denir; geçişlilik, böyle bir sömürüye karşı korumadır.

Süreklilik şu anlama gelir: A çok iyi, C çok kötü, B ise arada bir sonuç olsun. Örneğin A = 1000 TL, B = 500 TL, C = 0 TL. A'yı yüksek olasılıkla, C'yi düşük olasılıkla veren bir piyango B'den iyidir; A'yı düşük olasılıkla veren bir piyango B'den kötüdür. Arada bir yerde, karar vericinin B ile piyango arasında kayıtsız kaldığı bir p değeri vardır. Bu kayıtsızlık olasılığı aslında fayda fonksiyonunun kendisini verir: $u(A) = 1$ ve $u(C) = 0$ seçilirse $u(B) = p$ olur. Fayda sayılarının neden "ölçülebilir" olduğunun gerekçesi budur.

Bağımsızlık en çok tartışılan aksiyomdur. Söylediği şudur: iki seçeneğin ortak bir parçası varsa, karar o ortak parçaya bağlı olmamalıdır. Bir para atılsın; yazı gelirse A ile B arasında seçim yapılacak, tura gelirse iki durumda da C alınacak. Tura gelirse seçimin bir önemi yoktur. Yazı gelirse A tercih edilir. O

hâlde para atılmadan önce de “yazıda A” seçeneği tercih edilmelidir. İnsanların gerçek seçimleri ise bu aksiyomu kimi zaman çiğner.

1.1.7 Allais paradoksu

1953'te Fransız iktisatçı Maurice Allais, insanlara iki ayrı seçim sordu (Allais, 1953). Tutarlar milyon lira cinsindendir. Tablo 1.2 dört seçeneği gösterir.

Seçim	Seçenek	1 milyon	5 milyon	0
1. seçim	A	%100	—	—
	B	%89	%10	%1
2. seçim	C	%11	—	%89
	D	—	%10	%90

Tablo 1.2: Allais'nin iki seçimi. Hücrelerde her tutarın olasılığı yazılıdır.

Birinci seçimde çoğu kişi A'yı seçer. A kesin 1 milyon verir; B daha fazla kazandırabilir ama %1 olasılıkla hiçbir şey vermez. Kesin bir milyonu bırakıp sıfır riskine girmek insanlara kötü gelir. İkinci seçimde ise çoğu kişi D'yi seçer. Burada iki seçenekte de büyük olasılıkla hiçbir şey kazanılmaz; %10 ile %11 arasındaki fark küçük görünür, 5 milyon ise 1 milyondan çok büyüktür. İki seçim de tek başına anlaşılırdır. Sorun, ikisinin birlikte beklenen fayda kuramıyla çelişmesidir.

Önce seçenekler arasındaki ilişkiyi görelim. A ve B'de ortak bir parça vardır: ikisi de %89 olasılıkla 1 milyon verir (A'nın %100'ünün %89'u, B'nin %89'u). Bu ortak %89 “1 milyon” parçası, iki seçenekte de %89 “0” ile değiştirilirse A'dan C, B'den D elde edilir. A, %11 olasılıkla 1 milyon ve %89 olasılıkla 0'a dönüşür; bu C'dir. B, %10 olasılıkla 5 milyon, %1 + %89 = %90 olasılıkla 0'a dönüşür; bu D'dir. Bağımsızlık aksiyomuna göre ortak parçayı değiştirmek sıralamayı değiştirmemelidir.

Örnek 1.4

Herhangi bir u fayda fonksiyonu için iki farkı hesaplayalım. Kısalık için $u(1)$, $u(5)$ ve $u(0)$ sırasıyla 1 milyonun, 5 milyonun ve 0'ın faydasıdır.

1. $E[u(A)] = u(1)$, çünkü A kesindir.
2. $E[u(B)] = 0{,}89 \cdot u(1) + 0{,}10 \cdot u(5) + 0{,}01 \cdot u(0)$.
3. Fark: $E[u(A)] - E[u(B)] = (1 - 0{,}89) \cdot u(1) - 0{,}10 \cdot u(5) - 0{,}01 \cdot u(0) = 0{,}11 \cdot u(1) - 0{,}10 \cdot u(5) - 0{,}01 \cdot u(0)$.
4. $E[u(C)] = 0{,}11 \cdot u(1) + 0{,}89 \cdot u(0)$ ve $E[u(D)] = 0{,}10 \cdot u(5) + 0{,}90 \cdot u(0)$.
5. Fark: $E[u(C)] - E[u(D)] = 0{,}11 \cdot u(1) - 0{,}10 \cdot u(5) + (0{,}89 - 0{,}90) \cdot u(0) = 0{,}11 \cdot u(1) - 0{,}10 \cdot u(5) - 0{,}01 \cdot u(0)$.

İki fark tam olarak aynı ifadedir. Bu ifade pozitifse hem A ≻ B hem C ≻ D olmalıdır. Negatifse hem B ≻ A hem D ≻ C olmalıdır. "A ve D" seçimi, farkın aynı anda hem pozitif hem negatif olmasını gerektirir. Bu imkânsızdır; yani "A ve D" seçimini açıklayan hiçbir fayda fonksiyonu yoktur.

Hesap, hiçbir fayda fonksiyonu seçilmeden yapıldı. $u(x) = x$, $u(x) = \sqrt{x}$ ya da başka herhangi bir fonksiyon alınsa da sonuç değişmez. Örneğin $u(x) = x$ ile iki fark da $0{,}11 - 0{,}50 = -0{,}39$'dur; bu kişi B'yi ve D'yi seçer. Dört seçeneğin beklenen değerleri de birbirinden farklıdır: A 1 milyon, B 1,39 milyon, C 0,11 milyon, D 0,5 milyondur. Yani seçimler "her şey eşit olduğu için keyfi" de değildir.

Allais'nin seçimleri hangi aksiyomu çiğner? Geçişliliği değil, **bağımsızlığı**. İnsanlar kesinliğe, olasılık hesabının söylediğinden daha fazla ağırlık verir. %100 ile %99 arasındaki fark, %11 ile %10 arasındaki farktan çok daha büyük hissedilir. Bu olguya **kesinlik etkisi** (*certainty effect*) denir. Kesinlik etkisi, insanların gerçek seçimlerini açıklamaya çalışan **beklenti kuramı**nın (*prospect theory*) çıkış noktalarından biridir. Beklenti kuramı olasılıkları doğrudan değil, bir ağırlık fonksiyonundan geçirerek kullanır; küçük olasılıkları büyütür, kesinliğe yakın olasılıkları küçültür.

Allais paradoksu beklenen fayda kuramını geçersiz kılmaz. Kuram, iki farklı amaçla kullanılır. *Betimleyici* olarak, insanların nasıl karar verdiğini anlatmaya çalışır; burada Allais gibi sapmalar önemlidir. *Kural koyucu* (normatif) olarak ise tutarlı bir karar vericinin nasıl karar vermesi gerektiğini söyler. Algoritmalar ve yapay ajanlar için genellikle ikinci kullanım önemlidir: bir yazılım ajanının tercihlerinin tutarlı olması istenir.

1.1.8 Fayda sayıları neyi ölçer?

Beklenen fayda hesabında $u(400) = 20$ gibi sayılar kullanılır. Bu sayıların bir birimi var mıdır? "20 birim mutluluk" demek anlamlı mıdır? Cevap, fayda ölçeğinin bir termometreye benzediğidir. Celsius ve Fahrenheit ölçekleri farklı sıfır noktası ve farklı birim kullanır; suyun donma noktası birinde 0, ötekinde 32'dir. Ama iki ölçek de sıcaklıkları aynı sıraya dizer ve biri ötekinden $F = 1{,}8 \cdot C + 32$ formülüyle elde edilir. Fayda da böyledir: sıfır noktası ve birimi keyfîdir.

Tanım 1.4

$v = a \cdot u + b$ ($a > 0$) biçimindeki dönüşüme **pozitif afin dönüşüm** (*affine transformation*) denir. Dönüşüm ölçeği a kat büyütür ve b kadar kaydırır.

Teorem 1.2 (Fayda ölçeğinin tekliği)

u bir karar vericinin vNM fayda fonksiyonuysa, her pozitif afin dönüşüm $v = a \cdot u + b$ de aynı tercihleri temsil eder: hiçbir piyango kararını değiştirmez. Afin olmayan dönüşümler, artan olsalar bile, piyango kararlarını değiştirebilir.

Teoremin birinci yarısı kısa bir hesapla gösterilir. Bir piyangonun v ile beklenen faydası

$$E[v(L)] = \Sigma\, p_i\,(a \cdot u(x_i) + b) = a \cdot \Sigma\, p_i\, u(x_i) + b \cdot \Sigma\, p_i = a \cdot E[u(L)] + b$$

olur. İlk eşitlik tanımdır. İkincisi, toplamı iki parçaya ayırıp sabitleri dışarı çıkarmaktır. Üçüncüsü olasılıkların toplamının 1 olmasından gelir. Demek ki her piyangonun yeni beklenen faydası, eskisinin a ile çarpılıp b eklenmiş hâlidir. $a > 0$ olduğu için büyük olan büyük kalır; sıralama değişmez. $a < 0$ olsaydı sıralama tersine dönerdi. Bu yüzden koşul $a > 0$'dır.

Fayda	Piyango A	Kesin B	Seçim
$u = \sqrt{x}$	15	20	B
$v = 10\sqrt{x} + 5$	155	205	B
$w = (\sqrt{x})^2 = x$	450	400	A

Tablo 1.3: A (½: 900 TL; ½: 0 TL) ve kesin B (400 TL) üç farklı fayda ölçeğiyle.

Örnek 1.5

Tablo 1.3'teki sayılar şöyle bulunur.

1. $v = 10\sqrt{x} + 5$ afin bir dönüşümdür ($a = 10$, $b = 5$). $v(900) = 305$, $v(0) = 5$; $E[v(A)] = ½ \cdot 305 + ½ \cdot 5 = 155$. Formülle de: $10 \cdot 15 + 5 = 155$. Kesin B için $v(400) = 10 \cdot 20 + 5 = 205$. $205 > 155$, seçim yine B'dir.
2. $w = (\sqrt{x})^2$ de artan bir dönüşümdür: $\sqrt{x}$ büyüdükçe karesi de büyür. Ama afin değildir. $w(x) = x$ olduğu için $E[w(A)] = 450$ ve $w(B) = 400$ olur; seçim A'ya döner.

Kesin sonuçlar arasındaki sıralamayı her artan dönüşüm korur (400 TL, 0 TL'den hep iyidir). Piyangolar arasındaki sıralamayı ise yalnız afin dönüşümler korur. Bu yüzden vNM faydasına **kardinal fayda** (*cardinal utility*) denir: yalnız sıra değil, farkların oranları da anlamlıdır.

Bu sonucun oyun kuramında doğrudan bir karşılığı vardır. Bir oyuncunun kazanç tablosundaki bütün sayıları 2 ile çarpıp 5 eklemek, o oyuncunun hiçbir kararını değiştirmez. Bir oyunu incelerken kazançları daha kolay hesaplanan sayılara dönüştürmek bu yüzden serbesttir. Ama kazançların karesini almak gibi afin olmayan bir dönüşüm, riskli seçimleri değiştirebilir.

Sık yapılan hata

- **"u(B) − u(A) = 5 birim mutluluk."** Farkın büyüklüğü ölçek seçimine bağlıdır: v ölçeğinde aynı fark 50'dir. Anlamlı olan, aynı kişinin seçenekleri arasındaki sıralamadır.
- **"Ayşe'nin faydası 20, Bora'nınki 15; Ayşe daha mutlu."** Her kişinin ölçeği ayrı ayrı, keyfî olarak kaydırılıp büyütülebilir. vNM kuramı kişiler arasında fayda karşılaştırması yapmaz.

- **"Her artan dönüşüm kararı korur."** Kesin sonuçlar için doğrudur; piyangolar için yalnız pozitif afin dönüşümler kararı korur.

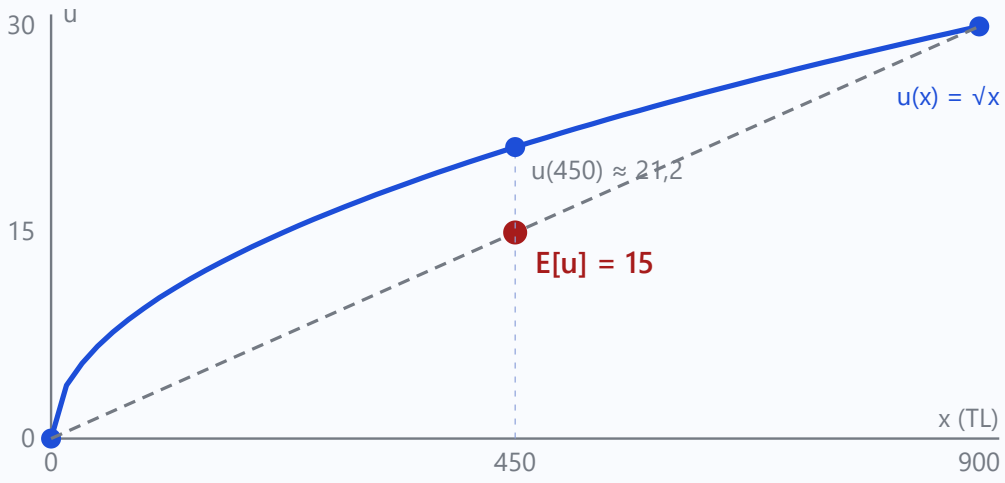
1.2 Risk tutumları

Aynı piyangoya farklı insanlar farklı tepki verir. Kimi kesin tutarı tercih eder, kimi şansını dener. Bu farkı fayda fonksiyonunun biçimi açıklar. Risk tutumu iki sayıyla ölçülür: kesinlik eşdeğeri ve risk primi. Bu iki sayı sigortanın neden var olduğunu da açıklar. Olasılıklar hiç bilinmediğinde ise başka ölçütler gerekir.

1.2.1 İçbükey fayda ve azalan marjinal fayda

$u(x) = \sqrt{x}$ fonksiyonunun değerlerine bakalım: $u(0) = 0$, $u(100) = 10$, $u(400) = 20$, $u(900) = 30$. İlk 100 TL faydayı 10 artırır. 100'den 400'e çıkmak, yani 300 TL daha eklemek, faydayı yine yalnız 10 artırır. 400'den 900'e çıkmak için 500 TL gerekir. Her ek 10 birim fayda için gereken para giderek büyür. Başka bir deyişle, her ek lira bir öncekinden daha az fayda katar. Buna **azalan marjinal fayda** (*diminishing marginal utility*) denir. "Marjinal", "bir birim daha eklenince" anlamındadır.

Azalan marjinal faydalı bir fonksiyonun grafiği aşağı doğru bükülür; böyle bir fonksiyona **içbükey** (*concave*) denir. İçbükeyliğin geometrik bir tanımı vardır: eğri üzerindeki iki noktayı birleştiren doğru parçası (**kiriş**), eğrinin altında kalır. Tersine, yukarı doğru bükülen fonksiyona **dışbükey** (*convex*) denir; kirişi eğrinin üstündedir. Düz bir doğrunun kirişi doğrunun kendisidir.



Şekil 1.3: İçbükey fayda $u(x) = \sqrt{x}$. Kesikli çizgi, eğrinin iki ucunu, (0, 0) ile (900, 30) noktalarını birleştiren kiriştir. Piyango A'nın beklenen faydası kirişin ortasıdır: $E[u] = 15$. Aynı 450 TL kesin verilse faydası $u(450) \approx 21{,}2$ olur ve eğri kirişin üstünde kalır.

Kiriş, piyangonun beklenen faydasını gözle görmeyi sağlar. Piyango A iki uçtan birini eşit olasılıkla verir. Beklenen faydası iki ucun faydalarının ortalamasıdır: $½ \cdot 0 + ½ \cdot 30 = 15$. Bu, kirişin tam ortasındaki noktanın yüksekliğidir ve yatay eksende $x = 450$ hizasına düşer; çünkü beklenen değer de iki ucun ortalamasıdır. Şekil 1.3'te aynı hizada eğrinin yüksekliği $\sqrt{450} \approx 21{,}2$'dir. Eğri kirişin üstündedir: 450 TL'yi kesin almak, ortalaması 450 TL olan piyangodan daha fazla fayda verir.

Teorem 1.3 (Jensen eşitsizliği)

u içbükey ise her piyango X için $E[u(X)] \le u(E[X])$'dir. u dışbükey ise eşitsizlik ters yöndedir: $E[u(X)] \ge u(E[X])$. u doğrusal ise iki taraf eşittir.

Jensen eşitsizliği, kiriş gözleminin genel hâlidir. Sol taraf, piyangonun beklenen faydasıdır (kiriş üzerindeki nokta). Sağ taraf, beklenen değerin kesin verilmesinin faydasıdır (eğri üzerindeki nokta). İçbükey faydalı biri için piyangonun kendisi, ortalamasının kesin verilmesinden kötüdür. Riskten kaçınmanın matematiksel tanımı budur.

1.2.2 Kesinlik eşdeğeri ve risk primi

Risk tutumunu sayıyla ifade etmenin yolu şu soruyu sormaktır: bu kişi piyangoyu hangi kesin tutara satmaya razıdır?

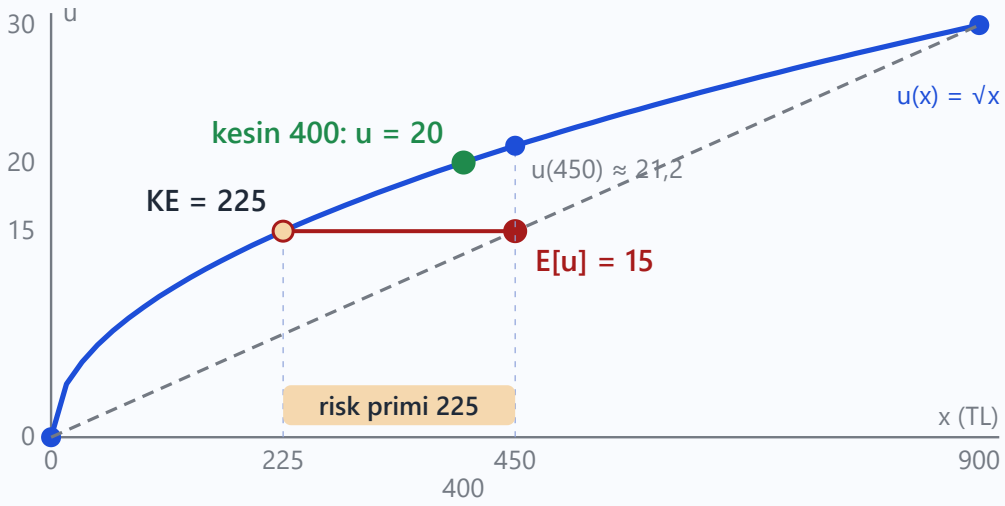
Tanım 1.5

Bir piyango X'in **kesinlik eşdeğeri** (*certainty equivalent*, KE), faydası piyangonun beklenen faydasına eşit olan kesin tutardır:

> $$u(KE) = E[u(X)], \text{ yani } KE = u^{-1}(E[u(X)]).$$

Risk primi (*risk premium*), beklenen değer ile kesinlik eşdeğeri arasındaki farktır: *Risk primi* $= E[X] - KE$.

u^{-1}, u'yu geri çözmek demektir: faydası verilen tutarı bulmak. $u(x) = \sqrt{x}$ için faydası 15 olan tutar, $\sqrt{x} = 15$ denkleminin çözümüdür. İki tarafın karesi alınır: $x = 15^2 = 225$. Yani $\sqrt{x}$ için $u^{-1}(y) = y^2$'dir.



Şekil 1.4: Kesinlik eşdeğeri ve risk primi. Kirişin ortasındaki *E[u] = 15* noktasından yatay gidilince eğriye *x = 225*'te varılır: KE = 225 TL. Beklenen değer 450 ile KE arasındaki 225 TL risk primidir. Kesin 400 TL'nin faydası 20'dir ve 15'ten büyüktür.

Örnek 1.6

Piyango A (½: 900 TL; ½: 0 TL) ve $u(x) = \sqrt{x}$ için kesinlik eşdeğeri ile risk primini adım adım bulalım. Şekil 1.4 her adımı gösterir.

1. **Uçların faydası.** $u(0) = 0$, $u(900) = 30$. Bunlar eğrinin iki uç noktasıdır.
2. **Beklenen fayda.** $E[u] = ½ \cdot 0 + ½ \cdot 30 = 15$. Bu, kirişin ortasıdır.
3. **Karşılaştırma.** Beklenen değer $E[X] = 450$ TL'dir. Kesin 450 TL'nin faydası $\sqrt{450} \approx 21{,}2 > 15$'tir. Kişi beklenen değeri kesin almayı piyangoya tercih eder.
4. **Kesinlik eşdeğeri.** Faydası 15 olan kesin tutar: $\sqrt{KE} = 15 \Rightarrow KE = 225$ TL. Bu kişi piyangoyu 225 TL'ye satmaya razıdır.
5. **Risk primi.** $450 - 225 = 225$ TL. Kişi, riski taşımamak için beklenen değerden 225 TL vazgeçmeye razıdır.
6. **Karar.** Kesin 400 TL'nin faydası 20'dir ve 15'ten büyüktür. Bu kişi B'yi seçer. Bir başka yoldan da aynı sonuca varılır: 400 TL, KE = 225 TL'den büyüktür.

Son adımdaki ikinci yol önemli bir kısayoldur. Bir piyangoyu kesin bir tutarla karşılaştırmak için faydaları karşılaştırmak yerine KE'yi kesin tutarla karşılaştırmak yeterlidir. KE kesin tutardan büyükse piyango, küçükse kesin tutar seçilir. Kısayol, *u* artan olduğu için işe yarar: büyük tutarın faydası da büyüktür.

Kesinlik eşdeğeri bir para tutarı olduğu için fayda sayılarından farklı olarak kişiler arasında karşılaştırılabilir. “Bu piyangoyu Ayşe 225 TL'ye, Bora 300 TL'ye satar” demek anlamlıdır ve Bora'nın riskten daha az kaçındığını gösterir. KE ayrıca ölçek seçiminden bağımsızdır: $v = 10\sqrt{x} + 5$ kullanılsa da fayda-

sı $E[v] = 155$ olan kesin tutar yine 225 TL çıkar ($10\sqrt{x} + 5 = 155 \Rightarrow \sqrt{x} = 15$).

1.2.3 Üç risk tutumu

Fayda fonksiyonunun biçimi üç tür risk tutumu tanımlar. Tablo 1.4 aynı piyango A ve kesin B için üç fayda fonksiyonunun kararını karşılaştırır.

Tutum	Fayda	E[u(A)]	u(400)	Seçim	KE (TL)	Risk primi
Riskten kaçınan	$\sqrt{x}$ (içbükey)	15	20	B	225	225
Risk nötr	x (doğrusal)	450	400	A	450	0
Risk seven	$x^2/900$ (dışbükey)	450	177,8	A	636,4	−186,4

Tablo 1.4: Üç risk tutumu. Piyango A: %50 olasılıkla 900 TL, %50 olasılıkla 0 TL ($E[X] = 450$); B: kesin 400 TL.

Örnek 1.7

Tablo 1.4'ün ikinci ve üçüncü satırları şöyle hesaplanır.

1. **Risk nötr, $u(x) = x$.** Fayda paranın kendisidir. $E[u(A)] = \tfrac{1}{2} \cdot 900 + \tfrac{1}{2} \cdot 0 = 450$, $u(400) = 400$. $450 > 400$, seçim A. Faydası 450 olan tutar 450'dir: KE = 450, risk primi $450 - 450 = 0$.
2. **Risk seven, $u(x) = x^2/900$.** Uçlar: $u(0) = 0$, $u(900) = 900^2/900 = 900$. $E[u(A)] = \tfrac{1}{2} \cdot 0 + \tfrac{1}{2} \cdot 900 = 450$.
3. Kesin 400'ün faydası: $400^2/900 = 160000/900 \approx 177{,}8$. $450 > 177{,}8$, seçim A.
4. KE: $x^2/900 = 450 \Rightarrow x^2 = 405000 \Rightarrow x = \sqrt{405000} \approx 636{,}4$ TL.
5. Risk primi: $450 - 636{,}4 = -186{,}4$ TL. Prim negatiftir.

Negatif risk primi ne demektir? Risk seven kişi, ortalaması 450 TL olan piyangoyu 636,4 TL'den aşağıya satmaz. Yani riski taşımak için beklenen değerin üstünde bir bedel ödemeye razıdır. Üç tutum şöyle özetlenir:

- **İçbükey → riskten kaçınan** (*risk-averse*). $KE < E[X]$, prim > 0. Riski üstlenmek için bir ödül ister. İnsanların çoğu, özellikle büyük tutarlarda, böyle davranır.
- **Doğrusal → risk nötr** (*risk-neutral*). $KE = E[X]$, prim = 0. Yalnız beklenen değere bakar. Büyük şirketler küçük tutarlı kararlarda yaklaşık böyle davranır.
- **Dışbükey → risk seven** (*risk-loving*). $KE > E[X]$, prim < 0. Risk için para öder. Beklenen değeri bilet fiyatından düşük bir piyango bileti alan kişi buna örnektir.

Bu üç durum Jensen eşitsizliğinden doğrudan çıkar. İçbükey u için $E[u(X)] < u(E[X])$'dir. KE'nin faydası sol tarafa eşittir, beklenen değerin faydası sağ taraftır. u artan olduğundan faydası küçük olan tutar da küçüktür: $KE < E[X]$. Dışbükey için her şey ters yöndedir.

Sık yapılan hata

- **KE'yi beklenen faydayla karıştırmak.** $E[u] = 15$ bir fayda sayısıdır; KE = 225 ise bir para tutarıdır. KE'yi bulmak için faydayı paraya geri çevirmek (u^{-1}) gerekir.
- **Risk primini *KE − E[X]* diye yazmak.** Tanım $E[X] - KE$'dir. Ters yazılırsa riskten kaçınan kişinin primi negatif çıkar ve yorum tersine döner.
- **"Risk seven her zaman piyangoyu seçer."** Risk seven, beklenen değerleri eşitken piyangoyu seçer. Piyangonun beklenen değeri kesin tutardan çok düşükse o da kesin tutarı seçebilir.

1.2.4 Sigorta neden var?

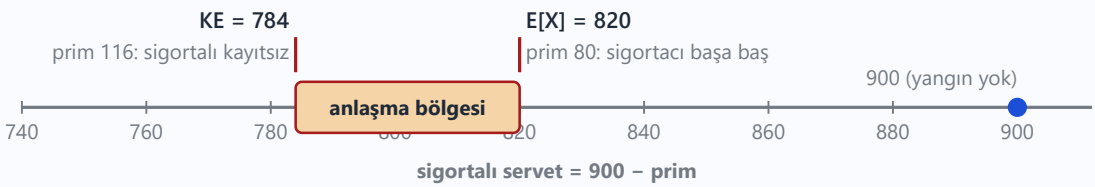
Risk tutumunun en önemli uygulamalarından biri sigortadır. Sigortalı, beklenen kaybından daha yüksek bir prim öder. Sigortacı da bu primden kâr eder. İki tarafın da kazançlı çıktığı bir anlaşma nasıl mümkündür? Cevap, iki tarafın risk tutumlarının farklı olmasıdır.

Örnek 1.8

Bir kişinin serveti 900 (bin TL) olsun. %10 olasılıkla bir yangın çıkar ve 800'lük kayıp olur; geriye 100 kalır. Fayda $u(x) = \sqrt{x}$'tir. Kişi, kaybın tamamını karşılayan bir sigortaya en fazla ne öder?

1. **Sigortasız beklenen fayda.** %90 olasılıkla servet 900, %10 olasılıkla 100'dür. $E[u] = 0{,}9 \cdot \sqrt{900} + 0{,}1 \cdot \sqrt{100} = 0{,}9 \cdot 30 + 0{,}1 \cdot 10 = 27 + 1 = 28$.
2. **Kesinlik eşdeğeri.** $\sqrt{KE} = 28 \Rightarrow KE = 28^2 = 784$. Kişi riskli 900 yerine kesin 784'e razıdır.
3. **En yüksek prim.** Sigortalı servet $900 - prim$'dir ve kesindir. Bu tutar 784'ten az olmamalıdır: $900 - prim \geq 784 \Rightarrow prim \leq 116$.
4. **Beklenen kayıp.** $0{,}1 \cdot 800 = 80$. Beklenen servet $900 - 80 = 820$'dir.
5. **Sigortacının alt sınırı.** Sigortacı çok sayıda bağımsız poliçe satar. Yangınların bir kısmı gerçekleşir, ama toplam ödeme beklenen kayıplar toplamına yakın kalır. Bu yüzden sigortacı yaklaşık risk nötrdür ve 80'den fazla her prim ona beklenen kâr getirir.

Sonuç: 80 ile 116 arasındaki her prim iki taraf için de kazançlıdır. Örneğin 100'lük primde sigortalının serveti kesin 800 olur; faydası $\sqrt{800} \approx 28{,}3 > 28$'dir. Sigortacı poliçe başına ortalama 20 kâr eder.



Şekil 1.5: Sigortanın anlaşma bölgesi. Yatay eksen sigortalının kesin servetidir (900 − prim). Prim 80 iken servet 820'dir ve sigortacı başa baş kalır. Prim 116 iken servet 784'tür (KE) ve sigortalı kayıtsız kalır. Arada kalan her prim iki tarafa da kazandırır.

Şekil 1.5 bu aralığı bir sayı doğrusu üzerinde gösterir. Anlaşma bölgesinin genişliği, *116 − 80 = 36*, tam olarak sigortalının risk primidir: beklenen servet 820 ile kesinlik eşdeğeri 784 arasındaki fark. Risk primi ne kadar büyükse, yani kişi riskten ne kadar çok kaçınıyorsa, sigortanın yaratabileceği kazanç da o kadar büyüktür.

Sigortanın ekonomideki işlevi buradan anlaşılır: risk, ona en kolay katlanabilen tarafa aktarılır. Bireyler için büyük bir kayıp felaket olabilir; sigortacı için ise binlerce poliçeden biridir. Aynı mantık, büyük şirketlerin neden küçük kararlarda risk nötr davranabildiğini de açıklar: tek bir kararın sonucu, toplam servetlerinin küçük bir parçasıdır.

1.2.5 Ellsberg torbası: belirsizlikten kaçınma

Şimdiye kadar olasılıkların bilindiğini varsaydık. Zar atarken her yüzün 1/6 olasılıkla geldiğini biliriz. Ama yeni bir ürünün başarı olasılığını kimse bilmez. Risk altında olasılıklar bilinir; belirsizlik altında bilinmez. İnsanlar bu iki durumda aynı davranmaz. Daniel Ellsberg 1961'de bunu basit bir deneyle gösterdi (Ellsberg, 1961).

Bir torbada 90 top vardır. 30'u kırmızıdır. Kalan 60 top siyah ya da sarıdır, ama kaçının siyah olduğu bilinmez. Torbadaki siyah top sayısına *s* diyelim; *s*, 0 ile 60 arasında herhangi bir sayı olabilir. Bir top çekilir ve kazandıran renk çekilirse 100 TL ödenir. İki bahis sorulur.

Bahis	Seçenek	Kazandıran top sayısı
1	kırmızı	30 (bilinen)
	siyah	*s* (bilinmiyor)
2	kırmızı ya da sarı	*30 + (60 − s)*
	siyah ya da sarı	60 (bilinen)

Tablo 1.5: Ellsberg torbası. *s*: torbadaki siyah top sayısı (0–60, bilinmiyor).

Tipik tercihler şöyledir: birinci bahiste çoğu kişi “kırmızı”yı, ikinci bahiste “siyah ya da sarı”yı seçer. İki tercihin ortak yanı, seçilen seçeneğin kazanma olasılığının bilinmesidir. Birinci bahiste kırmızının olasılığı 30/90'dır. İkinci bahiste siyah ya da sarının olasılığı 60/90'dır. Öteki seçeneklerin olasılığı ise *s*'ye bağlıdır ve bilinmez.

Örnek 1.9

Bu iki tercihi açıklayan bir s değeri var mıdır? Ödül iki bahiste de aynıdır (100 TL), bu yüzden daha fazla kazandıran topu olan seçenek tercih edilmelidir.

1. Birinci tercih "kırmızı $\succ$ siyah" ise kırmızı top siyahtan fazla olmalıdır: $30 > s$.
2. İkinci tercih "siyah ya da sarı $\succ$ kırmızı ya da sarı" ise $60 > 30 + (60 - s) = 90 - s$ olmalıdır.
3. İkinci eşitsizlik düzenlenir: $60 > 90 - s \Rightarrow s > 30$.
4. Birinci tercih $s < 30$, ikinci tercih $s > 30$ ister. İkisi aynı anda sağlanamaz.

İki tercihi birlikte açıklayan hiçbir olasılık ataması yoktur. Kişinin davranışı, torbadaki siyah top sayısına dair tutarlı bir inançla açıklanamaz.

Bu sonuç fayda fonksiyonunun biçimiyle de kurtarılamaz. Her seçenekte ödül aynıdır (100 TL ya da 0), yani fayda eğriliği yalnız iki sayıyı ölçekler; hangi seçeneğin daha olası olduğu algısını değiştirmez. Olasılığı bilinmeyen seçenekten uzak durma eğilimine **belirsizlikten kaçınma** (*ambiguity aversion*) denir. Riskten kaçınmadan farklıdır: riskten kaçınma bilinen olasılıklı sonuçların dağılımına, belirsizlikten kaçınma ise olasılıkların bilinmemesine verilen tepkidir.

Belirsizlikten kaçınmanın uç biçimi, olasılıklar hiç bilinmediğinde her seçenek için en kötü durumu düşünmek ve en kötü durumu en iyi olan seçeneği seçmektir. Bu kurala **maksimin** denir. Ellsberg torbasında maksimin şöyle düşünür: "siyah" bahsinde en kötü durum $s = 0$'dır ve kazanma olasılığı sıfırdır; "kırmızı"nın kazanma olasılığı ise hep 30/90'dır. Bu yüzden kırmızı seçilir. İkinci bahiste "kırmızı ya da sarı"nın en kötü durumu $s = 60$'tır ve olasılık 30/90'a düşer; "siyah ya da sarı" ise hep 60/90 verir. Maksimin iki tipik tercihi de açıklar.

1.2.6 Olasılıklar bilinmezse: belirsizlik ölçütleri

Bir yatırımcı üç eylemden birini seçecek: büyük yatırım, küçük yatırım ya da hiç yatırım yapmamak. Sonuç ekonominin durumuna bağlıdır: iyi, orta ya da kötü. Yatırımcı bu durumların olasılıklarını bilmiyor. Tablo 1.6 her eylemin her durumdaki kazancını gösterir. Satırlar eylemler, sütunlar durumlardır.

Eylem \ durum	İyi	Orta	Kötü
Büyük yatırım	200	50	−120
Küçük yatırım	90	40	−20
Yatırım yok	0	0	0

Tablo 1.6: Belirsizlik altında yatırım kararı. Kazançlar bin TL; durumların olasılıkları bilinmiyor.

Olasılıklar bilinmediği için beklenen fayda hesaplanamaz. Bunun yerine, "karamsarlık" dereceleri farklı ölçütler kullanılır. Bu ölçütlerin hepsi yalnız tablodaki sayılara bakar. Böyle bir probleme **belirsizlik altında karar** problemi denir.

Tanım 1.6

U[a][d], eylem *a*'nın durum *d*'deki kazancı olsun. Dört temel ölçüt şunlardır:

- **Maksimin** (*maximin*; Wald): Her eylemin en kötü sonucu bulunur; en kötü sonucu en büyük olan eylem seçilir. $max_a\ min_d\ U[a][d]$.
- **Maksimaks** (*maximax*): Her eylemin en iyi sonucu bulunur; en iyi sonucu en büyük olan seçilir. $max_a\ max_d\ U[a][d]$.
- **Laplace ölçütü** (*Laplace criterion*): Bütün durumlar eşit olası sayılır; ortalaması en büyük olan eylem seçilir.
- **Minimaks pişmanlık** (*minimax regret*; Savage): Her durumda, ulaşılabilecek en iyi sonuç ile elde edilen sonuç arasındaki farka **pişmanlık** denir. Her eylemin en büyük pişmanlığı bulunur; en büyük pişmanlığı en küçük olan eylem seçilir.

Her ölçütün arkasında bir bakış açısı vardır. Maksimin karamsardır: "Ne seçersem seçeyim doğa bana en kötüsünü yapacak" varsayımıyla hareket eder. Maksimaks iyimserdir: "Ne seçersem seçeyim en iyisi olacak" diye düşünür. Laplace "Hiçbir şey bilmiyorsam her duruma eşit olasılık vermeliyim" der; bu yüzden bilgisizliği eşit olasılıkla temsil eder. Savage'in ölçütü sonucun kendisine değil, sonradan duyulacak pişmanlığa bakar (Savage, 1951).

Örnek 1.10

Tablo 1.6'ya dört ölçütü uygulayalım.

1. **Maksimin.** Satır minimumları: Büyük −120, Küçük −20, Yok 0. En büyüğü 0'dır. Karar: *Yatırım yok*.
2. **Maksimaks.** Satır maksimumları: Büyük 200, Küçük 90, Yok 0. En büyüğü 200'dür. Karar: *Büyük yatırım*.
3. **Laplace.** Satır ortalamaları: Büyük $(200 + 50 - 120)/3 = 130/3 \approx 43{,}3$; Küçük $(90 + 40 - 20)/3 = 110/3 \approx 36{,}7$; Yok 0. En büyüğü 43,3'tür. Karar: *Büyük yatırım*.
4. **Pişmanlık tablosu.** Önce her durumda ulaşılabilecek en iyi sonuç bulunur (sütun maksimumları): İyi 200, Orta 50, Kötü 0. Pişmanlık = en iyi − elde edilen. Büyük: $(200 - 200, 50 - 50, 0 - (-120)) = (0, 0, 120)$. Küçük: $(110, 10, 20)$. Yok: $(200, 50, 0)$.
5. **Savage.** Satır başına en büyük pişmanlık: Büyük 120, Küçük 110, Yok 200. En küçüğü 110'dur. Karar: *Küçük yatırım*.

Aynı tablo üç farklı karar üretmiştir: maksimin “Yok”, maksimaks ve Laplace “Büyük”, Savage “Küçük”.

eylem \ durum	İyi	Orta	Kötü	min	max	ort.	maks. pişm.
Büyük yatırım	200	50	−120	−120	200	43,3	120
Küçük yatırım	90	40	−20	−20	90	36,7	110
Yatırım yok	0	0	0	0	0	0	200
sütun en iyisi:	**200**	**50**	**0**				

Şekil 1.6: Belirsizlik ölçütlerinin özet tablosu. Solda kazançlar ve her sütunun en iyisi (200, 50, 0); ortada satır minimumu, maksimumu ve ortalaması; sağda her eylemin en büyük pişmanlığı.

Şekil 1.6 bütün ara sonuçları tek tabloda toplar. Pişmanlık hesabında sık bir yanılgı, pişmanlığı satırın en iyisine göre hesaplamaktır. Oysa pişmanlık sütunun en iyisine göre hesaplanır: “Bu durumun gerçekleşeceğini bilseydim ne seçerdim ve ne kaybettim?” Kötü durumda en iyisi hiç yatırım yapmamaktır (0). Büyük yatırım yapan kişi kötü durumda −120 alır ve 120 pişmanlık duyar.

Bu dört ölçüte beşinci bir ölçüt eklenebilir. **Hurwicz ölçütü** (*Hurwicz criterion*) iki uç arasında durur. Bir **iyimserlik katsayısı** α (0 ile 1 arasında) seçilir ve her eylem

$$\alpha \cdot (\text{satır maks.}) + (1 - \alpha) \cdot (\text{satır min.})$$

ile puanlanır. $\alpha = 0$ maksimin, $\alpha = 1$ maksimakstır. $\alpha = 0{,}5$ ile puanlar: Büyük $0{,}5 \cdot 200 + 0{,}5 \cdot (-120) = 40$, Küçük $0{,}5 \cdot 90 + 0{,}5 \cdot (-20) = 35$, Yok 0. Karar Büyük yatırımdır.

Örnek 1.11

Hurwicz ölçütünün kararı α değiştikçe nasıl değişir?

1. Puanlar α'ya bağlı yazılır: Büyük $200\alpha - 120(1 - \alpha) = 320\alpha - 120$; Küçük $90\alpha - 20(1 - \alpha) = 110\alpha - 20$; Yok 0.
2. Küçük, Yok'tan iyi olur: $110\alpha - 20 > 0 \Rightarrow \alpha > 2/11 \approx 0{,}18$.
3. Büyük, Küçük'ten iyi olur: $320\alpha - 120 > 110\alpha - 20 \Rightarrow 210\alpha > 100 \Rightarrow \alpha > 10/21 \approx 0{,}48$.

Sonuç: $\alpha < 0{,}18$ iken Yok, $0{,}18 < \alpha < 0{,}48$ iken Küçük, $\alpha > 0{,}48$ iken Büyük yatırım seçilir. Ne kadar iyimser olunursa o kadar büyük risk alınır.

Hangi ölçüt doğrudur? Bu sorunun tek bir cevabı yoktur. Olasılıklar tahmin edilebiliyorsa en iyisi onları tahmin edip beklenen faydaya dönmektir. Tahmin edilemiyorsa ölçüt seçimi karar vericinin tutumunu yansıtır. Felaket riskinin kabul edilemediği durumlarda (bir köprünün tasarımı, bir hastanın te-

davisi) maksimin makul olabilir. Sık tekrarlanan küçük kararlarda Laplace ya da beklenen fayda daha uygundur. Oyun kuramında maksimin özel bir yer tutar: iki kişilik sıfır toplamlı oyunlarda rakip gerçekten en kötüsünü yapmak isteyen biridir ve maksimin doğal bir ölçüt hâline gelir.

Sık yapılan hata

- **Pişmanlığı satır içinde hesaplamak.** Pişmanlık, aynı *durumdaki* (sütundaki) en iyi sonuçla karşılaştırılır.
- **Savage'de en büyük pişmanlığı seçmek.** Önce her satırın en büyük pişmanlığı bulunur, sonra bunların en *küçüğü* seçilir. Adı da buradan gelir: minimaks pişmanlık.
- **Maksimin ile minimaksı karıştırmak.** Kazanç tablosunda maksimin kullanılır (en kötü kazancı büyütmek). Maliyet ya da kayıp tablosunda karşılığı minimakstır (en büyük kaybı küçültmek).

1.2.7 Kod izleme: olcutler()

Dört ölçüt kısa bir Python fonksiyonuyla hesaplanabilir. Fonksiyon kazanç tablosunu iç içe bir liste olarak alır: `U[i][j]`, eylem `i`'nin durum `j`'deki kazancıdır. Dört ölçütün seçtiği eylemlerin indislerini döndürür.

```
def olcutler(U):
    A, m = range(len(U)), len(U[0])
    mm = max(A, key=lambda i: min(U[i]))      # maximin
    mx = max(A, key=lambda i: max(U[i]))      # maximax
    lp = max(A, key=lambda i: sum(U[i]) / m)  # Laplace
    enb = [max(r[j] for r in U) for j in range(m)]
    pis = [max(enb[j] - r[j] for j in range(m))
           for r in U]            # en büyük pişmanlık
    sv = pis.index(min(pis))                  # Savage
    return mm, mx, lp, sv
```

Kod satır satır şöyle çalışır:

- `A` eylem indislerinin aralığıdır (0, 1, 2); `m` durum sayısıdır (3).
- `max(A, key=…)`, `A` içindeki indisler arasından `key` fonksiyonunun değerini en büyük yapanı döndürür. `lambda i: min(U[i])` küçük, adsız bir fonksiyondur: `i` indisini alır ve o satırın en küçük değerini verir. Bu satır maksimin kuralının birebir karşılığıdır.
- `mx` aynı yapıyı satır maksimumlarıyla, `lp` satır ortalamalarıyla kurar.
- `enb` listesi her sütunun en büyük değeridir: `r[j] for r in U`, bütün satırların `j`-inci öğesi üzerinde gezinir.
- `pis` listesi her satır için en büyük pişmanlığı tutar: her sütunda `enb[j] - r[j]` hesaplanır ve bunların en büyüğü alınır.

- `sv`, `pis` listesindeki en küçük değerin indisidir.

Adım	Hesaplanan	Değer
1	`A`, `m`	range(3), 3
2	satır min. −120, −20, 0	`mm = 2` (Yok)
3	satır maks. 200, 90, 0	`mx = 0` (Büyük)
4	ortalama 43,3, 36,7, 0,0	`lp = 0` (Büyük)
5	sütun maks.	`enb = [200, 50, 0]`
6	en büyük pişmanlık	`pis = [120, 110, 200]`
7	en küçük pişmanlığın indisi	`sv = 1` (Küçük)
8	dönüş	`(2, 0, 0, 1)`

Tablo 1.7: `olcutler(U)` izlemesi, Tablo 1.6'daki kazançlarla (0: Büyük, 1: Küçük, 2: Yok).

Fonksiyon `(2, 0, 0, 1)` döndürür. Bu dörtlü, elle bulunan sonuçların aynısıdır: maksimin Yok, maksimaks ve Laplace Büyük, Savage Küçük. Bir ayrıntı önemlidir: `max` eşitlik durumunda ilk bulduğu indisi döndürür. İki eylemin en kötü sonucu eşit olsaydı kod listede önce geleni seçerdi. Elle hesapta da eşitliklerin nasıl bozulacağına önceden karar vermek gerekir.

Laplace satırında toplam `m`'ye bölünmüştür. Bütün satırlar aynı sayıya bölündüğü için bölme sıralamayı değiştirmez; kod yalnız toplamlara bakarak da aynı kararı verirdi. Bölme, sonucu "ortalama kazanç" olarak yorumlanabilir kılmak içindir. Bu, 1.1.8'deki afin dönüşüm sonucunun küçük bir örneğidir: bütün puanları pozitif bir sayıyla çarpmak sıralamayı korur.

1.3 Bayesçi karar

Beklenen fayda hesabı olasılıkları kullanır. Peki olasılıklar yeni bir bilgi geldiğinde nasıl değişir? Bir test pozitif çıktığında hastalık olasılığı ne olur? Bir hava tahmini "yağmur" dediğinde yağmur olasılığı ne olur? İnancı kanıtla güncellemenin kuralı Bayes kuralıdır. Güncellenmiş olasılık, beklenen fayda hesabına doğrudan girer.

1.3.1 Koşullu olasılık

Bayes kuralını anlamak için önce koşullu olasılık gerekir. *P(A)*, A olayının olasılığıdır. *P(A | B)* ise "B olduğu bilindiğinde A'nın olasılığı" demektir ve "B verildiğinde A'nın olasılığı" diye okunur. Dikey çizginin sağı bilinen bilgidir, solu sorulan olaydır.

Tanım 1.7

P(B) > 0 iken **koşullu olasılık** (*conditional probability*)

$$P(A \mid B) = P(A \text{ ve } B) / P(B)$$

olarak tanımlanır. Buradan çarpım kuralı çıkar: $P(A \text{ ve } B) = P(B \mid A) \cdot P(A)$.

Tanımı bir sayma örneğiyle düşünmek kolaydır. Bir sınıfta 40 öğrenci olsun; 10'u gözlüklü, gözlüklülerin 4'ü kız olsun. Rastgele seçilen öğrencinin gözlüklü olduğu bilindiğinde kız olma olasılığı *4/10 = 0,4*'tür. Formülle: *P(kız ve gözlüklü) = 4/40*, *P(gözlüklü) = 10/40*; oran *(4/40)/(10/40) = 4/10*. Koşul, sayılan topluluğu daraltır: artık yalnız gözlüklülere bakılır.

En önemli nokta şudur: *P(A | B)* ile *P(B | A)* genellikle farklıdır. Gözlüklü bir öğrencinin kız olma olasılığı 0,4'tür. Bir kız öğrencinin gözlüklü olma olasılığı ise sınıftaki kız sayısına bağlıdır; 20 kız varsa *4/20 = 0,2*'dir. İki olasılık aynı "kız ve gözlüklü" grubunu farklı topluluklara böler. Bu iki olasılığı karıştırmak, Bayesçi akıl yürütmedeki en yaygın hatanın kaynağıdır.

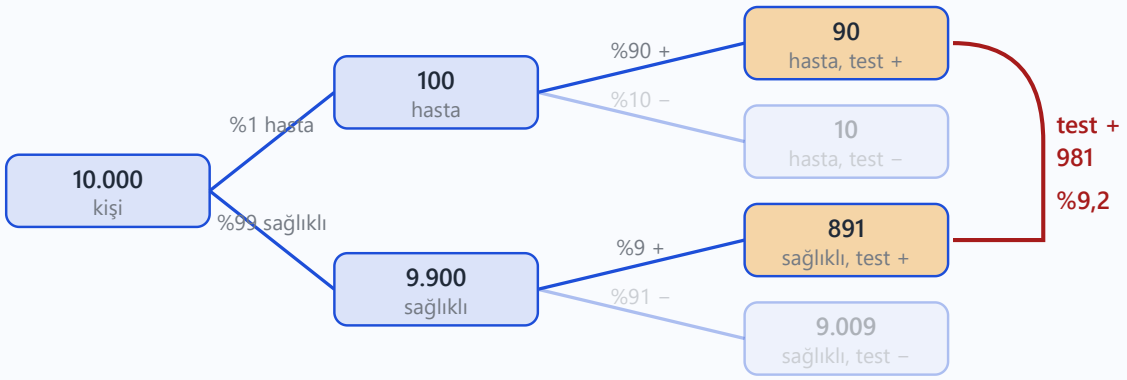
Bayesçi karar dilinde üç terim sık geçer:

- **Ön olasılık** (*prior*): Kanıt görülmeden önceki inanç. Örneğin bir hastalığın toplumdaki yaygınlığı, *P(hasta)*.
- **Olabilirlik** (*likelihood*): Gerçek durum bilindiğinde gözlenen kanıtın olasılığı. Örneğin *P(+ | hasta)*, hasta birinin testinin pozitif çıkma olasılığı.
- **Son olasılık** (*posterior*): Kanıt görüldükten sonraki güncellenmiş inanç. Örneğin *P(hasta | +)*.

1.3.2 Tıbbi test ve doğal sıklıklar

Bir hastalık toplumda %1 yaygındır. Hastalık için bir test vardır. Testin **duyarlılığı** (*sensitivity*) %90'dır: hastaların %90'ında test pozitif çıkar. Testin yanlış pozitif oranı %9'dur: sağlıklıların %9'unda da test pozitif çıkar. Bir kişinin testi pozitif çıktı. Bu kişinin gerçekten hasta olma olasılığı kaçtır?

Bu soruya çoğu kişi, hatta çoğu hekim, "%90 civarı" diye cevap verir. Doğru cevap yaklaşık %9'dur. Aradaki büyük farkı görmenin en kolay yolu olasılıklar yerine kişi sayılarıyla düşünmektir. Olasılıkları bir topluluktaki kişi sayılarına çevirmeye **doğal sıklıklar** (*natural frequencies*) yöntemi denir. Gigerenzer ve Hoffrage, aynı problem doğal sıklıklarla sorulduğunda çok daha fazla kişinin doğru çözdüğünü göstermiştir (Gigerenzer ve Hoffrage, 1995).



Şekil 1.7: 10.000 kişilik toplulukta tıbbi test. 100 hastanın 90'ı, 9.900 sağlıklının 891'i pozitif çıkar. Pozitif çıkan 981 kişiden yalnız 90'ı hastadır: *P(hasta | +) = 90/981 ≈ %9,2*.

Örnek 1.12

10.000 kişilik bir topluluk düşünelim. Şekil 1.7 her adımın sayılarını gösterir.

1. **Ön olasılık.** Yaygınlık %1'dir: *10.000 · 0,01 = 100* hasta, *9.900* sağlıklı.
2. **Hastalarda test.** Duyarlılık %90: 100 hastanın *100 · 0,9 = 90*'ı pozitif, 10'u negatif çıkar.
3. **Sağlıklılarda test.** Yanlış pozitif oranı %9: 9.900 sağlıklının *9.900 · 0,09 = 891*'i de pozitif çıkar. Sağlıklılar çok kalabalık olduğu için küçük bir oran büyük bir sayı verir.
4. **Pozitif çıkan herkes.** *90 + 891 = 981* kişi.
5. **Son olasılık.** Bu 981 kişiden yalnız 90'ı gerçekten hastadır: *P(hasta | +) = 90/981 ≈ 0,092*, yani yaklaşık %9,2.

Test %90 duyarlı olsa da pozitif sonuç çoğunlukla yanlış alarmdır. Pozitif çıkan her 11 kişiden yaklaşık 10'u sağlıklıdır.

Bu sonuç testin işe yaramadığı anlamına gelmez. Testten önce hastalık olasılığı %1'di; pozitif sonuçtan sonra %9,2'ye çıktı. Olasılık dokuz kattan fazla arttı. Ama başlangıç noktası çok düşük olduğu için son olasılık hâlâ %10'un altındadır. Bu yüzden tıpta pozitif bir tarama testinden sonra genellikle ikinci, daha kesin bir test yapılır.

1.3.3 Bayes kuralı ve taban oranı ihmali

Doğal sıklıklarla yapılan hesap, genel bir formülün özel hâlidir.

Teorem 1.4 (Bayes kuralı)

H bir hipotez (ör. "hasta"), K bir kanıt (ör. "test pozitif") olsun. ¬H, H'nin olmaması ("sağlıklı") demektir. O zaman

$$P(H \mid K) = P(K \mid H) \cdot P(H) \,/\, [\, P(K \mid H) \cdot P(H) + P(K \mid \neg H) \cdot P(\neg H)\,]$$

Payda, kanıtın toplam olasılığı $P(K)$'dır.

Formül koşullu olasılık tanımından iki adımda çıkar. Birinci adım: tanıma göre $P(H \mid K) = P(H \text{ ve } K) / P(K)$'dır. Çarpım kuralıyla pay $P(K \mid H) \cdot P(H)$ olarak yazılır. İkinci adım: K iki yoldan gerçekleşebilir, ya H ile birlikte ya ¬H ile birlikte. Bu iki yolun olasılıkları toplanınca $P(K) = P(K \mid H) \cdot P(H) + P(K \mid \neg H) \cdot P(\neg H)$ elde edilir. İki adım birleşince formül çıkar.

Tıbbi testte $P(H) = 0{,}01$, $P(K \mid H) = 0{,}9$, $P(K \mid \neg H) = 0{,}09$'dur. Pay $0{,}9 \cdot 0{,}01 = 0{,}009$, payda $0{,}009 + 0{,}09 \cdot 0{,}99 = 0{,}009 + 0{,}0891 = 0{,}0981$'dir. Oran $0{,}009 / 0{,}0981 \approx 0{,}0917$'dir. Bu, doğal sıklıklardaki $90/981$'in aynısıdır; tek fark her sayının 10.000'e bölünmüş olmasıdır. Doğal sıklıklar aslında Bayes kuralının kişi sayılarıyla yazılmış hâlidir.

Tanım 1.8

Taban oranı ihmali (*base-rate neglect*), ön olasılığı (taban oranını, yaygınlığı) hesaba katmadan olabilirliği son olasılık yerine koymaktır. Tipik biçimi, $P(+ \mid \text{hasta})$'yı $P(\text{hasta} \mid +)$ sanmaktır.

Taban oranı ihmali gerçek bir sorundur. Klasik bir çalışmada hekim ve tıp öğrencilerine şu soru soruldu: yaygınlığı 1/1000 olan bir hastalık için yanlış pozitif oranı %5 olan bir test pozitif çıkmıştır; kişinin hasta olma olasılığı nedir? 60 kişiden yalnız 11'i doğru cevaba (yaklaşık %2) ulaştı. En sık verilen cevap %95'ti (Casscells, Schoenberger ve Graboys, 1978).

Örnek 1.13

Casscells sorusunu doğal sıklıklarla çözelim. Testin duyarlılığı verilmemiştir; hastaların hepsinde testin pozitif çıktığını varsayalım.

1. 100.000 kişi alalım. Yaygınlık 1/1000: 100 hasta, 99.900 sağlıklı.
2. 100 hastanın hepsi pozitif çıkar.
3. Sağlıklıların %5'i pozitif çıkar: $99.900 \cdot 0{,}05 = 4.995$.
4. Pozitif çıkan herkes: $100 + 4.995 = 5.095$. Bunların 100'ü hasta: $100/5.095 \approx 0{,}0196$.

Doğru cevap yaklaşık %2'dir. “%95” cevabı, yanlış pozitif oranının tümleyenini (*1 − 0,05*) son olasılık sanmaktan gelir.

Taban oranının önemi, yaygınlık değişince açıkça görülür. Aynı test (duyarlılık %90, yanlış pozitif %9) yaygınlığı %10 olan bir toplulukta kullanılsın. 10.000 kişiden 1.000'i hasta, 900'ü pozitif; 9.000 sağlıklıdan 810'u pozitif. *P(hasta | +) = 900/1.710 = 10/19 ≈ %52,6*'dır. Aynı test, hastalığın daha yaygın olduğu bir toplulukta çok daha anlamlı bir sonuç verir. Bu yüzden tarama testleri genellikle risk grubundaki kişilere uygulanır.

Sık yapılan hata

- ***P(K | H)*'yı *P(H | K)* sanmak.** “Test hastaların %90'ını yakalar” ile “pozitif çıkanların %90'ı hastadır” farklı cümlelerdir.
- **Paydayı eksik yazmak.** Paydada yalnız doğru pozitifler değil, yanlış pozitifler de vardır. Yanlış pozitifler unutulursa son olasılık 1 çıkar.
- **Yanlış pozitifleri sağlıklılar yerine bütün topluluktan hesaplamak.** %9, 9.900 sağlıklıya uygulanır; 10.000'e değil.

1.3.4 Kod izleme: bayes()

Bir test sonucu tek başına yeterli değilse ikinci bir test yapılır. İkinci test için ön olasılık artık %1 değildir; birinci testin son olasılığıdır. Her yeni kanıt, bir önceki son olasılığı ön olasılık olarak kullanır. Buna **ardışık güncelleme** denir. Aşağıdaki fonksiyon bunu bir döngüyle yapar. Testlerin, kişinin gerçek durumu bilindiğinde birbirinden bağımsız sonuç verdiği varsayılır.

```
def bayes(p, duyarlilik, yp, testler):
  for sonuc in testler:
    if sonuc == "+":
      pay = duyarlilik * p
      payda = pay + yp * (1 - p)
    else:
      pay = (1 - duyarlilik) * p
      payda = pay + (1 - yp) * (1 - p)
    p = pay / payda
    print(sonuc, round(p, 4))
  return p
```

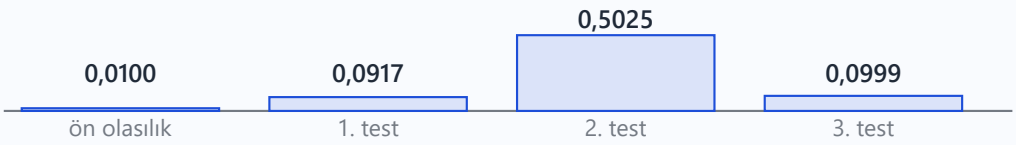
Fonksiyonun parametreleri şunlardır: `p` başlangıç ön olasılığı, `duyarlilik` *P(+ | hasta)*, `yp` yanlış pozitif oranı *P(+ | sağlıklı)*, `testler` ise "+" ve "-" sonuçlarından oluşan liste. Döngü her sonuç için şunu yapar:

- Sonuç "+" ise pozitif sonucun olabilirlikleri kullanılır: hastada `duyarlilik`, sağlıklıda `yp`.

- Sonuç "-" ise negatif sonucun olabilirlikleri kullanılır: hastada `1 - duyarlilik`, sağlıklıda `1 - yp`.
- `pay`, Bayes kuralının payıdır; `payda`, kanıtın toplam olasılığıdır.
- `p = pay / payda` ataması son olasılığı hesaplar ve aynı değişkene yazar. Böylece bir sonraki turda bu değer ön olasılık olur.

Tur	sonuc	ön p	pay	payda	yeni p
1	+	0,0100	*0,9 · 0,0100 = 0,0090*	*0,0090 + 0,09 · 0,9900 = 0,0981*	0,0917
2	+	0,0917	*0,9 · 0,0917 = 0,0826*	*0,0826 + 0,09 · 0,9083 = 0,1643*	0,5025
3	−	0,5025	*0,1 · 0,5025 = 0,0503*	*0,0503 + 0,91 · 0,4975 = 0,5030*	0,0999

Tablo 1.8: `bayes(0.01, 0.9, 0.09, ["+", "+", "-"])` izlemesi.



Şekil 1.8: Ardışık güncellemede hastalık olasılığı: ön olasılık 0,0100; birinci pozitiften sonra 0,0917; ikinci pozitiften sonra 0,5025; üçüncü test negatif çıkınca 0,0999.

Fonksiyon her turda yeni olasılığı yazdırır ve sonunda `0.0999` döndürür. Şekil 1.8 olasılığın seyrini gösterir. Birinci pozitif, olasılığı %1'den %9,2'ye çıkarır. İkinci bağımsız pozitif onu yaklaşık %50'ye taşır. Üçüncü test negatif çıkınca olasılık yeniden %10'a iner. Negatif sonuç güçlü bir kanıttır, çünkü hastaların yalnız %10'unda test negatif çıkar; sağlıklılarda ise %91'inde.

Ardışık güncellemenin iki önemli özelliği vardır. Birincisi, sıra önemli değildir: aynı üç sonuç ("+", "-", "+") sırasıyla işlense de son olasılık 0,0999 çıkar. İkincisi, iki pozitif sonucu birlikte tek seferde işlemek de aynı sonucu verir: iki bağımsız pozitifin olabilirliği hastada *0,9 · 0,9 = 0,81*, sağlıklıda *0,09 · 0,09 = 0,0081*'dir. Pay *0,81 · 0,01 = 0,0081*, payda *0,0081 + 0,0081 · 0,99 ≈ 0,0161*'dir; oran yine yaklaşık 0,5025'tir. Ama her testi baştan %1 ön olasılıkla değerlendirmek yanlıştır: o zaman iki pozitifin birikmiş kanıtı kaybolur.

1.3.5 Bayesçi karar kuralı: tedavi mi, bekle mi?

Son olasılığı bulmak işin yarısıdır. Asıl soru, bu olasılıkla ne yapılacağıdır. Bir hekim, pozitif testten sonra hastayı tedavi etmeli mi, yoksa beklemeli mi? Bu karar hem olasılığa hem de iki tür hatanın bedeline bağlıdır. Sağlıklı birini gereksiz yere tedavi etmek bir bedeldir (yan etkiler, maliyet). Hasta birini tedavi etmemek ise çok daha büyük bir bedeldir.

Tanım 1.9

Bayesçi karar kuralı (*Bayesian decision rule*): Ön olasılık kanıtla güncellenir; sonra her eylemin **beklenen kaybı** (*expected loss*) son olasılıkla hesaplanır. Beklenen kaybı en küçük olan eylem seçilir.

Kayıp, faydanın tersi gibi düşünülebilir: küçük olması iyidir. Kayıplarla çalışmak tıp ve mühendislikte daha doğaldır, çünkü hataların bedeli doğrudan ölçülür. Beklenen kaybı en küçük yapmak, kaybın eksi işaretlisini fayda sayıp beklenen faydayı en büyük yapmakla aynı şeydir.

Örnek 1.14

Kayıplar şöyle olsun: sağlıklıya gereksiz tedavi 30, hastayı tedavi etmemek 200. Doğru kararların kaybı 0'dır. Hastalık olasılığı p ise:

1. **Tedavi edilirse** kayıp yalnız kişi sağlıklıysa doğar: $E[\text{kayıp} \mid \text{tedavi}] = (1 - p) \cdot 30$.
2. **Beklenirse** kayıp yalnız kişi hastaysa doğar: $E[\text{kayıp} \mid \text{bekle}] = p \cdot 200$.
3. **Eşik.** Tedavi daha az kayıplıysa seçilir: $(1 - p) \cdot 30 < p \cdot 200 \Rightarrow 30 - 30p < 200p \Rightarrow 30 < 230p \Rightarrow p > 30/230 = 3/23 \approx \%13{,}0$.
4. **Bir pozitiften sonra** $p \approx 0{,}0917$: tedavi $0{,}9083 \cdot 30 \approx 27{,}2$, bekle $0{,}0917 \cdot 200 \approx 18{,}3$. Beklemek daha az kayıplıdır; yeniden test yapılır.
5. **İki pozitiften sonra** $p \approx 0{,}5025$: tedavi $0{,}4975 \cdot 30 \approx 14{,}9$, bekle $0{,}5025 \cdot 200 \approx 100{,}5$. Tedavi seçilir.

Durum	p	E[kayıp | tedavi]	E[kayıp | bekle]	Karar
1 pozitif	0,092	27,2	18,3	bekle, yeniden test
2 pozitif	0,503	14,9	100,5	tedavi

Tablo 1.9: Tedavi kararı. Kayıplar: gereksiz tedavi 30, tedavisiz hasta 200; eşik $p > 3/23$.

Eşik formülü genel olarak yazılabilir. Gereksiz tedavinin kaybı c, tedavisiz hastanın kaybı k ise tedavi $(1 - p)c < pk$ iken, yani $p > c/(c + k)$ iken seçilir. Formülün yorumu açıktır: hastalığı kaçırmak ne kadar pahalıysa (k büyüdükçe) eşik o kadar düşer. Ölümcül bir hastalıkta düşük bir olasılık bile tedaviye yeter. Tedavinin kendisi tehlikeliyse (c büyükse) eşik yükselir. Karar yalnız olasılığa değil, kayıpların oranına da bağlıdır.

Sık yapılan hata

- **"Olasılık %50'nin altındaysa tedavi etme."** Bu kural ancak iki hatanın bedeli eşitse doğrudur. Burada eşik %13'tür.

- **Kayıpları yanlış olasılıkla çarpmak.** Tedavinin kaybı, kişinin *sağlıklı* olma olasılığıyla $(1 - p)$ çarpılır; çünkü tedavi yalnız sağlıklı birine uygulanınca zarar verir.

1.3.6 Laboratuvara dönüş: yağmur tahmini

1.1.1'deki oyunun ilk sorusu artık cevaplanabilir. Yağmurun ön olasılığı 0,35, tahminin doğruluğu %90'dır. "Doğruluk %90" şu anlama gelir: yağmur yağacaksa tahmin %90 olasılıkla "yağmur" der, hava kuru gidecekse %90 olasılıkla "kuru" der. Yani $P(\text{"yağmur"} \mid \text{yağmur}) = 0{,}9$ ve $P(\text{"yağmur"} \mid \text{kuru}) = 0{,}1$'dir.

Örnek 1.15

1. **Son olasılık.** $P(\text{yağmur} \mid \text{"yağmur"}) = 0{,}9 \cdot 0{,}35 / (0{,}9 \cdot 0{,}35 + 0{,}1 \cdot 0{,}65) = 0{,}315 / 0{,}38 \approx 0{,}829$. Tahmin "yağmur" dediğinde yağmur olasılığı %90 değil, %82,9'dur.
2. **Eylemlerin beklenen faydası** (BF), yağmur olasılığı p iken: açık hava $-6p + 8(1 - p) = 8 - 14p$; kapalı $4p + 3(1 - p) = 3 + p$.
3. **Eşik.** Açık hava daha iyidir: $8 - 14p > 3 + p \Rightarrow 5 > 15p \Rightarrow p < 1/3$.
4. **Yalnız ön olasılıkla** ($p = 0{,}35$): açık 3,10, kapalı 3,35. $0{,}35 > 1/3$, kapalı seçilir; ama fark küçüktür.
5. **"Yağmur" sinyaliyle** ($p \approx 0{,}829$): açık −3,61, kapalı 3,83. Kapalı seçilir.
6. **"Kuru" sinyaliyle**: $P(\text{yağmur} \mid \text{"kuru"}) = 0{,}1 \cdot 0{,}35 / (0{,}1 \cdot 0{,}35 + 0{,}9 \cdot 0{,}65) = 0{,}035 / 0{,}62 \approx 0{,}056$. Açık 7,21, kapalı 3,06. Açık hava seçilir.

Bilgi	P(yağmur)	BF(açık)	BF(kapalı)	Karar
yalnız ön olasılık	0,35	3,10	3,35	kapalı
"yağmur"	0,829	−3,61	3,83	kapalı
"kuru"	0,056	7,21	3,06	açık

Tablo 1.10: Bayesçi Karar Laboratuvarı, birinci düzey (doğruluk %90). BF: beklenen fayda.

Tablo 1.10 bilginin neden değerli olduğunu gösterir. Sinyal olmadan kapalı mekân seçilir. Sinyal "kuru" derse karar açık havaya döner ve beklenen fayda 3,35'ten 7,21'e çıkar. Bilgi ancak kararı değiştirebildiği durumlarda işe yarar; "yağmur" sinyali kararı değiştirmez, ama "kuru" sinyali değiştirir.

Oyunun üçüncü düzeyinde sinyalin doğruluğu %78'dir ve sinyal ancak 1 birim ödenerek görülebilir. Satın almaya değer mi? Bunu yanıtlamak için sinyalin değeri hesaplanır: sinyalden sonra karar verildiğinde beklenen kazanç eksi yalnız ön olasılıkla karar verildiğinde beklenen kazanç.

Örnek 1.16

1. **"Yağmur" sinyalinin olasılığı.** *0,78 · 0,35 + 0,22 · 0,65 = 0,273 + 0,143 = 0,416*. "Kuru" sinyalinin olasılığı *0,584*'tür.
2. **"Yağmur" gelirse.** *P(yağmur) = 0,273/0,416 ≈ 0,656*. Kapalı seçilir: *3 + 0,656 ≈ 3,656*.
3. **"Kuru" gelirse.** *P(yağmur) = 0,22 · 0,35 / 0,584 = 0,077/0,584 ≈ 0,132*. Açık seçilir: *8 − 14 · 0,132 ≈ 6,154*.
4. **Sinyalle beklenen kazanç.** *0,416 · 3,656 + 0,584 · 6,154 ≈ 5,115*.
5. **Sinyalin değeri.** Sinyalsiz en iyi kazanç 3,35'tir. Değer *5,115 − 3,35 = 1,765*'tir.

Sinyalin bilgi değeri 1,765, fiyatı ise 1'dir. Değer fiyattan büyük olduğu için sinyali satın almaya değer.

Oyunda bir "EU-optimal" (beklenen faydaya göre en iyi) sayacı vardır. Bu sayaç şanslı sonuçları değil, eldeki bilgiyle beklenen faydayı maksimize eden kararları sayar. Örneğin "kuru" sinyalinden sonra açık havayı seçen biri, %5,6 olasılıkla yağmura yakalanır ve −6 alır. Bu kötü bir sonuçtur, ama karar iyiydi. Aynı durumda kapalıyı seçip şans eseri 4 alan biri ise kötü bir karar vermiş ama iyi bir sonuç almıştır. Uzun vadede iyi kararlar ortalamada iyi sonuç getirir; tek bir sonuç kararın kalitesini göstermez.

1.3.7 Rakibin tipini güncellemek

Bayes kuralı, tek kişilik kararlardan oyunlara geçişte bir köprüdür. Eksik bilgili bir oyunda bilinmeyen "durum" çoğu zaman rakibin kendisidir: rakibin elindeki kartlar, rakibin ne tür bir oyuncu olduğu. Rakibin bu gizli özelliğine **tip** denir. Tipe dair inanç ön olasılıktır; rakibin hamlesi ise kanıttır.

Örnek 1.17

Poker benzeri bir oyunda rakip %20 olasılıkla blöfçü, %80 olasılıkla dürüsttür. Blöfçü, elinden bağımsız olarak %90 olasılıkla yükseltir. Dürüst oyuncu yalnız güçlü eli varken yükseltir; güçlü el %30 olasılıkla geldiği için dürüst oyuncu %30 olasılıkla yükseltir. Rakip yükseltti. Görmek (*call*) blöfçüye karşı +30, dürüste karşı −30 getirsin; çekilmek 0 getirsin.

1. **Son olasılık.** *P(blöfçü | yükseltme) = 0,2 · 0,9 / (0,2 · 0,9 + 0,8 · 0,3) = 0,18 / 0,42 ≈ 0,429*.
2. **Görmenin beklenen değeri.** *E[gör] = 0,429 · 30 − 0,571 · 30 ≈ −4,3 < 0*. Çekilmek daha iyidir.
3. **İkinci yükseltme** (başka bir elde). Ön olasılık artık 0,429'dur: *0,429 · 0,9 / (0,429 · 0,9 + 0,571 · 0,3) ≈ 0,692*.
4. *E[gör] = 0,692 · 30 − 0,308 · 30 ≈ +11,5 > 0*. Artık görmek kârlıdır.

Gözlem	P(blöfçü)	E[gör]	Karar
yalnız ön olasılık	0,200	−18,0	çekil
1 yükseltme	0,429	−4,3	çekil
2 yükseltme	0,692	+11,5	gör

Tablo 1.11: Rakibin blöfçü olma olasılığı ve görme kararı.

Kazanç ve kayıp eşit büyüklükte (30) olduğu için görmek ancak $P(\text{blöfçü}) > 0{,}5$ iken kârlıdır. Tek yükseltme inancı 0,2'den 0,429'a çıkarır ama eşiğe ulaştırmaz. İkinci yükseltme eşiği aşar. Karar, kanıt biriktikçe değişir.

Bu örnekte sık yapılan hata, $P(\text{yükseltme} \mid \text{blöfçü}) = 0{,}9$'u $P(\text{blöfçü} \mid \text{yükseltme})$ sanmaktır. Bu, tıbbi testteki taban oranı ihmalinin aynısıdır. Dürüst oyuncular daha kalabalık olduğu için yükseltmelerin çoğu onlardan gelir: toplam yükseltme olasılığı 0,42'nin 0,24'ü dürüstlerden gelir, yani yükseltmelerin yaklaşık %57'si dürüst oyunculara aittir.

Bu bakış, kart oyunlarındaki davranışları açıklar. Poker ve Hokm'da bahis ve koz seçimleri, rakiplerin inancını güncellemesi için birer kanıttır. Blöf, rakibin güncellemesini yönlendirme çabasıdır: blöfçü, dürüst bir oyuncunun yapacağı hamleyi yaparak rakibin son olasılığını yanlış yöne çeker. Akıllı bir rakip ise bunu bilir ve olabilirlikleri buna göre ayarlar. Bu karşılıklı düşünme, tek kişilik karar kuramından oyun kuramına geçişin tam noktasıdır.

1.4 Karar ağaçları ve bilginin değeri

Gerçek kararlar çoğu zaman tek adımda verilmez. Önce bir bilgi toplanır, sonra o bilgiye göre bir karar verilir, sonra doğa sonucu belirler. Böyle sıralı kararları düzenlemenin aracı karar ağacıdır. Bir ürün lansmanı problemi, hem ağacın nasıl çözüldüğünü hem de bilginin parasal değerinin nasıl hesaplandığını gösterir.

1.4.1 Karar ağacının öğeleri

Tanım 1.10

Bir **karar ağacı** (*decision tree*) üç tür öğeden oluşur:

- **Karar düğümü** (□): Karar verici seçer. Değeri, dallarından en büyük değerdir.
- **Şans düğümü** (○; *chance node*): Doğa seçer. Değeri, dalların olasılıklarla ağırlıklı ortalamasıdır.
- **Yaprak**: Bir yolun net sonucu.

Karar ağacı, doğanın da hamle yaptığı bir oyun ağacıdır. Tek fark, doğanın bir oyuncu gibi kendi kazancını düşünmemesi, yalnız olasılıklarla hareket etmesidir. Ağaç **geriye doğru çıkarım** (*backward induction*) ile çözülür: yapraklardan başlanır ve köke doğru ilerlenir. Her düğümün değeri, çocuklarının değerleri bilindiğinde hesaplanabilir. Şans düğümünde ortalama, karar düğümünde en büyük değer alınır.

Neden geriye doğru? Çünkü bir karar düğümünde doğru seçimi yapmak için her dalın değerini bilmek gerekir. Bir dalın değeri ise o dalda ileride verilecek kararlara ve gerçekleşecek şans olaylarına bağlıdır. Sondan başlamak, her adımda gereken bilgiyi hazır tutar. Bu yöntem satranç gibi oyunların ağaçlarında da kullanılır.

1.4.2 Ürün lansmanı problemi

Bir firma yeni ürünü piyasaya sürüp sürmemeye karar verecek. Pazar iyi giderse (olasılık 2/5) +500, kötü giderse (olasılık 3/5) −200 bin TL kazanılır. Ürün sürülmezse kazanç 0'dır. Firma karar vermeden önce 20 bin TL'ye bir pazar araştırması yaptırabilir. Araştırma raporu kusurludur:

- Pazar gerçekten iyiyse rapor %80 olasılıkla “olumlu” der: $P(\text{olumlu} \mid \text{iyi}) = 0{,}8$.
- Pazar kötüyse rapor %10 olasılıkla yine “olumlu” der: $P(\text{olumlu} \mid \text{kötü}) = 0{,}1$.

Ağacı çözmek için şans düğümlerindeki olasılıklar gerekir. Araştırma yapılırsa ilk şans olayı raporun kendisidir. Rapordan sonra ise pazarın iyi olma olasılığı rapora göre güncellenmelidir. Bunlar Bayes kuralıyla bulunur.

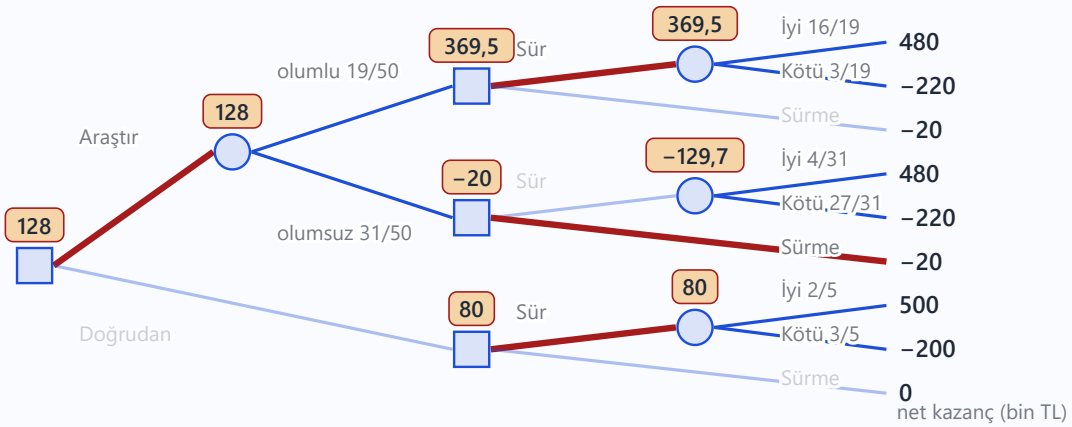
Örnek 1.18

1. **Olumlu raporun olasılığı.** $P(\text{olumlu}) = 0{,}4 \cdot 0{,}8 + 0{,}6 \cdot 0{,}1 = 0{,}32 + 0{,}06 = 0{,}38$. Olumsuz raporun olasılığı $0{,}62$'dir.
2. **Olumlu raporda pazar iyi.** $P(\text{iyi} \mid \text{olumlu}) = 0{,}32/0{,}38 = 16/19 \approx 0{,}842$.
3. **Olumsuz raporda pazar iyi.** İyi pazarda olumsuz rapor olasılığı $0{,}2$'dir: $0{,}4 \cdot 0{,}2 = 0{,}08$. $P(\text{iyi} \mid \text{olumsuz}) = 0{,}08/0{,}62 = 4/31 \approx 0{,}129$.

Olumlu rapor iyi pazar olasılığını 0,4'ten 0,842'ye çıkarır; olumsuz rapor 0,129'a indirir.

Yaprakların net kazançları da hesaplanmalıdır. Araştırma yapılan yollarda 20 bin TL düşülür: iyi pazarda sürmek $500 - 20 = 480$, kötü pazarda sürmek $-200 - 20 = -220$, sürmemek $0 - 20 = -20$ verir. Araştırma yapılmayan yollarda yapraklar 500, −200 ve 0'dır.

1.4.3 Geriye doğru çıkarım adım adım



Şekil 1.9: Ürün lansmanı karar ağacı ve geriye doğru çıkarımın sonucu. Kare karar düğümü, daire şans düğümüdür. Yapraklarda net kazanç (araştırma yapılırsa 20 düşülmüş) yazılıdır. Kalın kırmızı yollar seçilen kararlardır; turuncu kutular düğüm değerleridir.

Örnek 1.19

Şekil 1.9 ağacın tamamını gösterir. Yapraklardan köke doğru ilerleyelim.

1. **Araştırmasız sürme (şans).** $2/5 \cdot 500 + 3/5 \cdot (-200) = 200 - 120 = 80$.
2. **Araştırmasız karar.** $\max(\text{Sür } 80, \text{Sürme } 0) = 80 \Rightarrow$ Sür. Sürme dalı elenir.
3. **Olumlu raporda sürme (şans).** $16/19 \cdot 480 + 3/19 \cdot (-220) = (7680 - 660)/19 = 7020/19 \approx 369{,}5$.
4. **Olumlu raporda karar.** $\max(369{,}5, -20) = 369{,}5 \Rightarrow$ Sür.
5. **Olumsuz raporda sürme (şans).** $4/31 \cdot 480 + 27/31 \cdot (-220) = (1920 - 5940)/31 = -4020/31 \approx -129{,}7$.
6. **Olumsuz raporda karar.** $\max(-129{,}7, -20) = -20 \Rightarrow$ Sürme. Yalnız araştırma maliyeti kaybedilir.
7. **Rapor (şans).** $0{,}38 \cdot 369{,}5 + 0{,}62 \cdot (-20) = 140{,}4 - 12{,}4 = 128$.
8. **Kök (karar).** $\max(\text{Araştır } 128, \text{Doğrudan } 80) = 128 \Rightarrow$ Araştır.

En iyi strateji: araştırma yaptırmak; rapor olumluysa ürünü sürmek, olumsuzsa sürmemek. Bu stratejinin beklenen değeri 128 bin TL'dir.

Burada “strateji” sözcüğü önemlidir. Sonuç yalnız “araştır” değildir; her olası rapora karşı ne yapılacağını söyleyen tam bir plandır. Oyun kuramında da strateji böyle tanımlanır: oyuncunun karşılaşabileceği her durumda ne yapacağını önceden belirleyen eksiksiz bir plan.

Yedinci adımdaki hesap küçük bir kısayol içerir: *0,38 · 7020/19 = 0,02 · 7020 = 140,4*. Kesirleri sona kadar korumak yuvarlama hatalarını önler. Yuvarlanmış 369,5 değeri kullanılsa da sonuç 128'e çok yakın çıkar (140,41 − 12,4 = 128,01).

Sık yapılan hata

- **Kuralları ters çevirmek.** Karar düğümünde ortalama, şans düğümünde en büyük değer almak yanlıştır. Doğayı denetleyemeyiz, bu yüzden şans düğümünde ortalama alınır. Kendi kararımızı ise denetleriz, bu yüzden en iyisini seçeriz.
- **Doğayı da lehimize seçtirmek.** Her düğümde en büyük dalı almak aşırı iyimserdir: sanki pazar hep iyi gidecekmiş gibi.
- **Yalnız en olası dalı izlemek.** Olasılığı düşük ama sonucu büyük dallar (−220 gibi) ortalamaya mutlaka girer.
- **Araştırma maliyetini iki kez düşmek.** Maliyet ya yapraklardan düşülür ya da en sonda kökten; ikisi birden yapılırsa 20 iki kez sayılır.

1.4.4 Kod izleme: deger()

Geriye doğru çıkarım özyinelemeli (kendini çağıran) bir fonksiyonla kısaca yazılır. Ağacın her düğümü Python'da bir demet (`tuple`) ile, her yaprak ise bir sayıyla gösterilir. Şans düğümü `("S", [(olasılık, çocuk), …])`, karar düğümü `("K", [(ad, çocuk), …])` biçimindedir.

```
def deger(n):
    if not isinstance(n, tuple):    # yaprak
        return n
    tur, dallar = n
    if tur == "S":                  # şans düğümü
        return sum(p * deger(c) for p, c in dallar)
    return max(deger(c) for ad, c in dallar)  # karar
```

Kod satır satır şöyle çalışır:

- `n` bir demet değilse bir yapraktır; değeri sayının kendisidir ve doğrudan döndürülür. Özyinelemenin durma koşulu budur.
- Demetse iki parçaya ayrılır: `tur` ("S" ya da "K") ve `dallar` listesi.
- Şans düğümünde her dalın değeri `deger(c)` ile özyinelemeli olarak bulunur, olasılığıyla çarpılır ve toplanır.
- Karar düğümünde dalların değerlerinin en büyüğü döndürülür. Dalın adı (`ad`) hesapta kullanılmaz; yalnız okunabilirlik içindir.

Lansman ağacı bu biçimde şöyle yazılır:

```
agac = ("K", [
    ("Araştır", ("S", [
        (0.38, ("K", [("Sür", ("S", [(16/19, 480), (3/19, -220)])),
                      ("Sürme", -20)])),
        (0.62, ("K", [("Sür", ("S", [(4/31, 480), (27/31, -220)])),
                      ("Sürme", -20)]))])),
    ("Doğrudan", ("K", [("Sür", ("S", [(0.4, 500), (0.6, -200)])),
                        ("Sürme", 0)]))])
```

`deger(agac)` çağrısı ağaçta derinlik öncelikli gezinir: bir dala girer, o dalın en derinine iner, değeri hesaplayıp geri döner, sonra bir sonraki dala geçer. Tablo 1.12 çağrıların sırasını ve dönen değerleri gösterir. "Yığın" sütunu o anda açık olan çağrıları gösterir; her yeni çağrı yığının üstüne eklenir, her dönüş üstteki çağrıyı kaldırır.

Adım	Çağrı / dönüş	Hesap	Yığın derinliği
1	`deger(kök)` — karar	çocuklar sırayla	1
2	`deger(Araştır)` — şans	çocuklar sırayla	2
3	`deger(olumlu)` — karar	çocuklar sırayla	3
4	`deger(olumlu·Sür)` — şans	$0{,}842 \cdot 480 + 0{,}158 \cdot (-220) = 369{,}5$	4
5	`olumlu` döner	$\max(369{,}5, -20) = 369{,}5 \rightarrow$ Sür	3
6	`deger(olumsuz·Sür)` — şans	$0{,}129 \cdot 480 + 0{,}871 \cdot (-220) = -129{,}7$	4
7	`olumsuz` döner	$\max(-129{,}7, -20) = -20 \rightarrow$ Sürme	3
8	`Araştır` döner	$0{,}38 \cdot 369{,}5 + 0{,}62 \cdot (-20) = 128{,}0$	2
9	`deger(Doğrudan·Sür)` — şans	$0{,}4 \cdot 500 + 0{,}6 \cdot (-200) = 80{,}0$	3
10	`Doğrudan` döner	$\max(80, 0) = 80 \rightarrow$ Sür	2
11	`kök` döner	$\max(128{,}0, 80{,}0) = 128{,}0 \rightarrow$ Araştır	1

Tablo 1.12: `deger(agac)` izlemesi. Yaprak çağrıları ve bir düğüme giriş ile o düğümün ilk çocuğuna iniş, kısalık için tek satırda gösterilmiştir.

Fonksiyon `128.0` döndürür; elle bulunan değerin aynısıdır. Kod yalnız değeri döndürür, hangi dalın seçildiğini döndürmez. Stratejiyi de elde etmek için karar düğümünde `max` ile birlikte dalın adını da saklamak gerekir. Fonksiyonun çalışma süresi ağaçtaki düğüm sayısıyla orantılıdır, çünkü her düğüm tam bir kez ziyaret edilir. Bu ağaçta 17 düğüm vardır: 4 karar düğümü (kök, olumlu, olumsuz, Doğrudan), 4 şans düğümü ve 9 yaprak.

1.4.5 Mükemmel bilginin değeri (EVPI)

Araştırma yapmak, araştırmasız karara göre beklenen değeri 80'den 128'e çıkardı. Bilginin ne kadar değerli olduğunu sistemli olarak ölçmek için iki kavram kullanılır. Birincisi, hiçbir kusuru olmayan bir bilgi kaynağının değeridir.

Tanım 1.11

Mükemmel bilginin beklenen değeri (*expected value of perfect information*, EVPI), durumu önceden kesin olarak bilseydik elde edeceğimiz beklenen değer ile bilgi olmadan elde edilen en iyi beklenen değer arasındaki farktır:

> *EVPI = E[mükemmel bilgiyle değer] − E[bilgisiz en iyi değer]*

Mükemmel bilgiyi bir kâhin gibi düşünmek kolaydır. Kâhin pazarın iyi mi kötü mü olacağını önceden söyler. Ama kâhinin ne diyeceği de önceden bilinmez: %40 olasılıkla "iyi", %60 olasılıkla "kötü" der. Bu yüzden mükemmel bilgiyle değer de bir beklenen değerdir.

Örnek 1.20

1. **Bilgi yok.** En iyi eylem sürmektir: *0,4 · 500 + 0,6 · (−200) = 80*.
2. **Kâhin "iyi" derse** ürün sürülür ve 500 kazanılır. **"Kötü" derse** sürülmez ve 0 kazanılır.
3. **Mükemmel bilgiyle beklenen değer.** *0,4 · 500 + 0,6 · 0 = 200*.
4. **EVPI.** *200 − 80 = 120*.

Hiçbir bilgi kaynağına 120 bin TL'den fazla ödenmez. Kusurlu bir rapor, kâhinden daha değerli olamaz.

EVPI'nın nereden geldiğine bakmak öğreticidir. Bilgi olmadan ürün her durumda sürülür. Kâhin yalnız kötü pazarda kararı değiştirir: sürmek yerine sürmemeyi seçtirir ve −200'lük kaybı önler. Bu kazanç %60 olasılıkla gerçekleşir: *0,6 · 200 = 120*. EVPI tam olarak kararı değiştiren durumların beklenen kazancıdır.

1.4.6 Örnek bilginin değeri (EVSI)

Gerçek bilgi kaynakları kusurludur. Pazar araştırması bazen yanılır. Kusurlu bilginin değeri EVPI'dan küçüktür ve ayrıca hesaplanmalıdır.

Tanım 1.12

Örnek bilginin beklenen değeri (*expected value of sample information*, EVSI), kusurlu bir bilgi kaynağı maliyetsiz olsaydı elde edilecek beklenen değer ile bilgisiz en iyi beklenen değer arasındaki farktır. Bilgi kaynağı ancak maliyeti EVSI'dan küçükse kullanılmalıdır. Her zaman *0 ≤ EVSI ≤ EVPI*'dır.

Örnek 1.21

1. **Araştırmalı ağacın değeri** 128'dir, ama bu değerde araştırma maliyeti (20) düşülmüştür.
2. **Maliyetsiz araştırmanın değeri.** Maliyet geri eklenir: *128 + 20 = 148*. Bu, araştırma bedava olsaydı elde edilecek beklenen değerdir.
3. **EVSI.** *148 − 80 = 68*.
4. **Karar.** *EVSI = 68 > maliyet 20* ⇒ araştırma yapılmalıdır. Net kazanç *68 − 20 = 48*'dir; bu da *128 − 80*'e eşittir.
5. **Verimlilik.** *EVSI / EVPI = 68/120 ≈ %57*. Kusurlu rapor, mükemmel bilginin değerinin yarısından biraz fazlasını yakalar.

Durum	Hesap	Değer
Bilgi yok (hemen karar)	Sür: *0,4 · 500 + 0,6 · (−200)*	80
Mükemmel bilgi (kâhin)	iyi → sür 500, kötü → sürme 0: *0,4 · 500 + 0,6 · 0*	200
EVPI	*200 − 80*	120
Örnek bilgi (araştırma, maliyetsiz)	ağaçtaki 128 + araştırma maliyeti 20	148
EVSI	*148 − 80*	68

Tablo 1.13: Bilginin değeri, ürün lansmanı örneği (bin TL).

EVSI bir eşik görevi görür. Araştırmanın maliyeti 68'den küçükse araştırma yapılır, büyükse yapılmaz. Örneğin araştırma 70 bin TL olsaydı *EVSI = 68 < 70* olurdu. Araştırmalı yolun değeri *148 − 70 = 78*'e düşer ve doğrudan sürmenin 80'inden küçük kalır. En iyi strateji araştırmadan doğrudan sürmek olurdu (beklenen 80).

Bilgi neden değerlidir? Çünkü kararı değiştirebilir. Bu örnekte olumsuz rapor "sürme" kararına yol açar; bu yüzden araştırmanın değeri pozitiftir. Olumsuz rapor bile sürmeyi daha iyi gösterseydi, rapor ne derse desin aynı karar verilirdi ve araştırmanın değeri sıfır olurdu. Bu ilke, bilgi toplamanın genel kuralıdır: hiçbir sonucu kararı değiştirmeyecek bir test, ne kadar doğru olursa olsun, para etmez.

Sık yapılan hata

- **EVSI'yı maliyeti geri eklemeden hesaplamak.** *128 − 80 = 48* net kazançtır, EVSI değildir. EVSI, maliyetsiz araştırmanın değeriyle hesaplanır: *148 − 80 = 68*.
- **Araştırma kararını EVPI ile vermek.** EVPI yalnız bir üst sınırdır. Maliyet EVPI'dan küçük ama EVSI'dan büyükse araştırma yapılmaz.
- **Bilgisiz değeri sıfır almak.** Karşılaştırma "hiçbir şey yapmamak"la değil, bilgisiz en iyi kararla (burada 80) yapılır.

1.5 Tartışma: risk ve belirsizlik altında daha iyi karar

Bölümün araçları birlikte bir karar verme yöntemi oluşturur. Bu yöntem dört ilkeyle özetlenebilir.

- **Bilginin değeri hesaplanır.** Güncel veri kararı iyileştirir, ama bilgi toplamak ücretsiz değildir. Bilgi toplamak ancak EVSI maliyetinden büyükse kârlıdır. EVPI ise ödenebilecek en yüksek bedelin üst sınırıdır.
- **Senaryolar ve karar ağacı açıkça yazılır.** Olası durumlar ve olasılıkları yazılınca ağaç geriye doğru çözülebilir. Ağacı çizmek, gözden kaçan seçenekleri (ör. "önce araştır") görünür kılar.
- **Ön olasılık hesaba katılır.** Taban oranları Bayes kuralıyla son olasılığa girer. Doğal sıklıklarla düşünmek ("10.000 kişiden kaçı?") taban oranı ihmalini azaltır.
- **Rakipler ve çeşitlendirme düşünülür.** Sonucu doğa değil başka karar vericiler belirliyorsa, oyun kuramı ve simülasyon rakip davranışını modeller. Birbirinden bağımsız risklere yayılmak (portföy çeşitlendirmesi) toplam riski azaltır; sigortacının yaklaşık risk nötr davranabilmesinin nedeni de budur.

Bu ilkelerin hepsi bir ayrıma dayanır: *kararın kalitesi* ile *sonucun kalitesi* farklı şeylerdir. Beklenen faydayı maksimize eden bir karar, şans kötü giderse kötü bir sonuç verebilir. Kararları sonuçlarına göre yargılamak, şansın payını kararın payı sanmak demektir. Bir karar, verildiği anda mevcut olan bilgiyle değerlendirilmelidir.

1.5.1 Karar problemleri için kontrol listesi

Bir karar problemiyle karşılaşıldığında şu sorular sırayla sorulabilir:

1. Seçenekler (eylemler) nelerdir? Sonucu belirleyen durumlar nelerdir?
2. Durumların olasılıkları biliniyor mu? Biliniyorsa problem risk altında karardır (beklenen fayda); bilinmiyorsa belirsizlik altında karardır (maksimin, Laplace, Savage, Hurwicz).
3. Sonuçlar para ise karar verici risk nötr mü? Değilse fayda fonksiyonu ya da kesinlik eşdeğeri kullanılır.
4. Bir kanıt (test, rapor, sinyal) var mı? Varsa ön olasılık Bayes kuralıyla güncellenir; olabilirlik ile son olasılık karıştırılmaz.
5. Kararlar sıralı mı? Öyleyse karar ağacı çizilir ve geriye doğru çözülür.
6. Ek bilgi satın alınabiliyor mu? EVSI hesaplanır ve maliyetle karşılaştırılır.

1.5.2 Dört kısa problem

Dört problem, bölümün dört ana hesabını tekrar eder. Her biri, bölümdeki bir örneğin sayıları değiştirilerek elde edildi.

Örnek 1.22

Piyango: %20 olasılıkla 2.500 TL, aksi hâlde 100 TL. Fayda $u(x) = \sqrt{x}$. Beklenen değer, kesinlik eşdeğeri ve risk primi nedir?

1. $E[X] = 0{,}2 \cdot 2.500 + 0{,}8 \cdot 100 = 500 + 80 = 580$ TL.
2. $E[u] = 0{,}2 \cdot \sqrt{2.500} + 0{,}8 \cdot \sqrt{100} = 0{,}2 \cdot 50 + 0{,}8 \cdot 10 = 10 + 8 = 18$.
3. $KE = 18^2 = 324$ TL.
4. Risk primi $= 580 - 324 = 256$ TL.

Örnek 1.23

Sigorta örneğinde kayıp olasılığı %10 yerine %20 olsaydı en fazla ödenecek prim ne olurdu?

1. $E[u] = 0{,}8 \cdot 30 + 0{,}2 \cdot 10 = 24 + 2 = 26$.
2. $KE = 26^2 = 676$.
3. En yüksek prim $900 - 676 = 224$. Beklenen kayıp $0{,}2 \cdot 800 = 160$'tır.

Anlaşma bölgesi 160 ile 224 arasıdır ve genişliği 64'e çıkmıştır. Kayıp olasılığı artınca risk de artar, risk primi büyür.

Örnek 1.24

Tıbbi testte yaygınlık %10 olsaydı (duyarlılık %90, yanlış pozitif %9) $P(\text{hasta} \mid +)$ kaç olurdu?

1. Pay: $0{,}9 \cdot 0{,}1 = 0{,}09$. Payda: $0{,}09 + 0{,}09 \cdot 0{,}9 = 0{,}09 + 0{,}081 = 0{,}171$.
2. $P = 0{,}09/0{,}171 = 10/19 \approx \%52{,}6$.

Taban oranı yükselince pozitif sonuç çok daha anlamlı hâle gelir: %9,2 yerine %52,6.

Örnek 1.25

Karar ağacında araştırma 70 bin TL olsaydı en iyi strateji ne olurdu?

1. EVSI maliyetten bağımsızdır: 68.
2. $68 < 70$: araştırmanın beklenen katkısı maliyetini karşılamaz.
3. En iyi strateji doğrudan sürmektir; beklenen değer 80'dir.

1.5.3 Etkileşimli oyunlar

Bölümün kavramları ders sitesindeki birkaç oyunda doğrudan karşılık bulur:

- Bayesçi Karar Laboratuvarı: ön olasılık, olabilirlik ve kazançlara göre son olasılık ve karar; üçüncü düzeyde bilginin değeri.
- Blackjack: kart çekme ya da durma kararında batma olasılığı ve temel strateji.
- Mayın Tarlası: açılan sayılara göre hücrelerin mayın olasılığını güncelleme.
- Pişti: oynanan kartları hatırlayarak rakibin elindeki kartlar hakkında çıkarım.
- Hokm: eksik bilgili, takım hâlinde oynanan kart oyunu; koz seçimi ve sinyalleşme.
- Poker: eksik bilgi, olasılık ve blöf; rakibin elini olasılıkla tahmin etme.
- Tavla: zarlı oyunda hamleleri sezgisel bir değerlendirme fonksiyonuyla puanlama; üç farklı ağırlık profili.

Bu oyunların hepsinde aynı döngü vardır: bir inanç tutulur, yeni gözlemle güncellenir ve güncel inançla beklenen kazancı en büyük hamle seçilir. Tavla gibi şans düğümlü oyunlarda hamle seçimi, zarların olası sonuçları üzerinde ortalama alan bir karar ağacı aramasıdır.

1.6 Özet

- **Beklenen fayda**: $E[u] = \Sigma\ p_i\ u(x_i)$. Piyango A (½: 900; ½: 0) için $E[X] = 450$, ama $\sqrt{x}$ ile $E[u] = 15 < u(400) = 20$; beklenen değer A'yı, beklenen fayda B'yi seçer. St. Petersburg paradoksu, beklenen değerin tek başına yetmediğini gösterir.
- **vNM aksiyomları** (tamlık, geçişlilik, süreklilik, bağımsızlık) sağlanıyorsa karar verici beklenen faydayı maksimize eder. Allais seçimleri bağımsızlığı, Ellsberg seçimleri tutarlı bir olasılık inancını çiğner.
- **Fayda ölçeği**: $a \cdot u + b$ ($a > 0$) dönüşümü hiçbir piyango kararını değiştirmez; afin olmayan artan dönüşümler değiştirebilir. Fayda farklarının büyüklüğü ve kişiler arası fayda karşılaştırmaları anlamlı değildir.
- **Risk tutumu**: içbükey → kaçınan (KE 225, prim 225); doğrusal → nötr (prim 0); dışbükey → seven (prim negatif). Jensen eşitsizliği bu sınıflandırmanın temelidir. Sigortada 80 ile 116 arasındaki primler iki tarafa da kazandırır.
- **Belirsizlik ölçütleri**: aynı tabloda maksimin Yatırım yok, maksimaks ve Laplace Büyük, Savage Küçük yatırımı seçer. Hurwicz ölçütü iyimserlik katsayısıyla iki uç arasında geçiş yapar.
- **Bayesçi karar**: yaygınlık %1, duyarlılık %90, yanlış pozitif %9 ⇒ $P(\text{hasta} \mid +) \approx \%9{,}2$. Ardışık testlerde son olasılık bir sonraki testin ön olasılığıdır. Tedavi, beklenen kaybı küçük olduğunda, yani $p > 3/23$ iken seçilir.
- **Karar ağacı**: şans düğümünde ortalama, karar düğümünde en büyük değer alınır. Araştır (128) > araştırmasız (80); EVPI = 120, EVSI = 68. Bilgi ancak kararı değiştirebiliyorsa değerlidir.

Kavram	Ne söyler?	Nasıl hesaplanır?
Beklenen fayda	Riskli seçeneklerin değeri	$\Sigma\ p_i\ u(x_i)$
Kesinlik eşdeğeri	Piyangoya denk kesin tutar	$u^{-1}(E[u])$

Kavram	Ne söyler?	Nasıl hesaplanır?
Risk primi	Riskten kurtulmanın bedeli	*E[X] − KE*
Maksimin / Savage	Olasılıksız karar	En kötüyü büyüt / en büyük pişmanlığı küçült
Bayes kuralı	Kanıtla inanç güncelleme	Pay / (pay + yanlış alarm)
Bayesçi karar kuralı	Son olasılıkla en iyi eylem	En küçük beklenen kayıp
Geriye doğru çıkarım	Sıralı kararlarda en iyi strateji	Şansta ortalama, kararda maksimum
EVPI / EVSI	Bilginin parasal değeri	Bilgiyle değer − bilgisiz değer

Tablo 1.14: Bölümün kavramları bir bakışta.

1.7 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

karar alma *decision making* · risk altında karar *decision under risk* · belirsizlik altında karar *decision under uncertainty* · piyango *lottery* · fayda fonksiyonu *utility function* · beklenen fayda *expected utility* · St. Petersburg paradoksu *St. Petersburg paradox* · von Neumann–Morgenstern aksiyomları *von Neumann–Morgenstern axioms* · tamlık / süreklilik / bağımsızlık aksiyomları *completeness / continuity / independence axioms* · Allais paradoksu *Allais paradox* · kesinlik etkisi *certainty effect* · afin dönüşüm *affine transformation* · kardinal fayda *cardinal utility* · azalan marjinal fayda *diminishing marginal utility* · içbükey *concave* · dışbükey *convex* · Jensen eşitsizliği *Jensen's inequality* · risk tutumu *risk attitude* · riskten kaçınan *risk-averse* · risk nötr *risk-neutral* · risk seven *risk-loving* · kesinlik eşdeğeri (KE) *certainty equivalent* · risk primi *risk premium* · Ellsberg paradoksu *Ellsberg paradox* · belirsizlikten kaçınma *ambiguity aversion* · maksimin *maximin* · maksimaks *maximax* · Laplace ölçütü *Laplace criterion* · minimaks pişmanlık *minimax regret* · Hurwicz ölçütü *Hurwicz criterion* · iyimserlik katsayısı *coefficient of optimism* · koşullu olasılık *conditional probability* · ön olasılık *prior* · olabilirlik *likelihood* · son olasılık *posterior* · Bayes kuralı *Bayes' rule* · duyarlılık *sensitivity* · doğal sıklıklar *natural frequencies* · taban oranı ihmali *base-rate neglect* · kanıt *evidence* · Bayesçi karar kuralı *Bayesian decision rule* · beklenen kayıp *expected loss* · tip *type* · blöf *bluffing* · karar ağacı *decision tree* · karar düğümü *decision node* · şans düğümü *chance node* · geriye doğru çıkarım *backward induction* · mükemmel bilginin beklenen değeri (EVPI) *expected value of perfect information* · örnek bilginin beklenen değeri (EVSI) *expected value of sample information*

1.8 Kendini Dene

1. Piyango M: %40 olasılıkla 1.600 TL, %60 olasılıkla 100 TL. Fayda $u(x) = \sqrt{x}$. Beklenen değer, beklenen fayda, kesinlik eşdeğeri ve risk primi nedir? Bu kişi M ile kesin 500 TL arasında neyi seçer?

   Cevap

   $E[M] = 0{,}4 \cdot 1.600 + 0{,}6 \cdot 100 = 640 + 60 = 700$ TL. $E[u] = 0{,}4 \cdot 40 + 0{,}6 \cdot 10 = 16 + 6 = 22$. $KE = 22^2 = 484$ TL; risk primi $700 - 484 = 216$ TL. Kesin 500 TL, KE'den büyüktür ($\sqrt{500} \approx 22{,}4 > 22$); kişi kesin 500 TL'yi seçer. Risk nötr biri ise $700 > 500$ olduğu için M'yi seçerdi.

2. Bir karar vericinin fayda fonksiyonu u'dur. Aşağıdakilerden hangileri onun bütün piyango tercihlerini aynen korur: $v_1 = 4u + 7$, $v_2 = u^2$ ($u \geq 0$ iken), $v_3 = -2u + 1$, $v_4 = u/3 - 10$?

   Cevap

   v_1 ve v_4 korur: ikisi de $a \cdot u + b$ biçimindedir ve a pozitiftir (4 ve 1/3). v_3 afindir ama $a = -2 < 0$; bütün sıralamayı tersine çevirir. v_2 artan ama afin değildir; kesin sonuçların sırasını korur, piyangolar arasındaki sırayı değiştirebilir ($\sqrt{x}$'in karesi x'tir ve A ile B'nin kararını tersine çevirir).

3. Servet 2.500 (bin TL); %4 olasılıkla 2.400'lük kayıp olur ve geriye 100 kalır. Fayda $u(x) = \sqrt{x}$. Kişinin tam sigorta için ödemeye razı olduğu en yüksek prim ve risk nötr sigortacının isteyeceği en düşük prim nedir?

   Cevap

   Sigortasız $E[u] = 0{,}96 \cdot 50 + 0{,}04 \cdot 10 = 48 + 0{,}4 = 48{,}4$. $KE = 48{,}4^2 = 2.342{,}56$. En yüksek prim $2.500 - 2.342{,}56 = 157{,}44$. Beklenen kayıp $0{,}04 \cdot 2.400 = 96$; sigortacı en az bunu ister. 96 ile 157,44 arasındaki primler iki tarafa da kazandırır. Aradaki 61,44, kişinin risk primidir (beklenen servet 2.404 − KE 2.342,56). Olasılık küçük olsa da kayıp büyük olduğu için kişinin razı olduğu en yüksek prim beklenen kaybın yaklaşık 1,64 katıdır.

4. Olasılıklar bilinmiyor. Kazançlar: A (İyi 60, Orta 30, Kötü −40), B (40, 20, 0), C (100, −10, −60). Maksimin, maksimaks, Laplace, Savage ve $\alpha = 0{,}6$ ile Hurwicz hangi eylemi seçer?

   Cevap

   Satır minimumları −40, 0, −60 ⇒ maksimin B. Maksimumlar 60, 40, 100 ⇒ maksimaks C. Ortalamalar 16,7, 20, 10 ⇒ Laplace B. Sütun en iyileri 100, 30, 0; pişmanlıklar A (40, 0, 40), B (60, 10, 0), C (0, 40, 60); en büyükler 40, 60, 60 ⇒ Savage A. Hurwicz: A $0{,}6 \cdot 60 + 0{,}4 \cdot (-40) = 20$, B $0{,}6 \cdot 40 + 0 = 24$, C $0{,}6 \cdot 100 + 0{,}4 \cdot (-60) = 36$ ⇒ C. Beş ölçüt üç farklı eylem seçer.

5. Bir hastalığın yaygınlığı %2, testin duyarlılığı %95, yanlış pozitif oranı %5'tir. 10.000 kişilik toplulukta doğal sıklıklarla *P(hasta | +)* kaçtır? Test negatif çıkan birinin hasta olma olasılığı kaçtır?

Cevap

200 hasta: 190 pozitif, 10 negatif. 9.800 sağlıklı: 490 pozitif, 9.310 negatif. *P(hasta | +) = 190/(190 + 490) = 190/680 ≈ %27,9*. *P(hasta | −) = 10/(10 + 9.310) = 10/9.320 ≈ %0,11*. Negatif sonuç hastalığı neredeyse dışlar; pozitif sonuç ise hâlâ çoğunlukla yanlış alarmdır.

6. Önceki sorudaki pozitif testten sonra hekim karar veriyor. Sağlıklıya gereksiz tedavinin kaybı 40, hastayı tedavi etmemenin kaybı 160'tır. Tedavi eşiği nedir ve karar ne olur?

Cevap

Tedavi *(1 − p) · 40 < p · 160* iken seçilir: *40 < 200p ⇒ p > 0,2*. *p ≈ 0,279 > 0,2* olduğu için tedavi seçilir. Beklenen kayıplar: tedavi *(490/680) · 40 ≈ 28,8*, bekle *(190/680) · 160 ≈ 44,7*.

7. Bir firmanın yeni ürünü iyi pazarda (olasılık 0,5) +300, kötü pazarda (0,5) −250 bin TL getirir; sürmezse 0. (a) Bilgisiz en iyi karar ve EVPI nedir? (b) Maliyetsiz bir araştırma için *P(olumlu | iyi) = 0,8*, *P(olumlu | kötü) = 0,3*'tür. EVSI kaçtır?

Cevap

(a) Sür: *0,5 · 300 + 0,5 · (−250) = 25 > 0* ⇒ sür, 25. Kâhinle *0,5 · 300 + 0,5 · 0 = 150*; *EVPI = 150 − 25 = 125*. (b) *P(olumlu) = 0,4 + 0,15 = 0,55*. *P(iyi | olumlu) = 0,4/0,55 = 8/11*; sürmek *8/11 · 300 + 3/11 · (−250) = 1650/11 = 150* ⇒ sür. *P(iyi | olumsuz) = 0,1/0,45 = 2/9*; sürmek *2/9 · 300 + 7/9 · (−250) ≈ −127,8* ⇒ sürme (0). Araştırmalı değer *0,55 · 150 + 0,45 · 0 = 82,5*; *EVSI = 82,5 − 25 = 57,5 ≤ EVPI = 125*.

8. Rakip %25 olasılıkla blöfçüdür. Blöfçü bir elde %60, dürüst oyuncu %20 olasılıkla yükseltir. Bir yükseltmeden sonra ve (başka bir elde) ikinci yükseltmeden sonra rakibin blöfçü olma olasılığı nedir? Görmek blöfçüye karşı +20, dürüste karşı −40 getirirse hangi gözlemden sonra görmek kârlıdır?

Cevap

Birinci yükseltme: *0,25 · 0,6 / (0,15 + 0,75 · 0,2) = 0,15/0,30 = 0,5*. İkinci yükseltme: *0,5 · 0,6 / (0,3 + 0,5 · 0,2) = 0,3/0,4 = 0,75*. Görmek *20p − 40(1 − p) > 0 ⇒ p > 2/3* iken kârlıdır. Birinci yükseltmeden sonra *0,5 < 2/3* olduğu için çekilmek daha iyidir; ikinci yükseltmeden sonra *0,75 > 2/3*, görülür (*E[gör] = 15 − 10 = 5*).

1.9 İleri okuma

- S. Russell ve P. Norvig, *Artificial Intelligence: A Modern Approach* (2020), Bölüm 12–16: olasılık, Bayes kuralı, fayda kuramı, karar ağaçları ve bilginin değerini yapay ajanlar açısından adım adım anlatır.
- H. Raiffa, *Decision Analysis* (1968): karar ağaçlarını ve örnek bilginin değerini iş kararları üzerinden sunan klasik bir giriş kitabıdır.
- G. Gigerenzer ve U. Hoffrage, "How to improve Bayesian reasoning without instruction: Frequency formats" (1995): doğal sıklıkların Bayes hesabını neden kolaylaştırdığını deneylerle gösterir.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 16: Bayes kuralının, insanların birbirini izleyerek karar verdiği bilgi çağlayanlarında nasıl kullanıldığını anlatır.

1.10 Kaynaklar

1. M. Allais, "Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine", *Econometrica* 21(4), 503–546, 1953.
2. D. Bernoulli, "Specimen theoriae novae de mensura sortis", *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 5, 1738.
3. W. Casscells, A. Schoenberger, T. B. Graboys, "Interpretation by physicians of clinical laboratory results", *New England Journal of Medicine* 299, 999–1001, 1978.
4. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010 (Bölüm 16: bilgi çağlayanları ve Bayes kuralı).
5. D. Ellsberg, "Risk, ambiguity, and the Savage axioms", *Quarterly Journal of Economics* 75(4), 643–669, 1961.
6. G. Gigerenzer, U. Hoffrage, "How to improve Bayesian reasoning without instruction: Frequency formats", *Psychological Review* 102(4), 684–704, 1995.
7. H. Raiffa, *Decision Analysis: Introductory Lectures on Choices under Uncertainty*, Addison-Wesley, 1968.
8. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 4. baskı, Pearson, 2020 (Bölüm 12–16: belirsizlik, Bayes, karar kuramı).
9. L. J. Savage, "The theory of statistical decision", *Journal of the American Statistical Association* 46, 55–67, 1951.
10. J. von Neumann, O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

KISIM I · BÖLÜM 2

Oyunlar ve Temsiller

Bir durumu oyun olarak incelemek için önce onu kesin bir dille yazmak gerekir: kimler oynuyor, kim neyi seçebiliyor, kim neyi biliyor ve her sonuç kime ne getiriyor? Bu not, bu soruların cevaplarını veren temel kavramlarla başlar. Ardından oyunları yazmanın iki yolunu, matrisi (normal form) ve ağacı (genişletilmiş form) anlatır; oyunları altı boyutta sınıflandırır. Mahkûmun İkilemi ve altı klasik 2×2 oyun üzerinden denge ile verimlilik arasındaki farkı gösterir. 2/3 ortalaması oyunu, kombinatoryal oyunlar ve sıralı oyunlar ise kuramın algoritmalarla, yani minimaks, Nim'in XOR kuralı ve geriye doğru çıkarımla nasıl buluştuğunu gösterir. Minimaksın büyük ağaçlarda kullanılabilmesi için gereken alfa-beta budaması, hamle sıralaması ve derinlik sınırlı arama, XOX ve Dörtlü Bağla üzerinde ayrıntılı olarak işlenir.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum02`

ÖĞRENME ÇIKTILARI

Kavramlar
Oyuncu, strateji, kazanç, bilgi, rasyonellik ve ortak bilgiyi bir örnek üzerinde tanımlamak.

Temsil ve sınıflandırma
Normal form (matris) ve genişletilmiş form (ağaç); oyunları zamanlama, anlaşma, toplam kazanç, simetri, bilgi ve ufuk boyutlarında ayırt etmek.

Denge ve verimlilik
Nash dengesini sezgisel olarak açıklamak; Mahkûmun İkilemi'nde ihanetin baskın olduğunu ve dengenin Pareto optimal olmadığını göstermek.

Klasik oyunlar
Altı klasik 2×2 oyunun saf dengelerini en iyi tepkilerle bulmak; çatışma ve koordinasyon yapısını yorumlamak.

Algoritmalar
Seviye-k akıl yürütmeyi açıklamak; minimaks ve Nim'in XOR stratejisiyle küçük oyunları elle ve kodla çözmek.

Sıralı oyunlar
Geriye doğru çıkarımla alt oyun mükemmel dengeyi bulmak; inandırıcı olmayan tehdidi ve sonlu ile sonsuz ufuk farkını yorumlamak.

Oyun ağacı araması
Minimaksı ve alfa-beta budamasını elle ve kodla uygulamak; hamle sıralamasının etkisini, değerlendirme fonksiyonunu ve ufuk etkisini açıklamak.

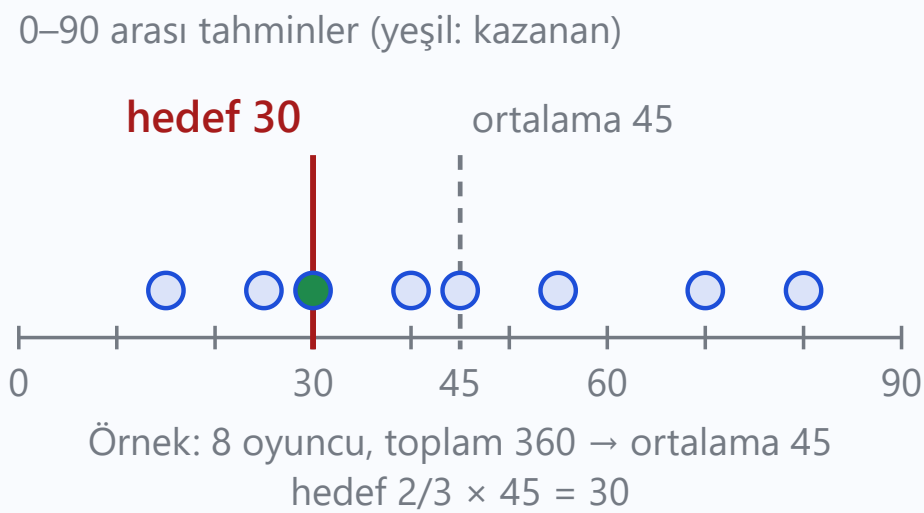
2.1 Temel kavramlar

Bir oyunu oluşturan parçalar basit görünür, ama sonraki bütün hesaplar bu parçaların tanımları üzerine kurulur. Bir kavram eksik ya da yanlış tanımlanırsa, örneğin strateji ile eylem karıştırılırsa, sonraki hesapların hepsi yanlış çıkar. Bu yüzden kavramlar önce küçük bir oyun üzerinden somutlaştırılır, sonra biçimsel olarak yazılır.

2.1.1 Bir oyunla başlangıç: 2/3 ortalaması

Bir sınıftaki herkes 0 ile 90 arasında gizlice bir sayı yazar. Sayı tam ya da ondalıklı olabilir. Bütün tahminlerin ortalaması hesaplanır ve bu ortalamanın 2/3'ü **hedef** olarak alınır. Hedefe en yakın tahmini yazan kazanır; birden çok kişi aynı uzaklıktaysa ödül paylaşılır. Bu oyuna **2/3 ortalaması oyunu** (*guess 2/3 of the average*) denir.

Oyunun ilginç yanı, doğru cevabın başkalarının ne yazacağına bağlı olmasıdır. Tek başına düşünen biri "sayılar 0 ile 90 arasında rastgele dağılırsa ortalama 45 olur, 45'in 2/3'ü 30'dur; o hâlde 30 yazarım" diyebilir. Ama aynı akıl yürütmeyi başkaları da yapıyorsa çoğu kişi 30 yazar; o zaman ortalama 30'a yakın, hedef 20'ye yakın olur. Bunu da öngören biri 20 yazar. Her adım, ötekilerin bir adım önceki düşüncesine verilen bir cevaptır.



Şekil 2.1: Sekiz oyunculu bir örnek. Tahminlerin toplamı 360, ortalaması 45'tir; hedef 2/3 × 45 = 30'dur. 30'a en yakın tahmin (yeşil) kazanır.

Şekil 2.1'de sekiz oyuncunun tahminleri bir sayı doğrusu üzerinde gösterilir. Tahminlerin toplamı 360 olduğundan ortalama 360 / 8 = 45'tir. Hedef 45'in 2/3'ü, yani 30'dur. Tam 30 yazan oyuncu yeşil noktadır ve kazanır. Bu sekiz kişinin tahminleri farklı olsaydı ortalama ve hedef de değişirdi. Kazanan sayı oyunun kurallarından değil, oyuncuların birlikte yaptığı seçimlerden çıkar.

Bu küçük oyun, bölümün neredeyse bütün sorularını içinde taşır. Oyuncular kimdir? Her oyuncunun seçenekleri nelerdir? Her seçim kime ne kazandırır? Oyuncular birbirinin ne kadar akıllı olduğunu biliyor mu? "Herkes çok akıllıysa" hangi sayı kazanır? Gerçek bir sınıfta hangi sayı kazanır? Bu soruların ilk dördü oyunun *tanımına*, son ikisi oyunun *çözümüne* aittir. Oyunun çözümü 2.6'da ayrıntılı olarak verilir.

2.1.2 Oyun kuramı nedir?

Oyun kuramı (*game theory*), karar vericilerin **stratejik etkileşim** içinde olduğu durumları inceleyen matematiksel bir araçtır. Stratejik etkileşim kısaca şöyle özetlenir: bir oyuncunun en iyi kararı, ötekinin ne yapacağına bağlıdır; ötekinin en iyi kararı da onunkine. Tek kişilik karar probleminde karar verici yalnız doğaya ya da şansa karşı oynar. Oyunda ise karşıda, kendisi de düşünen ve kendi çıkarını gözeten başka karar vericiler vardır.

Karar vericilere **oyuncu** (*player*) ya da **ajan** (*agent*) denir. Oyuncu bir insan, bir şirket, bir ülke, bir yazılım ya da bir ağ cihazı olabilir. Oyun kuramı, oyuncuların nasıl davrandığını ve sonucun nasıl şekillendiğini açıklamaya çalışır. Bunu yaparken oyuncuların **rasyonel** olduğu ve kendi çıkarlarına göre hareket ettiği varsayılır. Bu varsayımın tam anlamı 2.1.5'te açılır.

Kuram ekonomi, siyaset bilimi, biyoloji, psikoloji ve bilgisayar biliminde yaygın kullanılır. Üç örnek alan, kuramın ne tür sorulara cevap verdiğini gösterir:

- **Ekonomi:** Şirketlerin fiyat rekabeti, açık artırmalar ve pazarlık. Bir şirketin fiyatı, rakibin fiyatına göre anlam kazanır.
- **Bilgisayar ağları:** Her paketin ya da her sürücünün kendi yolunu seçtiği bencil yönlendirme. Herkes en hızlı yolu seçerse o yol tıkanabilir.
- **Yapay zekâ:** Çoklu ajan sistemleri, oyun oynayan programlar ve öğrenen ajanlar. Bir satranç programı rakibin en iyi cevabını hesaba katmadan hamle seçemez.

Algoritmik oyun kuramı (*algorithmic game theory*), klasik oyun kuramına bilgisayar biliminden gelen iki soru ekler. Birincisi hesaplamayla ilgilidir: bir oyunun dengesini hesaplamak ne kadar zordur, hangi algoritma bunu yapar? İkincisi tasarımla ilgilidir: bencil ajanlar arasında iyi sonuç veren kurallar, yani algoritmalar, protokoller ve açık artırmalar nasıl tasarlanır? Bu iki soru bölümün sonunda, 2.10'da yeniden ele alınır.

2.1.3 Kısa tarihçe

Oyun kuramı yirminci yüzyılın başında satranç gibi oyunlarla ilgili matematiksel sorularla başladı. Yüzyılın ortasında ekonominin temel araçlarından biri oldu. Yüzyılın sonunda bilgisayar bilimiyle buluşarak algoritmik oyun kuramına dönüştü. Tablo 2.1 bu yolculuğun önemli duraklarını gösterir.

Yıl	Gelişme
1913	Zermelo: satranç gibi sonlu, mükemmel bilgili oyunlarda bir taraf kazanmayı ya da iki taraf beraberliği garanti eder.
1928	von Neumann: iki kişilik sıfır toplamlı oyunlar için minimaks teoremi.
1944	von Neumann ve Morgenstern, *Theory of Games and Economic Behavior*: oyun kuramı ayrı bir alan olur.
1950	Nash: her sonlu oyunda karışık stratejilerde bir denge vardır. Aynı yıl RAND'daki Flood–Dresher deneyi yapılır; Tucker Mahkûmun İkilemi hikâyesini anlatır.
1961	Vickrey: ikinci fiyat açık artırmasında dürüst teklif baskındır.
1994	Nash, Harsanyi ve Selten'e Nobel ekonomi ödülü.
1999	Koutsoupias–Papadimitriou (en kötü denge, Anarşinin Bedeli) ve Nisan–Ronen (algoritmik mekanizma tasarımı): algoritmik oyun kuramı doğar.
2006	Genel oyunlarda Nash dengesini hızlı bulan bir yöntem bilinmez: problem PPAD-tam denen zor problemler sınıfındadır (Daskalakis–Goldberg–Papadimitriou; iki oyuncu: Chen–Deng).
2012, 2020	Nobel: Roth ve Shapley (kararlı eşleştirme, piyasa tasarımı); Milgrom ve Wilson (açık artırma tasarımı).

Tablo 2.1: Oyun kuramından algoritmik oyun kuramına önemli gelişmeler.

Tablodaki tarihler üç dönemi ayırır. 1913–1944 arası, kuramın matematiksel temellerinin atıldığı dönemdir. Zermelo'nun satranç sonucu ve von Neumann'ın minimaks teoremi bu dönemin iki büyük sonucudur; ikisi de bu notta yeniden karşımıza çıkar (2.2.8 ve 2.7.1). 1950–1994 arası, kuramın genel oyunlara ve ekonomiye yayıldığı dönemdir. Nash'in denge kavramı bu dönemin merkezindedir. 1999'dan sonrası ise bilgisayar biliminin iki sorusunu, yani “dengeyi hesaplamak ne kadar zor?” ve “iyi kurallar nasıl tasarlanır?” sorularını kuramın merkezine taşır.

Mahkûmun İkilemi'nin doğuşu da bu tabloda görünür. 1950'de RAND şirketinde Merrill Flood ve Melvin Dresher, iki kişiye aynı oyunu art arda oynatarak oyuncuların kuramın öngördüğü gibi davranıp davranmadığını sınadı. Aynı yıl Albert Tucker, bu oyunun yapısını iki mahkûmun hikâyesiyle anlattı ve oyun bu adla tanındı. Hikâye 2.4.2'de ayrıntılı olarak anlatılır.

2.1.4 Bir oyunun bileşenleri

Bir oyun, oyuncuların etkileşime girdiği ve her birinin seçiminin herkesin kazancını etkilediği bir durumdur. Bir durumu oyun olarak yazmak için altı soruya cevap vermek gerekir. Tablo 2.2 bu altı bileşeni ve gösterimlerini listeler.

Bileşen	Anlamı	Gösterim ya da örnek
Oyuncular	Karar alan kişi ya da kurumlar.	$N = \{1, \ldots, n\}$

Bileşen	Anlamı	Gösterim ya da örnek
Eylemler / stratejiler	Her oyuncunun seçebileceği eylemler kümesi.	S_i = {Taş, Kağıt, Makas}
Kazanç	Her sonucun oyuncu için değeri; herkesin seçimine bağlıdır.	$u_i(s_1, \ldots, s_n)$
Bilgi	Kim neyi, ne zaman biliyor? Rakibin hamlesi görülüyor mu?	satranç ≠ poker
Kurallar ve sıra	Eşzamanlı mı, sırayla mı? Kazançları eylemlere bağlayan kurallar.	normal / genişletilmiş form
Sonuç	Tüm oyuncuların seçimlerinin bir araya gelmesiyle oluşur.	$s = (s_1, \ldots, s_n)$

Tablo 2.2: Bir oyunun bileşenleri ve gösterimleri.

Gösterimdeki i harfi "herhangi bir oyuncu" anlamına gelir. S_i, oyuncu i'nin seçebileceği stratejilerin kümesidir; u_i ise oyuncu i'nin kazancını hesaplayan fonksiyondur. Kazanç fonksiyonunun içine bütün oyuncuların seçimleri yazılır, yalnız oyuncu i'ninki değil. Stratejik etkileşim tam olarak budur: oyuncu kendi kazancını tek başına belirleyemez.

Örnek 2.1

2/3 ortalaması oyunu, 30 kişilik bir sınıfta bileşenlerine şöyle ayrılır:

1. **Oyuncular:** sınıftaki 30 öğrenci, $N = \{1, \ldots, 30\}$.
2. **Stratejiler:** her öğrenci için 0 ile 90 arasındaki bütün sayılar, $S_i = [0, 90]$. Bu küme sonsuzdur, çünkü ondalıklı sayılar da serbesttir.
3. **Sonuç:** 30 sayılık bir liste, $s = (s_1, \ldots, s_{30})$.
4. **Kazanç:** listeden ortalama ve hedef hesaplanır. Hedefe en yakın olanlar ödülü eşit paylaşır, ötekiler 0 alır. Öğrenci i'nin kazancı bütün listeye bağlıdır.
5. **Bilgi:** herkes kuralları bilir, ama ötekilerin yazdığı sayıyı görmez.
6. **Kurallar ve sıra:** herkes aynı anda (gizlice) yazar; oyun eşzamanlıdır.

Bu yazım, oyunun matris olarak çizilemeyecek kadar büyük olduğunu da gösterir: her oyuncunun sonsuz sayıda stratejisi vardır. Yine de oyun tam olarak tanımlanmıştır ve 2.6'da çözülür.

2.1.5 Oyuncular, rasyonellik ve ortak bilgi

Oyun kuramı oyuncular hakkında üç varsayım yapar. Üçü de sade bir dille söylenebilir, ama her birinin ince bir yanı vardır.

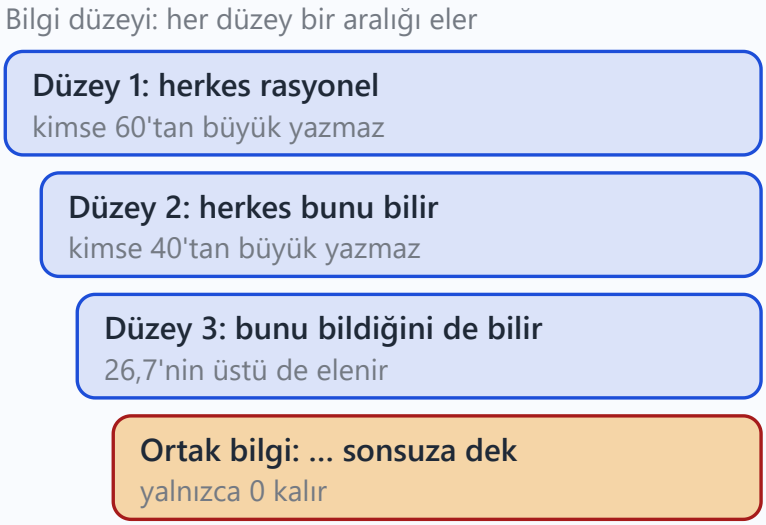
Tanım 2.1

- **Rasyonellik** (*rationality*): Her oyuncu, ötekilerin ne yapacağına dair inançlarına göre kendi beklenen faydasını maksimize eden seçimi yapar.
- **Kendi çıkarı** (*self-interest*): Oyuncu kendi fayda fonksiyonunu izler. Başkalarının kazancı, ancak kendi faydasına girerse önemlidir.
- **Ortak bilgi** (*common knowledge*): Herkes bir bilgiyi biliyorsa, herkes herkesin onu bildiğini biliyorsa, herkes bunun da herkesçe bilindiğini biliyorsa ve bu zincir sonsuza dek sürüyorsa o bilgi ortak bilgidir.

Rasyonellik tanımında iki önemli sözcük vardır. Birincisi “inançlarına göre” ifadesidir. Oyuncu ötekilerin ne yapacağını kesin olarak bilmez; onun hakkında bir tahmini, yani inancı vardır. Rasyonel oyuncu bu inanca göre en iyi seçimi yapar. İnanç yanlışsa sonuç kötü olabilir, ama seçim yine rasyoneldir. İkincisi “beklenen fayda” ifadesidir. Sonuç belirsizse oyuncu olası sonuçların faydalarını olasılıklarıyla tartıp toplar. Bu hesap 2.1.8'de ayrıntılı olarak yapılır.

Kendi çıkarı varsayımı çoğu zaman yanlış anlaşılır. **Rasyonel, bencil demek değildir.** Fayda fonksiyonu adalet, özgecilik ya da itibar kaygılarını da içerebilir. Kazancının bir kısmını gönüllü olarak bağışlayan biri, bağış yapmaktan fayda sağlıyorsa rasyoneldir. Kuram yalnızca şunu söyler: oyuncu neyi önemsiyorsa onu en iyi yapmaya çalışır. Neyin önemsendiği fayda fonksiyonuna yazılır. Sonuç ise hiçbir oyuncunun tek başına kontrol edemediği bir etkileşimin ürünüdür.

Ortak bilgi varsayımı üçünün en soyutudur; etkisi 2/3 ortalaması oyununda açıkça görülür. Şekil 2.2, bu oyunda her bilgi düzeyinin hangi tahminleri elediğini gösterir.



Şekil 2.2: 2/3 ortalaması oyununda bilgi düzeyleri. Her düzey bir aralığı eler: herkes rasyonelse kimse 60'tan büyük yazmaz; herkes bunu biliyorsa kimse 40'tan büyük yazmaz; bunu bildiklerini de biliyorlarsa 26,7'nin üstü elenir. Ortak bilgi (sonsuz düzey) yalnızca 0'ı bırakır.

Düzey 1'de yalnızca "herkes rasyonel" bilgisi vardır. Ortalama en fazla 90 olabileceğinden hedef en fazla 60'tır. 60'tan büyük bir sayı yazmak hiçbir durumda 60 yazmaktan daha iyi değildir; rasyonel oyuncu 60'ın üstünü yazmaz. Düzey 2'de herkes bunu bilir: kimsenin 60'ı aşmayacağını bilen biri ortalamanın en fazla 60, hedefin en fazla 40 olduğunu çıkarır. Düzey 3'te üst sınır 26,7'ye iner. Her düzey bir kat daha "herkes biliyor" ekler ve üst sınırı 2/3 ile çarpar. Ortak bilgi bu zincirin sonsuza dek sürmesidir ve sonunda yalnızca 0 kalır. Bu hesap 2.6.3'te kodla yapılır.

Sık yapılan hata

- **"Herkesin bilmesi yeter."** Herkesin rasyonel olması ile herkesin rasyonel olduğunun ortak bilgi olması farklıdır. Düzey 1'de herkes rasyoneldir, ama yalnızca 60'ın üstü elenir. 0 sonucuna ulaşmak için sonsuz düzey gerekir.
- **"Rasyonel oyuncu parasını hiç azaltmaz."** Fayda fonksiyonu para dışındaki kaygıları da içerebilir. Rasyonellik, kişinin kendi faydasını maksimize etmesidir; faydanın içeriği kişiden kişiye değişir.

2.1.6 Eylem ve strateji

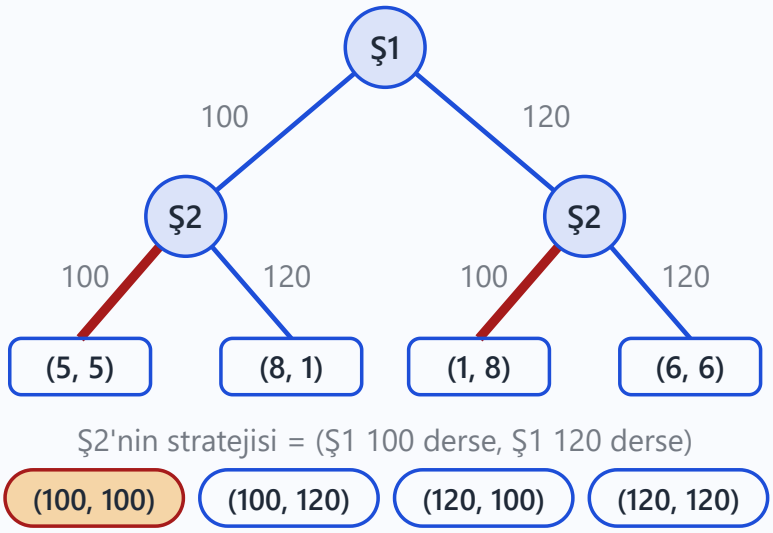
Günlük dilde "eylem" ve "strateji" sözcükleri birbirinin yerine kullanılır. Oyun kuramında ikisi farklıdır ve bu fark özellikle sıralı oyunlarda önemlidir.

Tanım 2.2

Eylem (*action*), oyuncunun bir karar anında yapabileceği tek bir hamledir. **Strateji** (*strategy*) ise oyuncunun seçtiği eylem planıdır. Sıralı oyunlarda strateji, oyuncunun karşılaşabileceği *her olası durumda* ne yapacağını söyleyen tam plandır.

Her oyuncudan bir strateji içeren listeye **strateji profili** (*strategy profile*) denir: $s = (s_1, \ldots, s_n)$. Oyunun sonucunu profil belirler.

Eşzamanlı bir oyunda her oyuncu tek bir kez ve ötekini görmeden karar verir. Bu durumda strateji ile eylem aynı şeydir: Taş-Kağıt-Makas'ta "Taş" hem bir eylem hem de bir stratejidir. Fark, bir oyuncu ötekinin hamlesini gördükten sonra karar verdiğinde ortaya çıkar. O zaman oyuncunun planı, gördüğü her hamle için ayrı bir cevap içermelidir.



Şekil 2.3: Sıralı fiyat oyunu. Şirket 1 (Ş1) fiyatını seçer; Şirket 2 (Ş2) bu fiyatı görüp cevap verir. Ş2'nin her stratejisi bir çift olarak yazılır: (Ş1 100 derse, Ş1 120 derse). Kalın dallar (100, 100) planını gösterir.

Şekil 2.3'te Ş2, Ş1'in fiyatını görüp yanıt verir. Ş2'nin iki karar düğümü vardır: biri Ş1'in 100 TL dediği, öteki 120 TL dediği durumdur. Her düğümde iki eylemden birini (100 ya da 120) seçer. Bir strateji, iki düğümün her biri için bir seçim yapmak demektir. Bu yüzden Ş2'nin strateji sayısı $2 \cdot 2 = 4$'tür: (100, 100), (100, 120), (120, 100) ve (120, 120). Oysa yalnız 2 eylemi vardır. Örneğin (120, 100) stratejisi şunu söyler: "Ş1 100 derse 120 derim, Ş1 120 derse 100 derim."

Genel kural şöyledir: bir oyuncunun k karar düğümü varsa ve her düğümde m eylemden birini seçiyorsa, strateji sayısı m^k'dır. Ş2'nin 3 düğümü olsaydı strateji sayısı $2 \cdot 2 \cdot 2 = 8$ olurdu. Bu sayı düğüm sayısıyla üstel olarak büyür. Bu yüzden büyük sıralı oyunları matris olarak yazmak pratik değildir; ağaç gösterimi (2.2.6) çok daha küçüktür.

Örnek 2.2

Bir öğrenci önce derse gelip gelmeyeceğine karar verir. Öğretmen öğrencinin gelip gelmediğini görür ve iki durumda da "ödev ver" ya da "ödev verme" seçeneklerinden birini seçer. Ödev verilirse öğrenci "yap" ya da "yapma" der. Öğretmenin ve öğrencinin strateji sayıları şöyle bulunur:

1. Öğretmenin karar düğümleri: "öğrenci geldi" ve "öğrenci gelmedi". İki düğüm, her birinde iki eylem: $2^2 = 4$ strateji.
2. Öğrencinin karar düğümleri: başlangıç (gel / gelme), "geldim ve ödev verildi" ve "gelmedim ve ödev verildi". Üç düğüm, her birinde iki eylem: $2^3 = 8$ strateji.
3. Öğrencinin bir stratejisi örneğin şöyledir: (gel; gelince ödev verilirse yap; gelmeyince ödev verilirse yapma). Plan, öğrencinin gerçekte hiç karşılaşmayacağı "gelmedim ve ödev verildi" durumunu da içerir.

> Son maddedeki gözlem önemlidir: strateji, ulaşılmayan durumlar için de bir seçim yapar. Bu ayrıntı, sıralı oyunlarda inandırıcı olmayan tehditleri incelerken belirleyici olur (2.8.2).

Strateji, ötekilerin davranışına ilişkin beklentilere dayanır. Bir oyuncu bir planı seçerken ötekilerin ne yapacağını tahmin eder. Bu tahminin yanında, oyuncunun seçimini nasıl yaptığı da önemlidir. İki tür strateji vardır:

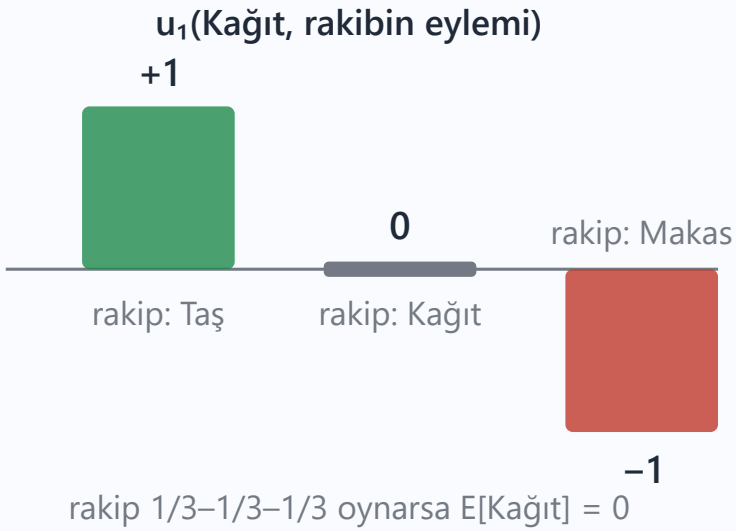
- **Saf strateji** (*pure strategy*): tek bir planı kesin olarak seçmek. Örnek: “her zaman Taş”.
- **Karışık strateji** (*mixed strategy*): saf stratejiler üzerinde bir olasılık dağılımı. Oyuncu hangi saf stratejiyi oynayacağını bir zar atarak belirler. Örnek: (Taş 1/3, Kağıt 1/3, Makas 1/3).

Karışık strateji ilk bakışta tuhaf görünebilir: rasyonel biri neden kararını zara bıraksın? Cevap, tahmin edilebilir olmanın bazı oyunlarda kaybettirmesidir. Taş-Kağıt-Makas'ta hep Taş diyen bir oyuncu, rakibi bunu fark edince hep kaybeder. Rastgele oynamak, rakibin sömürebileceği bir örüntü bırakmaz. Bu fikir 2.2.3'te sayılarla gösterilir.

2.1.7 Kazanç ve fayda fonksiyonu

Kazanç (*payoff*), bir oyuncunun elde ettiği değerdir ve bütün oyuncuların seçtiği strateji profiline bağlıdır. Kazancı ölçen fonksiyona **fayda fonksiyonu** (*utility function*) denir ve oyuncu i için u_i ile gösterilir. Oyuncular stratejilerini kazançlarını maksimize etmek için seçer.

Kazanç para, puan ya da memnuniyet düzeyi olabilir. Bazen kazanç kötü bir şeyin eksi işaretlisidir: hapis yılı ya da bekleme süresi gibi maliyetlerde az olan iyidir, bu yüzden kazanç olarak −(hapis yılı) yazılır. Böylece “büyük sayı iyidir” kuralı bütün oyunlarda geçerli kalır. Karışık stratejilerde ise sonuç belirsiz olduğu için oyuncu **beklenen kazancını** (*expected payoff*) hesaplar.



Şekil 2.4: Taş-Kağıt-Makas'ta Kağıt oynayan oyuncunun kazancı rakibin eylemine göre değişir: rakip Taş derse +1, Kağıt derse 0, Makas derse −1. Rakip üç eylemi 1/3'er olasılıkla oynarsa Kağıt'ın beklenen kazancı 0'dır.

Şekil 2.4 kazancın neden bir profilin fonksiyonu olduğunu gösterir. Oyuncu 1 hep aynı eylemi, Kağıt'ı seçer. Ama kazancı rakibin seçimine göre +1, 0 ya da −1 olur. Bu yüzden kazanç yalnız Kağıt'a değil, (Kağıt, rakibin eylemi) çiftine bağlıdır. Gösterimle: $u_1(\text{Kağıt, Taş}) = +1$, $u_1(\text{Kağıt, Kağıt}) = 0$, $u_1(\text{Kağıt, Makas}) = -1$.

Örnek 2.3

Rakip üç eylemi 1/3'er olasılıkla oynarsa Kağıt'ın beklenen kazancı şöyle hesaplanır. Her olası sonucun kazancı olasılığıyla çarpılır ve çarpımlar toplanır:

$$E[u_1(\text{Kağıt})] = \tfrac{1}{3}\cdot(+1) + \tfrac{1}{3}\cdot 0 + \tfrac{1}{3}\cdot(-1) = 0$$

Rakip daha çok Taş oynasaydı, örneğin (Taş ½, Kağıt ¼, Makas ¼), hesap değişirdi: $\tfrac{1}{2}\cdot 1 + \tfrac{1}{4}\cdot 0 + \tfrac{1}{4}\cdot(-1) = \tfrac{1}{4}$. Kağıt'ın beklenen kazancı artı olurdu, çünkü Kağıt Taş'ı yener. Beklenen kazanç, oyuncunun kendi seçimine ve rakibin olasılıklarına birlikte bağlıdır.

2.1.8 Sıralı ve kardinal fayda

Fayda sayıları iki farklı anlamda kullanılabilir. Bu fark, hangi hesapların anlamlı olduğunu belirler.

Tanım 2.3

Sıralı fayda (*ordinal utility*) yalnızca tercih sırasını söyler. Ayşe'nin tercihleri Opera $\succ$ Futbol $\succ$ Evde ise ($\succ$: "tercih edilir") bu sıra 3, 2, 0 ya da 30, 20, 0 ile yazılabilir; ikisi aynı bilgiyi taşır.

Kardinal fayda (*cardinal utility*) ise farkların büyüklüğünü de taşır; von Neumann–Morgenstern faydası diye de anılır. Belirsiz sonuçlar beklenen fayda ile karşılaştırıldığı için sayıların kendisi önemlidir.

Sıralı faydada sayıları sırayı bozmadan değiştirmek hiçbir şeyi değiştirmez. 3, 2, 0 yerine 30, 20, 0 yazılsa da Ayşe Opera'yı Futbol'a, Futbol'u Evde kalmaya tercih eder. Saf strateji analizi, baskınlık ve saf dengeler yalnızca "hangisi daha iyi?" sorusuna baktığı için bu değişiklikten etkilenmez.

Belirsizlik devreye girince durum değişir. **Piyango** (*lottery*), olası sonuçların olasılıklarıyla birlikte yazıldığı belirsiz bir seçenektir. "½ olasılıkla Opera, ½ olasılıkla Evde" bir piyangodur. Piyangonun değeri, sonuçların faydalarının olasılıklarla tartılmış toplamıdır ve bu toplam sayıların büyüklüğüne bağlıdır.

Örnek 2.4

Ayşe iki seçenek arasında karar verecektir: kesin Futbol ya da ½ Opera + ½ Evde piyangosu. Üç farklı fayda sayısı için karar Tablo 2.3'te hesaplanır.

Fayda (Opera, Futbol, Evde)	Piyango	Kesin Futbol	Seçim
(3, 2, 0)	½·3 + ½·0 = 1,5	2	Futbol
(3, 1, 0): aynı sıra	½·3 + ½·0 = 1,5	1	Piyango
2u + 1 = (7, 5, 1)	½·7 + ½·1 = 4	5	Futbol

Tablo 2.3: Aynı tercih sırasıyla üç fayda ataması ve piyango kararı.

1. İlk satırda piyango 1,5, kesin Futbol 2 verir. 2 > 1,5 olduğundan Ayşe Futbol'u seçer.
2. İkinci satırda tercih sırası aynıdır (Opera ≻ Futbol ≻ Evde), yalnız Futbol'un faydası 2'den 1'e inmiştir. Piyango yine 1,5'tir; ama kesin Futbol artık 1'dir. 1,5 > 1 olduğundan karar piyangoya döner.
3. Üçüncü satırda ilk satırdaki her fayda 2 ile çarpılıp 1 eklenmiştir: (3, 2, 0) → (7, 5, 1). Piyango 4, kesin Futbol 5 verir; karar ilk satırdaki gibi Futbol'dur.

Sonuç: sıra aynı kalsa da sayılar değişince piyango kararı değişebilir (satır 1 ile 2). Buna karşılık $a \cdot u + b$ biçimindeki bir dönüşüm ($a > 0$) kararı korur (satır 1 ile 3). Çünkü böyle bir dönüşümde her beklenen değer de aynı biçimde dönüşür: $½(2 \cdot 3 + 1) + ½(2 \cdot 0 + 1) = 2 \cdot 1{,}5 + 1 = 4$. İki seçeneğin farkı yalnızca a ile çarpılır ve $a > 0$ olduğu için işareti değişmez.

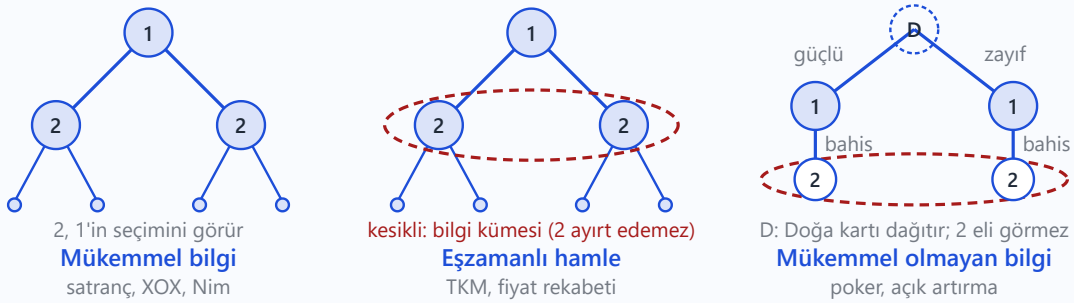
Sık yapılan hata

- **"Kazanç 4, kazanç 2'nin iki katı iyidir."** Bu yorum yalnızca kardinal faydada anlamlıdır. Sayılar yalnızca sırayı gösteriyorsa 4, 2'den iyidir; ama ne kadar iyi olduğu bilinmez.
- **Karışık strateji hesabında sıralı sayılar kullanmak.** Beklenen kazanç hesabı, faydaların kardinal (vNM) olduğunu varsayar. Yalnızca sıra biliniyorsa saf strateji analiziyle yetinilir.

2.1.9 Bilgi: kim neyi biliyor?

Oyuncuların oyun, kazançlar ve yapılan hamleler hakkında bildikleri, stratejiyi ve sonucu doğrudan etkiler. Satrançta iki oyuncu da tahtanın tamamını görür. Pokerde ise rakibin kartları kapalıdır. Bu fark, iki oyunda iyi oynamanın neden çok farklı beceriler istediğini açıklar.

Oyun ağacında bilgi, **bilgi kümesi** (*information set*) ile gösterilir. Bilgi kümesi, sırası gelen oyuncunun birbirinden ayırt edemediği karar düğümlerinin kümesidir. Ağaçta bu düğümler kesikli bir çizgiyle birleştirilir. Oyuncu hangi düğümde olduğunu bilmediği için kümedeki bütün düğümlerde aynı eylemi seçmek zorundadır.



Şekil 2.5: Üç bilgi durumu. Solda oyuncu 2, oyuncu 1'in seçimini görür (mükemmel bilgi; satranç, XOX, Nim). Ortada iki düğüm kesikli bir bilgi kümesindedir; oyuncu 2 bunları ayırt edemez (eşzamanlı hamle; Taş-Kağıt-Makas, fiyat rekabeti). Sağda D (Doğa) kartı dağıtır ve oyuncu 2 eli görmez (mükemmel olmayan bilgi; poker, açık artırma).

Şekil 2.5 üç durumu yan yana koyar. Solda her düğüm kendi başına bir bilgi kümesidir; oyuncu 2, oyuncu 1'in seçimini görür ve buna göre cevap verir. Ortada oyuncu 2'nin iki düğümü aynı bilgi kümesindedir. Oyuncu 1 bir seçim yapmıştır, ama oyuncu 2 bunu görmeden karar verir; bu, iki oyuncunun aynı anda karar vermesiyle aynı şeydir. Sağda ise ilk hamleyi **Doğa** (*Nature*) yapar: elin güçlü ya da zayıf olması bir şans olayıdır. Oyuncu 1 kendi elini görür, oyuncu 2 görmez. Oyuncu 2, oyuncu 1'in bahis yaptığını görür ama elin güçlü mü zayıf mı olduğunu bilmez.

Tanım 2.4

Bir oyunda her bilgi kümesi tek bir düğümden oluşuyorsa, yani her oyuncu karar verirken geçmiş hamlelerin hepsini görüyorsa, oyun **mükemmel bilgili** (*perfect information*) bir oyundur. En az bir bilgi kümesinde birden çok düğüm varsa oyun **mükemmel olmayan bilgili** (*imperfect information*) bir oyundur.

Mükemmel bilgi, *geçmişi* görmek demektir; geleceği bilmek değil. Satranç oyuncusu rakibinin bir sonraki hamlesini bilmez, ama o ana kadar oynanan bütün hamleleri görür. Bu yüzden satranç mükemmel bilgili bir oyundur. Mükemmel bilgi ile **tam bilgi** (*complete information*) de farklıdır; tam bilgi, herkesin kazanç fonksiyonlarını bilmesidir. Bu ayrım 2.3.2'de açılır.

2.1.10 Poker ve blöf

Texas Hold'em pokerinde her oyuncunun iki kapalı kartı vardır. Ortada herkesin görebildiği açık kartlar bulunur ve her oyuncu kendi kapalı kartlarıyla açık kartları birleştirerek en iyi eli oluşturmaya çalışır. Rakibin kapalı kartları görülmez: bilgi gizlidir. Güçlü bir eli olan oyuncu bile rakiplerin ellerini bahislerinden tahmin etmeye çalışır.

Bu oyunda rakibin davranışı, yani bahis yapması ya da beklemesi, eli hakkında bilgi taşır. Yalnızca güçlü elde bahis yapan bir oyuncu kolayca okunur: bahis yaptığında rakipler çekilir, beklediğinde rakipler saldırır. **Blöf** (*bluff*), elin gücünü gizlemek için yanıltıcı bir sinyal göndermektir: zayıf elde de

bahis yapmak.

Oyun kuramı açısından blöf bir **karışık stratejidir**. Zayıf elde belirli bir olasılıkla bahis yapmak, rakibin oyuncuyu okumasını engeller. Olasılık çok düşükse oyuncu yine okunur; çok yüksekse rakip her bahsi görmeye başlar ve blöfler pahalıya patlar. Doğru olasılık, rakibin bahsi görmekle çekilmek arasında kararsız kalacağı noktadır. Bilgisayar poker programları bu dengeleri hesaplayarak oynar. Libratus adlı program 2017'de dört profesyonel oyuncuya karşı uzun bir maç serisini kazandı.

Poker örneği, bilginin oyunun yapısını nasıl değiştirdiğini gösterir. Satrançta “en iyi hamle” kavramı anlamlıdır, çünkü her şey açıktır. Pokerde ise tek bir en iyi hamle yoktur; en iyi davranış, hangi elde hangi olasılıkla ne yapılacağını söyleyen bir karışımdır.

2.2 Oyunların temsili

Bir oyunu yazmanın iki temel yolu vardır. **Normal form** oyunu bir tablo, yani matris olarak yazar; oyuncuların aynı anda karar verdiği durumlar için doğaldır. **Genişletilmiş form** oyunu bir ağaç olarak çizer; hamlelerin sırasını ve her anda kimin neyi bildiğini gösterir. Aynı oyun iki gösterimle de yazılabilir.

2.2.1 Normal form

Tanım 2.5

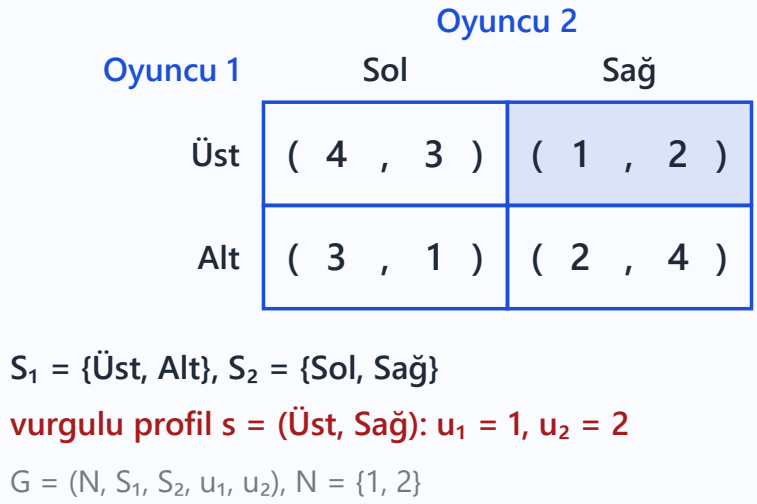
Bir **normal form oyun** (*normal form, strategic form*) $G = (N, (S_i), (u_i))$ üç bileşenden oluşur:

- $N = \{1, \ldots, n\}$: oyuncular kümesi.
- S_i: oyuncu i'nin strateji kümesi.
- $u_i : S_1 \times \ldots \times S_n \to \mathbb{R}$: oyuncu i'nin fayda fonksiyonu. Her strateji profiline bir gerçek sayı atar.

Strateji profili $s = (s_1, \ldots, s_n)$, her oyuncudan bir strateji içerir. Bütün profillerin kümesi $S_1 \times \ldots \times S_n$ ile gösterilir.

Tanımdaki “×” işareti **Kartezyen çarpım**dır: iki kümenin Kartezyen çarpımı, birinci kümeden bir eleman ile ikinci kümeden bir elemanın oluşturduğu bütün sıralı çiftlerin kümesidir. Örneğin $S_1 = \{Üst, Alt\}$ ve $S_2 = \{Sol, Sağ\}$ ise $S_1 \times S_2$ = {(Üst, Sol), (Üst, Sağ), (Alt, Sol), (Alt, Sağ)} olur. Her çift bir strateji profilidir. Fayda fonksiyonunun tanımı $u_i : S_1 \times \ldots \times S_n \to \mathbb{R}$ şunu söyler: u_i bir profili girdi olarak alır ve bir sayı verir. Girdi bütün oyuncuların seçimidir; yalnız oyuncu i'ninki değil.

İki oyunculu bir oyunda normal form bir **kazanç matrisi** (*payoff matrix*) ile yazılır. Satırlar oyuncu 1'in, sütunlar oyuncu 2'nin stratejileridir. Her hücrede iki sayı vardır: önce satır oyuncusunun, sonra sütun oyuncusunun kazancı, yani (u_1, u_2). Bu sıralama bütün not boyunca değişmez.



Şekil 2.6: İki oyunculu bir normal form oyun. S_1 = {Üst, Alt}, S_2 = {Sol, Sağ}. Vurgulu profil s = (Üst, Sağ) için $u_1 = 1, u_2 = 2$'dir.

Şekil 2.6 bu okumayı somutlaştırır. Oyun $G = (N, S_1, S_2, u_1, u_2)$, $N = \{1, 2\}$ biçiminde yazılır. Dört profil vardır ve her biri bir hücredir. Vurgulu hücre (Üst, Sağ) = (1, 2) şu anlama gelir: oyuncu 1 Üst'ü, oyuncu 2 Sağ'ı seçtiğinde oyuncu 1 bir birim, oyuncu 2 iki birim kazanır. Oyuncu 1 yalnızca satırı seçebilir; hangi hücreye düşüleceğini oyuncu 2'nin sütunu belirler.

Örnek 2.5

Şekil 2.6'daki oyunda dört profil için iki fayda fonksiyonunun değerleri şöyledir:

1. $u_1(\text{Üst, Sol}) = 4, u_2(\text{Üst, Sol}) = 3$.
2. $u_1(\text{Üst, Sağ}) = 1, u_2(\text{Üst, Sağ}) = 2$.
3. $u_1(\text{Alt, Sol}) = 3, u_2(\text{Alt, Sol}) = 1$.
4. $u_1(\text{Alt, Sağ}) = 2, u_2(\text{Alt, Sağ}) = 4$.

Oyuncu 1, oyuncu 2'nin Sol oynayacağını bilseydi Üst'ü seçerdi (4 > 3). Sağ oynayacağını bilseydi Alt'ı seçerdi (2 > 1). Demek ki oyuncu 1'in en iyi seçimi oyuncu 2'ye bağlıdır; tablo tek başına “oyuncu 1 şunu oynamalı” demez. Bu bağımlılık, 2.4.1'deki en iyi tepki kavramının çıkış noktasıdır.

Normal form, oyuncuların **eşzamanlı** karar verdiği ya da birbirinin seçimini bilmeden karar verdiği durumları modeller. İki şirketin fiyatlarını aynı gün kapalı zarfla bildirmesi buna bir örnektir. Önemli olan zaman değil bilgidir: biri sabah, öteki akşam karar verse de, akşam karar veren sabahki kararı görmüyorsa oyun eşzamanlıdır.

2.2.2 Taş-Kağıt-Makas normal formda

Taş-Kağıt-Makas'ta (TKM) iki oyuncu ($n = 2$) aynı anda Taş, Kağıt ya da Makas seçer. Taş Makas'ı, Makas Kağıt'ı, Kağıt Taş'ı yener. Kazanana +1, kaybedene −1, beraberlikte ikisine de 0 yazılır.



Şekil 2.7: Taş-Kağıt-Makas'ın kazanç matrisi. Kazanan +1, kaybeden −1, beraberlikte 0; her hücrede iki kazancın toplamı 0'dır.

Şekil 2.7'deki matriste köşegen hücreler (Taş, Taş), (Kağıt, Kağıt), (Makas, Makas) beraberliktir: (0, 0). Öteki hücrelerde biri kazanır, öteki kaybeder. Örneğin (Kağıt, Taş) = (1, −1): satır oyuncusu Kağıt ile Taş'ı yener. Her hücrede iki sayının toplamı 0'dır.

Matrise dikkatle bakınca bir şey daha görülür: hiçbir hücrede iki oyuncu da seçiminden memnun kalmaz. Hep biri seçimini değiştirmek ister. (Taş, Makas) = (1, −1) hücresinde sütun oyuncusu Kağıt'a geçerse kazanır. (Kağıt, Kağıt) = (0, 0) hücresinde oyunculardan her biri tek başına Makas'a geçerek kazanabilir. Dokuz hücrenin hiçbiri bu "kimse değiştirmek istemez" testini geçmez. Bu gözlem, TKM'nin saf stratejilerde bir dengesi olmadığını söyler. Denge kavramı 2.4.1'de tanımlanır.

2.2.3 Taş-Kağıt-Makas'ta strateji seçimi

TKM'yi tekrar tekrar oynayan bir oyuncu hangi kuralı izlemelidir? Üç olası kural karşılaştırılabilir.

1. Rastgele seçim. Her turda her sembol 1/3 olasılıkla seçilir. Bu strateji hiç sömürülemez: rakip ne yaparsa yapsın beklenen kazanç 0'dır. Örneğin rakip hep Taş derse oyuncunun kazancı 1/3 olasılıkla 0 (Taş), 1/3 olasılıkla +1 (Kağıt), 1/3 olasılıkla −1 (Makas) olur; ortalama 0'dır. Aynı hesap rakibin her seçimi için tekrarlanabilir.

2. Geçmişe göre seçim. "Rakip son turda Taş dediyse bu tur Kağıt de." Bu kural, örüntülü oynayan bir rakibe karşı kazandırır. Ama kuralın kendisi de bir örüntüdür. Rakip bu kuralı fark ederse, son turda Taş deyip bu tur Makas diyerek oyuncuyu yener.

3. Dengesiz karışık strateji. Örneğin zamanın yarısında Kağıt, yarısında Makas.

Örnek 2.6

Oyuncu ½ Kağıt, ½ Makas oynar. Rakibin üç saf seçeneğinin beklenen kazancı ayrı ayrı hesaplanır (rakibin bakış açısından):

1. Rakip Taş derse: Kağıt'a karşı −1, Makas'a karşı +1. Beklenen $½ \cdot (−1) + ½ \cdot (+1) = 0$.
2. Rakip Kağıt derse: Kağıt'a karşı 0, Makas'a karşı −1. Beklenen $½ \cdot 0 + ½ \cdot (−1) = −½$.
3. Rakip Makas derse: Kağıt'a karşı +1, Makas'a karşı 0. Beklenen $½ \cdot (+1) + ½ \cdot 0 = +½$.

Rakip hep Makas derse beklenen kazancı +1/2, oyuncununki −1/2 olur. Bu karışım sömürülebilir. Oyuncu tahmin edilemezdir, çünkü hangi turda Kağıt diyeceği bilinmez; ama Taş'ı hiç seçmediği için Makas ona karşı güvenlidir.

Üç kuralın karşılaştırması bir ders içerir: basit bir oyunda bile strateji seçimi önemlidir. Tahmin edilemez olmak yetmez; *doğru olasılıklarla* tahmin edilemez olmak gerekir. TKM'de doğru olasılıklar 1/3'tür ve bu olasılıklar herhangi bir rakibe karşı 0 garanti eder.

2.2.4 Sıfır toplamlı oyunlar

Tanım 2.6

İki oyunculu bir oyun, her strateji profilinde kazançların toplamı sıfırsa **sıfır toplamlı**dır (*zero-sum game*): her s için $u_1(s) + u_2(s) = 0$. Toplam sıfır değil de sabit bir sayıysa oyun **sabit toplamlı**dır (*constant-sum game*); aynı analiz bu oyunlarda da geçerlidir.

Sıfır toplamlı oyunda birinin kazancı, ötekinin kaybıdır. TKM, Yazı-Tura ve satranç (+1 / 0 / −1 puanlamasıyla) sıfır toplamlıdır. Bir puanın iki oyuncu arasında paylaşıldığı bir oyun da sabit toplamlıdır: biri 0,7 alırsa öteki 0,3 alır. Buna karşılık Mahkûmun İkilemi ve fiyat rekabeti sıfır toplamlı değildir; bu oyunlarda toplam kazanç sonuca göre değişir.

Sıfır toplamlı oyunda tek bir matris yeter, çünkü sütun oyuncusunun kazancı satır oyuncusununkinin eksi işaretlisidir. Satır oyuncusu kazancını büyütmek, sütun oyuncusu satır oyuncusunun kazancını küçültmek ister. Bu durumda iki doğal ölçüt ortaya çıkar. **Maksimin** (*maximin*): satır oyuncusu her satırın en kötü durumuna bakar ve en kötü durumu en iyi olan satırı seçer. **Minimaks** (*minimax*): sütun oyuncusu her sütunda satır oyuncusunun alabileceği en iyi sonuca bakar ve bunu en küçük yapan sütunu seçer. Oyun ağacındaki minimaks araması (2.7.2), aynı fikri sıralı oyunda yapraktan köke uygulayan algoritmadır.



Şekil 2.8: Yazı-Tura Eşleştirme. Eşleştiren iki para aynı yüzü gösterirse, ayrıştıran farklı yüzü gösterirse 1 kazanır. $u_2 = -u_1$ olduğundan yalnızca satır oyuncusunun kazançları yazılan A matrisi oyunu tam olarak tanımlar.

Şekil 2.8 bu kısaltmayı gösterir. Üstteki tablo iki kazancı da yazar; alttaki A matrisi yalnızca satır oyuncusunun kazancını yazar. Sütun oyuncusunun kazancı her hücrede A'daki sayının eksi işaretlisidir. Bu kısaltma sıfır toplamlı olmayan oyunlarda yapılamaz, çünkü orada bir oyuncunun kazancı ötekininkinden çıkarılamaz.

Örnek 2.7

Yazı-Tura'da $A = [[1, -1], [-1, 1]]$'dir. Saf stratejilerde maksimin ve minimaks şöyle bulunur:

1. Satır oyuncusunun her satırdaki en kötü sonucu: Yazı satırında $\min(1, -1) = -1$, Tura satırında $\min(-1, 1) = -1$. Maksimin = $\max(-1, -1) = -1$.
2. Sütun oyuncusu her sütunda satır oyuncusunun en iyisine bakar: Yazı sütununda $\max(1, -1) = 1$, Tura sütununda $\max(-1, 1) = 1$. Minimaks = $\min(1, 1) = 1$.
3. Maksimin (-1) ile minimaks (1) eşit değildir. Saf stratejilerle oynayan iki oyuncu da kendisi için ancak en kötü durumu garanti edebilir.

İki değer arasındaki boşluk, oyunda saf bir "kararlı nokta" olmadığını gösterir. Karışık stratejilere (½–½) geçildiğinde iki oyuncu da beklenen 0'ı garanti eder ve boşluk kapanır. von Neumann'ın 1928'de ispatladığı minimaks teoremi, karışık stratejilerde bu iki değerin her sonlu sıfır toplamlı oyunda eşit olduğunu söyler.

Sıfır toplamlı oyunda ortak çıkar yoktur; oyun saf rekabettir. Sıfır toplamlı olmayan oyunlarda ise herkesin kazandığı ya da herkesin kaybettiği sonuçlar olabilir. İşbirliği ile rekabet bir aradadır; iletişim ve müzakere sonucu değiştirebilir. Ticaret müzakereleri ve iklim anlaşmaları bu tür oyunlardır: anlaşma sağlanırsa iki taraf da kazanır, sağlanamazsa iki taraf da kaybeder.

2.2.5 Fiyat rekabeti oyunu

İki şirket aynı pazarda benzer bir ürün satar ve fiyatını aynı anda belirler: 100 TL ya da 120 TL. Ucuz olan müşteriyi kapar; ikisi de pahalı satarsa kâr marjı yüksek kalır. Tablo 2.4 kârları birim olarak verir.

Ş1 \ Ş2	100 TL	120 TL
100 TL	**(5, 5)**	(8, 1)
120 TL	(1, 8)	(6, 6)

Tablo 2.4: Fiyat rekabeti oyunu. Hücrede (Şirket 1, Şirket 2) kârı.

İkisi de 100 TL derse 5'er, ikisi de 120 TL derse 6'şar birim kâr ederler. Biri 100, öteki 120 derse ucuz olan müşterilerin çoğunu alır ve 8 kazanır; pahalı olan yalnızca 1 kazanır.

Örnek 2.8

Şirket 1'in bakış açısından iki durum ayrı ayrı incelenir:

1. Şirket 2 100 TL derse: Şirket 1, 100 ile 5, 120 ile 1 kazanır. 5 > 1 olduğundan 100 daha iyidir.
2. Şirket 2 120 TL derse: Şirket 1, 100 ile 8, 120 ile 6 kazanır. 8 > 6 olduğundan yine 100 daha iyidir.
3. Karşı taraf ne yaparsa yapsın 100 TL daha çok kazandırır. Böyle bir stratejiye **baskın strateji** (*dominant strategy*) denir.
4. Oyun simetrik olduğundan aynı hesap Şirket 2 için de geçerlidir: onun için de 100 TL baskındır.

Sonuç (100, 100) olur ve iki şirket 5'er kâr eder. Oysa (120, 120) ikisine de 6 verirdi. Bireysel olarak en iyi seçim, ikisini birden daha kötü bir sonuca götürür.

Bu yapı, 2.4'te ayrıntılı incelenen Mahkûmun İkilemi'nin yapısıdır. Fiyat rekabeti oyunu sıfır toplamlı değildir: hücrelerdeki toplamlar 10, 9, 9 ve 12'dir. Toplamın en büyük olduğu hücre (120, 120)'dir, ama oyuncular oraya ulaşamaz.

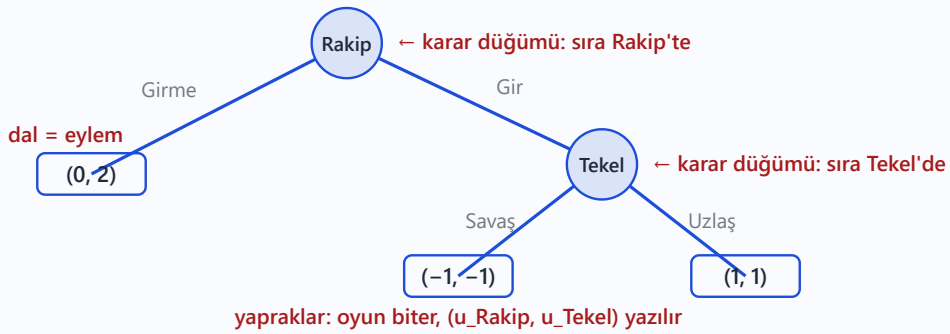
2.2.6 Genişletilmiş form

Normal form, kimin önce oynadığını göstermez. Sıralı etkileşimlerde (satranç, pazarlık, piyasaya giriş) sıra önemlidir: ikinci oynayan, birincinin hamlesini görerek karar verir. Bu durumları yazmanın doğal yolu oyun ağacıdır.

Tanım 2.7

Genişletilmiş form (*extensive form*), kararların sırasını ve her anda kimin neyi bildiğini gösteren bir **oyun ağacıdır** (*game tree*). Ağacın parçaları şunlardır:

- **Kök** (*root*): oyunun başladığı düğüm.
- **Karar düğümü** (*decision node*): bir oyuncunun karar verdiği nokta; düğümün üzerinde sırası gelen oyuncu yazılır.
- **Dal** (*branch*): bir eylem.
- **Yaprak** (*leaf*, *terminal node*): oyunun bittiği nokta; her oyuncunun kazancı yazılır.
- **Bilgi kümesi**: aynı oyuncunun ayırt edemediği düğümler (kesikli çizgiyle).



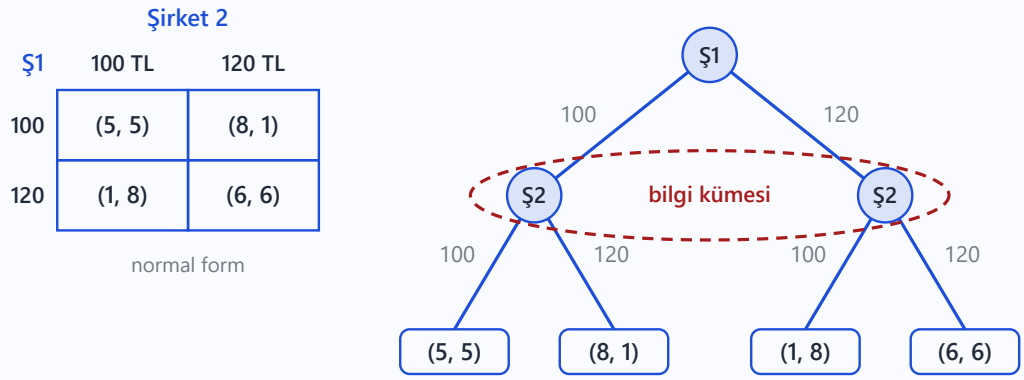
Şekil 2.9: Piyasaya giriş oyununun ağacı. Rakip (giren firma) pazara girer ya da girmez; girerse Tekel (mevcut firma) savaşır ya da uzlaşır. Yapraklarda (u_{Rakip}, u_{Tekel}) yazılır.

Şekil 2.9'da kök, Rakip'in karar düğümüdür. Rakip "Girme" derse oyun biter: Rakip 0 kazanır, Tekel pazarı tek başına tutar ve 2 kazanır. Rakip "Gir" derse sıra Tekel'e geçer. Tekel "Savaş" derse fiyat savaşı ikisine de zarar verir: (−1, −1). Tekel "Uzlaş" derse pazarı paylaşırlar: (1, 1). Ağaç, sırayı ve her karar anında kimin oynadığını açıkça gösterir. Bu oyunun çözümü 2.8.2'de verilir.

2.2.7 Normal formdan oyun ağacına

Eşzamanlı bir oyun da ağaç olarak çizilebilir. Bunun için ağaçta bilgi kümesi kullanılır. Fiyat rekabeti oyunu (Tablo 2.4) üzerinden dönüşüm adım adım şöyle yapılır:

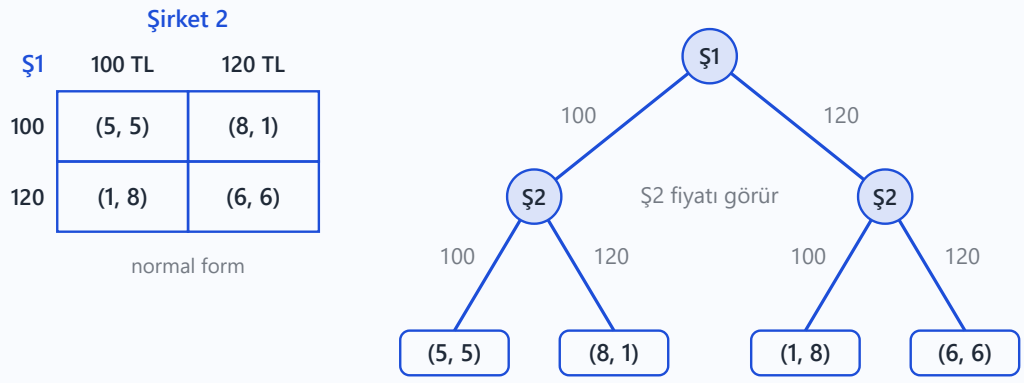
1. **Kök:** Şirket 1'in kararı. Her dal bir eylemdir: 100 TL ya da 120 TL.
2. **İkinci düzey:** her dalın sonunda Şirket 2'nin bir karar düğümü vardır. O da 100 ya da 120 seçer.
3. **Yapraklar:** dört yol vardır ve her biri matrisin bir hücresine karşılık gelir. Örneğin 100–120 yolu (8, 1) hücresidir.
4. **Bilgi kümesi:** iki şirket fiyatı aynı anda belirlediği için Ş2, Ş1'in seçimini görmez. Ş2'nin iki düğümü kesikli bir çizgiyle aynı bilgi kümesine konur.



Şekil 2.10: Fiyat rekabeti oyununun matrisi (solda) ve ağacı (sağda). Ş2'nin iki düğümü aynı bilgi kümesindedir: Ş2, Ş1'in fiyatını görmez. Bu ağaç, matrisle aynı oyundur.

Şekil 2.10'daki ağaç, matrisle tam olarak aynı oyunu anlatır. Ağaç Ş1'i önce çizer, ama bu bir zaman sırası değildir; bilgi kümesi Ş2'nin Ş1'in seçimini bilmediğini söyler. Ş1'i yukarıya, Ş2'yi aşağıya koymak yerine tersini yapmak da aynı oyunu verirdi.

Bilgi kümesi kaldırılırsa oyun değişir. O zaman Ş2, Ş1'in fiyatını gördükten sonra karar verir: oyun artık sıralıdır.



Şekil 2.11: Bilgi kümesi kaldırılınca Ş2, Ş1'in fiyatını görür. Yapraklar aynıdır, ama oyun artık sıralıdır ve geriye doğru çıkarımla çözülür.

Şekil 2.11'deki oyunda yapraklar aynıdır, ama Ş2'nin strateji sayısı değişmiştir. Eşzamanlı oyunda Ş2'nin iki stratejisi vardı (100 ya da 120), çünkü iki düğümünü ayırt edemiyordu ve ikisinde aynı şeyi seçmek zorundaydı. Sıralı oyunda iki düğüm ayrıdır ve Ş2'nin dört stratejisi vardır (Şekil 2.3). Bilgi, strateji kümesini doğrudan değiştirir.

Örnek 2.9

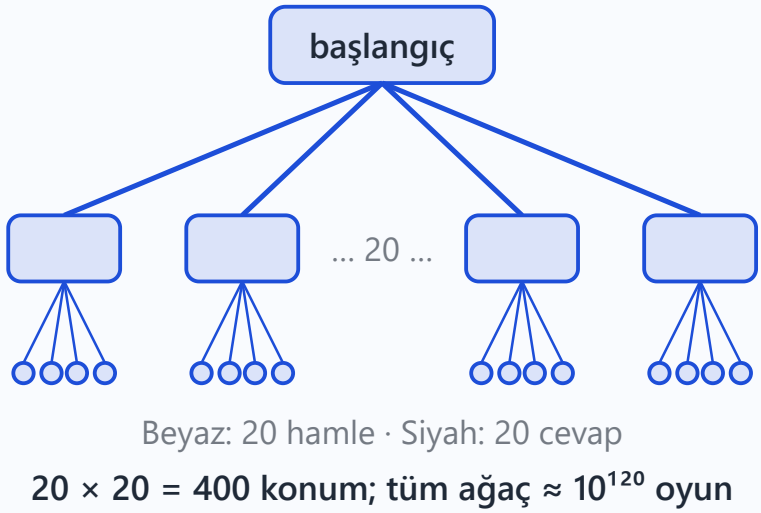
Sıralı fiyat oyunu, Şekil 2.11'deki ağaçta sondan başa doğru çözülür:

1. Ş1 100 dediyse Ş2'nin seçenekleri: 100 ile 5, 120 ile 1. Ş2 100 der; düğümün değeri (5, 5).
2. Ş1 120 dediyse Ş2'nin seçenekleri: 100 ile 8, 120 ile 6. Ş2 100 der; düğümün değeri (1, 8).
3. Ş1 bunları bilerek karşılaştırır: 100 ile 5, 120 ile 1. Ş1 100 der.

Sonuç yine (100, 100) = (5, 5)'tir. Ş2'nin stratejisi (100, 100), yani "Ş1 ne derse desin 100 derim" planıdır. Bu oyunda sıranın sonucu değiştirmemesinin nedeni, 100'ün iki oyuncu için de baskın olmasıdır. Sıra, baskın stratejisi olmayan oyunlarda sonucu değiştirebilir.

2.2.8 Satranç bir genişletilmiş form oyunudur

Satranç, genişletilmiş formun en tanınmış örneğidir. İki oyuncu sırayla hamle yapar; her hamle bir dal, her konum bir düğümdür. Oyun mükemmel bilgilidir: iki oyuncu da tahtayı ve bütün geçmiş hamleleri görür. Sonuç galibiyet, yenilgi ya da beraberliktir. Beraberlik pat, üç kez aynı konumun tekrarı ya da 50 hamle kuralıyla oluşabilir.



Şekil 2.12: Satranç ağacının ilk iki düzeyi. Beyaz'ın 20 açılış hamlesi ve Siyah'ın her birine 20 cevabı vardır: iki yarım hamlede 20 × 20 = 400 konum. Tüm ağaç yaklaşık 10^{120} oyun içerir.

Ernst Zermelo 1913'te bu tür oyunlar için temel bir sonuç ispatladı.

Teorem 2.1 (Zermelo)

İki oyunculu, sonlu, mükemmel bilgili ve şans içermeyen bir oyunda ya bir taraf kazanmayı garanti edebilir ya da iki taraf da en az beraberliği garanti edebilir.

Teorem, satrancın sonucunun “belirli” olduğunu söyler: iki taraf da hatasız oynarsa ya Beyaz kazanır, ya Siyah kazanır ya da oyun berabere biter. Ama hangisinin olduğu bilinmez, çünkü ağaç çok büyüktür. Şekil 2.12 bu büyüklüğü gösterir. İlk iki yarım hamlede 400 konum vardır. Her yarım hamlede ortalama 30 kadar seçenek olduğu ve oyunlar onlarca hamle sürdüğü için tüm ağaçtaki oyun sayısı yaklaşık 10^{120}'dir. Bu sayı, gözlemlenebilir evrendeki atom sayısından (yaklaşık 10^{80}) çok büyüktür.

Bu yüzden satranç programları ağacın yalnızca bir kısmını arar. Arama belirli bir derinlikte durur, oradaki konumlar bir değerlendirme fonksiyonuyla puanlanır ve gereksiz dallar budanır (minimaks ve budama). XOX, Nim ve “21'i Söyleme” gibi küçük oyunların ağacı ise tamamen çözülebilir. Bu oyunlar 2.7'de çözülür.

2.3 Oyunların sınıflandırılması

Oyunlar çok çeşitlidir; her birini ayrı ayrı incelemek yerine ortak özelliklerine göre gruplamak işe yarar. Bir oyunun hangi gruba girdiği, hangi çözüm yönteminin kullanılacağını belirler. Örneğin sıralı ve mükemmel bilgili bir oyun geriye doğru çıkarımla çözülür; eşzamanlı bir oyun matris üzerinde en iyi tepkilerle çözülür.

2.3.1 Altı boyut

Bir oyunu sınıflandırmak için altı soru sorulur. Her soru bir boyuttur ve her boyutun iki ucu vardır. Tablo 2.5 altı boyutu, her boyutta sorulan soruyu ve iki uca birer örneği verir.

Boyut	Bir uç	Diğer uç	Sorulan soru	Örnek
Zamanlama	Eşzamanlı	Sıralı	Rakibin hamlesini görerek mi oynuyorum?	kapalı zarf teklifi / piyasaya giriş
Anlaşma	İşbirlikçi	İşbirlikçi olmayan	Bağlayıcı anlaşma yapılabilir mi?	ortak girişim / Mahkûmun İkilemi
Toplam kazanç	Sıfır toplamlı	Sıfır toplamlı olmayan	Birinin kazancı diğerinin kaybı mı?	penaltı / iklim anlaşması
Simetri	Simetrik	Asimetrik	Oyuncuların yerini değiştirince oyun aynı mı?	Geyik Avı / Cinsiyetler Savaşı
Bilgi	Mükemmel bilgi	Mükemmel olmayan bilgi	Geçmiş hamlelerin hepsini görüyor muyum?	Dörtlü Bağla / poker
Ufuk	Sonlu	Sonsuz	Oyunun belirli bir sonu var mı?	Kırkayak / sürekli fiyat rekabeti

Tablo 2.5: Oyunların sınıflandırılmasında altı boyut.

Zamanlama. Eşzamanlı oyunda oyuncular birbirinin seçimini görmeden karar verir; kapalı zarf açık artırması buna örnektir. Sıralı oyunda biri ötekinin hamlesini görerek oynar; piyasaya giriş oyununda Tekel, Rakip'in girdiğini gördükten sonra karar verir. 2.2.7'de görüldüğü gibi iki tür arasındaki fark bir bilgi kümesidir.

Anlaşma. İşbirlikçi oyunda (*cooperative game*) oyuncular *bağlayıcı* anlaşmalar yapabilir: verilen söz dışarıdan, örneğin bir sözleşme ve mahkeme ile uygulatılabilir. İşbirlikçi olmayan oyunda (*non-cooperative game*) böyle bir zorlama yoktur; oyuncular konuşabilir ama söz bağlayıcı değildir. İki tür oyunun analiz birimi de farklıdır. İşbirlikçi oyunda soru şudur: hangi grup, yani hangi **koalisyon** (*coalition*) birlikte hareket eder ve kazancı nasıl paylaşır? Shapley değeri ve çekirdek bu sorunun cevaplarıdır. İşbirlikçi olmayan oyunda analiz birimi tek oyuncudur ve temel kavram Nash dengesidir. "İşbirlikçi olmayan" adı oyuncuların hiç işbirliği yapmadığı anlamına gelmez: işbirliği ortaya çıkabilir, ama ancak her oyuncunun kendi çıkarına uygunsa.

Toplam kazanç. Sıfır toplamlı oyunda (2.2.4) birinin kazancı ötekinin kaybıdır; penaltı atışı buna örnektir: gol atıcıya kazanç, kaleciye kayıptır. İklim anlaşmasında ise herkesin kazandığı ya da herkesin kaybettiği sonuçlar vardır.

Simetri ve **ufuk** boyutları kısa bir biçimsel tanım ister. **Bilgi** boyutu 2.1.9'da tanımlandı: Dörtlü Bağla'da iki oyuncu da tahtanın tamamını görür, pokerde rakibin kartları kapalıdır.

2.3.2 Tam bilgi, simetri ve ufuk

Tam bilgi (*complete information*), mükemmel bilgiden ayrı bir kavramdır. Tam bilgide herkesin kazanç fonksiyonu herkesçe bilinir: her oyuncu, ötekilerin neyi ne kadar istediğini bilir. Bilinmiyorsa oyunda **eksik bilgi** (*incomplete information*) vardır. Açık artırmada her teklifçi, ötekilerin ürüne biçtiği değeri bilmez; bu bir eksik bilgi durumudur. Eksik bilgili oyunlar **Bayesçi oyunlar** (*Bayesian games*) olarak modellenir: her oyuncunun bilinmeyen özelliği bir olasılık dağılımından çekilmiş gibi düşünülür.

İki kavramın farkı şöyle özetlenebilir. Mükemmel bilgi, *hamleler* hakkındadır: geçmişte kim ne oynadı? Tam bilgi, *oyunun kendisi* hakkındadır: herkesin kazançları ne? Satranç hem mükemmel hem de tam bilgilidir. Taş-Kağıt-Makas tam bilgilidir (kazançlar herkesçe bilinir), ama mükemmel bilgili değildir (hamleler aynı anda yapılır).

Tanım 2.8

İki oyunculu bir oyunda iki oyuncunun strateji kümeleri aynıysa ve her a, b için

$$u_1(a, b) = u_2(b, a)$$

sağlanıyorsa oyun **simetrik**tir (*symmetric game*). Yani kazanç kimin oynadığına değil, yalnızca hangi stratejinin hangi stratejiye karşı oynandığına bağlıdır.

Tanımdaki eşitlik dikkatle okunmalıdır: satır oyuncusu a'yı, sütun oyuncusu b'yi oynadığında satır oyuncusunun aldığı, roller değiştiğinde (satır b, sütun a) sütun oyuncusunun aldığına eşit olmalıdır. Matriste bu, (a, b) hücresindeki ilk sayının (b, a) hücresindeki ikinci sayıya eşit olması demektir.

Örnek 2.10

Fiyat rekabeti oyunu (Tablo 2.4) simetri tanımıyla denetlenir. Dört eşitlik gerekir:

1. $u_1(100, 100) = 5$ ve $u_2(100, 100) = 5$: eşit.
2. $u_1(100, 120) = 8$ ve $u_2(120, 100) = 8$: eşit.
3. $u_1(120, 100) = 1$ ve $u_2(100, 120) = 1$: eşit.
4. $u_1(120, 120) = 6$ ve $u_2(120, 120) = 6$: eşit.

Dört eşitlik de sağlandığı için oyun simetriktir. Cinsiyetler Savaşı'nda ise (Opera, Opera) = (3, 2)'dir: $u_1(\text{Opera, Opera}) = 3 \neq u_2(\text{Opera, Opera}) = 2$. İlk adımda tanım bozulur; oyun asimetriktir.

Ufuk (*horizon*), oyunun kaç tur sürdüğüdür. Sonlu ufuklu oyunun belirli bir sonu vardır; kırkayak oyunu (2.8.5) dört hamlede biter. Sonsuz ufuklu oyunda ise oyunun ne zaman biteceği belli değildir; iki şirketin her ay yeniden fiyat belirlemesi böyledir. Sonsuz ufukta gelecek kazançlar bir **iskonto faktörü** (*discount factor*) $\delta \in (0, 1)$ ile tartılır ve oyuncunun toplam kazancı şöyle yazılır:

$$U = u_0 + \delta\, u_1 + \delta^2\, u_2 + \ldots = \Sigma_t\, \delta^t\, u_t$$

Burada u_t, t. turun kazancıdır. δ küçükse gelecek az önemlidir: oyuncu sabırsızdır. δ 1'e yakınsa gelecek neredeyse bugün kadar önemlidir. δ'nın ikinci bir yorumu da vardır: her turdan sonra oyunun bir tur daha sürme olasılığı. Bu yorumda δ^t, oyunun t. tura ulaşma olasılığıdır.

Örnek 2.11

Bir oyuncu her tur 1 birim kazanır ve $\delta = 0{,}9$'dur. Toplam kazancı şöyle hesaplanır:

1. İlk birkaç terim: $1 + 0{,}9 + 0{,}81 + 0{,}729 + \ldots$
2. Bu bir geometrik seridir. Toplamı S olsun. $S = 1 + \delta + \delta^2 + \ldots$ ise $\delta S = \delta + \delta^2 + \ldots$ olur. İki eşitlik birbirinden çıkarılınca $S - \delta S = 1$, yani $S = 1 / (1 - \delta)$ bulunur.
3. $\delta = 0{,}9$ için $S = 1 / 0{,}1 = 10$.

Sonsuz tur oynansa da toplam sonludur, çünkü uzak turların ağırlığı giderek küçülür. Bir dönem sonraki 10 birim, bugün $0{,}9 \cdot 10 = 9$ birim değerindedir. Bu formül, sonsuz tekrarlı Mahkûmun İkilemi'nde (2.8.8) işbirliğinin ne zaman sürdüğünü hesaplamak için kullanılır.

2.3.3 Bir oyunu sınıflandırmak

Altı boyut birbirinden bağımsızdır; bir oyun her boyutta ayrı ayrı konumlandırılır. Satranç örneğinde altı cevap şöyledir: sıralı (oyuncular sırayla oynar), işbirlikçi olmayan (bağlayıcı anlaşma yoktur), sıfır toplamlı (biri kazanırsa öteki kaybeder), simetrik değil (Beyaz önce oynar; ancak kurallar iki taraf

için aynıdır), mükemmel bilgili (tahta açıktır) ve sonlu (beraberlik kuralları oyunun sonsuza dek sürmesini engeller).

Örnek 2.12

Bu bölümün üç oyunu altı boyutta sınıflandırılır:

Boyut	Taş-Kağıt-Makas	Mahkûmun İkilemi	Piyasaya giriş
Zamanlama	Eşzamanlı	Eşzamanlı	Sıralı
Anlaşma	İşbirlikçi olmayan	İşbirlikçi olmayan	İşbirlikçi olmayan
Toplam kazanç	Sıfır toplamlı	Sıfır toplamlı olmayan	Sıfır toplamlı olmayan
Simetri	Simetrik	Simetrik	Asimetrik (roller farklı)
Bilgi	Mükemmel olmayan	Mükemmel olmayan	Mükemmel bilgi
Ufuk	Sonlu (tek tur)	Sonlu (tek tur)	Sonlu

Tablo 2.6: Üç oyunun sınıflandırması.

Eşzamanlı oyunlar ağaçta bir bilgi kümesi gerektirdiği için mükemmel bilgili değildir. Piyasaya giriş oyununda ise Tekel, Rakip'in girdiğini görerek karar verir; her bilgi kümesi tek düğümdür. Bu oyunun sonuçlarındaki toplamlar 2, −2 ve 2'dir; toplam sabit olmadığından oyun sıfır toplamlı değildir. Rakip ile Tekel'in seçenekleri ve kazançları farklı olduğu için oyun asimetriktir.

Sık yapılan hata

- **Mükemmel bilgi ile tam bilgiyi karıştırmak.** Mükemmel bilgi geçmiş hamlelerle, tam bilgi kazanç fonksiyonlarıyla ilgilidir. TKM tam bilgilidir ama mükemmel bilgili değildir.
- **"Konuşabiliyorlarsa oyun işbirlikçidir."** Konuşmak yetmez; anlaşmanın *bağlayıcı* olması gerekir. Mahkûmlar sorgudan önce söz verse bile oyun işbirlikçi olmayan bir oyun olarak kalır.
- **Simetriyi yalnız köşegene bakarak denetlemek.** (a, a) hücrelerinde iki sayının eşit olması yetmez; köşegen dışındaki (a, b) ile (b, a) hücreleri de çapraz olarak eşleşmelidir.

2.4 Denge ve Mahkûmun İkilemi

Bir oyunu yazmak, onu çözmek demek değildir. Oyunun "çözümü", oyuncuların hangi stratejileri seçeceğine dair bir öngörüdür. Oyun kuramının en temel çözüm kavramı Nash dengesidir. Kavram burada sezgisel düzeyde kalır; Mahkûmun İkilemi ise denge ile verimliliğin farklı şeyler olduğunu gösterir.

2.4.1 En iyi tepki ve Nash dengesi

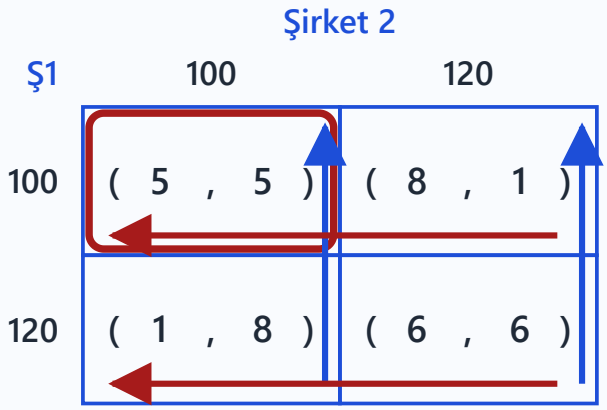
Tanım 2.9

Oyuncu i dışındaki oyuncuların stratejileri s_{-i} ile gösterilir. s_{-i} sabitken oyuncu i'ye en yüksek kazancı veren stratejiye **en iyi tepki** (*best response*) denir.

Herkesin stratejisinin ötekilerin stratejilerine bir en iyi tepki olduğu profile **Nash dengesi** (*Nash equilibrium*) denir ve s^* ile gösterilir. Dengede hiç kimse tek taraflı sapmak istemez (Nash, 1950).

Tanım bir "pişmanlık testi" olarak okunabilir. Oyun oynandıktan sonra her oyuncu kendine şunu sorar: "Ötekilerin ne yaptığını bilseydim, seçimimi değiştirir miydim?" Herkesin cevabı "hayır" ise profil bir Nash dengesidir. Tek bir oyuncunun cevabı "evet" ise değildir.

İki oyunculu bir matriste en iyi tepkiler ve dengeler oklarla görselleştirilebilir. Her hücreden, oyunculardan birinin sapmak istediği yöne bir ok çizilir. Hiçbir okun çıkmadığı hücre dengedir.



mavi ↑: Şirket 1 saparak kazanır · kırmızı ←: Şirket 2 sapar

okların hepsi (100, 100)'e akar: kimse ayrılmak istemez

Şekil 2.13: Fiyat rekabeti oyununda sapma okları. Mavi oklar Şirket 1'in (satır oyuncusunun) kazandıran sapmalarını, kırmızı oklar Şirket 2'nin (sütun oyuncusunun) kazandıran sapmalarını gösterir. Bütün oklar (100, 100)'e akar; bu hücreden ok çıkmaz.

Örnek 2.13

Şekil 2.13'teki oklar dört hücre için şöyle bulunur:

1. (120, 120) = (6, 6): Ş1 100'e geçerse 8 alır (8 > 6), yukarı ok. Ş2 100'e geçerse 8 alır, sola ok. İki ok çıkar.

2. (100, 120) = (8, 1): Ş2 100'e geçerse 5 alır (5 > 1), sola ok. Ş1 120'ye geçerse 6 alır (6 < 8); Ş1 sapmaz.
3. (120, 100) = (1, 8): Ş1 100'e geçerse 5 alır (5 > 1), yukarı ok. Ş2 sapmaz (6 < 8).
4. (100, 100) = (5, 5): Ş1 120'ye geçerse 1 alır (1 < 5); Ş2 120'ye geçerse 1 alır. Hiçbir ok çıkmaz.

Tek denge (100, 100)'dür. Sapma okları dengeye girer, dengeden çıkmaz.

Nash dengesinin *söylediği* şudur: ötekiler sabitken sapmak kazandırmaz; kimse pişman değildir. Denge kendi kendini sürdüren bir durumdur. Nash dengesinin *söylemediği* ise dengenin herkes için iyi olduğudur. Fiyat rekabetinde (120, 120) iki şirkete de 6 verirdi; denge ise 5'er verir. Mahkûmun İkilemi de aynı yapıdadır. Burada kavramın sezgisi ve klasik örneklerdeki kullanımı yeterlidir.

2.4.2 Mahkûmun İkilemi: hikâye ve matris

İki şüpheli, Sam ve John, ayrı odalarda sorgulanır ve birbirleriyle konuşamazlar. Her birinin iki seçeneği vardır: sessiz kalmak ya da suç ortağını ele verip itiraf etmek. Tablo 2.7 dört durumdaki cezaları verir.

Durum	Sam	John
İkisi de sessiz	1 yıl	1 yıl
Sam itiraf eder, John sessiz	serbest	3 yıl
Sam sessiz, John itiraf eder	3 yıl	serbest
İkisi de itiraf eder	2 yıl	2 yıl

Tablo 2.7: Mahkûmun İkilemi'nde hapis süreleri.

Terim birliği için iki adlandırma sabitlenir. **İşbirliği**, suç ortağıyla işbirliği yapmak, yani *sessiz kalmak*tır. **İhanet** ise itiraf etmektir (polisle işbirliği). Kısaca iki strateji “Sessiz” ve “İhanet” olarak yazılır.

Hapis yılı kötü bir şeydir; az olması iyidir. Kazanç olarak −(hapis yılı) yazılır: az ceza, büyük kazanç demektir. Tablo 2.8 oyunun kazanç matrisini verir. Her hücrede ilk sayı Sam'in (satır), ikinci sayı John'un (sütun) kazancıdır ve hücreler a, b, c, d olarak adlandırılmıştır.

Sam \ John	Sessiz	İhanet
Sessiz	a (−1, −1)	b (−3, 0)
İhanet	c (0, −3)	**d (−2, −2)**

Tablo 2.8: Sam ve John'un kazanç matrisi. Kazanç = −(hapis yılı).

Tablo şöyle okunur. İkisi de sessiz kalırsa (hücre a) ikisi de −1 alır. Sam sessiz kalıp John ihanet ederse (hücre b) Sam −3, John 0 alır: Sam 3 yıl yatar, John serbest kalır. Hücre c bunun tersidir. İkisi de ihanet ederse (hücre d) ikisi de −2 alır. Temel soru şudur: her mahkûm kendi cezasını en aza indirmek ister-

se ne olur?

2.4.3 İhanet neden baskın?

Sam, John'un ne yapacağını bilmez. Bu yüzden iki durumu ayrı ayrı inceler. Bu yönteme **durum analizi** denir: rakibin her olası seçimi için en iyi cevap bulunur.

Örnek 2.14

1. **Durum 1: John sessiz kalırsa.** Sam sessiz kalırsa −1 (hücre a), ihanet ederse 0 (hücre c) alır. 0 > −1 olduğundan ihanet daha iyidir.
2. **Durum 2: John ihanet ederse.** Sam sessiz kalırsa −3 (hücre b), ihanet ederse −2 (hücre d) alır. −2 > −3 olduğundan ihanet yine daha iyidir.
3. John ne yaparsa yapsın ihanet Sam'e daha az ceza getirir. İhanet Sam için **baskın stratejidir**.
4. Oyun simetriktir; John da aynı ikilemi yaşar ve onun için de ihanet baskındır.
5. İkisi de ihanet eder. Sonuç hücre d: (−2, −2), yani ikişer yıl hapis.

Tanım 2.10

Oyuncu i'nin bir stratejisi s_i, ötekilerin her olası seçimine karşı başka herhangi bir stratejiden kesin daha çok kazandırıyorsa **(kesin) baskın strateji**dir (*strictly dominant strategy*): her $s_i' \neq s_i$ ve her s_{-i} için $u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$.

Baskın strateji varsa oyuncu ötekinin ne yapacağını tahmin etmek zorunda kalmaz. Sam'in John'un olasılıklarını bilmesi gerekmez; her durumda aynı cevap en iyidir. Bu, baskın stratejinin en güçlü yanıdır: rakip hakkında hiçbir inanç gerektirmez.

Sık yapılan hata

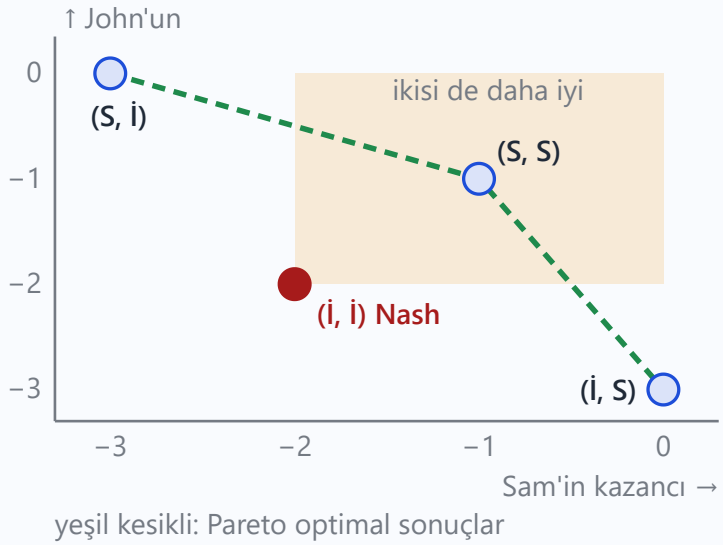
- **"İhanet ortalama cezayı azalttığı için seçilir."** Bu gerekçe yanlıştır. İhanet, John'un *her* seçimine karşı daha iyidir; ortalama almaya ve John'un olasılıklarını bilmeye gerek yoktur.
- **"İtiraf ikisi için en iyi sonuçtur."** Değildir. İkisi de sessiz kalsaydı (hücre a) (−1, −1) alacaklardı; bu, (−2, −2)'den ikisi için de daha iyidir. Bireysel en iyi seçim ikisini birden kötüleştirir.
- **Ceza ile kazancı karıştırmak.** Hapis yılları doğrudan kazanç gibi okunursa "büyük olanı seçmek" 3 yılı seçmek olur. Eksi işaret konmadan hesap yapılmamalıdır.

2.4.4 İkilemin mesajı

Oyunun tek Nash dengesi (İhanet, İhanet) = (−2, −2)'dir. Bunu denetlemek kolaydır: d hücresinden Sam Sessiz'e geçerse −3 alır, John Sessiz'e geçerse −3 alır; ikisi de sapmaz. Öteki üç hücrede en az bir oyuncu ihanete geçerek kazanır. Buna karşılık (Sessiz, Sessiz) = (−1, −1) ikisi için de daha iyidir. Demek ki denge, ikisini birden daha iyi yapan bir sonuçtan kötüdür; 2.4.5'te tanımlanacak dille, denge **Pareto optimal değildir**.

Mahkûmlar sorgudan önce anlaşsa bile durum değişmez. Oyun işbirlikçi olmayan bir oyundur: anlaşma bağlayıcı değildir. "İkimiz de sessiz kalacağız" sözünden sonra John'un sessiz kalacağına inanan Sam, ihanet ederek 0 alır (−1 yerine). İkisi için de sözü bozmak kazançlıdır. Söz kazançları değiştirmediği için baskın stratejiyi de değiştirmez.

Oyunun mesajı şudur: herkesin kendi çıkarına rasyonel davranması, toplum için iyi sonucu garanti etmez. Bu, "görünmez el" fikrinin, yani bireysel çıkar peşindeki davranışların kendiliğinden toplumsal iyiliğe yol açtığı fikrinin her zaman geçerli olmadığını gösterir.



Şekil 2.14: Mahkûmun İkilemi'nin dört sonucu kazanç uzayında. Yatay eksen Sam'in, dikey eksen John'un kazancıdır. Nash dengesi (İ, İ) kırmızıdır; turuncu bölgedeki (S, S) ikisine de daha iyidir. Yeşil kesikli çizgi Pareto optimal sonuçları birleştirir.

Mahkûmun İkilemi yapısı gündelik hayatta sık görülür:

- **Fiyat savaşları:** her şirket fiyat kırar, sonunda hepsinin kârı düşer (Tablo 2.4).
- **Silahlanma yarışı:** her ülke silahlanır, kimse daha güvende olmaz, hepsi daha yoksul olur.
- **İklim anlaşmaları:** her ülke emisyonu azaltmanın maliyetinden kaçınmak ister.
- **Sporda doping:** herkes doping yaparsa göreli sıra değişmez, ama herkesin sağlığı zarar görür.
- **Ortak kaynakların aşırı kullanımı:** balıkçılar stokun tükeneceğini bilse de daha çok avlanır.

İkilemden çıkış yolları da vardır. Birincisi **tekrarlanan oyun ve itibar**dır: aynı oyuncular tekrar tekrar karşılaşırsa bugünkü ihanet yarın cezalandırılabilir. Kısasa kısas (*tit-for-tat*) stratejisi ilk turda işbirliği yapar, sonra rakibin bir önceki hamlesini kopyalar. Sonsuz tekrarda işbirliğinin nasıl sürdüğü 2.8.8'de hesaplanır. İkincisi **sözleşme ve teşvik tasarımı**dır: kazançlar değiştirilirse, örneğin ihanete bir ceza eklenirse, ihanet baskın olmaktan çıkar. Oyunun kurallarını tasarlamak mekanizma tasarımının konusudur (2.10.3).

2.4.5 Pareto optimallik

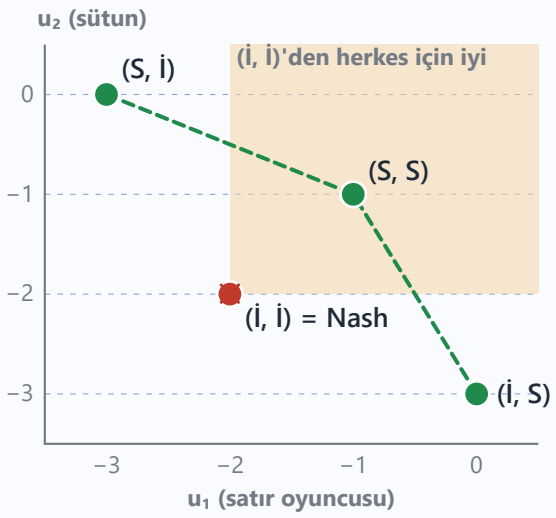
Nash dengesi bir *kararlılık* kavramıdır. Bir sonucun *verimli* olup olmadığını ölçmek için ayrı bir kavram gerekir. Bunun en zayıf ve en yaygın ölçütü Pareto optimalliktir.

Tanım 2.11

Bir x sonucu, bir y sonucunu **Pareto baskılar** (*Pareto dominates*) ancak ve ancak her oyuncu için $u(x) \geq u(y)$ ve en az bir oyuncu için $u(x) > u(y)$ ise.

Hiçbir sonucun Pareto baskılamadığı sonuç **Pareto optimal**dir (*Pareto optimal*): kimseyi kötüleştirmeden kimse iyileştirilemez.

Kazanç uzayında, yani her sonucun (u_1, u_2) noktası olarak çizildiği düzlemde, tanımın kolay bir görsel karşılığı vardır. Bir noktanın sağ üst çeyreği, iki oyuncuya da en az o kadar veren noktaların bölgesidir. Bu bölgede (noktanın kendisi dışında) başka bir sonuç yoksa nokta Pareto optimaldir.



Şekil 2.15: Pareto optimallik kazanç uzayında. (İ, İ) = (−2, −2) noktasının sağ üst çeyreğinde (turuncu) (S, S) vardır; Nash dengesi Pareto baskılanır. Yeşil üç nokta (S, İ), (S, S), (İ, S) Pareto optimaldir.

Örnek 2.15

Mahkûmun İkilemi'nin dört sonucu (S: Sessiz, İ: İhanet) tek tek denetlenir:

1. (İ, İ) = (−2, −2): (S, S) = (−1, −1) ikisine de daha çok verir (−1 > −2). Pareto baskılanır.
2. (S, S) = (−1, −1): onu baskılamak için iki oyuncuya da en az −1 veren ve birine daha fazlasını veren bir sonuç gerekir. (S, İ) = (−3, 0) Sam'i kötüleştirir, (İ, S) = (0, −3) John'u kötüleştirir. Pareto optimaldir.
3. (S, İ) = (−3, 0): John'a 0'dan fazlasını veren sonuç yoktur; 0 veren tek sonuç budur. Kimse onu baskılayamaz. Pareto optimaldir.
4. (İ, S) = (0, −3): aynı akıl yürütmeyle Sam'e 0 veren tek sonuçtur. Pareto optimaldir.

Sonuç: (S, S), (S, İ) ve (İ, S) Pareto optimaldir; tek Nash dengesi olan (İ, İ) ise Pareto baskılanır. Oyunun dengesi, dört sonuç içinde verimsiz olan tek sonuçtur.

Pareto optimallik adalet demek değildir. (S, İ) Pareto optimaldir, ama Sam 3 yıl yatarken John serbest kalır. Pareto optimallik yalnızca "boşa harcanan bir şey yok" der; kazancın nasıl dağıldığına bakmaz. Bir oyuncunun her şeyi aldığı, ötekinin hiçbir şey almadığı bir sonuç da Pareto optimal olabilir.

Denge (kararlılık) ile verimlilik ayrı sorulardır. Mahkûmun İkilemi bu iki sorunun farklı cevaplar verebileceğini gösterir: kararlı olan verimli değildir, verimli olan kararlı değildir. Aradaki kayıp, **Anarşinin Bedeli** (*Price of Anarchy*) ile ölçülür. Hapis yılları maliyet olarak sayılırsa dengede toplam maliyet 2 + 2 = 4 yıl, en iyi sonuçta (S, S) 1 + 1 = 2 yıldır; oran 4 / 2 = 2'dir. Bencil davranış, toplam maliyeti iki katına çıkarır.

2.5 Klasik 2×2 oyunlar

İki oyuncunun ikişer stratejisi olan oyunlara **2×2 oyunlar** denir. Matrisleri yalnızca dört hücreden oluşur, ama bu küçük oyunlar çok farklı stratejik durumları temsil eder: tam çatışma, güven sorunu, buluşma sorunu ve cesaret yarışı. Altı klasik oyun bu durumların hepsini kapsar.

2.5.1 Altı oyun ve saf dengeleri

Tablo 2.9 altı oyunu tek bir biçimde yazar. Her satırda bir oyun vardır. (1, 2) sütunu, satır oyuncusunun 1. eylemini, sütun oyuncusunun 2. eylemini oynadığı hücreyi gösterir; hücrede önce satır oyuncusunun, sonra sütun oyuncusunun kazancı yazılır.

Oyun	Eylemler (1 / 2)	(1, 1)	(1, 2)	(2, 1)	(2, 2)	Saf Nash dengeleri
Yazı-Tura Eşleştirme	Yazı / Tura	(1, −1)	(−1, 1)	(−1, 1)	(1, −1)	yok

Oyun	Eylemler (1 / 2)	(1, 1)	(1, 2)	(2, 1)	(2, 2)	Saf Nash dengeleri
Tavuk	Kır / Düz git	(0, 0)	(−1, 1)	(1, −1)	(−10, −10)	(Kır, Düz git), (Düz git, Kır)
Cinsiyetler Savaşı	Opera / Futbol	(3, 2)	(0, 0)	(0, 0)	(2, 3)	(Opera, Opera), (Futbol, Futbol)
Geyik Avı	Geyik / Tavşan	(4, 4)	(0, 3)	(3, 0)	(3, 3)	(Geyik, Geyik), (Tavşan, Tavşan)
Saf Koordinasyon	Parti / Ev	(10, 10)	(0, 0)	(0, 0)	(5, 5)	(Parti, Parti), (Ev, Ev)
Bedava Para	Yazı / Tura	(1, 1)	(0, 0)	(0, 0)	(0, 0)	(Yazı, Yazı) kesin; (Tura, Tura) zayıf

Tablo 2.9: Altı klasik 2×2 oyun ve saf Nash dengeleri.

Saf dengeleri bulmanın düzenli bir yolu vardır: **en iyi tepki işaretleme**. Her sütunda satır oyuncusunun en büyük kazancı işaretlenir; bu, satır oyuncusunun o sütuna en iyi tepkisidir. Her satırda sütun oyuncusunun en büyük kazancı işaretlenir; bu da sütun oyuncusunun o satıra en iyi tepkisidir. İki işareti birden taşıyan hücre, iki oyuncunun da birbirine en iyi tepki verdiği hücredir, yani bir saf Nash dengesidir. Eşitlik varsa eşit en büyüklerin hepsi işaretlenir.

Örnek 2.16

Tavuk oyununda iki sürücü dar bir yolda birbirine doğru gelir. Her biri ya kenara kırar (Kır) ya da düz gider (Düz git). İşaretleme şöyle yapılır:

1. Sütun “Kır” (sütun oyuncusu kırıyor): satır oyuncusu Kır ile 0, Düz git ile 1 alır. En iyisi Düz git (1 > 0); (Düz git, Kır) hücresine satır işareti.
2. Sütun “Düz git”: satır oyuncusu Kır ile −1, Düz git ile −10 alır. En iyisi Kır (−1 > −10); (Kır, Düz git) hücresine satır işareti.
3. Satır “Kır”: sütun oyuncusu Kır ile 0, Düz git ile 1 alır. En iyisi Düz git; (Kır, Düz git) hücresine sütun işareti.
4. Satır “Düz git”: sütun oyuncusu Kır ile −1, Düz git ile −10 alır. En iyisi Kır; (Düz git, Kır) hücresine sütun işareti.

1 \ 2	Kır	Düz git
Kır	(0, 0)	**(−1*, 1*)**
Düz git	**(1*, −1*)**	(−10, −10)

Tablo 2.10: Tavuk oyununda işaretleme. Yeşil hücreler iki işaretli, yani saf Nash dengesidir.

İki işaretli hücreler (Kır, Düz git) ve (Düz git, Kır)'dır. Oyunun iki saf dengesi vardır ve ikisinde de sürücülerden biri kırar, öteki düz gider. (Düz git, Düz git) en kötü sonuçtur: çarpışma.

Örnek 2.17

Yazı-Tura Eşleştirme'de işaretleme hiçbir hücrede iki işaret bırakmaz:

1. Sütun Yazı: satır Yazı ile 1, Tura ile −1 alır → (Yazı, Yazı)'ya satır işareti.
2. Sütun Tura: satır Yazı ile −1, Tura ile 1 alır → (Tura, Tura)'ya satır işareti.
3. Satır Yazı: sütun Yazı ile −1, Tura ile 1 alır → (Yazı, Tura)'ya sütun işareti.
4. Satır Tura: sütun Yazı ile 1, Tura ile −1 alır → (Tura, Yazı)'ya sütun işareti.

Satır işaretleri köşegende, sütun işaretleri köşegen dışındadır; hiçbir hücre iki işaret taşımaz. Saf Nash dengesi yoktur. Her hücrede kaybeden oyuncu sapmak ister ve sapma okları bir döngü çizer.

Örnek 2.18

Bedava Para oyununda iki oyuncu da Yazı derse birer birim kazanır; öteki üç hücrede kimse bir şey kazanmaz. İşaretleme eşitliklere dikkat ister:

1. Sütun Yazı: satır Yazı ile 1, Tura ile 0 → Yazı işaretlenir.
2. Sütun Tura: satır Yazı ile 0, Tura ile 0 → eşitlik; *ikisi de* işaretlenir.
3. Satır Yazı: sütun Yazı ile 1, Tura ile 0 → Yazı işaretlenir.
4. Satır Tura: sütun Yazı ile 0, Tura ile 0 → ikisi de işaretlenir.

İki işaretli hücreler (Yazı, Yazı) ve (Tura, Tura)'dır. İki denge farklı türdendir. (Yazı, Yazı) **kesin dengedir**: sapan oyuncu 1'den 0'a düşer, yani kaybeder. (Tura, Tura) **zayıf dengedir**: sapan oyuncu 0'dan 0'a geçer; sapmak kazandırmaz ama kaybettirmez de. Yazı, Tura'yı **zayıf baskılar**: hiçbir durumda daha kötü değildir ve en az bir durumda daha iyidir.

Geri kalan üç oyunun işaretlemesi aynı yolla yapılır. Cinsiyetler Savaşı'nda (Opera, Opera) = (3, 2) ve (Futbol, Futbol) = (2, 3) iki işaret taşır; köşegen dışı (0, 0) hücrelerinde ikisi de buluşma noktasına geçmek ister. Geyik Avı'nda (Geyik, Geyik) = (4, 4) ve (Tavşan, Tavşan) = (3, 3) dengedir: ötekinin Geyik avladığını bilen biri Geyik'ten 4, Tavşan'dan 3 alır; ötekinin Tavşan avladığını bilen biri Geyik'ten 0, Tavşan'dan 3 alır. Saf Koordinasyon'da (Parti, Parti) ve (Ev, Ev) dengedir.

Sık yapılan hata

- **Tek işaret yeterli sanmak.** Bir hücrenin denge olması için *iki* oyuncunun da en iyi tepkisi olması gerekir. Yalnız satır işareti taşıyan hücrede sütun oyuncusu sapar.
- **Yanlış yönde aramak.** Satır oyuncusunun en iyi tepkisi *sütun içinde* (yukarı-aşağı) aranır, çünkü satır oyuncusu yalnız satırını değiştirebilir. Sütun oyuncusununki *satır içinde* (sağa-sola) aranır.

- **Yanlış sayıya bakmak.** Satır oyuncusu için her hücrenin ilk sayısı, sütun oyuncusu için ikinci sayısı karşılaştırılır.
- **Eşitlikte birini seçmek.** Eşit en büyüklerin hepsi işaretlenir; yoksa Bedava Para'daki zayıf denge gözden kaçar.

2.5.2 Çatışma mı, koordinasyon mu?

Altı oyun, iki oyuncunun çıkarlarının ne kadar örtüştüğüne göre bir yelpazeye dizilir. Bir uçta çıkarların tamamen zıt olduğu Yazı-Tura, öbür uçta hiç çatışma olmayan Saf Koordinasyon ve Bedava Para vardır.

Yazı-Tura Eşleştirme: tam çatışma. Çıkarlar tamamen zıttır ve oyun sıfır toplamlıdır. Her hücrede kaybeden sapar. Bir oyuncunun kestirilebilen her eğilimi sömürülür: eşleştiren biraz daha sık Yazı derse ayrıştıran Tura'ya kayar. Tek çıkış rastgele oynamaktır, yani karışık strateji (½–½). Penaltı atışı bu yapının gerçek hayattaki örneğidir: atıcı ve kaleci birbirini tahmin etmeye çalışır.

Tavuk: anti-koordinasyon. Oyuncular *farklı* eylemler seçmek ister: biri kırmalı, öteki düz gitmelidir. İki denge vardır ve her biri bir sürücüyü kayırır. Kim kazanır? Geri adım atamayacağını inandırıcı biçimde gösteren. Örneğin direksiyonunu pencereden atan bir sürücü, Düz git'ten başka seçeneği olmadığını gösterir; öteki sürücünün en iyi tepkisi kırmak olur. Bu tür bir davranışa **taahhüt** (*commitment*) denir. Taahhüt, seçenekleri azaltarak kazandırır; tek kişilik karar probleminde seçenek azalması hiç kazandırmazken oyunda kazandırabilir.

Cinsiyetler Savaşı: koordinasyon ve çatışma birlikte. İki kişi akşamı birlikte geçirmek ister; buluşmak ortak çıkardır. Ama biri Opera'yı (3 > 2), öteki Futbol'u (3 > 2) tercih eder; nerede buluşulacağı çatışmadır. Baskın strateji yoktur. Çözüm iletişim (“bu akşam operaya gidiyorum”), gelenek ya da sırayla seçmektir (bir hafta opera, bir hafta futbol).

Geyik Avı: güven ikilemi. İki avcı birlikte geyik avlarsa büyük ödül alır (4, 4). Biri tek başına tavşan avlarsa 3 alır; geyik peşindeki öteki eli boş kalır (0). (Geyik, Geyik) hem dengedir hem de Pareto optimaldir. Ama Tavşan her durumda 3 garanti eder: öteki ne yaparsa yapsın tavşan avcısı 3 alır. Geyik ise ötekine güvenmeyi gerektirir. Mahkûmun İkilemi'nden farkı şudur: Geyik Avı'nda işbirliği bir dengedir. Kimse işbirliğinden tek başına sapmak istemez; sorun sapma isteği değil, ötekinin sapacağı korkusudur (Skyrms, 2004).

Saf Koordinasyon: kötü denge de kalıcıdır. Çıkar çatışması yoktur: iki oyuncu da Parti'de buluşmayı tercih eder (10 > 5). Ama (Ev, Ev) de bir dengedir: ötekinin evde kalacağını bilen biri partiye gitmekle 0 alır, evde kalmakla 5. Kötü denge kendini sürdürür. Oyuncular çoğu zaman belirgin, “herkesin aklına gelecek” olana yönelir. Thomas Schelling bu tür seçimlere **odak noktası** (*focal point*) adını verdi (Schelling, 1960). Örneğin bir şehirde buluşma yeri söylenmemiş iki kişi, çoğu zaman şehrin en bilinen meydanına gider.

Bedava Para: zayıf denge. (Yazı, Yazı) kesin dengedir: sapan kaybeder. (Tura, Tura) zayıf dengedir: sapmak kazandırmaz, kaybettirmez de. Yazı zayıf baskındır; makul bir oyuncunun Tura'yı seçmesi için bir neden yoktur. Bu oyun, Nash dengesi tanımının "≥" işaretinin bazen akla yatkın olmayan dengeleri de içeri aldığını gösterir.

Örnek 2.19

Altı oyunun hangilerinin sıfır toplamlı, hangilerinin simetrik olduğu tanımlarla denetlenir:

1. **Sıfır toplamlı:** her hücrede toplam 0 olmalıdır. Yazı-Tura'da bütün toplamlar 0'dır. Tavuk'ta (−10, −10) toplamı −20'dir; ötekilerde de toplamı sıfırdan farklı hücreler vardır (ör. Geyik Avı'nda (4, 4)). Yalnızca Yazı-Tura sıfır toplamlıdır.
2. **Simetrik:** $u_1(a, b) = u_2(b, a)$ olmalıdır. Tavuk'ta $u_1(Kır, Düz) = -1 = u_2(Düz, Kır)$, $u_1(Düz, Kır) = 1 = u_2(Kır, Düz)$ ve köşegenler eşittir: simetriktir. Geyik Avı, Saf Koordinasyon ve Bedava Para da aynı denetimi geçer.
3. **Simetrik olmayanlar:** Cinsiyetler Savaşı'nda $u_1(Opera, Opera) = 3 \neq 2 = u_2(Opera, Opera)$. Yazı-Tura'da $u_1(Yazı, Yazı) = 1 \neq -1 = u_2(Yazı, Yazı)$; rollerden biri eşleştirmek, öteki ayrıştırmak ister.

Sonuç: sıfır toplamlı yalnızca Yazı-Tura'dır; simetrik olanlar Tavuk, Geyik Avı, Saf Koordinasyon ve Bedava Para'dır.

Oyun	Yapı	Saf denge sayısı	Temel sorun
Yazı-Tura	Tam çatışma	0	Tahmin edilemez olmak
Tavuk	Anti-koordinasyon	2	Kim geri adım atacak? (taahhüt)
Cinsiyetler Savaşı	Koordinasyon + çatışma	2	Nerede buluşulacak?
Geyik Avı	Güven	2	Ötekine güvenilecek mi?
Saf Koordinasyon	Saf ortak çıkar	2	Kötü dengeden kaçmak (odak noktası)
Bedava Para	Ortak çıkar, zayıf denge	2	Zayıf baskın stratejiyi seçmek

Tablo 2.11: Altı oyunun yapısı bir bakışta.

2.6 2/3 ortalaması oyunu

Bölümün başındaki oyun (2.1.1) artık çözülebilir. Oyunun iki "cevabı" vardır. Biri kuramın cevabıdır: herkes tam rasyonel ve bu ortak bilgiyse ne olur? Öteki pratik cevaptır: gerçek bir sınıfta en iyi tahmin nedir? İki cevap birbirinden farklıdır.

2.6.1 Seviye-k düşünme

Gerçek oyuncular sonsuz adım düşünmez; birkaç adımda dururlar. Bu davranışı modellemenin bir yolu **seviye-k düşünme**dir (*level-k reasoning*). Model, oyuncuları kaç adım düşündüklerine göre sınıflandırır.

Tanım 2.12

Seviye 0 oyuncu stratejik düşünmez ve rastgele bir sayı yazar; 0–90 aralığında bu tahminlerin ortalaması 45'tir. **Seviye k** oyuncu ($k \geq 1$), diğer herkesin seviye $k-1$ olduğunu varsayar ve bu varsayıma en iyi tepkiyi verir: seviye $k-1$'in tahmininin 2/3'ünü yazar.

Düşünme seviyesi	Varsayım	Tahmin
Seviye 0	Herkes rastgele yazar, ortalama 45	— (rastgele)
Seviye 1	Diğerleri seviye 0	$2/3 \cdot 45 = 30$
Seviye 2	Diğerleri seviye 1	$2/3 \cdot 30 = 20$
Seviye 3	Diğerleri seviye 2	$2/3 \cdot 20 \approx 13{,}3$
Seviye ∞	Herkes tam rasyonel	0

Tablo 2.12: Seviye-k düşünmede tahminler.

“Ortalama 45 olur, 2/3'ü 30” akıl yürütmesi yalnızca seviye 1'dir. Başkalarının da 30 diyeceğini düşünen biri daha düşük bir sayı, örneğin 20 yazar. Herkesin fazla düşük yazacağını düşünen ise biraz yukarı kayabilir. Seviye k'nın tahmini genel olarak $45 \cdot (2/3)^k$'dır ve k büyüdükçe 0'a yaklaşır.

Önemli nokta şudur: *en iyi tahmin, ötekilerin kaçıncı seviyede düşündüğüne dair inanca bağlıdır*. Sınıfın çoğu seviye 1 ise en iyi cevap seviye 2'dir; daha fazla düşünmek kaybettirir. Seviye kavramı, ortak bilgideki “herkes biliyor” katlarını sayan bilgi düzeyiyle (Şekil 2.2) karıştırılmamalıdır. Seviye, bir oyuncunun kaç adım düşündüğüdür; bilgi düzeyi ise oyuncular hakkında neyin bilindiğidir.

2.6.2 Kod izleme: seviye-k tahmini

Aşağıdaki Python kodu, sınıftaki seviyelerin dağılımına dair bir inanç verildiğinde her seviyenin hedefe ne kadar uzak kalacağını hesaplar.

```
def tahmin(k, L0=45):     # seviye-k
    g = L0
    for _ in range(k):
        g = g * 2 / 3
    return g
pay = {0: 0.3, 1: 0.4, 2: 0.3}   # inanç
ort = sum(p * tahmin(k) for k, p in pay.items())
hedef = ort * 2 / 3
for k in range(5):
    print(k, round(tahmin(k), 1),
          round(abs(tahmin(k) - hedef), 1))
```

Kod şöyle çalışır:

- `tahmin(k, L0=45)` seviye-k oyuncunun tahminini döndürür. `g`, 45 ile başlar ve döngü `k` kez çalışarak her seferinde `g`'yi 2/3 ile çarpar. `k = 0` için döngü hiç çalışmaz ve 45 döner: seviye 0'ın ortalaması.
- `pay` sözlüğü inancı tutar: sınıfın %30'u seviye 0, %40'ı seviye 1, %30'u seviye 2.
- `ort`, beklenen ortalamadır: her seviyenin tahmini o seviyenin payıyla çarpılıp toplanır.
- `hedef`, ortalamanın 2/3'üdür; kazanan sayı budur.
- Son döngü seviye 0'dan 4'e kadar her seviyenin tahminini ve hedefe uzaklığını yazdırır.

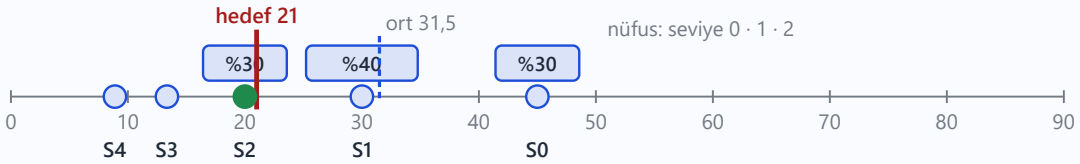
Örnek 2.20

1. Seviye tahminleri: `tahmin(0) = 45`, `tahmin(1) = 30`, `tahmin(2) = 20`.
2. Beklenen ortalama: $0{,}3 \cdot 45 + 0{,}4 \cdot 30 + 0{,}3 \cdot 20 = 13{,}5 + 12 + 6 = 31{,}5$.
3. Hedef: $2/3 \cdot 31{,}5 = 21$.
4. Her seviyenin uzaklığı Tablo 2.13'te listelenir.

k	tahmin(k)	|tahmin – hedef|	Yorum
0	45	24	ortalamayı yazmak çok yüksek
1	30	9	klasik "45 → 30" cevabı, hâlâ uzak
2	**20**	**1**	**hedefe en yakın: en iyi tepki**
3	13,3	7,7	bir seviye fazla düşünmek kaybettirir
4	8,9	12,1	0'a yaklaştıkça uzaklık artar

Tablo 2.13: Kodun çıktısı (hedef = 21).

Bu nüfusa karşı 20 yazan kazanır. Nash dengesi cevabı olan 0 yazan ise 21 birim uzakta kalır ve kaybeder. En iyi tepki inanca bağlıdır.



Şekil 2.16: Seviye-k tahminleri (S0–S4) ve hedef. Nüfus %30 seviye 0, %40 seviye 1, %30 seviye 2'dir; beklenen ortalama 31,5, hedef 21'dir. Hedefe en yakın tahmin seviye 2'nin (20) tahminidir.

Şekil 2.16 aynı sonucu sayı doğrusu üzerinde gösterir. Hedef (21), seviye 2'nin tahminine (20) çok yakındır. Seviye 3 ve 4 hedefin solunda, seviye 0 ve 1 sağında kalır. Hedefin yeri nüfusa bağlıdır: nüfusun daha büyük kısmı seviye 0 olsaydı hedef sağa kayardı.

2.6.3 Yinelemeli eleme

Kuramın cevabı farklı bir yöntemle bulunur. Bu yöntem, hiçbir durumda iyi olamayacak tahminleri adım adım siler.

Tanım 2.13

Bir strateji, başka bir strateji ötekilerin her seçimine karşı daha iyi sonuç veriyorsa **baskılanır** (*dominated*). Rasyonel oyuncu baskılanan stratejiyi oynamaz. **Yinelemeli baskın strateji eleme** (*iterated elimination of dominated strategies*), baskılanan stratejileri silip küçülen oyunda yeniden baskılananları aramaktır; yeni bir silme yapılamayana kadar sürer.

2/3 oyununda eleme şöyle işler. Başlangıçta her tahmin 0 ile 90 arasındadır ve hiçbir şey elenmemiştir.

1. **Tur 1.** Ortalama en fazla 90 olabilir, bu yüzden hedef en fazla 60'tır. 60'ın üstündeki her tahmin, hiçbir durumda 60 yazmaktan daha iyi değildir: hedef 60'ın altındaysa 60 ona daha yakındır. Bu tahminler elenir.
2. **Tur 2.** Herkes rasyonel ve bunu herkes biliyor: kimse 60'ı aşmaz. Ortalama ≤ 60 ⇒ hedef ≤ 40. 40'ın üstü elenir.
3. **Tur 3.** Ortalama ≤ 40 ⇒ hedef ≤ 26,7. Bu adımı atmak için "herkesin, herkesin rasyonel olduğunu bildiğini" bilmek gerekir.
4. **Tur 4–6.** Ortalama ≤ 26,7 ⇒ hedef ≤ 17,8; ortalama ≤ 17,8 ⇒ hedef ≤ 11,9; ortalama ≤ 11,9 ⇒ hedef ≤ 7,9.

Her turda üst sınır 2/3 ile çarpılır; k tur sonra üst sınır $90 \cdot (2/3)^k$'dır. Aşağıdaki kod bu çarpımı altı tur tekrarlar.

```
ust = 90                # olası en büyük tahmin
for tur in range(1, 7):
    # ortalama ≤ ust  ⇒  hedef ≤ (2/3)·ust
    ust = ust * 2 / 3
    print(tur, round(ust, 1))
# ust = 90·(2/3)^k → 0 : tek denge herkes 0
```

tur	Önceki ust	Yeni ust = ust · 2/3	Anlamı
1	90	60	herkes rasyonel
2	60	40	+ herkes bunu biliyor
3	40	26,7	+ bunu bildiklerini biliyorlar
4	26,7	17,8	bir kat daha
5	17,8	11,9	bir kat daha
6	11,9	7,9	bir kat daha

Tablo 2.14: Eleme kodunun izlemesi.

Kodda `ust` değişkeni, "anlamlı en büyük tahmin"i tutar. Döngünün her turunda yorum satırındaki çıkarım uygulanır: ortalama `ust`'u aşamıyorsa hedef `(2/3)·ust`'u aşamaz. Yeni `ust` bu değerdir. Döngü 6 turda durur, ama süreç kuramda sonsuza dek sürer.



Şekil 2.17: Yinelemeli elemenin sonu. Her turda üst sınır 2/3 ile çarpılır ve 0'a yakınsar; elemeden sağ çıkan tek tahmin 0'dır.

Üst sınır geometrik olarak küçülür: her tur bir öncekinin 2/3'ü kadardır. Kaç tur sonra sınır 1'in altına iner? $90 \cdot (2/3)^k < 1$ eşitsizliği $k = 12$ için ilk kez sağlanır ($90 \cdot (2/3)^{12} \approx 0{,}70$; $k = 11$ için ≈ 1,04). Ama hiçbir sonlu turda tam 0'a ulaşılmaz. Süreç sonsuza dek sürer ve üst sınır 0'a yakınsar: yinelemeli elemeden sağ çıkan tek profil, herkesin 0 söylemesidir (Şekil 2.17).

2.6.4 Tek Nash dengesi: herkes 0

Yinelemeli eleme, herkesin 0 söylediği profili bırakır. Bu profilin bir Nash dengesi olduğu ayrıca denetlenir.

Örnek 2.21

n oyuncunun hepsi 0 yazsın. Pişmanlık testi uygulanır:

1. Herkes 0 derse ortalama 0, hedef 0'dır. Herkes hedefe tam isabet eder ve ödül paylaşılır.
2. Bir oyuncu $x > 0$'a sapsın. Toplam x olur, ortalama x/n, hedef $2x/(3n)$'dir.
3. Ötekilerin 0'ı hedefe $2x/(3n)$ uzaklıktadır. Sapan oyuncunun x'i hedefe $x - 2x/(3n) = x(3n - 2)/(3n)$ uzaklıktadır.
4. $n \geq 2$ için $3n - 2 > 2$ olduğundan sapanın uzaklığı daha büyüktür. Örneğin $n = 10$, $x = 30$ için hedef 2'dir; 0 hedefe 2, 30 ise 28 uzaktadır.
5. Sapan oyuncu kaybeder; paydan da mahrum kalır. Sapma kârsızdır, profil bir Nash dengesidir.

Başka hiçbir profil yinelemeli elemeden sağ çıkmaz; bu yüzden denge tektir. Bunun nedeni, kesin baskılanan stratejileri silmenin hiçbir Nash dengesini silmemesidir: bir dengede herkes en iyi tepkisini oynar, oysa baskılanan bir strateji hiçbir durumda en iyi tepki olamaz. Dolayısıyla her denge elemeden sağ çıkar; sağ çıkan tek profil herkesin 0 söylemesi olduğuna göre denge de tektir.

Bir ayrıntı, oyunun kurallarının dengeyi nasıl etkilediğini gösterir. Yalnız tam sayılara izin verilseydi $n \geq 4$ oyuncuda "herkes 1" de bir denge olurdu. Ondalıklı tahmine izin vermek bu ikinci dengeyi ortadan kaldırır.

Örnek 2.22

Yalnız tam sayılar serbest, n oyuncunun hepsi 1 yazıyor. Ortalama 1, hedef 2/3'tür ve herkes paylaşır (her biri $1/n$ ödül alır).

1. Bir oyuncu 0'a sapsın. Ortalama $(n - 1)/n$, hedef $2(n - 1)/(3n)$ olur.
2. Sapanın uzaklığı $2(n - 1)/(3n)$; ötekilerin uzaklığı $1 - 2(n - 1)/(3n) = (n + 2)/(3n)$'dir.
3. Sapan tek başına kazanır ancak $2(n - 1) < n + 2$, yani $n < 4$ ise. $n = 4$'te iki uzaklık eşittir (6/12 = 6/12) ve ödül yine dörde bölünür; sapma kazancı artırmaz. $n > 4$'te sapan kaybeder.
4. Bir oyuncu 2'ye sapsın. Ortalama $(n + 1)/n$, hedef $2(n + 1)/(3n)$ olur; bu sayı $n \geq 2$ için en fazla 1'dir. 1, hedefe 2'den daha yakındır; sapan kaybeder.

Demek ki $n \geq 4$ iken "herkes 1" kârlı bir sapmaya izin vermez ve bir dengedir. Ondalıklı tahminlerde ise "herkes 1" durumunda bir oyuncu 0,7 gibi bir sayıya saparak hedefe daha yakın olur (10 oyuncuda hedef yaklaşık 0,65 olur; 0,7 ona 0,05, 1 ise 0,35 uzaktadır) ve tek başına kazanır; denge bozulur.

2.6.5 Kuram ve deney

Kuramın cevabı 0'dır. Ama deneylerde kazanan sayı 0 çıkmaz. 1997'de *Financial Times* okurlarıyla yapılan 0–100 sürümünde kazanan tahmin 13'tü. Rosemarie Nagel'in 1995'te yayımlanan laboratuvar deneylerinde ilk tur tahminleri 1–2 düşünme adımının çevresinde yığıldı: tahminlerin çoğu seviye 1

ya da seviye 2'nin tahminlerine yakındı (Nagel, 1995). Oyun aynı grupla tekrarlandıkça tahminler 0'a doğru indi.

Kuram: yinelemeli eleme	Anlamlı en büyük tahmin	Gerçek: seviye-k	Tahmin
başlangıç	90	seviye 0 (rastgele, ortalama 45)	45
adım 1	2/3 × 90 = 60	seviye 1 (diğerleri seviye 0)	30
adım 2	2/3 × 60 = 40	seviye 2 (diğerleri seviye 1)	20
adım 3	26,7	seviye 3 (diğerleri seviye 2)	13,3
adım 4	17,8	…	…
…	→ 0	seviye ∞	0

Tablo 2.15: Kuramın cevabı ile gerçekteki düşünme adımları.

İki sütun farklı soruları cevaplar. Sol sütun, ortak bilginin her katının neyi elediğini gösterir: her adım bir kat daha “herkes rasyonel” varsayımı ekler ve sonsuz adımın sonu tek Nash dengesidir. Sağ sütun ise oyuncuların gerçekte ne yazdığını modeller. Kuram ile deney arasındaki fark, “herkes tam rasyonel ve bu ortak bilgi” varsayımının gerçekte tam tutmamasından kaynaklanır. Birkaç kişi bile yüksek sayılar yazarsa ortalama yükselir ve 0 yazan kaybeder.

Bu oyunda Nash dengesini belirlemek zor değildir: denge tek ve saf stratejilidir. Zor olan, gerçek oyuncular karşısında en iyi tepkiyi tahmin etmektir. Kazanmak için 0'ı değil, sınıfın kaç adım düşündüğünü bilmek gerekir. Sınıf seviye 1 oynar ve ortalama 30 çıkarsa hedef 20 olur; seviye 2 kazanır.

Örnek 2.23

Oyun 0–100 aralığında ve hedef “ortalamanın yarısı” olsun. Yinelemeli eleme şöyle ilerler:

1. Ortalama en fazla 100 ⇒ hedef en fazla 50. 50'nin üstü elenir.
2. Ortalama en fazla 50 ⇒ hedef en fazla 25. 25'in üstü elenir.
3. Sonraki sınırlar 12,5; 6,25; 3,125; … Her turda sınır yarıya iner.

Üst sınır 0'a yakınsar ve tek denge yine herkesin 0 söylemesidir. Çarpan 1'den küçük olduğu sürece (2/3, 1/2, …) sonuç değişmez; yalnızca yakınsama hızı değişir. Çarpan küçüldükçe sınır daha hızlı küçülür.

2.7 Kombinatoryal oyunlar

Satranç, XOX ve Nim gibi oyunların ortak bir yapısı vardır: iki oyuncu sırayla oynar, her şey açıktır, zar ya da kart yoktur. Bu oyunlar oyun kuramının en “algoritmik” köşesidir: kuramsal olarak her biri, ağacı sondan başa taranarak çözülebilir. Bu taramanın üç biçimi vardır: minimaks, geriye doğru etiketleme ve Nim'in XOR kuralı.

2.7.1 Kombinatoryal oyun ve Zermelo teoremi

Tanım 2.14

Bir **kombinatoryal oyun** (*combinatorial game*) şu özellikleri taşır: iki oyuncu sırayla oynar; bilgi mükemmeldir; şans yoktur; oyun sonlu sayıda adımda biter.

Normal oyun kuralı (*normal play*): son hamleyi yapan kazanır; hamle yapamayan kaybeder.

Nim normal oyun kuralıyla oynanır: son jetonu alan kazanır. 21'i Söyleme'de ise kural terstir: 21'i söyleyen, yani son hamleyi yapmak zorunda kalan kaybeder. XOX ve satrançta beraberlik de mümkündür. Zermelo teoremi (Teorem 2.1) bu oyunların hepsine uygulanır: böyle bir oyunda ya bir taraf kazanmayı ya da iki taraf beraberliği garanti eder.

Bazı oyunlarda kimin kazandığı, ağaç hiç taranmadan bulunabilir. Hex bunun ünlü örneğidir. Hex'te beraberlik yoktur: tahta dolduğunda oyunculardan biri mutlaka kendi kenarlarını birleştirmiştir. **Strateji çalma argümanı** (*strategy-stealing argument*) şöyle işler. İkinci oyuncunun kazanan bir stratejisi olduğunu varsayalım. İlk oyuncu rastgele bir taş koyar ve sonra kendini ikinci oyuncu yerine koyarak o kazanan stratejiyi oynar. Tahtadaki fazladan taş ona hiçbir zaman zarar vermez; stratejinin istediği kare zaten doluysa yine rastgele bir taş koyar. Böylece ilk oyuncu da kazanır. İki oyuncunun aynı anda kazanması imkânsız olduğu için varsayım yanlıştır: ikinci oyuncunun kazanan stratejisi yoktur. Beraberlik de olmadığı için ilk oyuncu kazanır. Argüman, ilk oyuncunun *nasıl* kazandığını söylemez; yalnızca kazandığını gösterir.

Kombinatoryal oyunlarda her konum iki türden birindedir:

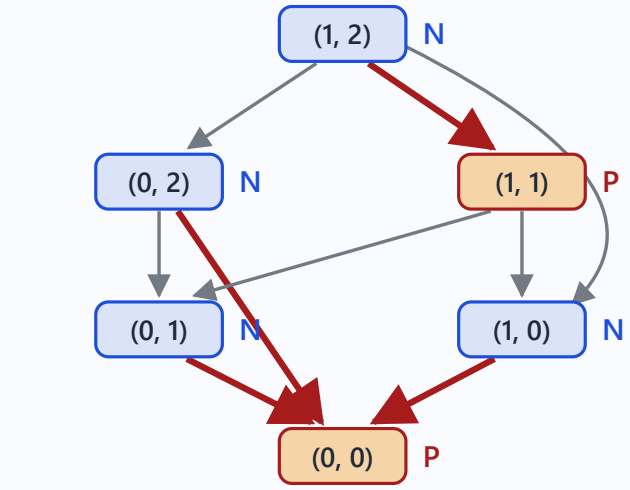
Tanım 2.15

Normal oyun kuralıyla oynanan bir oyunda bir konum, sırası gelen oyuncu kazanmayı garanti edebiliyorsa **N-konumu** (*N-position*, *next*: sıradaki kazanır), garanti edemiyorsa **P-konumu**dur (*P-position*, *previous*: önceki oyuncu kazanır). Etiketler şu iki kuralla belirlenir:

- P-konumundan yapılan *her* hamle bir N-konumuna gider.
- N-konumundan *en az bir* hamle bir P-konumuna gider.

Hamle yapılamayan son konum, normal oyun kuralında P-konumudur: sırası gelen oyuncu hamle yapamaz ve kaybeder.

Kuralların mantığı şudur. Sırası gelen oyuncu rakibini bir P-konumuna gönderebiliyorsa bunu yapar; rakip ne yaparsa yapsın ona tekrar bir N-konumu bırakır ve oyuncu her seferinde rakibi yeniden P'ye gönderir. Sonunda hamle yapamayan rakip olur. Oyuncunun bütün hamleleri N-konumlarına gidiyorsa ne yaparsa yapsın rakibe kazanan bir konum bırakır.



Şekil 2.18: İki yığınlı Nim (1 ve 2 jeton): bütün konumlar ve etiketleri. P (turuncu) sıradakinin kaybettiği, N sıradakinin kazandığı konumdur. Kalın oklar kazandıran hamlelerdir.

Örnek 2.24

İki yığınlı Nim'de yığınlarda 1 ve 2 jeton vardır. Sırası gelen oyuncu tek bir yığından en az bir jeton alır; son jetonu alan kazanır. Konumlar (birinci yığın, ikinci yığın) olarak yazılır ve etiketler yapraktan köke hesaplanır (Şekil 2.18):

1. (0, 0): hamle yok. Sırası gelen kaybeder → **P**.
2. (0, 1) ve (1, 0): tek hamle (0, 0)'a gider, bu P'dir → **N**.
3. (0, 2): (0, 1)'e ya da (0, 0)'a gidilebilir. (0, 0) P'dir → **N**. Kazandıran hamle iki jetonu birden almaktır.
4. (1, 1): hamleler (0, 1) ve (1, 0)'a gider; ikisi de N'dir → **P**.
5. (1, 2): hamleler (0, 2) [N], (1, 1) [P], (1, 0) [N]. (1, 1) P olduğundan → **N**. Kazandıran hamle, ikinci yığından bir jeton alıp (1, 1) bırakmaktır.

Başlangıç konumu (1, 2) N'dir: ilk oyuncu doğru oynarsa kazanır. Etiketleri yapraktan köke hesaplamak, minimaksın kazanma/kaybetme değerleriyle çalışan özel hâlidir.

2.7.2 Kod izleme: oyun ağacında minimaks

Kazanç değerleri yalnızca “kazan / kaybet” değil de sayılarsa aynı fikir **minimaks** algoritmasıyla uygulanır. İki oyuncu MAX ve MIN olarak adlandırılır. MAX büyük değer ister, MIN küçük değer ister. Yapraklardaki sayılar MAX'ın kazancıdır; oyun sıfır toplamlı olduğu için MIN'in kazancı bunun eksi işaretlisidir. Bir düğümün değeri şöyle tanımlanır: yaprakta oyun sonu kazancıdır; MAX düğümünde çocukların değerlerinin en büyüğü, MIN düğümünde en küçüğüdür.

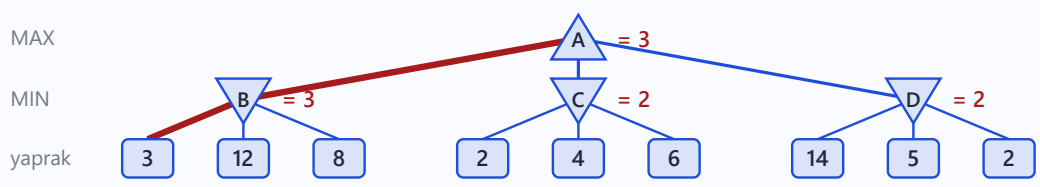
```
T = {"A": ["B", "C", "D"], "B": [3, 12, 8],
     "C": [2, 4, 6], "D": [14, 5, 2]}
def minimax(n, maxi):
  if isinstance(n, int):          # yaprak
    return n
  best = -99 if maxi else 99
  for c in T[n]:
    v = minimax(c, not maxi)
    best = max(best, v) if maxi else min(best, v)
  return best
print(minimax("A", True))
```

Kodun parçaları şunlardır:

- `T` sözlüğü ağacı tutar. Kök A'nın çocukları B, C, D'dir. B, C, D'nin çocukları yapraklardır ve tam sayılarla (`int`) gösterilir.
- `minimax(n, maxi)` düğüm `n` 'nin değerini döndürür. `maxi` doğruysa sıra MAX'tadır.
- `isinstance(n, int)` doğruysa `n` bir yapraktır ve değeri kendisidir; fonksiyon hemen döner.
- `best` , MAX için −99, MIN için 99 ile başlar. Bu sayılar "henüz hiçbir çocuk görülmedi" anlamındadır: MAX'ın ilk gördüğü değer −99'dan büyük, MIN'in ilk gördüğü 99'dan küçük olacaktır.
- Döngü her çocuğu sırayla değerlendirir. Çocuğun sırası öteki oyuncudadır; bu yüzden özyinelemeli çağrı `not maxi` ile yapılır. MAX `max` , MIN `min` ile `best` 'i günceller.

Adım	Düğüm (sıra)	Görülen değer	best
1	B (MIN)	başlangıç	99
2	B	yaprak 3	min(99, 3) = 3
3	B	yaprak 12	min(3, 12) = 3
4	B	yaprak 8	min(3, 8) = 3 → B = 3 döner
5	A (MAX)	B = 3	max(−99, 3) = 3
6	C (MIN)	yapraklar 2, 4, 6	99 → 2 → 2 → 2; C = 2
7	A (MAX)	C = 2	max(3, 2) = 3
8	D (MIN)	yapraklar 14, 5, 2	99 → 14 → 5 → 2; D = 2
9	A (MAX)	D = 2	max(3, 2) = 3 → sonuç 3

Tablo 2.16: `minimax("A", True)` izlemesi.



Şekil 2.19: Minimaks ağacının sonucu. MIN düğümleri B = 3, C = 2, D = 2 değerini alır; kök A (MAX) B'yi seçer

ve oyunun değeri 3'tür.

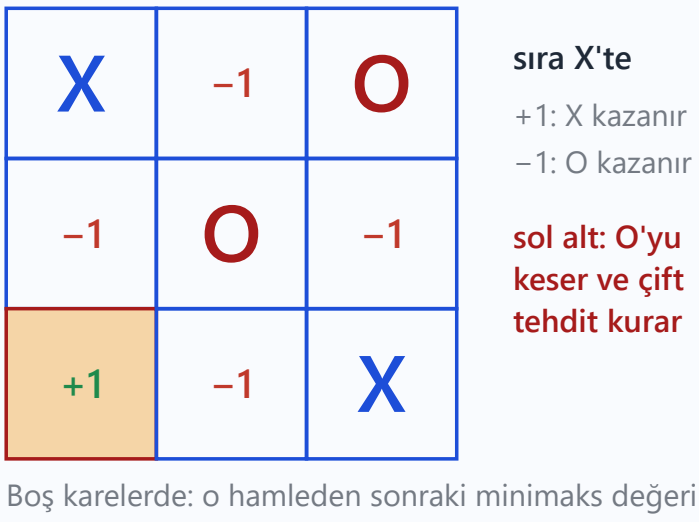
İzlemede dikkat çeken nokta D düğümüdür. D'nin ilk yaprağı 14'tür ve MAX için cazip görünür. Ama D bir MIN düğümüdür: MIN, 14 yerine 2'yi seçer. MAX bunu bilerek D'ye gitmez. Oyunun değeri 3'tür: MAX B'yi seçer ve MIN ne yaparsa yapsın en az 3 garanti eder (Şekil 2.19). Kod 9 yaprağın hepsine bakar. Alfa-beta budaması bazı yaprakları hiç değerlendirmeden atlayabilir: örneğin C'nin ilk yaprağı 2 görüldüğünde C'nin değeri en fazla 2 olacaktır ve 3'ten küçüktür; C'nin öteki yapraklarına bakmaya gerek yoktur.

Sık yapılan hata

- **En büyük yaprağı aramak.** MAX, ağaçtaki en büyük sayıya (14) ulaşamaz, çünkü aradaki seçimi MIN yapar. Değerler aşağıdan yukarı, her düğümde sırası gelen oyuncunun gözüyle hesaplanır.
- **Düzeyleri karıştırmak.** Bir düğümün çocuklarında sıra her zaman öteki oyuncudadır. Kodda bu, özyinelemeli çağrıdaki `not maxi` ile sağlanır.

2.7.3 XOX: tüm ağaç aranabilir

XOX'un (*tic-tac-toe*) ağacı küçüktür. Bütün oyunları sayan bir program 255.168 farklı oyun ve 5.478 erişilebilir konum bulur. Bu sayılar bir bilgisayar için çok küçüktür: tüm ağaç bir saniyeden kısa sürede taranır. Minimaks değerleri +1 (X kazanır), 0 (berabere) ve −1 (O kazanır) olarak alınır. Boş tahtada minimaks değeri 0'dır: iki taraf da hatasız oynarsa oyun berabere biter.



Şekil 2.20: Sıra X'te. Boş karelerde, X oraya oynarsa konumun minimaks değeri yazılıdır: +1 X kazanır, −1 O kazanır. Yalnızca sol alt kare kazandırır.

Şekil 2.20'deki konumda X yalnızca sol alt kareyle kazanır. Bu hamle iki iş yapar: O'nun çaprazını (sağ üst – orta – sol alt) keser ve aynı anda iki tehdit kurar. Sol sütunda (X, boş, X) ve alt satırda (X, boş, X) iki ayrı kazanma tehdidi oluşur. O yalnız birini kapatabilir; X ötekini tamamlar (+1). Öteki her karede O, kendi çaprazını tamamlayarak kazanır (−1).

Satrançta tam arama imkânsızdır; ağaçta yaklaşık 10^{120} oyun vardır. Satranç programları üç teknikle çalışır: **derinlik sınırı** (ağaç belirli sayıda hamleden sonra kesilir), **değerlendirme fonksiyonu** (kesilen yerdeki konuma bir puan verilir: taş değerleri, şah güvenliği vb.) ve **alfa-beta budaması** (sonucu değiştiremeyecek dallar atlanır). Bu üç teknik birlikte, tam aramanın imkânsız olduğu oyunlarda da minimaksı güçlü bir oyuncuya dönüştürür. Minimaks XOX rakibi ise değerleri tam olarak hesaplayarak oynar: onu yenmek mümkün değildir; en iyi ihtimalle berabere kalınır.

2.7.4 Kod izleme: 21'i Söyleme

21'i Söyleme oyununun kuralı şöyledir: iki oyuncu sırayla, 1'den başlayarak 1, 2 ya da 3 ardışık sayı söyler. Örneğin ilk oyuncu “1, 2” der, ikinci “3, 4, 5” der, ilk oyuncu “6” der. 21'i söyleyen kaybeder. Oyunun durumu, son söylenen sayı *c* ile tanımlanır; başlangıçta *c = 0*'dır.

Her durum K (sırası gelen **k**azanır) ya da Y (sırası gelen **y**itirir, yani kaybeder) olarak etiketlenir. Taban durum *c = 20*'dir: sıradaki oyuncu 21'i söylemek zorundadır ve kaybeder (Y). Öteki durumlar sondan başa hesaplanır: bir durumdan rakibi bir Y durumuna itecek bir hamle varsa durum K'dir; yoksa Y'dir.

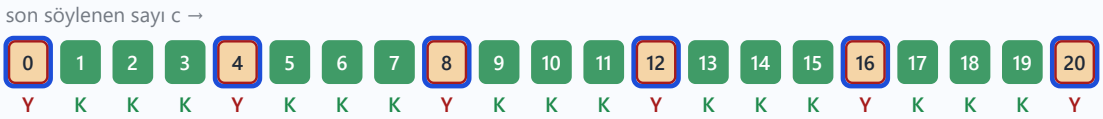
```
HEDEF = 21
kazanir = {HEDEF - 1: False}   # c=20: kayıp
for c in range(HEDEF - 2, -1, -1):
    kazanir[c] = any(not kazanir[c + k]
        for k in (1, 2, 3) if c + k < HEDEF)
kontrol = [c for c in kazanir if not kazanir[c]]
print(sorted(kontrol))
```

Kod şöyle çalışır:

- `kazanir` sözlüğü her *c* için “sırası gelen kazanır mı?” sorusunun cevabını tutar. `kazanir[20] = False` : taban durum.
- Döngü *c* üzerinde 19'dan 0'a kadar geriye doğru gezinir. Her *c* için 1, 2 ya da 3 sayı söylemek *c + k* durumuna götürür (21'e ulaşmadan; `c + k < HEDEF`).
- `any(not kazanir[c + k] …)` : en az bir hamle rakibi kaybeden bir duruma götürüyorsa *c* kazanan durumdur.
- `kontrol` listesi kaybeden durumları toplar. Bunlar, oyuncunun rakibine bırakmak istediği sayılardır: **kontrol sayıları**.

c	Ulaşılabilen durumlar	Rakibi Y'ye itebilir mi?	Etiket
20	— (21'i söylemek zorunda)	hayır	Y
19, 18, 17	20'ye ulaşabilir	evet, 20 söyler	K
16	**17, 18, 19 (hepsi K)**	**hayır**	**Y**
15, 14, 13	16'ya ulaşabilir	evet, 16 söyler	K
12	**13, 14, 15 (hepsi K)**	**hayır**	**Y**
11, 10, 9	12'ye ulaşabilir	evet	K
8	**9, 10, 11 (hepsi K)**	**hayır**	**Y**
7, 6, 5	8'e ulaşabilir	evet	K
4	**5, 6, 7 (hepsi K)**	**hayır**	**Y**
3, 2, 1	4'e ulaşabilir	evet	K
0	**1, 2, 3 (hepsi K)**	**hayır**	**Y**

Tablo 2.17: 21'i Söyleme'de geriye doğru etiketleme.



Şekil 2.21: 21'i Söyleme'de 0–20 durumlarının etiketleri. K: sırası gelen kazanır; Y (vurgulu): sırası gelen kaybeder. Y durumları 4'ün katlarıdır ve rakibe bırakılacak kontrol sayılarıdır.

Kod `[0, 4, 8, 12, 16, 20]` yazdırır. Y durumları 4'ün katlarıdır (Şekil 2.21). Başlangıç durumu *c = 0* Y'dir: ilk oyuncu kaybeder. İkinci oyuncunun stratejisi basittir: rakip *x* sayı söylerse kendisi *4 − x* sayı söyler. Böylece her turun sonunda toplam 4 artar ve ikinci oyuncu hep 4, 8, 12, 16, 20'ye tamamlar. Sonunda ilk oyuncu 21'i söylemek zorunda kalır.

Örnek 2.25

"16'yı Söyleme" oyununda kurallar aynıdır, ama 16'yı söyleyen kaybeder. Kontrol sayıları aynı yöntemle bulunur:

1. Taban durum: *c = 15* Y'dir; sıradaki 16'yı söylemek zorundadır.
2. 14, 13, 12: 15'e ulaşılabilir → K. *c = 11*: 12, 13, 14 hepsi K → Y.
3. Örüntü 4'lük adımlarla sürer: Y durumları 15, 11, 7, 3'tür.
4. *c = 0*: 1, 2 ya da 3 söylenebilir ve 3 bir Y durumudur. Başlangıç K'dir.

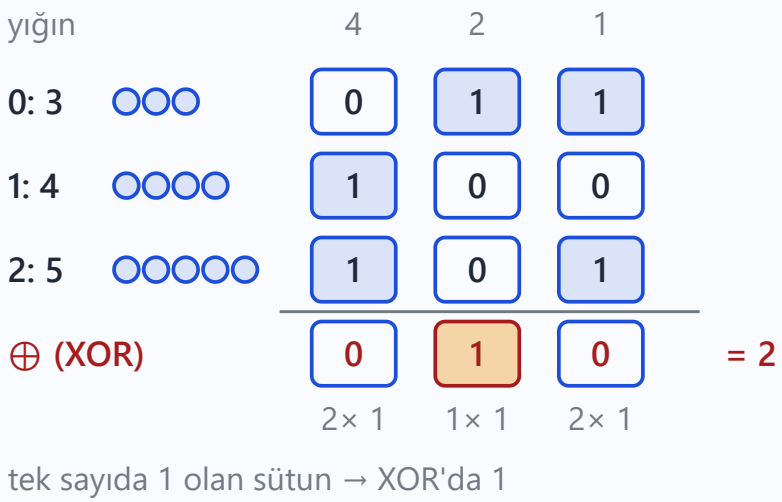
İlk oyuncu kazanır: “1, 2, 3” der ve rakibine 3 bırakır. Sonra her seferinde 4'ün katı artı 3'e (7, 11, 15) tamamlar. Hedef sayı değişince kontrol sayıları kayar ve kazanan oyuncu değişebilir. Genel kural: *H*'yi söyleyen kaybediyorsa Y durumları *H* − *1*'den 4'er geriye giden sayılardır.

2.7.5 Nim ve Bouton teoremi

Nim'de birkaç jeton yığını vardır. Sırası gelen oyuncu tek bir yığından en az bir jeton alır (isterse yığının hepsini). Son jetonu alan kazanır. Küçük konumlar Şekil 2.18'deki gibi elle etiketlenebilir; ama yığın sayısı ve boyu büyüdükçe konum sayısı hızla artar. Charles Bouton 1901'de bütün Nim konumlarını tek bir hesapla etiketleyen bir kural buldu.

Tanım 2.16

Yığın boylarının bit bit XOR'una **nim toplamı** (*nim-sum*) denir ve $\oplus$ ile gösterilir. XOR, eldesiz ikili toplamadır: her bit sütununda 1'lerin sayısı tekse sonuç 1, çiftse 0'dır.



Şekil 2.22: [3, 4, 5] konumunda nim toplamı. Yığınlar ikili yazılır (3 = 011, 4 = 100, 5 = 101); her sütunda 1'ler sayılır. Yalnızca 2'ler sütununda tek sayıda 1 vardır, bu yüzden 3 ⊕ 4 ⊕ 5 = 010 = 2'dir.

Örnek 2.26

3 ⊕ 4 ⊕ 5 sütun sütun hesaplanır (Şekil 2.22):

1. İkili yazım: 3 = 011, 4 = 100, 5 = 101 (sütunlar 4'ler, 2'ler, 1'ler).
2. 4'ler sütunu: 0, 1, 1 → iki tane 1, çift → 0.
3. 2'ler sütunu: 1, 0, 0 → bir tane 1, tek → 1.
4. 1'ler sütunu: 1, 0, 1 → iki tane 1, çift → 0.

5. Sonuç 010 = 2.

Teorem 2.2 (Bouton)

Bir Nim konumu, ancak ve ancak nim toplamı 0 ise kaybeden konumdur (P).

İspat. P ve N konumlarının iki kuralı (Tanım 2.15) denetlenir. Son konum (bütün yığınlar boş) nim toplamı 0'dır ve P'dir; bu, tabanı sağlar.

(1) Nim toplamı 0 olan bir konumdan yapılan her hamle, nim toplamı 0 olmayan bir konuma gider. Bir hamle tek bir yığını değiştirir: h boyunu daha küçük bir h''ye indirir. $h' \neq h$ olduğu için ikisi en az bir bitte farklıdır. O bit sütununda 1'lerin sayısı bir artar ya da bir azalır, yani tek-çift durumu değişir. Önceki toplamda o bit 0'dı; şimdi 1 olur. Toplam artık 0 değildir.

(2) Nim toplamı $x \neq 0$ olan bir konumdan, nim toplamı 0 olan bir konuma giden bir hamle vardır. x'in en soldaki 1 biti alınsın. O bit sütununda 1'lerin sayısı tektir, dolayısıyla en az bir yığının h boyunda o bit 1'dir. Bu yığın $h \oplus x$'e indirilir. $h \oplus x$, h'den küçüktür: o bit 1'den 0'a döner ve daha soldaki bitler değişmez (çünkü x'in o bitten soldaki bitleri 0'dır). Hamle geçerlidir. Yeni toplam $x \oplus h \oplus (h \oplus x) = 0$'dır, çünkü her sayı kendisiyle XOR'lanınca 0 verir.

Toplamı 0 olan konumlar P'nin, toplamı 0 olmayan konumlar N'nin iki kuralını sağlar. Etiketler sondan başa tek türlü belirlendiğinden P-konumları tam olarak nim toplamı 0 olanlardır. □

Teorem bir strateji de verir. Nim toplamı 0 olmayan bir konumda sırası gelen oyuncu, toplamı 0 yapan hamleyi yapar. Rakip toplamı 0 olan konumda her hamlesiyle toplamı bozar; oyuncu her seferinde yeniden 0 yapar. Bütün yığınlar boşken toplam 0'dır ve bu konumu rakibe bırakan, yani son jetonu alan, oyuncudur. İki oyuncuya aynı hamleleri tanıyan (tarafsız) birçok oyun da Nim'e çevrilerek çözülebilir; Sprague–Grundy kuramı bu çevirinin genel yoludur.

2.7.6 Kod izleme: Nim'de kazandıran hamle

Aşağıdaki kod, bir Nim konumunda nim toplamını hesaplar ve toplamı 0 yapan hamleyi bulur.

```
yiginlar = [3, 4, 5]
x = 0
for h in yiginlar:
    x ^= h            # nim toplamı (XOR)
for i, h in enumerate(yiginlar):
    if h ^ x < h:     # küçültülebilir mi?
        yiginlar[i] = h ^ x
        print("yığın", i, ":", h, "->", h ^ x)
        break
print(yiginlar)
```

İlk döngü `x`'i 0'dan başlatır ve her yığını `^=` (XOR atama) ile ekler. İkinci döngü yığınları sırayla dener: `h ^ x < h` ise yığın `h ^ x`'e küçültülebilir ve bu hamle toplamı 0 yapar. İlk uygun yığın bulunca hamle yapılır ve döngü `break` ile durur.

Adım	İşlem	İkili	Sonuç
1	x = 0 XOR 3	000 ⊕ 011 = 011	x = 3
2	x = 3 XOR 4	011 ⊕ 100 = 111	x = 7
3	x = 7 XOR 5	111 ⊕ 101 = 010	x = 2
4	yığın 0: 3 XOR 2	011 ⊕ 010 = 001	1 < 3 → uygun
5	yığın 0: 3 → 1	—	[1, 4, 5], toplam 0

Tablo 2.18: `yiginlar = [3, 4, 5]` için izleme.

Kod "yığın 0 : 3 -> 1" ve `[1, 4, 5]` yazdırır: ilk yığından 2 jeton alınır. Bu yığının seçilmesi rastlantı değildir. `x = 2`'nin en soldaki 1 biti 2'ler bitidir ve 3'ün (011) de bu biti 1'dir. Öteki yığınlar denense ne olur? *4 ⊕ 2 = 6* ve *5 ⊕ 2 = 7* yığından büyüktür; jeton eklemek yasak olduğu için bu hamleler yapılamaz. Yeni konum [1, 4, 5]'te *1 ⊕ 4 ⊕ 5 = 0*'dır (Şekil 2.23).



Şekil 2.23: Hamleden sonra [1, 4, 5]. Her bit sütununda 1'lerin sayısı çifttir ve nim toplamı 0'dır; rakip kaybeden bir konumdadır.

Örnek 2.27

Konum [2, 5, 6] olsun. Sıra bizde.

1. Nim toplamı: *2 ⊕ 5 = 010 ⊕ 101 = 111 = 7*; *7 ⊕ 6 = 111 ⊕ 110 = 001 = 1*. Toplam 1 ≠ 0: kazanan bir konumdur.
2. Yığınlar denenir: *2 ⊕ 1 = 3* (2'den büyük, olmaz); *5 ⊕ 1 = 4* (5'ten küçük, olur); *6 ⊕ 1 = 7* (6'dan büyük, olmaz).
3. Tek kazandıran hamle: ikinci yığını 5'ten 4'e indirmek. Yeni konum [2, 4, 6]: *2 ⊕ 4 ⊕ 6 = 010 ⊕ 100 ⊕ 110 = 000*.

x = 1'in tek 1 biti 1'ler bitidir; bu biti 1 olan tek yığın 5'tir (101). Bu yüzden yalnız bir kazandıran hamle vardır.

2.7.7 Çözülmüş oyunlar

2/3 oyunu ve Mahkûmun İkilemi dışında bu bölümün etkileşimli oyunları iki kişilik, sıralı, mükemmel bilgili ve şans içermeyen oyunlardır. Zermelo teoremine göre hepsi kuramsal olarak geriye doğru çıkarımla çözülebilir; ama ağaç çoğu zaman çok büyüktür. Bir oyunun "çözülmesi", mükemmel oyunda sonucun (ilk oyuncu kazanır, ikinci kazanır ya da berabere) bulunması demektir. Tablo 2.19 bilinen sonuçları özetler.

Oyun	Mükemmel oyunda sonuç	Nasıl?
XOX	Beraberlik	Tüm ağaç aranır (minimaks)
Nim	Nim toplamı ≠ 0 ise sıradaki kazanır	Bouton'un XOR kuralı (1901)
21'i Söyleme	İkinci oyuncu kazanır	Geriye doğru etiketleme; kontrol sayıları 4'ün katları
Dörtlü Bağla	İlk oyuncu kazanır	1988'de bilgisayar aramasıyla çözüldü
Hex	İlk oyuncu kazanır (beraberlik yok)	Strateji çalma; kazanan strateji büyük tahtalarda bilinmiyor
Othello (8×8)	Beraberlik	2023'te bilgisayar aramasıyla
Dama	Beraberlik	2007'de çözüldü (Schaeffer vd., 2007)
Satranç	Belirli ama bilinmiyor	Zermelo; ağaç $\approx 10^{120}$ oyun

Tablo 2.19: Bazı kombinatoryal oyunlar ve çözüm durumları.

Mangala gibi oyunlarda hamle ağacında birkaç adım ileriye bakmak oyuncuya önemli bir üstünlük sağlar. Tilki ve Kazlar'da roller asimetriktir: bir tarafta tek tilki, öbür tarafta çok sayıda kaz vardır; yine de bilgi mükemmeldir. Dokuz Taş da mükemmel bilgili klasik bir strateji oyunudur. Bu oyunların hepsinde aynı ilke geçerlidir: oyun sonlu ve bilgi mükemmelse, sonuç kuramsal olarak bellidir; pratikteki zorluk ağacın büyüklüğüdür.

2.8 Sıralı oyunlar ve geriye doğru çıkarım

Kombinatoryal oyunlarda kullanılan "sondan başa" fikri, kazançları sıfır toplamlı olmayan sıralı oyunlara da uygulanır. Bu oyunlarda yeni bir sorun ortaya çıkar: bazı Nash dengeleri, hiçbir zaman uygulanmayacak tehditlere dayanır. Alt oyun mükemmel denge kavramı bu sorunu çözer; geriye doğru çıkarım da bu dengeyi bulan algoritmadır.

2.8.1 Alt oyun ve alt oyun mükemmel denge

Tanım 2.17

Alt oyun (*subgame*), oyun ağacında tek bir karar düğümünden başlayıp o düğümün altındaki her şeyi içeren ve hiçbir bilgi kümesini bölmeyen oyun parçasıdır.

Alt oyun mükemmel denge (*subgame perfect equilibrium, SPE*), her alt oyunda bir Nash dengesi oluşturan strateji profilidir (Selten, 1965).

Alt oyun tanımındaki "bilgi kümesini bölmeme" koşulu şunu söyler: bir düğümden başlayan parça, oyuncunun o noktada ne bildiğini bozmamalıdır. Bir bilgi kümesinin bir düğümü parçanın içinde, öteki dışında kalırsa parça bir alt oyun değildir. Mükemmel bilgili oyunlarda her bilgi kümesi tek düğüm olduğu için her karar düğümü bir alt oyun başlatır. Bütün oyun da kendi kendinin bir alt oyunudur.

Neden yeni bir denge kavramı gerekir? Sıradan Nash dengesi, oyunun hiç ulaşılmayan düğümlerinde ne olduğuna bakmaz. Bir oyuncu, gerçekte hiç uygulamayacağı bir tehditle ötekini korkutabilir; tehdit hiç sınanmadığı için Nash dengesi bunu yakalayamaz. Böyle bir tehdide **inandırıcı olmayan tehdit** (*non-credible threat*) denir. SPE, her alt oyunda, yani ulaşılmayanlar dâhil her düğümde, oyuncuların en iyi seçimi yapmasını ister. Bu yüzden inandırıcı olmayan tehditleri eler.

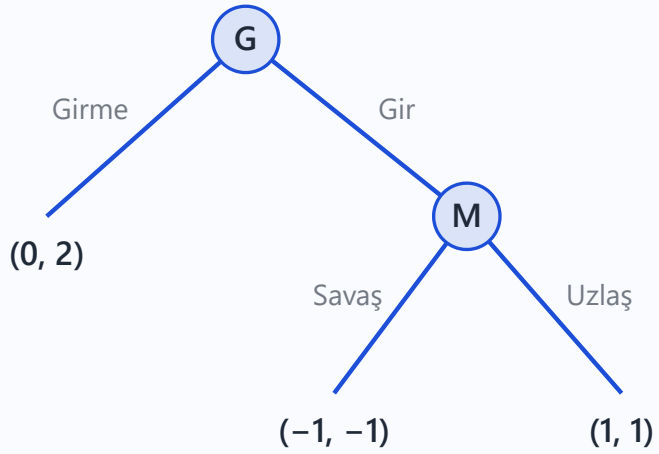
Sonlu ve mükemmel bilgili oyunlarda SPE, **geriye doğru çıkarım** (*backward induction*) ile bulunur (Kuhn/Zermelo). Algoritma dört adımdan oluşur:

1. Yalnızca yapraklara bağlı son karar düğümleri bulunur.
2. Orada karar veren oyuncunun en iyi dalı seçilir: yalnız o oyuncunun kazancı karşılaştırılır.
3. Düğüm, seçilen yaprağın kazanç vektörüyle değiştirilir; düğüm artık bir yapraktır.
4. Köke ulaşılana kadar bu adımlar tekrarlanır.

Algoritma her düğümü tam bir kez işler; bu yüzden çalışma süresi ağacın boyuyla doğru orantılıdır. Ağaç küçükse çözüm hızlıdır. Satrançtaki zorluk algoritmada değil, ağacın büyüklüğündedir.

2.8.2 Piyasaya giriş: inandırıcı olmayan tehdit

Piyasaya giriş oyunu (Şekil 2.9) burada kısa adlarla yazılır: G (giren firma, Rakip) önce seçer; M (mevcut firma, Tekel) yalnızca G girerse karar verir.



Şekil 2.24: Piyasaya giriş oyunu. G girmezse $(0, 2)$; girerse M savaşır $(-1, -1)$ ya da uzlaşır $(1, 1)$. Yapraklarda (u_G, u_M).

Önce oyunun normal formu yazılır. G'nin iki stratejisi vardır: Gir, Girme. M'nin tek karar düğümü ve iki stratejisi vardır: Savaş, Uzlaş (yani "G girerse savaşırım" ve "G girerse uzlaşırım"). G girmezse M'nin stratejisi sonucu etkilemez.

G \ M	Savaş	Uzlaş
Gir	$(-1, -1)$	**$(1, 1)$**
Girme	**$(0, 2)$**	$(0, 2)$

Tablo 2.20: Piyasaya giriş oyununun normal formu. Yeşil hücreler saf Nash dengeleridir.

Örnek 2.28

Tablo 2.20'de en iyi tepkiler işaretlenir:

1. M Savaş derse G için Gir -1, Girme 0 verir → Girme. M Uzlaş derse G için Gir 1, Girme 0 verir → Gir.
2. G Gir derse M için Savaş -1, Uzlaş 1 verir → Uzlaş. G Girme derse M'nin iki stratejisi de 2 verir → ikisi de en iyi tepkidir.
3. İki işaretli hücreler: (Gir, Uzlaş) ve (Girme, Savaş).

Oyunun iki saf Nash dengesi vardır. İkincisinde M "girersen savaşırım" der ve G, bu tehdide inanıp girmez. Tehdit hiç sınanmaz; M'nin kazancı 2'dir ve sapmak ona bir şey kazandırmaz.

Geriye doğru çıkarım bu iki dengeden birini eler:

1. **Son düğüm M:** Savaş → −1, Uzlaş → 1. M Uzlaş'ı seçer. M'nin düğümünün değeri (1, 1) olur.
2. **Kök G:** Girme → 0, Gir → 1 (M'nin uzlaşacağını bilerek). G Gir'i seçer.
3. **SPE:** (Gir, Uzlaş); sonuç (1, 1).

(Girme, Savaş) bir Nash dengesidir, ama alt oyun mükemmel değildir. M'nin düğümünden başlayan alt oyunda, yani G girdikten sonra, savaşmak M'ye −1, uzlaşmak 1 verir. Bu alt oyunda Savaş bir en iyi tepki değildir. "Girersen savaşırım" tehdidi inandırıcı değildir: G girdiği anda M'nin çıkarı uzlaşmaktır. Rasyonel bir G bunu öngörür ve tehdide aldırmadan girer.

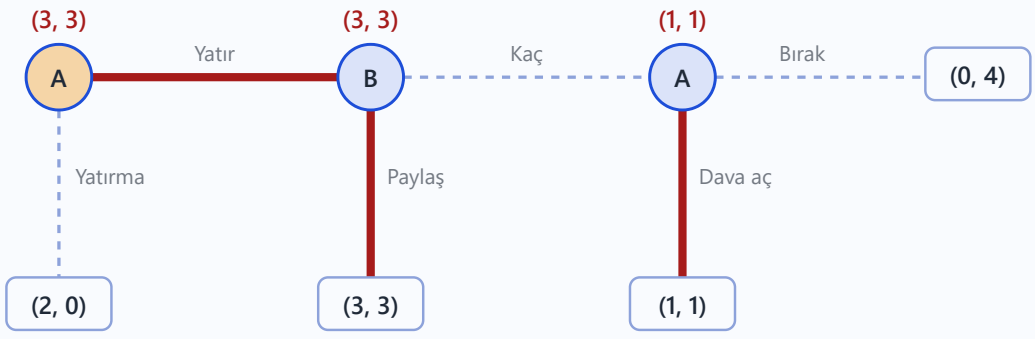
2.8.3 Yatırım oyunu adım adım

Üç aşamalı bir oyun, geriye doğru çıkarımın birden çok adımda nasıl işlediğini gösterir. Ayşe (A) yatırım yapar ya da yapmaz. Yatırım yaparsa girişimci Bora (B) kârı paylaşır ya da parayla kaçar. Bora kaçarsa Ayşe dava açar ya da bırakır. Yapraklarda (Ayşe, Bora) kazançları yazılır: Yatırma (2, 0); Paylaş (3, 3); Kaç ve Dava aç (1, 1); Kaç ve Bırak (0, 4).

Örnek 2.29

1. **Son karar düğümü** (Ayşe, Bora kaçtıktan sonra): Dava aç 1, Bırak 0. 1 > 0 olduğundan Ayşe dava açar. Düğümün değeri (1, 1).
2. **Bora'nın düğümü:** Bora, kaçarsa dava edileceğini bilir. Paylaş 3, Kaç 1 (dava düğümünün değerinden). 3 > 1 olduğundan Bora paylaşır. Düğümün değeri (3, 3).
3. **Kök** (Ayşe): Yatır 3, Yatırma 2. 3 > 2 olduğundan Ayşe yatırır.

Alt oyun mükemmel denge: Ayşe'nin stratejisi (Yatır, Dava aç), Bora'nın stratejisi Paylaş; sonuç (3, 3). Ayşe'nin stratejisi "Dava aç" bileşenini de içerir, çünkü strateji ulaşılmayan düğümlerde ne yapılacağını da söyler. Bora'nın paylaşmasının nedeni tam olarak bu ulaşılmayan seçimdir.



A: Ayşe (yatırımcı), B: Bora (girişimci); yaprakta (Ayşe, Bora)

Şekil 2.25: Yatırım oyununun çözümü. Kalın kırmızı dallar her düğümde seçilen eylemdir; düğümlerin üstünde

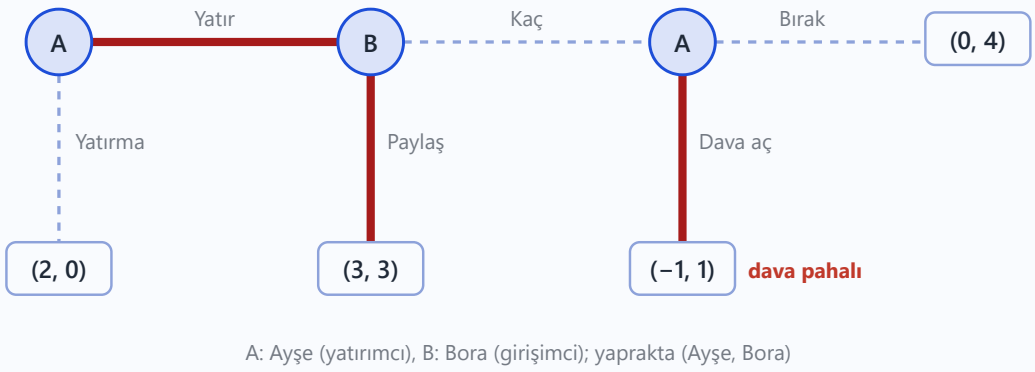
geriye doğru çıkarımla bulunan değerler yazılıdır. Sonuç (3, 3).

Dava pahalı olsun: dava açan Ayşe 1 yerine −1 alır. Ağaç aynıdır, yalnızca bir yaprak (1, 1)'den (−1, 1)'e değişmiştir. Geriye doğru çıkarım baştan yapılır.

Örnek 2.30

1. **Son düğüm:** Bırak 0, Dava aç −1. 0 > −1 olduğundan Ayşe bırakır. Düğümün değeri (0, 4).
2. **Bora:** Kaç 4, Paylaş 3. 4 > 3 olduğundan Bora kaçar. Düğümün değeri (0, 4).
3. **Kök:** Yatır 0, Yatırma 2. 2 > 0 olduğundan Ayşe yatırmaz.

Sonuç (2, 0) ikisi için de (3, 3)'ten daha kötüdür. Tek bir yaprağın değişmesi, zincirleme olarak her düğümün kararını değiştirmiştir.



Şekil 2.26: Dava pahalı olduğunda (yaprak (−1, 1)) "kaçarsan dava açarım" tehdidi inandırıcı değildir. (Yatır, Paylaş) yolu yine bir Nash dengesiyle desteklenebilir, ama alt oyun mükemmel değildir.

Şekil 2.26 ince bir noktaya işaret eder. Ayşe "kaçarsan dava açarım" planını açıklarsa (Ayşe: Yatır, Dava aç; Bora: Paylaş) profili yine bir Nash dengesidir. Bu profilde Ayşe 3, Bora 3 alır. Bora kaçarsa dava edilir ve 1 alır; sapmaz. Ayşe yatırmazsa 2 alır; sapmaz. Dava düğümüne hiç ulaşılmadığı için Ayşe'nin oradaki seçimi kazancını etkilemez. Ama tehdit inandırıcı değildir: Bora gerçekten kaçarsa Ayşe dava açarak −1, bırakarak 0 alır ve bırakır. Bora bunu bilir. Bu profil alt oyun mükemmel denge değildir.

Buradan çıkan ders şudur: söz vermek yetmez; tehdidin uygulanması o anda kârlı olmalıdır. Ayşe'nin dava açmayı kârlı kılmasının yolları vardır: avukatla önceden sabit ücretli anlaşmak ya da Bora'dan teminat almak. Bu tür önlemler oyunun kazançlarını değiştirerek tehdidi inandırıcı yapar.

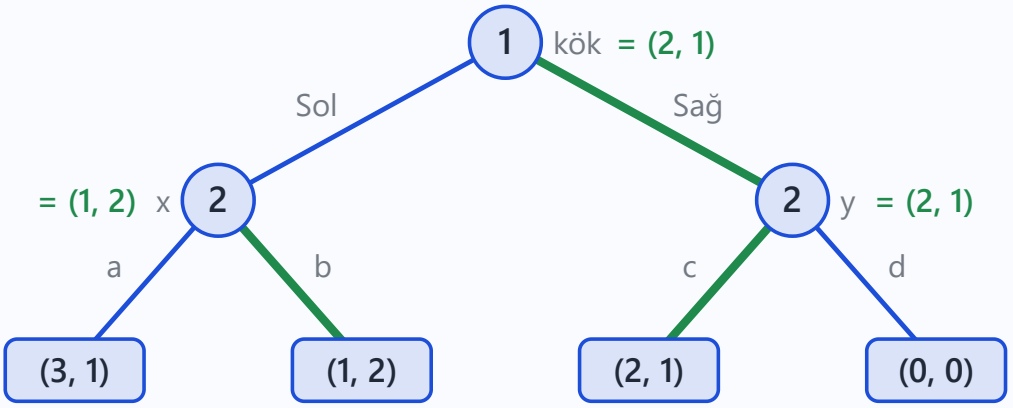
2.8.4 Kod izleme: cozum()

Geriye doğru çıkarım kısa bir özyinelemeli fonksiyonla yazılabilir. Ağaç iç içe listelerle gösterilir: her karar düğümü `[ad, oyuncu, dallar]` biçimindedir; `dallar`, `(hamle, cocuk)` çiftlerinin listesidir. Yapraklar kazanç demetleridir (`tuple`). Oyuncular 0 ve 1 indisleriyle gösterilir: oyuncu 1 indis 0, oyuncu 2 indis 1'dir.

```
def cozum(d):
    if isinstance(d, tuple):         # yaprak: kazanç
        return d
    ad, oyuncu, dallar = d
    en_iyi = None
    for hamle, cocuk in dallar:
        v = cozum(cocuk)
        if en_iyi is None or v[oyuncu] > en_iyi[oyuncu]:
            en_iyi, secim = v, hamle
    print(ad, secim)
    return en_iyi
kok = ['kök', 0, [('Sol', ['x', 1, [('a', (3, 1)), ('b', (1, 2))]]),
                  ('Sağ', ['y', 1, [('c', (2, 1)), ('d', (0, 0))]])]]
print(cozum(kok))
```

Fonksiyon şöyle çalışır:

- Düğüm bir demetse yapraktır ve kazanç vektörü olduğu gibi döner.
- Değilse düğüm üç parçaya açılır: `ad`, karar veren `oyuncu` ve `dallar`.
- Her dal için çocuğun değeri özyinelemeli olarak hesaplanır: `v = cozum(cocuk)`.
- Karşılaştırmada yalnızca *karar veren oyuncunun* kazancı kullanılır: `v[oyuncu] > en_iyi[oyuncu]`. Kesin büyüktür (`>`) kullanıldığı için eşitlikte ilk görülen dal korunur.
- Düğümün seçimi yazdırılır ve seçilen dalın kazanç vektörü döner; üst düğüm bu vektörü bir yaprak gibi kullanır.



Şekil 2.27: `cozum(kok)` ağacı ve sonucu. Kökte oyuncu 1, x ve y düğümlerinde oyuncu 2 seçer; x'te b, y'de c, kökte Sağ seçilir. Sonuç (2, 1).

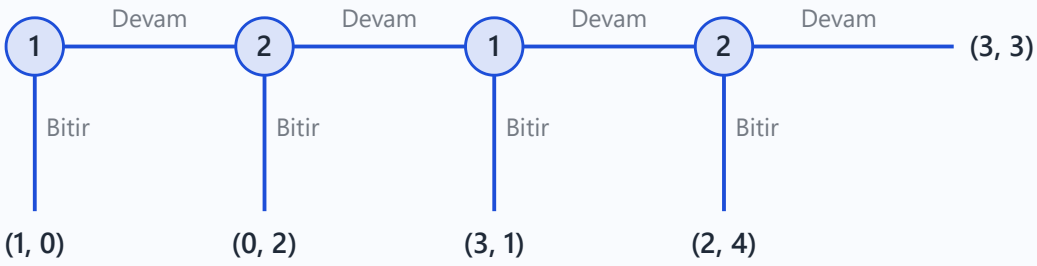
Adım	Düğüm (oyuncu)	Dal ve değer v	Karşılaştırma	en_iyi, secim
1	x (1)	a → (3, 1)	en_iyi yok	(3, 1), a
2	x (1)	b → (1, 2)	v[1] = 2 > 1	(1, 2), b → “x b”
3	kök (0)	Sol → (1, 2)	en_iyi yok	(1, 2), Sol
4	y (1)	c → (2, 1)	en_iyi yok	(2, 1), c
5	y (1)	d → (0, 0)	v[1] = 0 > 1? hayır	(2, 1), c → “y c”
6	kök (0)	Sağ → (2, 1)	v[0] = 2 > 1	(2, 1), Sağ → “kök Sağ”

Tablo 2.21: `cozum(kok)` izlemesi.

Program sırasıyla “x b”, “y c”, “kök Sağ” ve `(2, 1)` yazdırır. Adım 2 önemlidir: x'te oyuncu 2 kendi kazancına bakar ve b'yi seçer (2 > 1). Böylece oyuncu 1'in en sevdiği yaprak (3, 1) ulaşılamaz olur. Kökte oyuncu 1, Sol'un gerçek değerini (1, 2) ve Sağ'ın gerçek değerini (2, 1) karşılaştırır ve Sağ'ı seçer. SPE: (Sağ; x'te b, y'de c). Ulaşılmayan x düğümündeki b de stratejinin parçasıdır; oyuncu 1'in Sol'u reddetmesinin nedeni odur.

2.8.5 Kırkayak oyunu

Kırkayak oyunu (*centipede game*), Robert Rosenthal'ın 1981'de tanımladığı bir oyundur (Rosenthal, 1981). İki oyuncu sırayla karar verir. Her düğümde karar veren oyuncu ya “Bitir” der ve o anki paylaşım gerçekleşir, ya da “Devam” der ve toplam büyür. Toplam kazanç 1 → 2 → 4 → 6 diye büyür, sonda da 6'dır. Ama devam eden oyuncu, rakibi bir sonraki adımda bitirirse daha az alır.



Şekil 2.28: Dört düğümlü kırkayak oyunu. Düğümdeki sayı karar veren oyuncudur. Aşağı doğru dallar Bitir'dir; yapraklarda (Oyuncu 1, Oyuncu 2) kazancı yazılır. Herkes devam ederse sonuç (3, 3).

Yaprakları sırayla okumak oyunun gerilimini gösterir. Oyuncu 1 ilk düğümde bitirirse (1, 0) olur. Devam ederse oyuncu 2 ikinci düğümde bitirerek (0, 2) alabilir: oyuncu 2 için 2 > 0. Oyuncu 2 devam ederse oyuncu 1 üçüncü düğümde (3, 1) ile bitirebilir. Dördüncü düğümde oyuncu 2 (2, 4) ile bitirebilir ya da devam edip (3, 3)'e gidebilir.

Geriye doğru çıkarım son düğümden başlar. Tablo 2.22 dört adımın sonucunu gösterir.

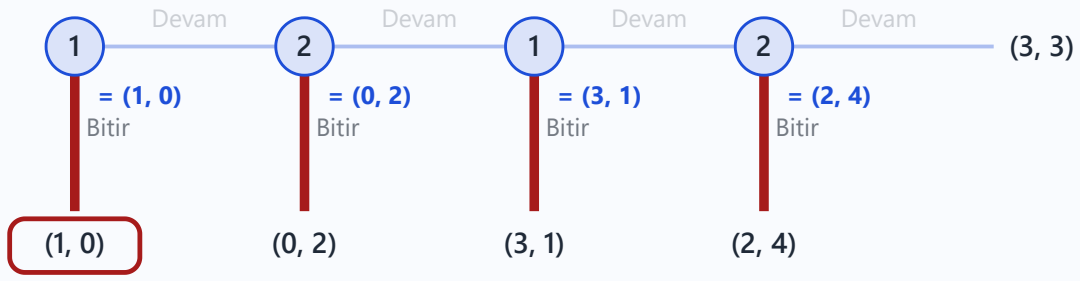
Düğüm	Karar veren	Bitir	Devamın değeri	Karar	Düğüm değeri
4	Oyuncu 2	(2, 4)	(3, 3)	Bitir	(2, 4)
3	Oyuncu 1	(3, 1)	(2, 4)	Bitir	(3, 1)
2	Oyuncu 2	(0, 2)	(3, 1)	Bitir	(0, 2)
1	Oyuncu 1	(1, 0)	(0, 2)	Bitir	(1, 0)

Tablo 2.22: Kırkayak oyununda geriye doğru çıkarım.

Örnek 2.31

1. **4. düğüm, Oyuncu 2:** Bitir 4, Devam 3 (devamın değeri (3, 3)). 4 > 3 ⇒ Bitir; düğümün değeri (2, 4).
2. **3. düğüm, Oyuncu 1:** Bitir 3, Devam 2 (devamın değeri (2, 4)). 3 > 2 ⇒ Bitir; düğümün değeri (3, 1).
3. **2. düğüm, Oyuncu 2:** Bitir 2, Devam 1 (devamın değeri (3, 1)). 2 > 1 ⇒ Bitir; düğümün değeri (0, 2).
4. **1. düğüm, Oyuncu 1:** Bitir 1, Devam 0 (devamın değeri (0, 2)). 1 > 0 ⇒ Bitir; düğümün değeri (1, 0).

Alt oyun mükemmel denge: her oyuncu her düğümde Bitir der. Oyun ilk hamlede (1, 0) ile biter; oysa (3, 3) ikisi için de daha iyiydi.



Şekil 2.29: Kırkayak oyununun çözümü. Her düğümde Bitir (kalın kırmızı) seçilir; düğüm değerleri sağdan sola (2, 4), (3, 1), (0, 2), (1, 0) olarak bulunur. Oyun ilk düğümde (1, 0) ile biter.

Örnek 2.32

En sağdaki kazanç (3, 3) yerine (5, 5) olsun. Geriye doğru çıkarım yeniden yapılır:

1. 4. düğüm, Oyuncu 2: Devam 5 > Bitir 4 ⇒ Devam; değer (5, 5).
2. 3. düğüm, Oyuncu 1: Devam 5 > Bitir 3 ⇒ Devam; değer (5, 5).
3. 2. düğüm, Oyuncu 2: Devam 5 > Bitir 2 ⇒ Devam; değer (5, 5).
4. 1. düğüm, Oyuncu 1: Devam 5 > Bitir 1 ⇒ Devam; değer (5, 5).

Herkes devam eder ve sonuç (5, 5) olur. Oyunun çözülmesinin anahtarı son düğümdür: orada oyuncu 2 bitirmeyi tercih etmediği anda bütün zincir tersine döner.

2.8.6 Kırkayak: teori ve deney

Teori. SPE'de her oyuncu her düğümde Bitir der ve oyun ilk hamlede (1, 0) ile biter. Daha güçlü bir sonuç da vardır: oyunun herhangi bir Nash dengesinde sonuç aynıdır. Nedeni şudur. Oyunun 1. düğümden sonra bittiği bir Nash dengesi olsaydı, oyunun bittiği yerden bir önceki düğümde karar veren oyuncu sapmak isterdi. Örneğin oyun 3. düğümde (3, 1) ile bitseydi, 2. düğümdeki oyuncu 2 devam ederek 1 alırken bitirerek 2 alabilirdi. Oyun sonuna kadar sürüp (3, 3) olsaydı, 4. düğümdeki oyuncu 2 bitirerek 4 alabilirdi. Her durumda kârlı bir tek taraflı sapma vardır; demek ki her Nash dengesinde oyun ilk düğümde biter.

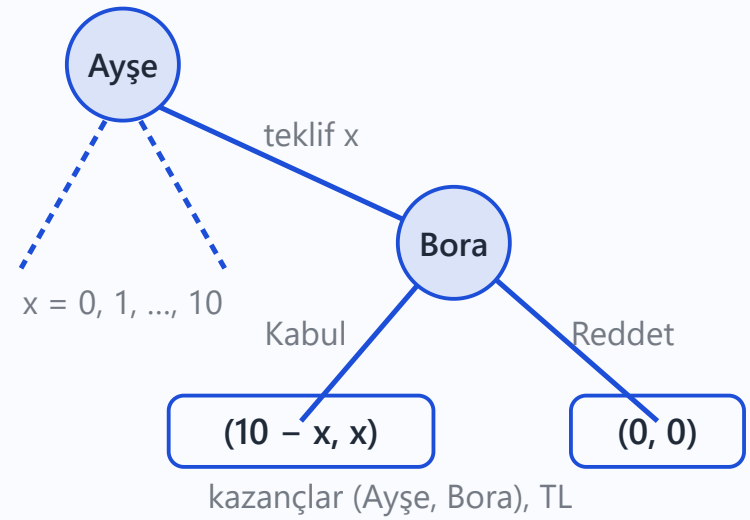
Verimsizlik. (3, 3) iki oyuncu için de (1, 0)'dan iyidir: SPE sonucu Pareto optimal değildir. Düğüm sayısı arttıkça kayıp büyür; yüz düğümlü bir kırkayakta oyuncular büyük bir toplamı ilk adımda feda eder.

Deney. Laboratuvarda oyuncuların çoğu ilk düğümde bitirmez; birkaç tur devam eder (McKelvey ve Palfrey, 1992). Bunun nedeni, geriye doğru çıkarımın rasyonelliğin ortak bilgi olmasını ve uzun bir akıl yürütme zincirini gerektirmesidir. Rakibin biraz "iyi niyetli" olma ihtimali bile devam etmeyi kârlı kılar: rakip bir tur daha devam ederse oyuncu daha büyük bir paydan bitirebilir.

Kırkayak, sonlu tekrarlanan Mahkûmun İkilemi ile aynı mantığı paylaşır: son turdan geriye bakınca işbirliği çözülür. Sonsuz ufukta ise işbirliği dengede kalabilir (2.8.8).

2.8.7 Últimatom oyunu

Últimatom oyunu, pazarlığın en basit modelidir (Güth, Schmittberger ve Schwarze, 1982). Ayşe 10 TL'nin x TL'sini Bora'ya önerir; x tam TL'dir (0, 1, ..., 10). Bora kabul ederse paylaşım $(10 - x, x)$ olur. Reddederse ikisi de 0 alır.



Şekil 2.30: Últimatom oyununun ağacı. Ayşe $x = 0, 1, ..., 10$ teklif eder; Bora her teklif için kabul eder ya da reddeder. Kazançlar (Ayşe, Bora), TL.

Örnek 2.33

1. **Bora'nın düğümleri.** Her x için ayrı bir düğüm vardır (11 düğüm). $x > 0$ ise kabul x, ret 0 verir; $x > 0$ olduğundan Bora kabul eder. $x = 0$'da kabul de ret de 0 verir; Bora kayıtsızdır.
2. **Ayşe'nin kararı (Bora 0'ı reddediyorsa).** Ayşe, Bora'nın her pozitif teklifi kabul edeceğini bilir. Payı $10 - x$'i en büyük yapan pozitif teklif $x = 1$'dir. SPE sonucu (9, 1).
3. **Ayşe'nin kararı (Bora 0'ı da kabul ediyorsa).** Ayşe $x = 0$ teklif eder ve 10 alır. Bu da bir SPE'dir; sonuç (10, 0).

Bora'nın stratejisi 11 düğümün her birinde bir karar içerir. Bu yüzden Bora'nın $2^{11} = 2048$ stratejisi vardır.

Bu oyunda da inandırıcı olmayan bir tehdit vardır. Bora "4 TL'den azını reddederim" derse ve Ayşe buna inanırsa Ayşe 4 teklif eder. (Ayşe: 4 teklif et; Bora: 4 ve üstünü kabul, altını ret) profili bir Nash dengesidir: Ayşe daha az teklif ederse 0 alır, daha çok teklif ederse daha az kalır; Bora'nın ulaşılan dü-

ğümdeki seçimi (kabul) en iyisidir. Ama 1 TL önerildiğinde reddetmek Bora'ya $0 < 1$ verir; tehdit inandırıcı değildir ve profil alt oyun mükemmel değildir.

Teklif (toplamın oranı)	Kuram (yalnız para)	Deneylerde
En sık teklif	en küçük birim	%40–50
%20'nin altındaki teklifler	kabul edilir	yaklaşık yarısı reddedilir

Tablo 2.23: Ültimatom oyununda kuram ve deney (Camerer, 2003).

Deney sonuçları kuramdan çok farklıdır. Teklif verenlerin çoğu toplamın %40–50'sini önerir; %20'nin altındaki tekliflerin yaklaşık yarısı reddedilir. Bu sonuç "oyuncular geriye doğru çıkarım yapamıyor" diye okunmamalıdır. Ret, adalet kaygısının fayda fonksiyonunda yer aldığını gösterir: haksız bir teklifi reddetmek, birine 1 TL'den daha değerli gelebilir. Teklif verenler de bunu öngörüp cömert davranır (Camerer, 2003). Bu, 2.1.5'teki ilkenin bir örneğidir: rasyonel, bencil demek değildir.

2.8.8 Sonsuz tekrarlı Mahkûmun İkilemi

Mahkûmun İkilemi aynı iki oyuncu arasında tekrar tekrar oynanırsa işbirliği mümkün olur mu? Cevap, oyunun ne zaman biteceğinin bilinip bilinmediğine bağlıdır. Aynı tek atışlık oyun tekrarlanır; sonucu belirleyen ufuktur.

Tanım 2.18

Kinci tetik stratejisi (*grim trigger*): İlk turda Sessiz oynanır; rakip bir kez ihanet ederse sonsuza dek İhanet oynanır.

Sonlu ve bilinen ufuk. Oyun 10 tur sürsün ve bu bilinsin. Son turda gelecek yoktur, dolayısıyla ceza da yoktur: 10. tur tek atışlık bir Mahkûmun İkilemi'dir ve orada İhanet baskındır. 9. turda oyuncular 10. turda ne olacağını bilir: ne yaparlarsa yapsınlar 10. turda ikisi de ihanet edecektir. Bu yüzden 9. turdaki seçimin bir geleceği yoktur ve 9. tur da tek atışlık bir oyun gibidir: İhanet. Geriye doğru çıkarım bunu ilk tura kadar taşır. Her turda İhanet oynanır; kırkayaktaki gibi işbirliği sondan başa çözülür.

Sonsuz ufuk. Her turdan sonra oyun δ olasılıkla bir tur daha sürsün. Oyunun bilinen bir son turu yoktur. Bugünkü ihanetin kazancı, gelecekteki kalıcı cezanın kaybıyla tartılır. δ yeterince büyükse, yani oyuncular sabırlıysa ya da oyunun sürme olasılığı yüksekse, sapmak kârlı değildir ve işbirliği dengede kalır.

Örnek 2.34

Sam ve John'un kazançlarıyla (Tablo 2.8) iki oyuncu da kinci tetik oynasın. Sam'in sapmasının kârlı olup olmadığı hesaplanır.

1. **Uymak.** Her tur (Sessiz, Sessiz) oynanır ve Sam her tur −1 alır. Toplam: $-1 - \delta - \delta^2 - \ldots = -1 / (1 - \delta)$ (Örnek 2.11 formülüyle).
2. **Sapmak.** Sam bir turda ihanet eder ve o tur 0 alır (John sessizdir). Sonraki her tur John ihanet eder; Sam'in en iyisi de ihanettir ve her tur −2 alır. Toplam: $0 + \delta(-2) + \delta^2(-2) + \ldots = -2\delta / (1 - \delta)$.
3. **Karşılaştırma.** Uymak sapmaktan kötü değilse: $-1 / (1 - \delta) \geq -2\delta / (1 - \delta)$. İki taraf $(1 - \delta) > 0$ ile çarpılır: $-1 \geq -2\delta$, yani $\delta \geq 1/2$.

Oyunun bir tur daha sürme olasılığı en az 1/2 ise ihanet kârsızdır ve iki kinci tetik stratejisi birlikte işbirliğini sürdürür. Örneğin $\delta = 0{,}9$'da uymak −10, sapmak −18 verir.

Ceza inandırıcıdır, çünkü (İhanet, İhanet) tek atışlık oyunun bir dengesidir. Ceza aşamasında hiçbir oyuncu tek başına sessiz kalarak kazanamaz: rakip ihanet ederken sessiz kalmak −3, ihanet −2 verir. Bu yüzden iki kinci tetik stratejisi birlikte alt oyun mükemmel dengedir: hem işbirliği aşamasında ($\delta \geq 1/2$ iken) hem de ceza aşamasında kimse sapmak istemez.

2.8.9 Sıralı oyunlarda sık hatalar

Sık yapılan hata

- **Kökten başlamak.** İlk hamle, sonraki hamlelere verilecek tepkiye bağlıdır; çözüm her zaman son karar düğümlerinden başlar.
- **Yanlış kişinin kazancına bakmak.** Her düğümde yalnız orada karar veren oyuncunun sayısı karşılaştırılır. Yatırım oyununda Bora'nın düğümünde ikinci sayıya bakılır.
- **Stratejiyi yol sanmak.** Strateji, ulaşılmayan düğümler dâhil her düğüm için seçimi içerir: Ayşe'nin stratejisi (Yatır, Dava aç)'tır, yalnız "Yatır" değil.
- **Nash = alt oyun mükemmel sanmak.** Her alt oyun mükemmel denge bir Nash dengesidir; tersi doğru değildir. İnandırıcı olmayan tehdit içeren Nash dengeleri vardır (piyasaya giriş, ültimatom).
- **Eşzamanlıyı sıralı çözmek.** Hamleler görülmüyorsa (Mahkûmun İkilemi) ağaçta bilgi kümesi vardır; geriye doğru çıkarım doğrudan uygulanmaz, matris çözülür.
- **Söz = taahhüt sanmak.** "Dava açarım" demek yetmez; tehdidin uygulanması o anda kârlı olmalı ya da önceden bağlanmalıdır (sözleşme, teminat).

2.9 Minimaks ve alfa-beta budaması

Minimaks değerinin tanımı birkaç satırdır: yaprakta oyunun sonucu, MAX düğümünde çocukların en büyüğü, MIN düğümünde en küçüğü. Bu tanım küçük ağaçlarda elle, XOX'ta bir bilgisayarla kolayca uygulanır. Zorluk tanımda değil, ağacın büyüklüğündedir. Bir hamle daha ileriye bakmak, bakılacak

konum sayısını dallanma etkeni kadar katlar. Bu yüzden oyun oynayan programların gücü üç soruya verdikleri cevaba bağlıdır: hangi dallar hiç okunmadan atlanabilir, hamleler hangi sırayla denenmelidir ve ağaç yapraklara kadar inemiyorsa yarıda kalan konumlara hangi değer verilecektir? Alfa-beta budaması birinci soruyu, hamle sıralaması ikincisini, değerlendirme fonksiyonu ile derinlik sınırlı arama üçüncüsünü cevaplar.

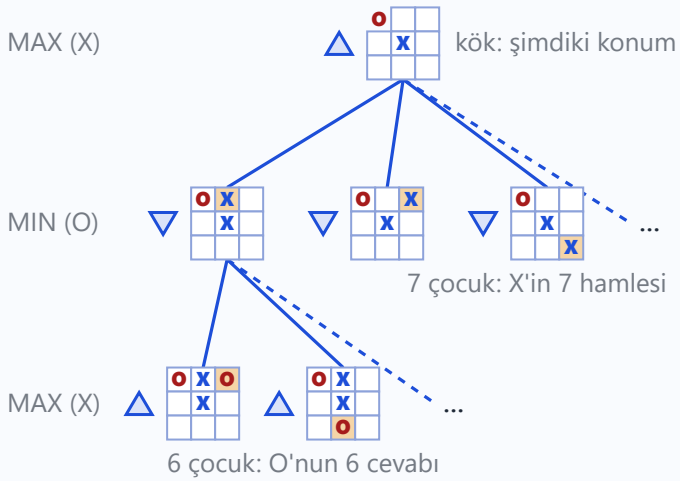
2.9.1 Oyun ağacı ve ağacın büyüklüğü

Genişletilmiş formda bir oyun bir ağaçla yazılır. İki kişilik, sıfır toplamlı ve mükemmel bilgili oyunlarda bu ağacın her katmanında yalnız bir oyuncu karar verir ve kazançları yazmak için tek bir sayı yeter. Bu tür oyunlar için kullanılan dil aşağıdaki tanımda toplanmıştır.

Tanım 2.19

İki kişilik, sıfır toplamlı, mükemmel bilgili bir oyunun **oyun ağacında** (*game tree*) her **düğüm** bir konumdur: tahtanın durumu ve sıranın kimde olduğu. Her **kenar** bir hamledir. **Kök** şimdiki konumdur; **yaprak** ise oyunun bittiği konumdur ve çocuğu yoktur. Kökte sırası olan oyuncuya **MAX** (▲), rakibine **MIN** (▽) denir. MAX değeri büyütmek, MIN küçültmek ister. Katmanlar sırayla MAX ve MIN olur; bir katman aşağı inmek bir **yarım hamle**dir (*ply*), yani bir oyuncunun tek hamlesidir.

Her yaprağın bir **faydası** vardır ve bu sayı MAX'ın gözünden yazılır. Oyun sıfır toplamlı olduğu için MIN'in faydası bunun eksi işaretlisidir.



Şekil 2.31: XOX oyun ağacının ilk iki katmanı. Kökte iki taş konmuş, sıra X'tedir (MAX). Kökün yedi çocuğu X'in yedi hamlesidir; her birinin altında O'nun altı cevabı vardır (MIN).

Şekil 2.31, ağacın nasıl büyüdüğünü gösterir. Kökte boş kare sayısı 7 olduğu için X'in 7 hamlesi vardır. Her hamleden sonra 6 boş kare kalır, dolayısıyla O'nun 6 cevabı vardır. Yalnız iki yarım hamle sonra kökün altında $7 \cdot 6 = 42$ konum bulunur. Aynı konuma farklı hamle sıralarıyla da ulaşılabilir: X'in önce sol üste, sonra sağ alta oynaması ile önce sağ alta, sonra sol üste oynaması aynı tahtayı verir. Ağaçta bu konum iki ayrı düğüm olarak görünür; ağaç, konumların değil oyun geçmişlerinin ağacıdır.

XOX'ta fayda üç değer alır: X kazanırsa +1, berabere 0, O kazanırsa −1 (Şekil 2.32). O'nun faydası X'inkinin eksi işaretlisidir: $u_O = -u_X$. Bu yüzden her yaprağa tek bir sayı yazmak yeter. İç düğümlerin değeri başlangıçta bilinmez; minimaks bu değerleri yapraklardan hesaplar. “Merkez iyi bir karedir” gibi bir yargı da ancak ağaç yapraklara kadar arandığında bir sayıya dönüşür: merkezin değeri, merkeze oynandıktan sonra iki tarafın en iyi oyunuyla varılan yaprağın faydasıdır.



Şekil 2.32: Üç oyun sonu ve faydaları (X'in gözünden): X kazanır +1, berabere 0, O kazanır −1.

Ağacın büyüklüğünü iki sayı belirler.

Tanım 2.20

Dallanma etkeni b (*branching factor*), bir konumdaki yasal hamle sayısıdır; oyun boyunca değiştiği için çoğu zaman ortalaması ya da en büyüğü kullanılır. **Derinlik** d, oyunun yarım hamle cinsinden uzunluğudur. Her iç düğümün b çocuğu varsa d derinlikte b^d yaprak vardır; ağacın büyüklüğü yaklaşık b^d'dir.

Bu sayı üsteldir: derinlik bir artınca iş b katına çıkar. Satrançta $b \approx 35$ olduğundan bir yarım hamle daha ileri bakmak işi yaklaşık 35 kat, bir tam hamle (iki yarım hamle) daha ileri bakmak yaklaşık 1.225 kat artırır. Tablo 2.24 dört oyunun ağacını karşılaştırır.

Oyun	b	d (yarım hamle)	Ağaç ya da konum sayısı
XOX	≤ 9	≤ 9	9! = 362.880 üst sınır; gerçekte 255.168 oyun, 549.946 düğüm
Dörtlü Bağla	≤ 7	≤ 42	4.531.985.219.092 ≈ $4{,}5 \cdot 10^{12}$ yasal konum (Tromp)
Satranç	≈ 35	≈ 80	$35^{80} \approx 10^{123}$; Shannon'un (1950) kaba tahmini ≈ 10^{120} oyun
Go (19×19)	≈ 250	≈ 150	≈ $2{,}1 \cdot 10^{170}$ yasal konum (Tromp, 2016)

Tablo 2.24: Dört oyunda dallanma etkeni, derinlik ve ağaç ya da konum sayısı.

Örnek 2.35

XOX ağacının büyüklüğü dört farklı sayıyla anlatılabilir. Dördünün farkı, ağacın neyinin sayıldığındadır.

1. **Üst sınır.** Oyun hiç erken bitmeseydi ilk hamlede 9, ikincide 8, …, sonuncuda 1 seçenek olurdu: $9 \cdot 8 \cdot \ldots \cdot 1 = 9! = 362.880$ farklı oyun.
2. **Gerçek oyun sayısı.** Oyunların bir kısmı üçlü tamamlanınca erken biter; o oyunların devamı yoktur. Bütün oyunları sayan bir program 255.168 farklı oyun bulur. Her oyun ağacın bir yaprağıdır.
3. **Düğüm sayısı.** Boş tahtadan başlayan tam bir minimaks araması yalnız yaprakları değil, yolda geçtiği her iç düğümü de ziyaret eder. Kök dâhil ziyaret edilen düğüm sayısı 549.946'dır; bunların 255.168'i yaprak, kalan 294.778'i iç düğümdür.
4. **Farklı konum sayısı.** Aynı tahtaya farklı sıralarla gelindiği için ağaçtaki 549.946 düğüm yalnız 5.478 farklı konuma karşılık gelir.

Son iki sayı arasındaki yaklaşık 100 katlık fark, aynı konumların tekrar tekrar arandığını gösterir. Bu tekrar, geçiş tablosu denen bir bellekle ortadan kaldırılabilir.

XOX'un tüm ağacı bir saniyenin altında aranır. Dörtlü Bağla'da bile tam arama ancak oyuna özgü bilgilerle ve çok güçlü programlarla yapılabilmiştir. Satranç ve Go'da ise ağacın tamamını aramak söz konusu değildir; arama belli bir derinlikte durdurulur ve orada konumun değeri tahmin edilir.

2.9.2 Minimaks değeri ve elle çözüm

Minimaks değeri, iki tarafın da en iyi oynadığı varsayımıyla MAX'ın oyunun sonunda alacağı faydadır. Tanım, bir konumun değerini çocuklarının değerlerine bağlar; bu yüzden özyinelemelidir.

Tanım 2.21

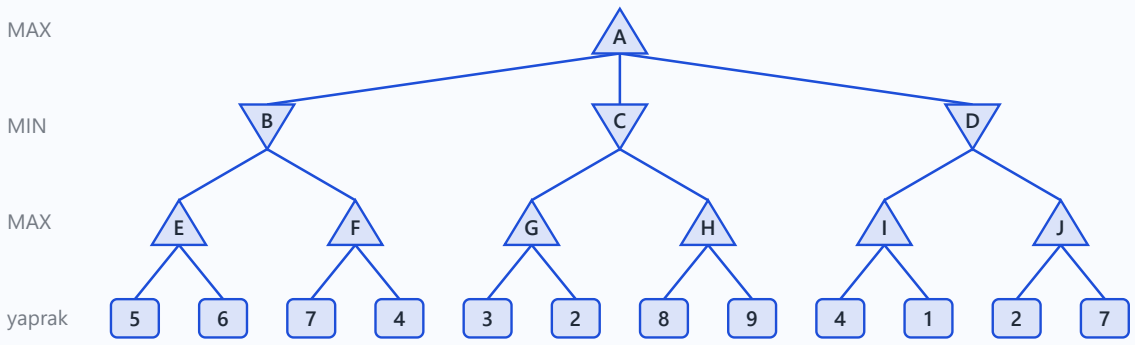
Bir *s* konumunun **minimaks değeri** şöyle tanımlanır:

```
minimaks(s) =
  fayda(s)                        s oyun sonuysa
  max  minimaks(sonuç(s, a))      sıra MAX'taysa
   a
  min  minimaks(sonuç(s, a))      sıra MIN'deyse
   a
```

Burada *a*, *s*'deki yasal hamleleri, *sonuç(s, a)* da *a* hamlesinden sonraki konumu gösterir. Kökte MAX, değeri en büyük çocuğa giden hamleyi seçer.

Tanımı uygulayan arama **derinlik öncelikli** ilerler: kökten bir yaprağa kadar iner, yaprağın faydasını okur, bir üst düğüme döner ve o düğümün öteki çocuklarına iner. Değerler bu sırada yapraktan köke doğru **yükselir**. Bir düğümün değeri ancak bütün çocukları bittiğinde kesinleşir.

Aramanın maliyeti iki kalemde ölçülür. **Zaman**: her düğüm bir kez ziyaret edildiği için $O(b^d)$. **Bellek**: aynı anda yalnız kökten o anki düğüme giden yol ve bu yol üzerindeki düğümlerin hangi çocuklarının denendiği tutulur; bu da $O(b \cdot d)$ eder. Bellek bu kadar küçük olduğu için minimaksın sınırı hiçbir zaman bellek değil, zamandır.



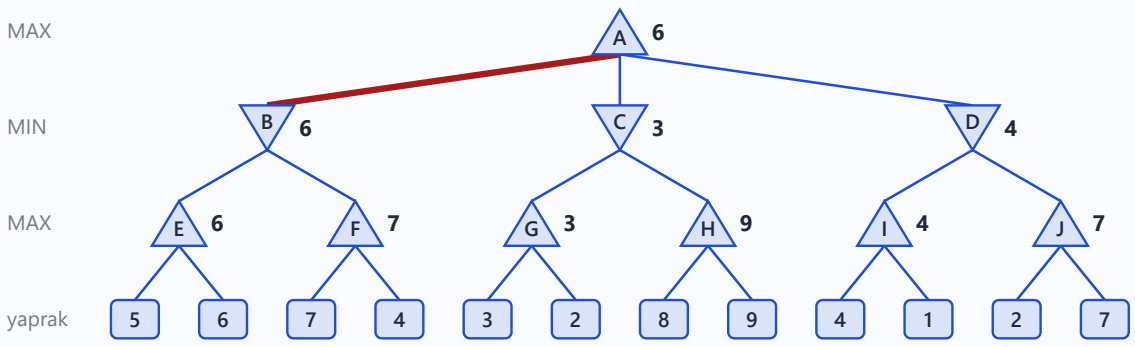
Şekil 2.33: Üç katmanlı oyun ağacı: kök A (MAX), ikinci katmanda B, C, D (MIN), üçüncü katmanda E–J (MAX) ve on iki yaprak. İç düğümlerin değerleri henüz hesaplanmamıştır.

Örnek 2.36

Şekil 2.33'teki ağaçta yapraklar soldan sağa E = [5, 6], F = [7, 4], G = [3, 2], H = [8, 9], I = [4, 1], J = [2, 7] biçiminde dağılmıştır. Kökün değeri ve MAX'ın hamlesi aşağıdan yukarı katman katman bulunur:

1. **Üçüncü katman (MAX).** Her MAX düğümü yapraklarının en büyüğünü alır: E = max(5, 6) = 6, F = max(7, 4) = 7, G = max(3, 2) = 3, H = max(8, 9) = 9, I = max(4, 1) = 4, J = max(2, 7) = 7.
2. **İkinci katman (MIN).** Her MIN düğümü çocuklarının en küçüğünü alır: B = min(E, F) = min(6, 7) = 6, C = min(G, H) = min(3, 9) = 3, D = min(I, J) = min(4, 7) = 4.
3. **Kök (MAX).** A = max(B, C, D) = max(6, 3, 4) = 6.

Kökün değeri 6'dır ve MAX, B'ye giden hamleyi seçer. Ağaçtaki en büyük yaprak (9) H'nin altındadır, ama MAX oraya ulaşamaz: C'de seçimi MIN yapar ve G'yi, yani 3'ü seçer. B'nin altındaki 7 de ulaşılamaz bir sayıdır; MIN, B'de F yerine E'yi seçer ve MAX'ı 6'da tutar. Değer 6, MAX'ın rakip ne yaparsa yapsın garanti ettiği en iyi sonuçtur.



Şekil 2.34: Minimaksın sonucu. Her iç düğümün sağında değeri yazılıdır; kalın kenar kökün seçtiği hamledir (A → B). Kökün değeri 6'dır.

Minimaks, geriye doğru çıkarımın iki kişilik, sıfır toplamlı oyunlardaki özel hâlidir. Genel bir sıralı oyunda her düğümde sırası gelen oyuncu *kendi* kazancını büyütür. Sıfır toplamlı oyunda ikinci oyuncunun kazancı birincininkinin eksi işaretlisi olduğundan "ikinci oyuncu kendi kazancını büyütür" ile "birincinin kazancını küçültür" aynı şeydir. Ağaçtaki tek sayının MAX katmanında büyütülüp MIN katmanında küçültülmesi bundan gelir.

Matris oyunlarında tanımlanan maksimin ve minimaks ölçütleriyle (Örnek 2.7) ilişki de buradan kurulur. Maksimin, MAX'ın kendi başına garanti edebileceği en iyi değerdir: $\max_s \min_t u(s, t)$. Minimaks, MIN'in MAX'ı tutabileceği en düşük değerdir: $\min_t \max_s u(s, t)$. Yazı-Tura gibi eşzamanlı oyunlarda saf stratejilerle bu iki sayı ayrışabilir. Mükemmel bilgili sonlu bir oyunda ise ikisi her zaman eşittir ve eşitlik için saf stratejiler yeter (Zermelo, 1913; Kuhn, 1953). Ağaçta hesaplanan minimaks değeri bu ortak değerdir. Sebebi, mükemmel bilgide her oyuncunun karar verirken rakibin önceki hamlelerini görmesidir: eşzamanlı oyundaki "rakip ne seçti?" belirsizliği yoktur.

Minimaks hamlesi, rakibin hatasız oynadığı varsayımıyla bulunur. Bu varsayımın getirdiği güvence ve sınırlar şöyledir:

- **Garanti ettiği:** MAX, rakip ne yaparsa yapsın en az minimaks değerini alır. Rakip hata yaparsa daha fazlasını alabilir; daha azını hiçbir zaman almaz. Bu yüzden minimaks hamlesi, en kötü durumdaki kazancı en büyük yapan hamledir.
- **Garanti etmediği:** Zayıf bir rakibe karşı en çok kazanç getiren hamleyi bulmaz. Değeri eşit iki hamleden birinin rakibe tuzak kurduğunu, ötekinin kurmadığını ayırt etmez; ikisini de aynı sayıyla görür.
- **Şans varsa:** Tavla gibi zarlı oyunlarda ağaca **şans düğümleri** eklenir. Şans düğümünde en büyük ya da en küçük değil, olasılıklarla ağırlıklı ortalama alınır (*expectiminimax*, beklenen minimaks). Bu tür ağaçlarda budama zorlaşır, çünkü bir ortalamanın sınırını bilmek için çocukların çoğunu görmek gerekir.

2.9.3 Kod izleme: minimaks fonksiyonu

Aşağıdaki kod, tanımı daha küçük bir ağaçta uygular. Kök A'nın üç MIN çocuğu vardır: B = [3, 5], C = [2, 9], D = [6, 4].

```
from math import inf
T = {"A": ["B", "C", "D"], "B": [3, 5],
     "C": [2, 9], "D": [6, 4]}
def minimaks(n, maks):
  if isinstance(n, int):        # yaprak
    return n
  deger = -inf if maks else inf
  for c in T[n]:
    v = minimaks(c, not maks)
    if maks:
      deger = max(deger, v)
    else:
      deger = min(deger, v)
  return deger
print(minimaks("A", True))
```

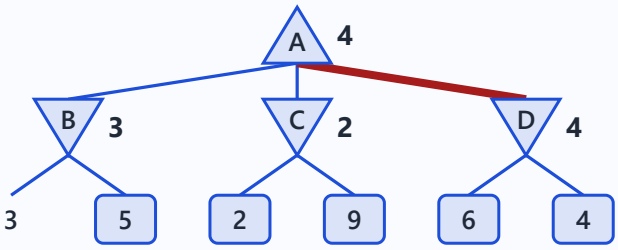
Kodun yapısı şöyledir:

- `T` sözlüğü her iç düğümün çocuklarını sırayla tutar. Yapraklar tam sayılardır; bir düğümün yaprak olup olmadığı `isinstance(n, int)` denetimiyle anlaşılır ve yaprağın faydası hemen döndürülür.
- `deger`, MAX düğümünde `-inf`, MIN düğümünde `inf` ile başlar. Sonsuzluklar "henüz hiçbir çocuk görülmedi" anlamındadır: ilk çocuğun değeri ne olursa olsun MAX için $-\infty$'dan büyük, MIN için $+\infty$'dan küçüktür ve `deger`'in yerini alır.
- Döngü çocukları `T`'deki sırayla dener. Çocukta sıra öteki oyuncudadır; özyinelemeli çağrının `not maks` ile yapılmasının sebebi budur.
- MAX düğümü `max`, MIN düğümü `min` ile günceller. Döngü bitince düğümün kesin değeri döndürülür.

Adım	Yığın	Olay	deger
1	A	kök A (MAX) açılır	$-\infty$
2	A ▸ B	B (MIN) açılır	$+\infty$
3	A ▸ B	yaprak 3: min($+\infty$, 3)	3
4	A ▸ B	yaprak 5: min(3, 5); B biter, 3 döner	3
5	A	B = 3: max($-\infty$, 3)	3
6	A ▸ C	C açılır; yapraklar 2 ve 9: min($+\infty$, 2) = 2, min(2, 9) = 2	2
7	A	C = 2: max(3, 2)	3
8	A ▸ D	D açılır; yapraklar 6 ve 4: min($+\infty$, 6) = 6, min(6, 4) = 4	4

Adım	Yığın	Olay	deger
9	A	D = 4: max(3, 4); A biter	4

Tablo 2.25: `minimaks("A", True)` izlemesi. Yığın, o anda açık olan çağrılardır.



Şekil 2.35: Kod izlemesinin sonu. B = 3, C = 2, D = 4; kök A = 4 ve A, D'ye giden hamleyi seçer. Altı yaprağın hepsi okunmuştur.

Program `4` yazdırır. İzlemede iki nokta öğreticidir. Birincisi, kökün `deger`'i adım adım 3'ten 4'e çıkar: A'nın değeri, son çocuk D bitene kadar kesin değildir. İkincisi, C'nin ikinci yaprağı 9'un okunması gereksizdir. C'nin ilk yaprağı 2 görüldüğünde C'nin değeri en fazla 2 olacaktır; A ise B'den 3'ü zaten garanti etmiştir. C, 9 ne olursa olsun seçilmez. Minimaks bunu bilmez ve altı yaprağın hepsini okur. Alfa-beta budaması tam olarak bu tür okumaları atlar.

2.9.4 Nim: minimaks, P ve N konumları, XOR kuralı

Nim'de konumlar P ve N olarak etiketlenir (Tanım 2.15): N konumunda sıradaki oyuncu, P konumunda bir önceki oyuncu kazanır. Bu etiketler minimaks değerinin başka bir yazılışıdır. Bunu görmek için değer, MAX'ın değil *sıradaki oyuncunun* gözünden yazılır: sıradaki kazanırsa +1, kaybederse −1. Bir hamleden sonra sıra rakibe geçtiği için çocuğun değeri rakibin gözündendir ve işareti çevrilerek okunur:

```
değer(s) = -1                              hamle yoksa
değer(s) = max  -değer(sonuç(s, a))        öteki durumlarda
           a
```

Bu, minimaksın tek bir fonksiyonla yazılışıdır ve **negamaks** (*negamax*) adını taşır. MIN düğümünde en küçüğü seçmek, değerin eksi işaretlisinde en büyüğü seçmekle aynıdır; negamaks iki ayrı dalı bu gözlemle birleştirir.

Formül P ve N kurallarını kelimesi kelimesine tekrarlar. Hamlesi olmayan konum −1, yani P'dir. Bir konumun değeri, ancak ve ancak en az bir çocuğun değeri −1 ise +1'dir; yani en az bir hamle bir P konumuna gidiyorsa konum N'dir. Bütün çocuklar +1 ise değer −1'dir; yani her hamle bir N konumuna gidiyorsa konum P'dir. Sonuç olarak N ⇔ +1 ve P ⇔ −1 eşlemesi geçerlidir.

Konum	Sıradakine değer	Tür
(0, 0)	−1	P
(1, 0)	+1	N
(1, 1)	−1	P
(2, 0)	+1	N
(1, 2)	+1	N
(2, 2)	−1	P
(1, 2, 3)	−1	P

Tablo 2.26: Bazı Nim konumlarının sıradaki oyuncuya göre minimaks değeri ve türü.

Örnek 2.37

Nim(1, 3) konumunun değeri negamaksla bulunur. Konumdan dört hamle vardır: birinci yığından tek jetonu almak (0, 3) ya da ikinci yığından 1, 2 veya 3 jeton almak: (1, 2), (1, 1), (1, 0).

1. (1, 1): iki hamle de (0, 1) ya da (1, 0) bırakır. Bunlardan (0, 0)'a gidilir ve (0, 0) = −1 olduğundan (0, 1) = (1, 0) = +1'dir. (1, 1) = max(−(+1), −(+1)) = −1: P.
2. (1, 2) = +1, çünkü (1, 1)'e gidebilir: −(−1) = +1. (0, 3) = +1, çünkü üç jetonu birden alıp (0, 0) bırakabilir. (1, 0) = +1.
3. (1, 3) = max(−(0, 3), −(1, 2), −(1, 1), −(1, 0)) = max(−1, −1, +1, −1) = +1.

Konum N'dir ve tek kazandıran hamle ikinci yığından iki jeton alıp (1, 1) bırakmaktır. Nim toplamı $1 \oplus 3 = 2$'dir; toplamı 0 yapan hamle de $3 \oplus 2 = 1$, yani ikinci yığını 1'e indirmektir. İki yöntem aynı hamleyi bulur. Tam ağaç 28 düğüm içerir.

Bouton teoremine göre (Teorem 2.2) bir Nim konumu ancak ve ancak nim toplamı 0 ise P'dir ve kazandıran hamle toplamı 0 yapan hamledir. Minimaks ise bunu hiçbir kuram bilmeden, ağacın tamamını arayarak bulur. Tablo 2.27 iki yöntemi beş konumda karşılaştırır.

Konum	XOR	Minimaksın kazandıran hamlesi	XOR'u 0 yapan hamle	Tam ağaç
(1, 2)	3	→ (1, 1)	→ (1, 1)	12 düğüm
(1, 3)	2	→ (1, 1)	→ (1, 1)	28 düğüm
(2, 3)	1	→ (2, 2)	→ (2, 2)	86 düğüm
(1, 2, 3)	0	yok: P konumu	yok	447 düğüm
(3, 4, 5)	2	→ (1, 4, 5)	→ (1, 4, 5)	1.038.768 düğüm

Tablo 2.27: Minimaksın tam ağaçta bulduğu hamle ile XOR kuralının hamlesi. Son sütun, tam ağacın kök dâhil düğüm sayısıdır.

İki yöntem her satırda aynı hamleyi bulur; fark maliyettedir. Minimaksın ağacı jeton sayısıyla üstel büyür: (1, 2)'de 12 düğüm olan ağaç, toplam 12 jetonluk (3, 4, 5)'te bir milyonu aşar. XOR kuralı ise yığın sayısı kadar işlemle biter. Bir oyuna özgü kuram bulunduğunda aramaya gerek kalmaz; böyle bir kuram yoksa minimaks her oyunda çalışan genel yöntemdir.

2.9.5 XOX'ta değer ölçeği ve derinlik cezası

XOX'ta +1 / 0 / −1 ölçeği oyunun sonucunu doğru verir, ama bir eksiği vardır: hızlı bir kazanç ile yavaş bir kazanç aynı değeri alır. Şekil 2.36'daki konumda sıra X'tedir. X sol alt kareye oynarsa sol sütunu tamamlar ve hemen kazanır. Merkeze oynarsa iki ayrı tehdit kurar; O birini kapatır, X ötekini tamamlar ve iki hamle sonra kazanır. İki hamlenin değeri de +1'dir.



Şekil 2.36: Aynı XOX konumu iki ölçekle. ±1 ölçeğinde merkez ve sol alt eşittir (+1); derinlik cezasıyla sol alt +9 (hemen kazanç), merkez +7 (iki hamle sonra kazanç) alır. Öteki kareler iki ölçekte de kaybettirir.

Değerler eşitse seçimi, karelerin hangi sırayla denendiği belirler: program eşitlikte ilk bulduğu kareyi tutar. Merkez önce deneniyorsa program hazır kazancı bırakıp oyunu uzatır. Sonuç değişmez, X yine kazanır; ama oyun gereksiz yere uzar ve hatasız olmayan bir programda yanlışa yer açılır. Kaybeden tarafta durum daha kötüdür: bütün hamleler −1 aldığında program en çabuk kaybettiren hamleyi seçebilir ve rakibe hata yapma fırsatı bile bırakmaz.

Çözüm, faydaya oyunun ne kadar sürdüğünü katmaktır. Kökten itibaren yapılan hamle sayısı *d* olsun. Yaprakların faydası şöyle tanımlanır:

```
X kazanırsa   +(10 − d)
O kazanırsa   −(10 − d)
berabere       0
```

Buna **derinlik cezası** denir. Kazanç ne kadar erken gelirse değer o kadar büyüktür: kökten bir hamle sonra kazanç +9, üç hamle sonra kazanç +7'dir ve +9 > +7 olduğundan hızlı kazanç seçilir. Kayıp ne kadar geç gelirse değer o kadar büyüktür: iki hamle sonra kayıp −8, dört hamle sonra kayıp −6'dır ve

$-6 > -8$ olduğundan program kaybı geciktirir. XOX'ta oyun en çok 9 hamle sürdüğü için $10 - d$ her zaman pozitiftir; kazanç ile kayıp hiçbir zaman beraberliğin (0) öbür tarafına geçmez. Ceza, sonucun türünü değil yalnız aynı türden sonuçların sırasını değiştirir.

Örnek 2.38

Şekil 2.36'daki konumda X sol üstte ve sol ortada, O üst ortada ve alt ortadadır; sıra X'tedir. Beş boş karenin değerleri iki ölçekte de hesaplanır:

1. **Sol alt.** Sol sütun tamamlanır, X kökten bir hamle sonra kazanır: ±1 ölçeğinde +1, ceza ile $10 - 1 = +9$.
2. **Merkez.** X'in iki tehdidi oluşur: sol sütun (sol alt) ve çapraz (sol üst – merkez – sağ alt). O birini kapatır, X üçüncü hamlesinde ötekini tamamlar: +1 ya da $10 - 3 = +7$.
3. **Sağ üst, sağ orta, sağ alt.** O'nun da bir tehdidi vardır: üst orta ve alt orta O'dadır, merkez boştur. X bu üç kareden birine oynarsa O merkeze oynar ve orta sütunu tamamlar. O kökten iki hamle sonra kazanır: ±1 ölçeğinde −1, ceza ile $-(10 - 2) = -8$.

±1 ölçeğinde sol alt ve merkez eşittir; derinlik cezasıyla sol alt açıkça önde gelir. Program hemen kazanır.

2.9.6 Hamlenin gerekçesi ve boş tahtanın değeri

Minimaks yalnız sayılarla seçer; seçimin neden yapıldığını söylemez. Seçilen hamlenin gerekçesi, hamlenin ne işe yaradığı sırayla sorularak bulunabilir:

1. Hamle bir üçlüyü tamamlıyor mu? Öyleyse hamle **kazanır**.
2. Rakibin bir hamlede tamamlayacağı bir üçlüyü kapatıyor mu? Öyleyse hamle **engeller**.
3. Aynı anda iki ayrı kazanma tehdidi kuruyor mu? Öyleyse hamle **çatal kurar**: rakip iki tehdidi birden kapatamaz.
4. Bunların hiçbiri değilse gerekçeyi değerin işareti söyler: pozitifse kazanmayı zorlar, sıfırsa **beraberliği tutar**, negatifse kaybı geciktirir.



Şekil 2.37: Dört konum ve seçilen hamlenin gerekçesi (değerler X'in gözünden, derinlik cezasıyla). Soldan sağa: X üçlüyü tamamlayıp kazanır (+9); O tek berabere tutan kareyle tehdidi engeller (0, öteki kareler +8); X çatal kurar (+7); köşe açılışına O merkezle cevap verip beraberliği tutar (0, öteki yedi kare +4).

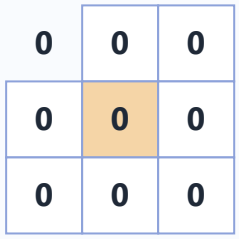
Örnek 2.39

Şekil 2.37'deki orta iki konum ayrıntılı incelenir.

1. **Engel.** X sol üstte ve üst ortada, O merkezdedir; sıra O'dadır. X'in üst satırda bir tehdidi vardır: sağ üst boş kalırsa X onu doldurup kazanır. O'nun sağ üste oynaması tehdidi kapatır ve oyunun değerini 0'da tutar. Öteki beş karenin her biri +8 alır: O sağ üstü bırakırsa X bir sonraki hamlesinde, yani kökten iki hamle sonra kazanır (10 − 2 = 8). O, MIN olduğu için değerlerin en küçüğünü, 0'ı seçer.
2. **Çatal.** X sol üstte ve merkezde, O üst ortada ve sağ alttadır; sıra X'tedir. Sol orta ve sol alt +7, öteki kareler 0 alır. X sol alta oynarsa sol sütunda (sol üst – sol orta – sol alt) ve çaprazda (sol alt – merkez – sağ üst) iki tehdit birden oluşur. O yalnız birini kapatabilir; X üçüncü hamlesinde ötekini tamamlar: 10 − 3 = +7.

“Önce kazan, sonra engelle, sonra çatal kur” gibi kurallarla da iyi bir XOX oyuncusu yazılabilir. Minimaks bu kuralların hiçbirini bilmez; hepsini ağacı arayarak kendisi bulur. Kurallar oyuna özgüdür ve başka bir oyuna taşınmaz; aynı minimaks programı ise Dörtlü Bağla'da da çalışır.

Boş tahtadan yapılan tam bir minimaks araması, X'in dokuz ilk hamlesinin hepsine 0 değerini verir (Şekil 2.38). Hiçbir açılış kazancı zorlamaz: iki taraf da hatasız oynarsa oyun berabere biter. Bütün değerler eşit olduğu için X'in hangi kareyle açtığını değerler değil, karelerin deneniş sırası belirler. Merkez, köşeler, kenarlar sırası kullanılırsa program merkezle açar.

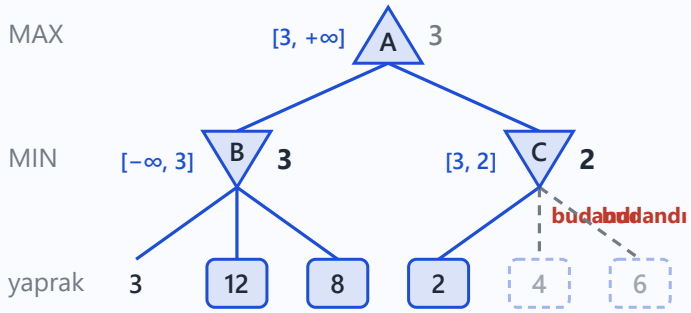


Şekil 2.38: Boş XOX tahtası: dokuz ilk hamlenin hepsinin minimaks değeri 0'dır.

Beraberliği tutmak iki taraf için aynı zorlukta değildir. X bir köşeyle açarsa O'nun sekiz cevabından yalnız biri, merkez, oyunu berabere tutar; öteki yedi cevabın her birinin değeri +4'tür (Şekil 2.37, en sağdaki konum). Değer +4, X'in kökten altı hamle sonra, yani dördüncü X taşıyla kazandığını gösterir: *10 − 6 = 4*. Köşe açılışına kenarla ya da başka bir köşeyle cevap veren O, X doğru oynadıkça kaçınılmaz olarak bir çatala düşer.

2.9.7 Alfa-beta budaması: α, β ve budama koşulu

Kod izlemesinde C'nin ikinci yaprağının gereksiz yere okunduğu görüldü. Aynı durum her ağaçta ortaya çıkar. Şekil 2.39'daki ağaçta A (MAX) ilk çocuğu B'den 3'ü garanti etmiştir: B = min(3, 12, 8) = 3. C'nin ilk yaprağı 2'dir. C bir MIN düğümü olduğundan değeri en fazla 2 olacaktır: MIN, 2'den daha büyüğünü seçmek için bir sebep bulamaz. A ise elinde 3 varken en fazla 2 veren C'yi hiçbir zaman seçmez. C'nin kalan yaprakları, 4 ve 6, ne olursa olsun kökün değeri değişmez. Bu iki yaprak hiç okunmaz; **budanır**.



Şekil 2.39: MIN düğümünde budama. A, B'den 3'ü garanti eder. C'nin ilk yaprağı 2 görülünce C'nin değeri en fazla 2'dir; A, C'yi seçmez ve C'nin kalan yaprakları (4 ve 6) budanır. Düğümlerin solunda [α, β] aralığı yazılıdır.

Küçük bir ağaçta iki yaprağı atlamak önemsiz görünebilir. Ama gerçek oyunlarda bir "yaprak" çoğu zaman kendisi derin bir alt ağaçtır: satrançta budanan tek bir dal milyonlarca konumu aramadan kurtarır. **Alfa-beta budaması** (*alpha–beta pruning*) bu gözlemi iki sayıyla sistemli hâle getirir.

Tanım 2.22

Arama sırasında kökten bir düğüme giden yol üzerinde:

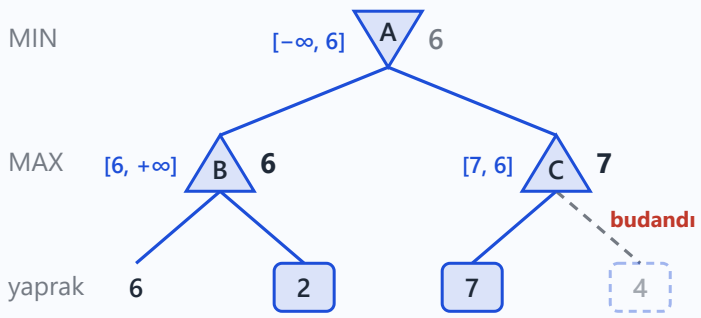
- **α (alfa)**, MAX'ın bu yol üzerinde *şimdiden garanti ettiği* en iyi değerdir: MAX'ın alt sınırı. Başlangıçta −∞'dur ve yalnız MAX düğümlerinde büyür: $\alpha = \max(\alpha, \text{değer})$.
- **β (beta)**, MIN'in aynı yol üzerinde şimdiden garanti ettiği en iyi (en küçük) değerdir: MIN'in üst sınırı. Başlangıçta +∞'dur ve yalnız MIN düğümlerinde küçülür: $\beta = \min(\beta, \text{değer})$.

Her düğüm [α, β] aralığını ebeveyninden alır ve çocuklarına aynen aktarır; ebeveyne geri dönen yalnız düğümün değeridir. Bir düğümde $\alpha \geq \beta$ olduğunda düğümün kalan çocukları aranmaz: **budama koşulu**.

Aralığın anlamı şudur: bir düğümün değeri [α, β] aralığının dışına düşerse kökün sonucunu etkilemez. Değer α'dan küçükse yukarıdaki bir MAX düğümü bu dalı seçmez, çünkü α'yı başka bir daldan zaten garanti etmiştir. Değer β'dan büyükse yukarıdaki bir MIN düğümü bu dalı seçmez, çünkü β'yı

başka bir daldan zaten garanti etmiştir. Aralık boşaldığında, yani $\alpha \geq \beta$ olduğunda, düğümün değerinin "ilginç" olabileceği bir bölge kalmamıştır.

Şekil 2.39'da C'ye gelen aralık [3, +∞]'dur: α = 3, A'nın B'den garanti ettiği değerdir. C ilk yaprağı okuyunca β = min(+∞, 2) = 2 olur. Artık α = 3 ≥ β = 2'dir ve C'nin kalan çocukları budanır. Budama MAX düğümünde de aynı koşulla olur. Şekil 2.40'taki ağaçta kök MIN'dir ve B'den 6'yı garanti etmiştir; kökte β = 6 olur ve C'ye [−∞, 6] aralığı gelir. C bir MAX düğümüdür ve ilk yaprağı 7'dir: α = 7 ≥ β = 6. C en az 7 verir; kök MIN ise 6'yı zaten garanti ettiği için C'yi seçmez. C'nin ikinci yaprağı 4 budanır.



Şekil 2.40: MAX düğümünde budama. Kök MIN, B'den 6'yı garanti eder (β = 6). C bir MAX düğümüdür; ilk yaprağı 7 görülünce α = 7 ≥ β = 6 olur. C en az 7 verir, MIN C'yi seçmez ve yaprak 4 budanır.

	MIN düğümünde	MAX düğümünde
Güncellenen	β = min(β, değer)	α = max(α, değer)
Budama	β ≤ α olunca	α ≥ β olunca
Anlamı	bu düğüm en fazla β verir; yukarıdaki MAX α'yı zaten garanti ediyor	bu düğüm en az α verir; yukarıdaki MIN β'yı zaten garanti ediyor
Döndürülen	bir üst sınır (gerçek değer ≤ β)	bir alt sınır (gerçek değer ≥ α)

Tablo 2.28: MIN ve MAX düğümünde budama. İki durum aynı koşuldur: α ≥ β.

Bazı kitaplar MIN düğümündeki kesmeye *α kesmesi*, MAX düğümündekine *β kesmesi* der: adlandırma, kesmeye sebep olan sınırın hangisi olduğuna göredir. İki kesme de aynı eşitsizlikle denetlenir; bu yüzden kodda tek bir `if a >= b` satırı yeter.

2.9.8 Budama sonucu neden değiştirmez?

Alfa-beta, minimaksla aynı kök değerini ve aynı hamleyi bulur; yalnız daha az yaprak okur. Bu iddianın doğruluğu, iç düğümlerin döndürdüğü sayıların ne anlama geldiği bilinerek gösterilir. Budanan bir düğümün döndürdüğü sayı gerçek minimaks değeri olmayabilir; ama gerçek değerle karşılaştırıldığında hangi tarafta durduğu bellidir.

Teorem 2.3

Bir düğümün gerçek minimaks değeri v, alfa-betanın [α, β] aralığıyla çağrıldığında döndürdüğü değer r olsun ($\alpha < \beta$). O zaman:

- $\alpha < v < \beta$ ise $r = v$ (kesin değer);
- $v \le \alpha$ ise $r \le \alpha$ (yalnız "yeterince kötü" olduğu bilinir);
- $v \ge \beta$ ise $r \ge \beta$ (yalnız "yeterince iyi" olduğu bilinir).

Özel olarak kök $(-\infty, +\infty)$ aralığıyla çağrıldığında döndürülen değer kökün minimaks değeridir.

İspat. İspat, düğümün yüksekliğine göre tümevarımla yapılır. Yaprakta $r = v$'dir ve üç durum da sağlanır. Bir MAX düğümü ele alınsın (MIN düğümü simetriktir). Düğüm çocuklarını sırayla dener ve her çocuktan sonra α'yı o ana kadarki en büyük döndürülen değerle günceller.

Budama olmadığı sürece her çocuk o anki $[\alpha', \beta]$ aralığıyla çağrılır ($\alpha' \ge \alpha$). Tümevarım varsayımına göre bir çocuğun gerçek değeri α"den büyük ve β'dan küçükse döndürülen değer kesindir; gerçek değer α"yü aşmıyorsa döndürülen değer de α"yü aşmaz ve düğümün en büyüğünü değiştirmez. Yani düğümün en büyüğünü belirleyebilecek her çocuk kesin değeriyle görülür. Bütün çocuklar budamasız biterse düğümün döndürdüğü değer, gerçek değerin α'dan büyük olduğu durumda gerçek değerin kendisidir; gerçek değer α'yı aşmıyorsa döndürülen değer de aşmaz.

Budama olursa bir çocuktan dönen değer β'ya ulaşmıştır. MAX düğümünün gerçek değeri bu çocuğun gerçek değerinden küçük olamaz ve bu çocuğun gerçek değeri, tümevarım varsayımına göre β'dan küçük değildir. Demek ki $v \ge \beta$'dır ve döndürülen değer de β'dan küçük değildir: üçüncü durum sağlanır. Okunmayan çocukların değeri ne olursa olsun bu sonuç değişmez, çünkü onlar MAX düğümünün değerini yalnız artırabilir. Kökte $\alpha = -\infty$, $\beta = +\infty$ olduğundan gerçek değer her zaman aralığın içindedir ve birinci durum uygulanır. □

Teoremin günlük dildeki karşılığı şudur: budanan bir dalın gerçek değeri ne olursa olsun, ebeveynin max ya da min işleminin sonucunu değiştiremez. MAX için α'yı aşamayan bir değer α'nın yerini alamaz; MIN için β'nın altına inemeyen bir değer β'nın yerini alamaz. İç düğümlerde döndürülen sayı bir sınır olabilir, ama kökte seçilen hamle yine minimaksın seçtiği hamledir.

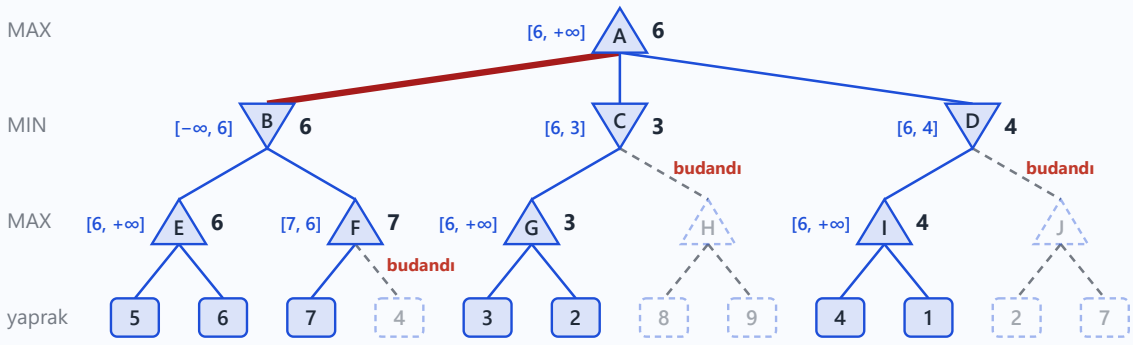
2.9.9 Alfa-beta adım adım: on iki yapraklı ağaç

Minimaksla elle çözülen on iki yapraklı ağaç (Şekil 2.33) şimdi alfa-betayla, çocuklar soldan sağa denenerek aranır. Her satır bir olaydır: bir düğüme inilmesi, bir yaprağın okunması ya da bir budama.

Adım	Düğüm	Olay	Değer	[α, β]
1	A (MAX)	arama başlar	$-\infty$	$[-\infty, +\infty]$
2	B (MIN)	aralık A'dan gelir	$+\infty$	$[-\infty, +\infty]$

Adım	Düğüm	Olay	Değer	[α, β]
3	E (MAX)	yapraklar 5 ve 6 okunur: max(5, 6)	6	[6, +∞]
4	B	E = 6: β = min(+∞, 6)	6	[−∞, 6]
5	F (MAX)	aralık B'den gelir; yaprak 7: α = 7	7	[7, 6]
6	**F**	**α = 7 ≥ β = 6: yaprak 4 budanır**	**7**	**—**
7	B	F = 7: min(6, 7); B biter	6	[−∞, 6]
8	A	B = 6: α = max(−∞, 6)	6	[6, +∞]
9	C (MIN)	aralık A'dan gelir	+∞	[6, +∞]
10	G (MAX)	yapraklar 3 ve 2 okunur: max(3, 2); α 6 kalır	3	[6, +∞]
11	C	G = 3: β = min(+∞, 3)	3	[6, 3]
12	**C**	**β = 3 ≤ α = 6: H'nin tamamı (8 ve 9) budanır**	**3**	**—**
13	A	C = 3: max(6, 3)	6	[6, +∞]
14	D (MIN)	aralık A'dan gelir	+∞	[6, +∞]
15	I (MAX)	yapraklar 4 ve 1 okunur: max(4, 1); α 6 kalır	4	[6, +∞]
16	D	I = 4: β = min(+∞, 4)	4	[6, 4]
17	**D**	**β = 4 ≤ α = 6: J'nin tamamı (2 ve 7) budanır**	**4**	**—**
18	A	D = 4: max(6, 4); arama biter	6	[6, +∞]

Tablo 2.29: On iki yapraklı ağaçta alfa-beta. [α, β] sütunu, olaydan sonra o düğümdeki aralıktır.



Şekil 2.41: Alfa-beta aramasının sonu. Düğümlerin solunda son [α, β] aralığı, sağında değeri yazılıdır. Kesik çizgiler ve soluk düğümler budanan dallardır: F'nin ikinci yaprağı, H ve J'nin tamamı. On iki yaprağın yedisi okunmuştur.

Sonuç minimaksla aynıdır: kökün değeri 6, seçilen hamle A → B. Okunan yapraklar 5, 6, 7, 3, 2, 4, 1'dir; on iki yaprağın yedisi okunur, beşi budanır. İzlemede üç noktaya dikkat etmek gerekir:

- **F'nin döndürdüğü 7 bir sınırdır.** F, 4'ü okumadan durmuştur; döndürdüğü sayı "F en az 7'dir" bilgisini taşır. Bu ağaçta F'nin gerçek değeri de 7'dir (max(7, 4) = 7), ama alfa-beta bunu bilmez ve bilmesine gerek de yoktur: B için önemli olan, F'nin 6'dan büyük olmasıdır.

- **G ve I'da α yükselmez.** G'ye [6, +∞] aralığı gelir; yaprakları 3 ve 2'dir ve *max(6, 3) = 6* olduğundan α değişmez. G'nin döndürdüğü 3, α'nın altında kalan bir değerdir: Teorem 2.3'ün "yeterince kötü" durumu. C'nin β'sı 3'e iner ve aralık boşalır.
- **Budama bütün bir alt ağacı silebilir.** C'de budanan H, tek bir yaprak değil iki yapraklı bir alt ağaçtır. Derin ağaçlarda budanan alt ağaçlar çok daha büyüktür; kazanç da buradan gelir.

2.9.10 Kod izleme: alfa-beta fonksiyonu

Minimaks koduna iki parametre ve üç satır eklemek alfa-betayı verir. Aynı altı yapraklı ağaç kullanılır.

```
from math import inf
T = {"A": ["B", "C", "D"], "B": [3, 5],
     "C": [2, 9], "D": [6, 4]}
def alfabeta(n, a, b, maks):
  if isinstance(n, int):        # yaprak
    return n
  deger = -inf if maks else inf
  for c in T[n]:
    v = alfabeta(c, a, b, not maks)
    if maks:
      deger = max(deger, v); a = max(a, deger)
    else:
      deger = min(deger, v); b = min(b, deger)
    if a >= b:
      break                     # budama
  return deger
print(alfabeta("A", -inf, inf, True))
```

Minimaksla farklar şunlardır:

- `a` ve `b` parametreleri α ve β'dır. Kök `-inf` ve `inf` ile çağrılır. Özyinelemeli çağrı, düğümün o anki `a` ve `b`'sini çocuğa aktarır.
- MAX düğümü `deger`'i güncelledikten sonra `a`'yı, MIN düğümü `b`'yi günceller. Güncellemeler yerel değişkenlerdedir; Python'da bir fonksiyon içindeki `a`'nın değişmesi çağıran fonksiyonun `a`'sını değiştirmez. Aralığın yalnız aşağıya aktarılması bu sayede kendiliğinden sağlanır.
- `if a >= b: break` satırı budamadır: kalan çocuklar denenmeden döngüden çıkılır.

Adım	Çağrı (gelen a, b)	Olay	deger	a, b (sonra)
1	A (−∞, +∞)	kök açılır	−∞	−∞, +∞
2	B (−∞, +∞)	yaprak 3: min; a < b, devam	3	−∞, 3
3	B	yaprak 5: min(3, 5); B biter, 3 döner	3	−∞, 3
4	A	B = 3: max; a = 3	3	3, +∞
5	C (3, +∞)	yaprak 2: min; b = 2	2	3, 2
6	**C**	**a = 3 ≥ b = 2: `break`, yaprak 9 okunmaz; 2 döner**	**2**	**—**

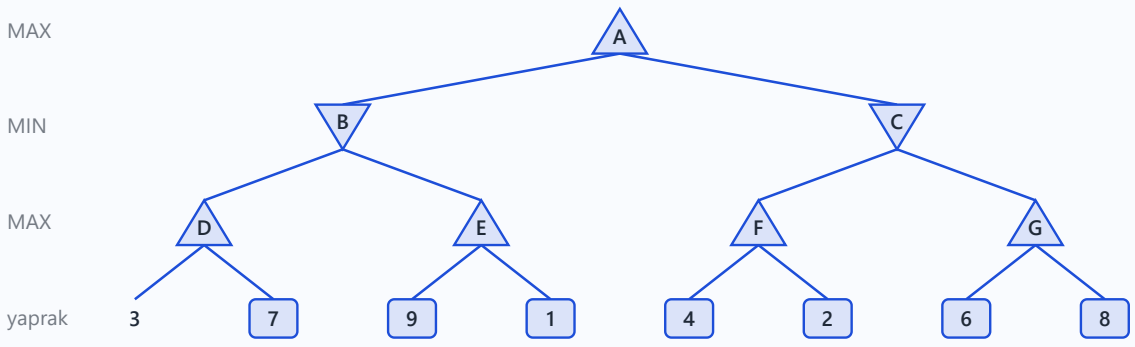
Adım	Çağrı (gelen a, b)	Olay	deger	a, b (sonra)
7	A	C = 2: max(3, 2); a 3 kalır	3	3, +∞
8	D (3, +∞)	yaprak 6: min; b = 6; a < b, devam	6	3, 6
9	D	yaprak 4: min(6, 4); b = 4; a < b; D biter, 4 döner	4	3, 4
10	A	D = 4: max(3, 4); a = 4; A biter	4	4, +∞

Tablo 2.30: `alfabeta("A", -inf, inf, True)` izlemesi.

Program yine `4` yazdırır. Altı yaprağın beşi okunur; minimaksın gereksiz yere okuduğu 9 budanmıştır. D'de budama olmaz: D'ye gelen aralık [3, +∞]'dur ve D'nin değeri 4, aralığın içindedir. D'nin değeri kesin olarak hesaplanır ve kökün değerini 3'ten 4'e çıkarır. Teorem 2.3'ün birinci durumu budur: değer aralığın içindeyse kesin değer döner.

2.9.11 Elle alfa-beta: sekiz yapraklı ağaç

Alfa-betayı elle uygularken her iç düğüm için üç şey izlenir: düğüme yukarıdan gelen aralık, düğümün o anki değeri ve aralığın boşalıp boşalmadığı. Şekil 2.42'deki ağaç bu izlemeyi baştan sona gösterir.



Şekil 2.42: Sekiz yapraklı ağaç: kök A (MAX), B ve C (MIN), D, E, F, G (MAX). Yapraklar D = [3, 7], E = [9, 1], F = [4, 2], G = [6, 8].

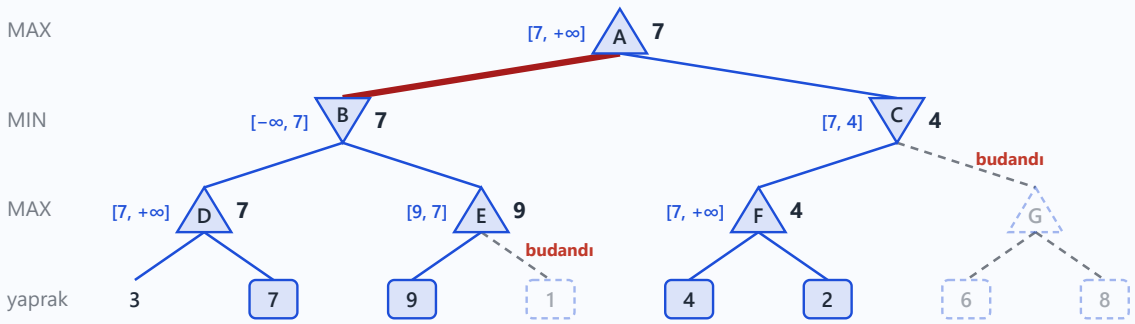
Örnek 2.40

Önce minimaksla kökün değeri bulunur: D = max(3, 7) = 7, E = max(9, 1) = 9, B = min(7, 9) = 7; F = max(4, 2) = 4, G = max(6, 8) = 8, C = min(4, 8) = 4; A = max(7, 4) = 7. Kökün değeri 7'dir ve MAX B'yi seçer. Ardından çocuklar soldan sağa denenerek alfa-beta uygulanır:

1. B'ye [−∞, +∞] gelir, D'ye de. D (MAX) 3'ü ve 7'yi okur: D = 7. B (MIN) β = min(+∞, 7) = 7 olur.
2. E'ye [−∞, 7] gelir. İlk yaprak 9'dur: α = 9 ≥ β = 7. E en az 9 verir; B, 7'yi zaten garanti ettiği için E'yi seçmez. Yaprak 1 budanır. B = min(7, 9) = 7.
3. Kökte α = 7 olur. C'ye [7, +∞] gelir, F'ye de. F (MAX) 4'ü ve 2'yi okur: F = 4 (α 7 olarak kalır, çünkü max(7, 4) = 7).

4. C'de β = min(+∞, 4) = 4 ≤ α = 7. C en fazla 4 verir; kök, 7'yi zaten garanti ettiği için C'yi seçmez. G'nin tamamı (6 ve 8) budanır.
5. Kök = max(7, 4) = 7: MAX B'yi seçer.

Sekiz yapraktan beşi (3, 7, 9, 4, 2) okunur, üçü (1, 6, 8) budanır. Sonuç minimaksla aynıdır. G'nin yaprakları arasındaki 8, C'nin gerçek değerini değiştirebilecek bir sayı değildir: C'nin değeri en fazla F'nin değeri olan 4'tür ve bu, kökün elindeki 7'nin altındadır.



Şekil 2.43: Elle alfa-beta çözümünün sonu. E'nin ikinci yaprağı (1) ve G'nin tamamı budanır; kök 7 değeriyle B'yi seçer.

Tablo 2.31 üç küçük ağaçta ve iki gerçek oyunda minimaks ile alfa-betayı karşılaştırır. Her satırda kök değeri ve seçilen hamle aynıdır; değişen yalnız okunan yaprak ya da ziyaret edilen düğüm sayısıdır. Kazanç ağaç büyüdükçe artar: küçük ağaçlarda birkaç yaprak, XOX'ta yaklaşık 70 kat, Dörtlü Bağla'da derinlik 7'de 189 kat.

Ağaç	Minimaks	Alfa-beta	Kök değeri	Budanan ya da sıra
On iki yapraklı ağaç	12 yaprak	7 yaprak	6 ve 6	F'nin 4'ü; H ve J'nin tamamı
Kod izlemedeki ağaç	6 yaprak	5 yaprak	4 ve 4	C'nin 9'u
Sekiz yapraklı ağaç	8 yaprak	5 yaprak	7 ve 7	E'nin 1'i; G'nin tamamı
XOX, boş tahta	549.946 düğüm	7.865 düğüm	0 ve 0	merkez, köşeler, kenarlar sırasıyla
Dörtlü Bağla, başlangıç, derinlik 7	960.793 düğüm	5.086 düğüm	+8 ve +8	ortadan dışa sırayla

Tablo 2.31: Aynı ağaçta minimaks ve alfa-beta.

2.9.12 Hamle sıralaması

Alfa-betanın ne kadar budayacağı, çocukların hangi sırayla denendiğine bağlıdır; kökün değeri ise sıraya bağlı değildir. Sekiz yapraklı ağaçta B'den önce C denenseydi kök, C'nin değerini kesin olarak hesaplamak zorunda kalırdı; çünkü o anda elinde bir alt sınır yoktur. En iyi çocuk önce denendiğinde α ve β erken daralır ve sonraki kardeşler çabucak elenir.

Bu etki Knuth ve Moore (1975) tarafından ölçülmüştür. Her iç düğümün b çocuğu olan d derinlikli düzgün bir ağaçta:

- **En iyi sırada** (MAX düğümünde en büyük, MIN düğümünde en küçük değerli çocuk önce) okunan yaprak sayısı $b^{\lceil d/2 \rceil} + b^{\lfloor d/2 \rfloor} - 1 \approx b^{d/2}$'dir.
- **En kötü sırada** hiç budama olmaz ve b^d yaprağın hepsi okunur.

Formülün yapısı bir gözlemden gelir. En iyi sırada, kökten en iyi çocuğa giden yol üzerindeki düğümlerin bütün çocukları açılmak zorundadır; çünkü bu düğümlerin değeri kesin olarak gerekir. Bunlara karşılık, bir kardeşin daha iyi olmadığını göstermek için o kardeşin altında tek bir "çürütücü" cevap yeter. Bu yüzden ağacın katmanlarının yarısında yalnız bir çocuk, öbür yarısında bütün çocuklar açılır ve okunan yaprak sayısı b'nin yarı üssüyle büyür. Aynı sürede yaklaşık **iki kat derinliğe** inilir; başka bir deyişle etkin dallanma etkeni b'den $\sqrt{b}$'ye iner.

d	35^d (budamasız)	En iyi sıra
2	1.225	69
4	1.500.625	2.449
6	1.838.265.625	85.749
8	$\approx 2{,}25 \cdot 10^{12}$	3.001.249

Tablo 2.32: b = 35 (satrançta ortalama hamle sayısı) için okunan yaprak sayısı.

Tablodaki sayılar formülden doğrudan hesaplanır. Örneğin $d = 4$ için $35^2 + 35^2 - 1 = 1.225 + 1.225 - 1 = 2.449$; $d = 2$ için $35 + 35 - 1 = 69$. Derinlik 8'de budamasız arama 2 trilyondan fazla yaprak okurken en iyi sıralı alfa-beta 3 milyon yaprakla yetinir. En iyi sıralı alfa-betanın derinlik 8'deki işi, budamasız aramanın derinlik 4'teki işinin iki katı kadardır.

Küçük bir ağaçta da etki görülür. Derinliği 3, dallanma etkeni 3 olan 27 yapraklı bir ağaçta en iyi sırada $3^2 + 3^1 - 1 = 11$ yaprak okunur; en kötü sırada budama hiç olmaz ve 27 yaprağın hepsi okunur. Sıra yalnız çocukların deneniş düzenini değiştirir; ağaç aynı ağaçtır ve kökün değeri her sırada aynıdır.

En iyi sırayı bilmek, zaten en iyi hamleyi bilmek demektir; pratikte sıra tahmin edilir. İyi tahminlerin üç kaynağı vardır:

- **Oyun bilgisi:** Dörtlü Bağla'da orta sütun, XOX'ta merkez ve köşeler, satrançta önce taş alan hamleler.
- **Önceki arama:** bir önceki, daha sığ aramanın en iyi bulduğu hamle önce denenir.
- **Kardeş düğümlerdeki deneyim:** başka bir dalda budamaya yol açmış bir hamle, bu dalda da önce denenir.

XOX'ta sıranın etkisi doğrudan sayılabilir. Tablo 2.33, boş tahtadan yapılan tam aramaların ziyaret ettiği düğüm sayılarını verir (derinlik cezalı ölçekle).

Yöntem	Ziyaret edilen düğüm	Kök değeri
Minimaks (sıra fark etmez)	549.946	0
Alfa-beta, kareler 0–8 sırasıyla (soldan sağa, yukarıdan aşağı)	20.866	0
Alfa-beta, önce merkez, sonra köşeler, sonra kenarlar	7.865	0
Geçiş tablolu minimaks (her konum bir kez)	5.478	0

Tablo 2.33: Boş XOX tahtasından tam arama: ziyaret edilen düğüm sayısı ve kök değeri.

İyi sıralama düğüm sayısını yaklaşık 2,7 kat azaltır (20.866 / 7.865); minimaksa göre kazanç yaklaşık 70 kattır (549.946 / 7.865). Değer ölçeği de budamayı etkiler: ±1 ölçeğiyle aynı iki alfa-beta araması 18.297 ve 7.275 düğüm ziyaret eder. Ölçek kabalaştıkça daha çok değer eşitlenir; eşit değerler $\alpha \geq \beta$ koşulunu daha sık sağladığı için budama artar. Derinlik cezası ise daha ince bir sıralama ister ve birkaç bin düğüm fazla arar; karşılığında hızlı kazancı ve geç kaybı ayırt eder.

2.9.13 Dörtlü Bağla: tam arama neden olmaz?

Dörtlü Bağla (*Connect Four*) 7 sütunlu, 6 satırlı dik bir tahtada oynanır. Oyuncu bir sütun seçer ve taşı o sütundaki en alttaki boş yere düşer. Yatay, dikey ya da çapraz dört taşını ilk dizen kazanır; tahta dolarsa oyun berabere biter. Dallanma etkeni en çok 7, oyunun uzunluğu en çok 42 yarım hamledir. John Tromp'un hesabına göre yasal konum sayısı 4.531.985.219.092, yani yaklaşık $4{,}5 \cdot 10^{12}$'dir.

Oyun 1988'de birbirinden bağımsız olarak James D. Allen ve Victor Allis tarafından çözüldü: ilk oyuncu ortadaki sütundan başlarsa kazanmayı garanti eder. Bu çözüm kaba kuvvetle değil, oyuna özgü kurallar ve çok güçlü aramalarla bulundu. Bir tarayıcıda, bir saniyede yapılabilecek bir iş değildir. Başlangıç konumunda yalnız 7 yarım hamle ileriye bakan bir minimaks bile 960.793 düğüm ziyaret eder.

Tam arama yapılamadığında arama belli bir derinlikte **durdurulur**. Durulan yerdeki konumlar oyun sonu değildir; faydaları bilinmez. Onların yerine bir **değerlendirme fonksiyonu** konumun ne kadar iyi olduğunu tahmin eder.

2.9.14 Derinlik sınırlı arama ve değerlendirme fonksiyonu

Tanım 2.23

Bir **değerlendirme fonksiyonu** (*evaluation function*) *E(s)*, oyun sonu olmayan bir konuma MAX'ın gözünden bir puan verir: konum MAX için iyi görünüyorsa büyük, kötü görünüyorsa küçük. **Derinlik sınırlı arama**, minimaksı en çok *d* yarım hamle ileriye bakacak biçimde keser:

```
H(s, d) =
  fayda(s)                    s oyun sonuysa
  E(s)                        d = 0 ise
  max  H(sonuç(s, a), d − 1)  sıra MAX'taysa
   a
  min  H(sonuç(s, a), d − 1)  sıra MIN'deyse
   a
```

Tanımdaki ilk iki satıra birlikte **kesme testi** denir: oyun bittiyse ya da derinlik sınırına gelindiyse aşağı inmek bırakılır. *E(s)* kesin bir değer değil, bir tahmindir; minimaks bu tahminleri yapraklarmış gibi kullanır ve köke kadar yükseltir. Alfa-beta da aynen çalışır: sınırdaki *E* değerleri yaprak gibi okunur.

Değerlendirme fonksiyonunun değerleri ile gerçek oyun sonlarının değerleri ayrı aralıklarda durmalıdır. Oyun sonunun değeri *E*'nin alabileceği her değerden büyük (kazançta) ya da küçük (kayıpta) olmalıdır; yoksa program "iyi görünen" bir konumu gerçek bir kazanca tercih edebilir. Burada kullanılan fayda şöyledir: Kırmızı (MAX) kökten *k*. yarım hamlede kazanırsa *+(1000 − k)*, Mavi (MIN) kazanırsa *−(1000 − k)*, tahta dolarsa 0. Değerlendirme fonksiyonu ise en fazla birkaç yüz değer alır. XOX'taki derinlik cezası burada da vardır: *k* küçüldükçe kazancın değeri büyür, kayıp geciktikçe kaybın değeri büyür.

Dörtlü Bağla için basit ve etkili bir değerlendirme fonksiyonu, tahtadaki **dörtlük pencerelere** dayanır. Pencere, yan yana dört kareden oluşan ve içinde bir dörtlü tamamlanabilecek bir yerdir. 7 × 6 tahtada 69 pencere vardır: 24 yatay (her satırda 4, altı satırda 24), 21 dikey (her sütunda 3, yedi sütunda 21) ve 24 çapraz (iki yönde 12'şer). Her pencere içindeki taşlara göre bir puan alır.

Penceredeki taşlar	p(w)
3 kırmızı + 1 boş	+5
2 kırmızı + 2 boş	+2
3 mavi + 1 boş	−5
2 mavi + 2 boş	−2

Penceredeki taşlar	p(w)
iki renk birden, tek taş ya da boş	0

Tablo 2.34: Dörtlük pencerenin puanı p(w).

Değerlendirme fonksiyonu pencere puanlarının toplamına orta sütun için bir ek puan katar:

```
E(s) = Σ p(w) + 3 · (K_orta − M_orta)
```

Burada K_{orta} ve M_{orta}, orta sütundaki kırmızı ve mavi taş sayısıdır. Orta sütunun ödüllendirilmesinin sebebi geometriktir: orta sütundaki bir kare 7 ile 13 arasında pencereye girer, kenar sütundaki bir kare ise yalnız 3 ile 5 arasında. Ortadaki bir taş daha çok dörtlünün parçası olabilir. İki renk birden içeren pencerenin puanı 0'dır, çünkü o pencerede artık kimse dörtlü kuramaz.



Şekil 2.44: Dörtlü Bağla tahtasında örnek pencereler ve vurgulu orta sütun. İki mavi ve iki boş kare içeren pencere −2, iki rengi birden içeren pencereler 0 puan alır.

Örnek 2.41

Kırmızı orta sütuna (4. sütun), Mavi onun solundaki sütuna (3. sütun), Kırmızı da orta sütunun sağına (5. sütun) oynamıştır; üç taş da alt satırdadır. *E(s)* şöyle hesaplanır:

1. **Alt satırın yatay pencereleri.** Sütun 1–4, 2–5 ve 3–6 pencereleri mavi taşı ve en az bir kırmızı taşı birlikte içerir: 0. Sütun 4–7 penceresinde iki kırmızı (4 ve 5) ve iki boş kare (6 ve 7) vardır: +2.
2. **Öteki pencereler.** Bütün dikey ve çapraz pencereler, taşlardan en çok birini içerir; tek taşlı pencere 0 alır. Yalnız mavi taşı içeren pencereler de tek taşlıdır: 0.
3. **Orta sütun.** Orta sütunda bir kırmızı taş vardır: *3 · (1 − 0) = +3*.

Toplam *E(s) = 2 + 3 = +5*'tir: konum Kırmızı için biraz iyi görünür. Aynı fonksiyon, boş tahtadan derinlik 1 ile yapılan aramanın değerini de açıklar: Kırmızı orta sütuna oynadığında tek taş hiçbir pencerede puan almaz, yalnız orta sütun ek puanı kalır ve *E = +3* olur. Kenar sütundaki bir taş ise 0 alır. Derinlik 1'de kök değeri +3'tür ve program orta sütunu seçer.

Derinlik sınırlı alfa-betanın kodu, soyut ağaçtaki koda yalnız kesme testini ve hamle üretimini ekler:

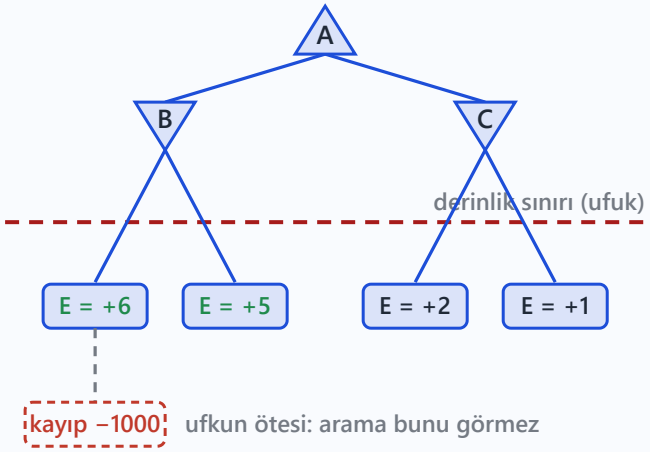
```
from math import inf
def alfabeta(s, d, a, b, maks):
  if s.bitti():                # oyun sonu
    return s.fayda()           # ±(1000 − k) ya da 0
  if d == 0:
    return E(s)                # değerlendirme
  deger = -inf if maks else inf
  for m in sirala(s.hamleler()):
    v = alfabeta(s.oyna(m), d-1, a, b, not maks)
    if maks:
      deger = max(deger, v); a = max(a, deger)
    else:
      deger = min(deger, v); b = min(b, deger)
    if a >= b:
      break
  return deger

def en_iyi_hamle(s, d):     # kök: MAX
  a, en = -inf, None
  for m in sirala(s.hamleler()):
    v = alfabeta(s.oyna(m), d-1, a, inf, False)
    if v > a:
      a, en = v, m
  return en
```

Kodun oyuna özgü parçaları beş tanedir: `bitti`, `fayda`, `hamleler`, `oyna` ve `E`. Gerisi oyundan bağımsızdır; aynı iskelet XOX, Dörtlü Bağla ya da satranç için kullanılabilir. `sirala` hamleleri denenecek sıraya koyar; Dörtlü Bağla'da bu sıra ortadan dışadır: 4, 3, 5, 2, 6, 1, 7. `en_iyi_hamle` fonksiyonu kökte yalnız değeri değil hamleyi de tutar. Kökteki α kökün çocukları arasında taşınır: ilk çocuk tam aralıkla aranır, sonrakiler daralan [α, +∞] penceresiyle aranır ve ilk çocuktan iyi olmadıkları çabucak gösterilir.

2.9.15 Ufuk etkisi

Derinlik sınırlı arama, sınırın hemen ötesindeki olayları göremez. Buna **ufuk etkisi** (*horizon effect*) denir. Şekil 2.45'teki ağaçta derinlik sınırındaki değerlendirmelere göre B'nin altındaki konum E = +6 ile en iyi görünür. Oysa o konumdan bir hamle sonra kaçınılmaz bir kayıp vardır (−1000 civarı). Arama sınırda durduğu için bu kaybı görmez ve B'yi seçer.



Şekil 2.45: Ufuk etkisi. Derinlik sınırında (kesik çizgi) yapılan değerlendirmeye göre B'nin altındaki konum E = +6 ile en iyisidir; sınırın bir hamle ötesindeki kayıp aramaya görünmez.

Ufuk etkisinin daha sinsi bir hâli de vardır: program, kaçınılmaz bir kaybı ufkun ötesine iten anlamsız hamleler yapar. Satrançta bir taşın kaybedileceğini gören program, boş yere şah çekerek ya da bir piyonu feda ederek kaybı birkaç yarım hamle erteler. Kayıp ortadan kalkmaz, yalnız arama derinliğinin dışına çıktığı için görünmez olur; üstelik feda edilen piyon da kaybedilmiştir.

Ufuk etkisinin tam bir çaresi yoktur; etkisini azaltan yöntemler vardır:

- **Derinliği artırmak:** ufuk uzaklaşır ama ortadan kalkmaz.
- **Sakinlik araması** (*quiescence search*): sınırdaki konum "hareketliyse", örneğin bir taş alımı ya da bir tehdit sürüyorsa, arama yalnız bu tür hamlelerle sürdürülür ve konum sakinleşince değerlendirilir.
- **Tehdit uzatması:** bir tarafın bir hamlede kazanma tehdidi varken arama bir katman daha iner.

Dörtlü Bağla'da derinlik 1 ya da 2 ile oynayan bir program, iki yönlü tehditleri göremez ve kolayca yenilir: rakibin iki hamle sonra kuracağı çift tehdit, programın ufkunun dışındadır. Derinlik artırıldıkça program bu tuzakları görmeye başlar. Derinliğe göre kök değerinin +3, 0, +6, 0, +6, 0, +8 diye inip çıkması da sınırlı bakışın bir sonucudur: tek derinliklerde ufuktaki son hamleyi Kırmızı, çift derinliklerde Mavi yapar ve son hamleyi yapan taraf değerlendirmede bir taş fazlasıyla görünür.

2.9.16 Sıralama, yinelemeli derinleştirme ve geçiş tablosu

Hamle sıralamasının Dörtlü Bağla'daki etkisi, başlangıç konumundan farklı derinliklerde yapılan aramalarla ölçülür (Tablo 2.35).

Derinlik	Minimaks	Alfa-beta (soldan sağa)	Alfa-beta (ortadan dışa)	Kök değeri
1	8	8	8	+3
2	57	33	21	0
3	400	200	76	+6

Derinlik	Minimaks	Alfa-beta (soldan sağa)	Alfa-beta (ortadan dışa)	Kök değeri
4	2.801	966	175	0
5	19.608	4.340	627	+6
6	137.257	16.409	1.285	0
7	960.793	67.549	5.086	+8

Tablo 2.35: Dörtlü Bağla başlangıç konumundan aramada ziyaret edilen düğüm sayısı.

Minimaks yaklaşık 7^d ile büyür: derinlik 1'de 8 düğüm (kök ve yedi çocuk), derinlik 2'de $1 + 7 + 49 = 57$, derinlik 7'de 960.793. Alfa-beta derinlik 7'de soldan sağa sırayla yaklaşık 14 kat (960.793 / 67.549), ortadan dışa sırayla 189 kat (960.793 / 5.086) daha az düğüm ziyaret eder. Kök değeri her satırda üç yöntemde de aynıdır: sıra yalnız maliyeti değiştirir. Orta sütun en çok dörtlüye girdiği için iyi hamleler çoğunlukla ortadadır; önce onlar denenince α ve β erken daralır.

Yinelemeli derinleştirme (*iterative deepening*), arama derinliğini önceden seçme sorununu çözer. Önce derinlik 1, sonra 2, 3, … ile arama yapılır; süre dolunca son tamamlanan aramanın hamlesi oynanır. İlk bakışta sığ aramaların tekrarlanması israf gibi görünür, ama tekrarlanan iş azdır: üstel büyüyen bir ağaçta son katman, öncekilerin toplamından büyüktür. $b = 7$ için bütün derinliklerin toplam işi, yalnız son derinliğin işinin yaklaşık $7/6$ katıdır:

```
7 + 7² + … + 7^d ≈ 7^d · (1 + 1/7 + 1/49 + …) = 7^d · 7/6
```

Yinelemeli derinleştirmenin ikinci kazancı sıralamadadır: bir önceki aramanın en iyi hamlesi bir sonraki aramada önce denenir. Hamle sıralaması böylece kendiliğinden iyileşir ve çoğu zaman sığ aramaların maliyeti, derin aramada kazanılan budamayla fazlasıyla geri alınır.

Geçiş tablosu (*transposition table*), aynı konumun tekrar tekrar aranmasını önler. Aynı konuma farklı hamle sıralarıyla ulaşılabilir (*transposition*). Konum bir kez aranır ve sonucu bir hash tablosuna yazılır; konumla yeniden karşılaşılınca sonuç tablodan okunur. XOX'ta tam arama böylece 549.946 düğüm yerine 5.478 konumla yetinir (Tablo 2.33). Tablodaki bir kayıt şunları içerir: değer, konumun hangi derinlikle arandığı, değerin kesin mi yoksa bir alt ya da üst sınır mı olduğu ve en iyi hamle. Sınır bilgisi alfa-beta yüzünden gereklidir: budanmış bir aramadan dönen değer kesin olmayabilir (Teorem 2.3). Satranç programları konumu, Zobrist'in (1970) önerdiği bir yöntemle tek bir sayıya, bir hash anahtarına çevirir; bir hamle bu anahtarı birkaç XOR işlemiyle günceller.

Güçlü oyun programlarının iskeleti bu parçalardan oluşur: yinelemeli derinleştirme, alfa-beta, iyi hamle sıralaması, geçiş tablosu ve iyi bir değerlendirme fonksiyonu. Gerçek sistemler bu iskeleti farklı ölçülerde kullanır:

- **Deep Blue (1997).** IBM'in satranç bilgisayarı, dünya şampiyonu Garri Kasparov'u altı partilik maçta 3½–2½ yendi. Özel satranç yongalarında paralel alfa-beta çalıştırıyor, saniyede yaklaşık 200 milyon konum değerlendiriyordu.

- **Satranç motorları.** Stockfish gibi programlar hâlâ alfa-beta türevi bir arama kullanır: yinelemeli derinleştirme, geçiş tablosu ve güçlü budama yöntemleri. 2020'den beri değerlendirme fonksiyonu bir sinir ağıdır (NNUE).
- **Go ve Monte Carlo ağaç araması.** Go'da $b \approx 250$'dir ve iyi bir $E(s)$ yazmak çok zordur. Monte Carlo ağaç araması (Coulom, 2006; UCT: Kocsis ve Szepesvári, 2006) konumları rastgele oynanan oyunlarla değerlendirir. AlphaGo bu aramayı derin sinir ağlarıyla birleştirdi ve Mart 2016'da Lee Sedol'u 4–1 yendi.

Bu sistemlerin ortak fikri, minimaksla aynıdır: rakibin en iyi cevabını varsayarak ileriye bakmak. Değişen, ağacın ne kadarının arandığı ve yaprakların nasıl değerlendirildiğidir.

2.9.17 Arama algoritmalarında sık hatalar

Sık yapılan hata

- **α ve β'yı yanlış yöne aktarmak.** Aralık ebeveynden çocuğa aynen iner; ebeveyne yalnız değer döner. Çocukta güncellenen α ya da β ebeveyne taşınmaz. MAX yalnız α'yı, MIN yalnız β'yı günceller. On iki yapraklı ağaçta F'de α = 7 olur, ama B'ye dönen yalnız F'nin değeridir; B'nin aralığı $[-\infty, 6]$ olarak kalır.
- **Budanan dalı hesaba katmak.** Budanan yaprakların değeri okunmaz; elle çözümde onları min ya da max işlemine sokmak yanlıştır. Kesilen bir düğümün döndürdüğü sayı da kesin değer değil, bir sınırdır.
- **Kök değerinin sıraya bağlı olduğunu sanmak.** Hamle sıralaması yalnız okunan yaprak sayısını değiştirir. Farklı sırada farklı bir kök değeri çıkıyorsa kodda bir hata vardır.
- **Bakış açısı ve işaret hatası.** $E(s)$ hep aynı oyuncunun gözünden yazılmalıdır (burada Kırmızı, yani MAX). Program ikinci oyuncuysa en küçüğü seçer. Negamaks yazılırsa her katmanda işaret çevrilir ve değerlendirme sıradaki oyuncunun gözünden yapılır; iki yazılış karıştırılırsa program rakibi için oynar.
- **Derinlik cezasını unutmak.** Bütün kazançlar aynı değeri alırsa program hazır kazancı erteler; kaybederken de en çabuk kaybedeni seçebilir. Fayda, kazancın kaç hamle sonra geldiğini de içermelidir.
- **Değerlendirmeyi oyun sonuyla aynı ölçekte tutmak.** $E(s)$ gerçek bir kazancın değerine ulaşabiliyorsa program iyi görünen bir konumu kesin bir kazanca tercih edebilir. Oyun sonu değerleri, değerlendirme fonksiyonunun alabileceği her değerin dışında kalmalıdır.
- **Derinlik sınırlı değeri kesin sanmak.** $H(s, d)$, d yarım hamlelik bir bakışın tahminidir. Dörtlü Bağla'da derinlik 7'nin +8'i "Kırmızı kazanır" demek değildir; oyunun gerçek sonucu ancak tam aramayla ya da bir ispatla bilinir.

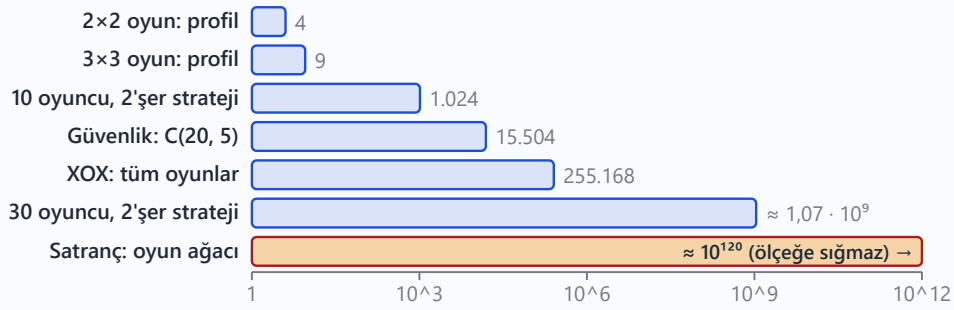
2.10 Algoritmik bakış ve mekanizma tasarımı

Buraya kadarki kavramların çoğu klasik oyun kuramına aittir. Ama bölüm boyunca bir çizgi de belirdi: minimaks, geriye doğru çıkarım ve Nim'in XOR kuralı birer algoritmadır; 2/3 oyununda eleme bir döngüdür. Oyun kuramının bilgisayar bilimiyle buluştuğu bu yer, algoritmik oyun kuramının sorularını doğurur.

2.10.1 Algoritmik oyun kuramı neden “algoritmik”?

Üç gözlem algoritmik bakışı gerekli kılar.

Büyüklük. Oyunun tanımı bile üstel olarak büyür. n oyuncunun her birinin 2 stratejisi varsa strateji profili sayısı 2^n'dir. Her profil için her oyuncunun kazancını yazmak gerektiğinden matris $n \cdot 2^n$ sayı içerir. 30 oyunculu bir oyunda profil sayısı $2^{30} \approx 1{,}07 \cdot 10^9$'dur. Bütün profilleri tek tek denetleyen kaba kuvvet yöntemi ölçeklenmez.



Şekil 2.46: Oyunların büyüklüğü logaritmik ölçekte. 2×2 oyunun 4, 3×3 oyunun 9 profili vardır; 10 oyunculu ikişer stratejili bir oyunun 1.024, 20 hedeften 5'ini korumanın C(20, 5) = 15.504, XOX'un 255.168 oyunu, 30 oyunculu ikişer stratejili bir oyunun yaklaşık $1{,}07 \cdot 10^9$ profili vardır. Satranç ağacı ($\approx 10^{120}$) ölçeğe sığmaz.

Şekil 2.46'daki çubukların her biri bir öncekinden kat kat büyüktür. Güvenlik örneğinde 20 hedeften 5'inin korunacağı her seçim bir stratejidir; seçim sayısı $C(20, 5) = 20! / (5! \cdot 15!) = 15.504$'tür. Bu sayılar, oyunların çoğunun ancak yapılarından yararlanan algoritmalarla çözülebileceğini gösterir.

Karmaşıklık. Nash'in teoremine göre her sonlu oyunda karışık stratejilerde bir denge vardır. Ama dengenin varlığını bilmek onu bulmak demek değildir. İki oyunculu oyunlarda bile dengeyi hızlı bulan bir yöntem bilinmez: problem PPAD-tam denen zor problemler sınıfındadır (Daskalakis, Goldberg ve Papadimitriou; iki oyuncu için Chen ve Deng).

Yapı ve tasarım. Özel yapı verimlilik getirir. Sıfır toplamlı oyunlar doğrusal programlamayla hızlı çözülür; Nim'de XOR, üstel sayıda konumu tek bir hesapla etiketler. Kurallar da algoritmadır: bir açık artırmada kazananı ve ödemeyi belirleyen kural bir programdır ve doğru tasarlanması gerekir.

2.10.2 Dört soru

Algoritmik oyun kuramının dört ana sorusu vardır. Bu bölümün kavramları, yani oyunların sınıflandırılması, klasik oyunların dengeleri ve sıralı oyunlarda geriye doğru çıkarım, dört sorunun da ortak dilidir.

1. **Hesaplama: denge bulunabilir mi?** Sıfır toplamlı oyunlar doğrusal programlamayla hızlı çözülür. Genel oyunlarda Nash dengesini hızlı bulan bir yöntem bilinmez; problem PPAD-tamdır. Kombinatoryal oyunlar da zor olabilir: Hex'i büyük tahtalarda çözmek PSPACE-tamdır (Reisch, 1981).
2. **Verimlilik: bencillik ne kaybettirir? Anarşinin Bedeli**, en kötü dengenin maliyetinin en iyi sonucun maliyetine oranıdır. Mahkûmun İkilemi'nde hapis yılı maliyet sayılırsa denge toplam 4 yıl, (Sessiz, Sessiz) 2 yıldır; oran 4 / 2 = 2'dir.
3. **Tasarım: hangi kurallar?** Mekanizma tasarımı, oyuncuların özel bilgisini dürüstçe bildirmesini sağlayan, yani **teşvik uyumlu** (*incentive compatible*) kuralları arar: açık artırmalar, eşleştirme, oylama.
4. **Dinamik: oyuncular dengeye nasıl ulaşır?** Tekrarlanan ve çevrim içi oyunlarda bilgi zamanla gelir; oyuncular geçmişten öğrenerek stratejilerini uyarlar. 2/3 oyununda tekrarlanan turlarda tahminlerin 0'a inmesi bunun bir örneğidir.

Örnek 2.42

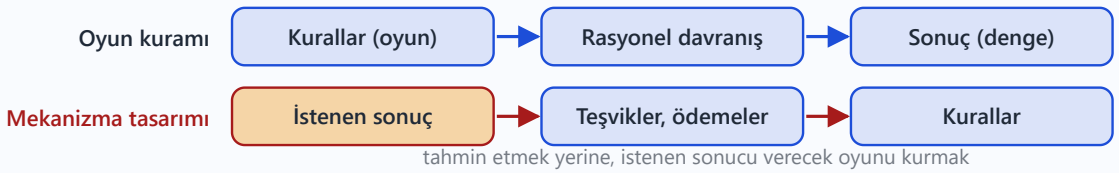
Fiyat rekabeti oyununda (Tablo 2.4) kârlar kazançtır, maliyet değildir. Bu durumda verimlilik kaybı, en iyi sonucun toplam kârının dengedeki toplam kâra oranıyla ölçülür:

1. Dengedeki toplam kâr: (100, 100) = 5 + 5 = 10.
2. Toplam kârı en büyük yapan sonuç: (120, 120) = 6 + 6 = 12. (Öteki hücrelerde 8 + 1 = 9.)
3. Oran: 12 / 10 = 1,2.

Bencil fiyatlama, şirketlerin toplam kârının yaklaşık %17'sini (12'den 10'a, 2/12) kaybettirir. Mahkûmun İkilemi'ndeki oran (2) daha büyüktür; kayıp oyunun sayılarına bağlıdır.

2.10.3 Mekanizma tasarımı: tersine oyun kuramı

Oyun kuramı verilen bir oyunun sonucunu tahmin eder: kurallar bellidir, soru ne olacağıdır. **Mekanizma tasarımı** (*mechanism design*) soruyu tersine çevirir: istenen sonuç bellidir, soru o sonucu verecek kuralların ve teşviklerin ne olduğudur. Bu yüzden mekanizma tasarımına “tersine oyun kuramı” da denir.



Şekil 2.47: Oyun kuramı ile mekanizma tasarımı. Üst sıra: kurallardan rasyonel davranışa, oradan sonuca (dengeye). Alt sıra: istenen sonuçtan teşvik ve ödemelere, oradan kurallara.

Tasarımın zorluğu şudur: ajanların özel bilgisi, örneğin bir ürüne biçtikleri değer, konumları ya da tercihleri, tasarımcıya kapalıdır. Ajanlar bu bilgiyi bildirirken yalan söyleyebilir. İyi bir mekanizma, yalan söylemeyi kârsız kılar. Vickrey'nin 1961'de gösterdiği gibi ikinci fiyat açık artırmasında dürüst teklif baskın stratejidir: kazanan, kendi teklifini değil ikinci en yüksek teklifi öder; bu yüzden teklifi şişirmek ya da kısmak kazanca bir şey eklemez.

Mekanizmalar iki gruba ayrılır:

- **Para ile:** tasarımcı ödeme alabilir ya da yapabilir; ödemeler doğru söylemeyi teşvik eder. Örnekler: açık artırmalar, reklam yerleşimi.
- **Para olmadan:** teşvikler yalnızca kuralla sağlanır; genellikle hedeften biraz ödün verilir. Örnekler: seçimler, oda takası, tesis yeri seçimi.

2.10.4 Gerçek dünya uygulamaları

Mekanizma tasarımı ve algoritmik oyun kuramı birçok gerçek sistemin içinde çalışır:

- **Arama reklamları:** reklam alanları her aramada genelleştirilmiş ikinci fiyat (*generalized second price*, GSP) açık artırmasıyla dağıtılır. Tek alanlı ikinci fiyat açık artırmasının aksine, çok alanlı GSP'de dürüst teklif baskın değildir.
- **Spektrum açık artırmaları:** devletler frekans lisanslarını tasarlanmış açık artırmalarla satar; ABD'de bu uygulama 1994'ten beri sürer.
- **Okul yerleştirme:** öğrenci önerili Gale–Shapley algoritmasında öğrencinin gerçek tercihlerini yazması en iyisidir; sonuç kararlı bir eşleştirmedir.
- **Oda ve vardiya takası:** herkes elindekini getirir; takas döngüleriyle kimse eskisinden kötü bir şey almaz ve para kullanılmaz.
- **Trafik ve yönlendirme:** her sürücü kendi en hızlı yolunu seçer; bencil seçimlerin toplam gecikmeye maliyeti Anarşinin Bedeli ile ölçülür.
- **Platformlar ve teşvikler:** yoğunluk fiyatlandırması sürücüleri yoğun saatlere çeker; Bitcoin madencileri işlemleri ödül için doğrular.

Bu uygulamaların ortak fikri şudur: kurallar öyle tasarlanır ki herkes kendi çıkarına davransa bile sistem iyi çalışır. Mahkûmun İkilemi'nde bireysel çıkar ortak kötü sonuca götürüyordu; iyi tasarlanmış bir mekanizmada ise bireysel çıkar istenen sonuca götürür.

2.11 Çözümlü problemler

Beş problem, bölümün ana yöntemlerini bir araya getirir; çözümler ara adımlarıyla verilir.

Örnek 2.43

Problem. Fiyat rekabeti oyununda (Tablo 2.4) baskın stratejiler ve denge nedir? Denge Pareto optimal midir?

Çözüm. Şirket 1 için 100 TL, Şirket 2'nin 100 demesine karşı 5 > 1, 120 demesine karşı 8 > 6 verir: 100 baskındır. Oyun simetrik olduğundan Şirket 2 için de 100 baskındır. Denge (100, 100) = (5, 5)'tir. (120, 120) = (6, 6) iki şirkete de daha çok verdiği için dengeyi Pareto baskılar; denge Pareto optimal değildir.

Örnek 2.44

Problem. 2/3 oyununda (0–90) sınıfın %20'si seviye 0, %80'i seviye 1 düşünüyorsa hangi seviye kazanır?

Çözüm. Beklenen ortalama $0{,}2 \cdot 45 + 0{,}8 \cdot 30 = 9 + 24 = 33$, hedef $2/3 \cdot 33 = 22$'dir. Uzaklıklar: seviye 0 (45) 23, seviye 1 (30) 8, seviye 2 (20) 2, seviye 3 (13,3) 8,7. Seviye 2 en yakındır. Nash cevabı 0 ise 22 uzakta kalır.

Örnek 2.45

Problem. İki oyuncu A ya da B seçer. (A, A) = (4, 1), (B, B) = (1, 4), öteki hücreler (0, 0). Saf dengeler hangileridir? Oyun hangi klasik oyuna benzer?

Çözüm. Sütun A'da satır oyuncusu A ile 4, B ile 0 alır → A. Sütun B'de A ile 0, B ile 1 → B. Satır A'da sütun oyuncusu A ile 1, B ile 0 → A. Satır B'de A ile 0, B ile 4 → B. İki işaretli hücreler (A, A) ve (B, B)'dir. Biri satır oyuncusunu (4 > 1), öteki sütun oyuncusunu (4 > 1) kayırır. Buluşmak ortak çıkar, yer çatışmadır: Cinsiyetler Savaşı yapısı.

Örnek 2.46

Problem. MAX kökünün üç MIN çocuğu var: B = [7, 2, 9], C = [4, 6], D = [1, 8]. Oyunun değeri ve MAX'ın seçimi nedir?

Çözüm. MIN değerleri: B = min(7, 2, 9) = 2, C = min(4, 6) = 4, D = min(1, 8) = 1. Kök MAX = max(2, 4, 1) = 4; MAX C'yi seçer. B'deki 9 ve D'deki 8 yanıltıcıdır: MIN onlara izin vermez.

Örnek 2.47

Problem. Piyasaya giriş oyununda (Girme, Savaş) neden alt oyun mükemmel değildir?

Çözüm. G girdikten sonra başlayan alt oyunda M için Savaş −1, Uzlaş 1 verir. Savaş bu alt oyunda en iyi tepki değildir; dolayısıyla profil o alt oyunda bir Nash dengesi oluşturmaz. "Girersen savaşırım" inandırıcı olmayan bir tehdittir. Profil bütün oyunda bir Nash dengesidir, ama her alt oyunda değildir.

2.12 Özet

- Bir **oyun**, oyuncular, stratejiler, kazançlar, bilgi ve kurallardan oluşur. **Rasyonel** oyuncu inancına göre beklenen faydasını maksimize eder; rasyonel, bencil demek değildir. **Ortak bilgi**, "herkes biliyor" zincirinin sonsuza dek sürmesidir.
- **Strateji**, her karar düğümünde ne yapılacağını söyleyen tam plandır; k düğümde m'şer eylem m^k strateji verir. **Karışık strateji**, saf stratejiler üzerinde olasılık dağılımıdır.
- **Sıralı fayda** yalnız sırayı, **kardinal fayda** farkların büyüklüğünü de taşır. Piyango ve karışık strateji hesapları kardinal fayda ister; $a \cdot u + b$ ($a > 0$) dönüşümü kararları korur.
- **Normal form** $G = (N, (S_i), (u_i))$ eşzamanlı oyunları matrisle; **genişletilmiş form** sıralı oyunları ağaçla yazar. Bilgi kümesi, eşzamanlı oyunu ağaçta gösterir.
- Oyunlar altı boyutta sınıflandırılır: zamanlama, anlaşma, toplam kazanç, simetri, bilgi ve ufuk. **Mükemmel bilgi** geçmiş hamlelerle, **tam bilgi** kazanç fonksiyonlarıyla ilgilidir.
- **Nash dengesi**, kimsenin tek taraflı sapmak istemediği profildir. Mahkûmun İkilemi'nde ihanet baskındır ve tek denge (−2, −2) **Pareto optimal** değildir: kararlılık verimlilik değildir.
- **Altı klasik 2×2 oyun** çatışmadan koordinasyona uzanır: Yazı-Tura'da saf denge yoktur; Tavuk, Cinsiyetler Savaşı, Geyik Avı ve Saf Koordinasyon'da iki saf denge vardır; Bedava Para'da biri zayıftır.
- **2/3 oyununda** yinelemeli eleme 90 → 60 → 40 → … → 0 ile tek dengeyi (herkes 0) verir; gerçek oyuncular 1–2 adım düşünür ve en iyi tahmin inanca bağlıdır (**seviye-k**).
- **Minimaks** değerleri yapraktan köke hesaplar; XOX berabere, örnek ağacın değeri 3'tür. **Nim**'de konum, nim toplamı 0 ise kaybedendir (Bouton); 21'i Söyleme'de kontrol sayıları 4'ün katlarıdır.
- **Alt oyun mükemmel denge** her alt oyunda Nash dengesidir ve geriye doğru çıkarımla bulunur; inandırıcı olmayan tehditleri eler (piyasaya giriş, yatırım, ültimatom). Kırkayakta SPE (1, 0)'dır.
- **Oyun ağacı** yaklaşık b^d büyüklüğe ulaşır. **Minimaks** değeri yapraktan köke yükselir: $O(b^d)$ zaman, $O(b \cdot d)$ bellek. Nim'de minimaks, P/N etiketleri ve XOR kuralı aynı hamleyi bulur (**negamaks**); XOX'ta boş tahtanın değeri 0'dır ve **derinlik cezası** $\pm(10 - d)$ hızlı kazancı ve geç kaybı seçtirir.
- **Alfa-beta budaması** [α, β] aralığını yukarıdan aşağı aktarır ve $\alpha \geq \beta$ olunca kalan çocukları atlar; kök değeri ve seçilen hamle değişmez (on iki yapraklı örnekte 12 yerine 7 yaprak). En iyi **hamle sıralamasıyla** yaklaşık $b^{d/2}$ yaprak okunur (Knuth–Moore, 1975); XOX'ta 549.946 yerine 7.865

düğüm yeter.

- Tam arama yapılamayan oyunlarda arama derinlik sınırında durur ve konumu bir **değerlendirme fonksiyonu** tahmin eder. Sınırın ötesini görememek **ufuk etkisidir**; **yinelemeli derinleştirme** ve **geçiş tablosu** aramayı hızlandırır. Dörtlü Bağla'da derinlik 7'de ortadan dışa sıralı alfa-beta, minimaksa göre 189 kat az düğüm ziyaret eder.
- **Kinci tetik stratejisi** sonsuz tekrarda ihaneti kalıcı cezayla caydırır; Sam–John kazançlarıyla $\delta \geq 1/2$ iken işbirliği dengededir. Sonlu ve bilinen ufukta işbirliği çözülür.
- Algoritmik oyun kuramı dört soru sorar: hesaplama, verimlilik (Anarşinin Bedeli), tasarım (mekanizma tasarımı) ve dinamik.

Kavram	Kısa tanım	Örnek
Normal / genişletilmiş form	Matris (eşzamanlı) ya da sırayı ve bilgiyi gösteren oyun ağacı	TKM, fiyat rekabeti; satranç
Sınıflandırma	Zamanlama, anlaşma, toplam kazanç, simetri, bilgi, ufuk	Satranç: sıralı, sıfır toplamlı, mükemmel bilgili
Baskın strateji, Nash dengesi	Karşı taraf ne yaparsa yapsın daha iyi; tek taraflı sapma kârsız	İhanet; (İhanet, İhanet)
Pareto optimallik	Kimseyi kötüleştirmeden iyileştirilemez	Denge (−2, −2) değil
Klasik 2×2 oyunlar	Çatışma ya da koordinasyon; saf dengeler en iyi tepkiyle bulunur	Yazı-Tura: yok; Geyik Avı: iki
Yinelemeli eleme, seviye-k	Baskılananları tekrar tekrar silmek; inanca göre en iyi tahmin	2/3 oyunu: 90 → 60 → 40 → … → 0
Minimaks, Nim	Yapraktan köke en iyi seçim; P/N konumları	Ağaç değeri 3; XOX berabere; Nim XOR
Alfa-beta, derinlik sınırlı arama	Sonucu etkilemeyecek dalları [α, β] ile atlamak; derinlik sınırında değerlendirme fonksiyonu	12 yapraktan 7'si; XOX: 549.946 → 7.865 düğüm
Alt oyun mükemmel denge	Her alt oyunda Nash; geriye doğru çıkarımla bulunur	Piyasaya giriş; kırkayak (1, 0)
Kinci tetik stratejisi	Sonsuz tekrarda ihaneti kalıcı cezayla caydırmak	Sabırlı oyuncularda (δ büyük) işbirliği

Tablo 2.36: Bölümün kavram haritası.

2.13 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

oyun kuramı *game theory* · oyuncu *player* · rasyonellik *rationality* · ortak bilgi *common knowledge* · strateji *strategy* · strateji profili *strategy profile* · saf strateji *pure strategy* · karışık strateji *mixed strategy* · kazanç *payoff* · fayda fonksiyonu *utility function* · beklenen fayda *expected utility* · sıralı fayda *ordinal utility* · kardinal fayda *cardinal utility* · piyango *lottery* · bilgi kümesi *information set*

mükemmel bilgi *perfect information* · tam bilgi *complete information* · eksik bilgi *incomplete information* · Bayesçi oyun *Bayesian game* · normal form *normal form* · kazanç matrisi *payoff matrix* · genişletilmiş form *extensive form* · oyun ağacı *game tree* · yaprak *leaf* · sıfır toplamlı oyun *zero-sum game* · sabit toplamlı oyun *constant-sum game* · maksimin *maximin* · minimaks *minimax* · simetrik oyun *symmetric game* · koalisyon *coalition* · iskonto faktörü *discount factor* · en iyi tepki *best response* · Nash dengesi *Nash equilibrium* · baskın strateji *dominant strategy* · Mahkûmun İkilemi *Prisoner's Dilemma* · Pareto optimal *Pareto optimal* · Pareto baskınlık *Pareto dominance* · Anarşinin Bedeli *Price of Anarchy* · taahhüt *commitment* · odak noktası *focal point* · seviye-k düşünme *level-k reasoning* · yinelemeli (baskın strateji) eleme *iterated elimination of dominated strategies* · kombinatoryal oyun *combinatorial game* · Zermelo teoremi *Zermelo's theorem* · strateji çalma argümanı *strategy-stealing argument* · nim toplamı *nim-sum* · alt oyun *subgame* · alt oyun mükemmel denge *subgame perfect equilibrium* · geriye doğru çıkarım *backward induction* · inandırıcı olmayan tehdit *non-credible threat* · kırkayak oyunu *centipede game* · Ültimatom oyunu *ultimatum game* · kinci tetik stratejisi *grim trigger* · dallanma etkeni *branching factor* · yarım hamle *ply* · negamaks *negamax* · alfa-beta budaması *alpha–beta pruning* · hamle sıralaması *move ordering* · değerlendirme fonksiyonu *evaluation function* · derinlik sınırlı arama *depth-limited search* · ufuk etkisi *horizon effect* · sakinlik araması *quiescence search* · yinelemeli derinleştirme *iterative deepening* · geçiş tablosu *transposition table* · Monte Carlo ağaç araması *Monte Carlo tree search* · mekanizma tasarımı *mechanism design* · teşvik uyumlu *incentive compatible*

2.14 Kendini Dene

1. Şirket 1 dört fiyattan birini seçer (80, 90, 100, 120 TL). Şirket 2 bu fiyatı görüp 90, 100 ya da 120 TL der. İki şirketin kaçar saf stratejisi vardır?

   Cevap

   Şirket 1'in tek karar düğümü ve 4 eylemi vardır: 4 strateji. Şirket 2'nin 4 karar düğümü vardır (Şirket 1'in her fiyatı için bir tane) ve her birinde 3 eylem seçer: $3^4 = 81$ strateji. Eylem sayısı (3) ya da düğüm × eylem (12) strateji sayısı değildir.

2. Hücrelerde (u_1, u_2): (Üst, Sol) = (2, 2), (Üst, Sağ) = (0, 3), (Alt, Sol) = (3, 0), (Alt, Sağ) = (1, 1).

Baskın stratejiler, Nash dengesi ve Pareto optimal sonuçlar nelerdir?

Cevap

Satır oyuncusu için Alt baskındır: Sol'a karşı 3 > 2, Sağ'a karşı 1 > 0. Sütun oyuncusu için Sağ baskındır: Üst'e karşı 3 > 2, Alt'a karşı 1 > 0. Tek Nash dengesi (Alt, Sağ) = (1, 1)'dir. (Üst, Sol) = (2, 2) dengeyi Pareto baskılar. Pareto optimal sonuçlar (2, 2), (0, 3) ve (3, 0)'dır. Oyun bir Mahkûmun İkilemi'dir.

3. 2/3 oyununda (0–90) sınıfın %60'ı seviye 0, %40'ı seviye 1 düşünüyorsa hedef kaçtır? Seviye 1 mi, seviye 2 mi hedefe daha yakındır?

Cevap

Beklenen ortalama $0{,}6 \cdot 45 + 0{,}4 \cdot 30 = 27 + 12 = 39$, hedef $2/3 \cdot 39 = 26$'dır. Seviye 1 (30) hedefe 4, seviye 2 (20) hedefe 6 uzaktadır. Bu nüfusta seviye 1 kazanır: ötekilerin az düşündüğü bir sınıfta fazla düşünmek kaybettirir.

4. MAX kökünün üç MIN çocuğu var: B = [5, 9], C = [6, 1, 8], D = [7, 4]. Oyunun minimaks değeri ve MAX'ın seçimi nedir? Ağaçtaki en büyük yaprak neden seçilemez?

Cevap

MIN değerleri: B = min(5, 9) = 5, C = min(6, 1, 8) = 1, D = min(7, 4) = 4. Kök MAX = max(5, 1, 4) = 5; MAX B'yi seçer. En büyük yaprak 9 B'dedir, ama B'de seçimi MIN yapar ve 5'i seçer.

5. Nim konumu [1, 3, 6]. Sırası gelen oyuncu kazanan bir konumda mıdır? Kazandıran hamle hangisidir?

Cevap

$1 \oplus 3 = 001 \oplus 011 = 010 = 2$; $2 \oplus 6 = 010 \oplus 110 = 100 = 4$. Nim toplamı $4 \neq 0$: kazanan konumdur. Yığınlar denenir: $1 \oplus 4 = 5$ (1'den büyük), $3 \oplus 4 = 7$ (3'ten büyük), $6 \oplus 4 = 2$ (6'dan küçük). Tek kazandıran hamle üçüncü yığından 4 jeton alıp [1, 3, 2] bırakmaktır: $1 \oplus 3 \oplus 2 = 0$.

6. "25'i Söyleme"de oyuncular sırayla 1'den 4'e kadar ardışık sayı söyler; 25'i söyleyen kaybeder. Kaybeden durumlar (son söylenen sayı) hangileridir ve kim kazanır?

Cevap

Taban durum 24'tür: sıradaki 25'i söylemek zorundadır (Y). 23, 22, 21, 20 durumlarından 24'e ulaşılır (K). 19'dan ulaşılabilen 20–23 durumlarının hepsi K'dir: 19 Y'dir. Örüntü 5'lik adımlarla sürer: Y durumları 24, 19, 14, 9, 4'tür. Başlangıç 0'dan 4'e ulaşılabildiği için ilk oyuncu kazanır: "1, 2, 3, 4" der, sonra rakip x sayı söylerse kendisi $5 - x$ sayı söyler.

7. Piyasaya giriş oyununda Tekel'in savaşmaktan itibar kazandığını varsayalım: Gir–Savaş yaprağı (−1, −1) yerine (−1, 3) olsun; öteki yapraklar aynıdır (Girme (0, 2), Gir–Uzlaş (1, 1)). Alt oyun mükemmel denge nedir? "Girersen savaşırım" tehdidi artık inandırıcı mıdır?

   Cevap

   M'nin düğümünde Savaş 3, Uzlaş 1 verir; M savaşır ve düğümün değeri (−1, 3) olur. Kökte G için Gir −1, Girme 0 verir; G girmez. SPE (Girme, Savaş), sonuç (0, 2). Tehdit artık inandırıcıdır, çünkü G girdiğinde savaşmak gerçekten M'nin çıkarınadır.

8. Tekrarlanan bir oyunda her tur kazançlar şöyledir: ikisi de işbirliği yaparsa 3'er; biri sapıp öteki işbirliği yaparsa sapan 6, öteki 0; ikisi de saparsa 1'er. İki oyuncu kinci tetik oynuyor. Sapmanın kârsız olması için oyunun bir tur daha sürme olasılığı δ en az kaç olmalıdır?

   Cevap

   Uymak: $3 / (1 - \delta)$. Sapmak: bu tur 6, sonra her tur 1: $6 + \delta / (1 - \delta)$. Koşul $3 / (1 - \delta) \geq 6 + \delta / (1 - \delta)$. İki taraf $(1 - \delta)$ ile çarpılır: $3 \geq 6(1 - \delta) + \delta = 6 - 5\delta$, yani $\delta \geq 3/5$. Sapmanın cazibesi büyüdükçe (6 yerine daha büyük) eşik de büyür.

9. MAX kökünün üç MIN çocuğu var: B = [4, 8, 6], C = [2, 7], D = [5, 3, 9]. Çocuklar soldan sağa denenirse alfa-beta hangi yaprakları okur, hangilerini budar? Kökün değeri kaçtır?

   Cevap

   B'nin üç yaprağı da okunur: B = min(4, 8, 6) = 4 ve kökte $\alpha = 4$ olur. C'ye [4, +∞] gelir; ilk yaprak 2 ile $\beta = 2 \leq \alpha = 4$ olur ve 7 budanır. D'ye de [4, +∞] gelir; 5 ile $\beta = 5$ olur ve aralık hâlâ doludur; 3 ile $\beta = 3 \leq 4$ olur ve 9 budanır. Okunan yapraklar 4, 8, 6, 2, 5, 3'tür (sekizden altısı). Kökün değeri max(4, 2, 3) = 4'tür ve MAX B'yi seçer. Minimaksın sonucu da aynıdır: C = 2, D = 3.

10. Her iç düğümün 4 çocuğu olan, derinliği 4 olan düzgün bir ağaçta minimaks kaç yaprak okur? En iyi hamle sıralamasıyla alfa-beta kaç yaprak okur?

    Cevap

    Minimaks bütün yaprakları okur: $4^4 = 256$. En iyi sırada Knuth–Moore formülü $4^2 + 4^2 - 1 = 31$ verir. En kötü sırada budama olmaz ve yine 256 yaprak okunur; kök değeri üç durumda da aynıdır.

11. XOX'ta derinlik cezası $\pm(10 - d)$ kullanılıyor. X'in üç hamlesinden biri kökten 3 yarım hamle sonra kazancı zorluyor, biri 7 yarım hamle sonra kazancı zorluyor, biri beraberliğe gidiyor. Hamlelerin

değerleri nedir ve X hangisini seçer? ±1 ölçeğinde ne değişir?

Cevap

Değerler *10 − 3 = +7*, *10 − 7 = +3* ve 0'dır; X en büyüğü, yani hızlı kazancı (+7) seçer. ±1 ölçeğinde iki kazanç da +1 alır ve aralarındaki seçimi yalnız hamlelerin deneniş sırası belirler; program kazancı gereksiz yere uzatabilir.

12. Dörtlü Bağla'da Kırmızı orta sütuna iki taş koymuştur (alttan birinci ve ikinci satır); Mavi'nin tek taşı en soldaki sütunun altındadır. Pencere puanları ve orta sütun ek puanıyla *E(s)* kaçtır?

Cevap

Orta sütunda alttan 1–4. satırları kapsayan dikey pencerede iki kırmızı ve iki boş kare vardır: +2. Öteki bütün pencereler kırmızı taşlardan en çok birini ya da yalnız mavi taşı içerir: 0. Orta sütun ek puanı *3 · (2 − 0) = 6*'dır. Toplam *E(s) = 2 + 6 = +8*.

13. Dallanma etkeni 3 olan bir oyunda yinelemeli derinleştirme derinlik 1'den 8'e kadar arama yapıyor. Toplam iş, yalnız derinlik 8'deki aramanın yaklaşık kaç katıdır? b = 7 ile karşılaştırıldığında ne görülür?

Cevap

Toplam iş $3 + 3^2 + \ldots + 3^8 \approx 3^8 \cdot (1 + 1/3 + 1/9 + \ldots) = 3^8 \cdot 3/2$'dir: yaklaşık 1,5 kat. b = 7 için oran 7/6 ≈ 1,17'dir. Genel olarak oran $b / (b - 1)$'dir: dallanma etkeni büyüdükçe son katman öncekilerin toplamını daha çok geçer ve yinelemeli derinleştirmenin ek maliyeti azalır.

2.15 İleri okuma

- M. J. Osborne, *An Introduction to Game Theory* (2004), Bölüm 2–5: normal form oyunları, Nash dengesini, genişletilmiş form oyunları ve geriye doğru çıkarımı az matematikle ve çok sayıda örnekle anlatır.
- A. R. Karlin ve Y. Peres, *Game Theory, Alive* (2017): kombinatoryal oyunları, Nim'i ve strateji çalma argümanını öğrenci düzeyinde ayrıntılı ispatlarla verir.
- C. F. Camerer, *Behavioral Game Theory* (2003): 2/3 oyunu, kırkayak ve ültimatom gibi oyunlarda gerçek oyuncuların davranışını deney sonuçlarıyla inceler.
- S. Russell ve P. Norvig, *Artificial Intelligence: A Modern Approach* (4. baskı, 2020), Bölüm 5: minimaksı, alfa-beta budamasını, değerlendirme fonksiyonlarını ve Monte Carlo ağaç aramasını oyun oynayan programlar üzerinden anlatır.
- D. E. Knuth ve R. W. Moore, “An analysis of alpha-beta pruning” (1975): alfa-betanın doğruluğunu ve en iyi sıralamada okunan yaprak sayısını ispatlarıyla veren klasik makale.

- N. Nisan, T. Roughgarden, É. Tardos ve V. Vazirani (ed.), *Algorithmic Game Theory* (2007), Bölüm 1: oyun kuramının temel kavramlarını algoritmik oyun kuramının soruları açısından tanıtır.

2.16 Kaynaklar

1. L. V. Allis, *A Knowledge-Based Approach of Connect-Four*, yüksek lisans tezi, Vrije Universiteit Amsterdam, 1988.
2. C. L. Bouton, "Nim, a game with a complete mathematical theory", *Annals of Mathematics* 3, 1901–1902.
3. C. F. Camerer, *Behavioral Game Theory*, Princeton University Press, 2003.
4. M. Campbell, A. J. Hoane, F. Hsu, "Deep Blue", *Artificial Intelligence* 134, 57–83, 2002.
5. R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search", *Computers and Games*, 2006.
6. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010.
7. W. Güth, R. Schmittberger, B. Schwarze, "An experimental analysis of ultimatum bargaining", *Journal of Economic Behavior & Organization* 3(4), 1982.
8. R. A. Hearn, E. D. Demaine, *Games, Puzzles, and Computation*, A K Peters, 2009.
9. A. R. Karlin, Y. Peres, *Game Theory, Alive*, American Mathematical Society, 2017.
10. D. E. Knuth, R. W. Moore, "An analysis of alpha-beta pruning", *Artificial Intelligence* 6(4), 293–326, 1975.
11. L. Kocsis, C. Szepesvári, "Bandit based Monte-Carlo planning", *ECML*, 2006.
12. H. W. Kuhn, "Extensive games and the problem of information", *Contributions to the Theory of Games* II, Princeton University Press, 1953.
13. R. D. McKelvey, T. R. Palfrey, "An experimental study of the centipede game", *Econometrica* 60(4), 1992.
14. R. Nagel, "Unraveling in guessing games: An experimental study", *American Economic Review* 85(5), 1995.
15. J. F. Nash, "Equilibrium points in n-person games", *PNAS* 36(1), 1950.
16. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 1).
17. M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2004 (Bölüm 2–5).
18. S. Reisch, "Hex ist PSPACE-vollständig", *Acta Informatica* 15, 1981.
19. R. W. Rosenthal, "Games of perfect information, predatory pricing and the chain-store paradox", *Journal of Economic Theory* 25, 1981.
20. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press, 2016.
21. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 4. baskı, Pearson, 2020 (Bölüm 5).
22. J. Schaeffer vd., "Checkers is solved", *Science* 317(5844), 2007.
23. T. C. Schelling, *The Strategy of Conflict*, Harvard University Press, 1960.
24. R. Selten, "Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit", *Zeitschrift für die gesamte Staatswissenschaft* 121, 301–324 ve 667–689, 1965.
25. C. E. Shannon, "Programming a computer for playing chess", *Philosophical Magazine* 41(314), 1950.

26. Y. Shoham, K. Leyton-Brown, *Multiagent Systems*, Cambridge University Press, 2009 (Bölüm 3 ve 5).

27. D. Silver vd., "Mastering the game of Go with deep neural networks and tree search", *Nature* 529, 484–489, 2016.

28. B. Skyrms, *The Stag Hunt and the Evolution of Social Structure*, Cambridge University Press, 2004.

29. J. Tromp, John's Connect Four Playground (Dörtlü Bağla konum sayısı).

30. J. Tromp, G. Farnebäck, "Combinatorics of Go", *Computers and Games*, 2006.

31. J. Tromp, "The Number of Legal Go Positions", *Computers and Games*, 2016.

32. J. von Neumann, "Zur Theorie der Gesellschaftsspiele", *Mathematische Annalen* 100, 1928.

33. E. Zermelo, "Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels", *Proceedings of the Fifth International Congress of Mathematicians*, cilt 2, 501–504, Cambridge University Press, 1913.

34. A. L. Zobrist, "A new hashing method with application for game playing", Tech. Rep. 88, University of Wisconsin, 1970.

KISIM I · BÖLÜM 3

Nash Dengesi

Bir oyunda her oyuncunun kazancı yalnız kendi seçimine değil, ötekilerin seçimine de bağlıdır. Bu yüzden "ne yapmalıyım?" sorusu "onlar ne yapacak?" sorusundan ayrılamaz. Nash dengesi bu döngüyü kıran kavramdır: herkes ötekilerin seçimini doğru bildiğinde kendi seçimini değiştirmek istemez. Not, tanımdan başlar ve hesaplama yöntemlerine, varlık teoremine, çoklu dengede seçim sorusuna ve sıfır toplamlı oyunlara uzanır. Son konu, dengeyi bulmanın bilgisayar için ne kadar zor olduğudur.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum03`

ÖĞRENME ÇIKTILARI

Tanım
Nash dengesini biçimsel olarak tanımlamak ve en iyi tepki (*best response*) kavramıyla ilişkilendirmek.

Saf denge bulma
Kazanç matrisinde en iyi tepkileri işaretlemek, yinelemeli baskın strateji eleme uygulamak; Cournot ve Bertrand gibi sürekli stratejili oyunlarda dengeyi bulmak.

Karışık denge
Kayıtsızlık ilkesiyle karışık dengeyi elle ve kodla hesaplamak; sıfır toplamlı oyunlarda ve güvenlik oyunlarında minimaks değerini bulmak.

Varlık ve karmaşıklık
Nash'in varlık teoremini, çoklu denge sorununu ve denge hesaplamanın PPAD-tamlığını yorumlamak.

3.1 Temeller

Bu bölümün bütün kavramları tek bir sorunun etrafında döner: birden çok karar vericinin kararları birbirine bağlıysa, "makul" bir sonuç nedir? Küçük bir oyun ve gündelik bir durum bu soruyu somutlaştırır; biçimsel tanım bunların üzerine kurulur.

3.1.1 Bir oyunla başlangıç: Mahkûmun İkilemi

Sam ve John adlı iki zanlı ayrı odalarda sorgulanır. Her biri, ötekinin ne yaptığını bilmeden iki seçenekten birini seçer: *Sessiz* kalmak ya da ötekine *İhanet* etmek. İkisi de sessiz kalırsa yetersiz delil yüzünden birer yıl yatarlar. İkisi de ihanet ederse ikişer yıl yatarlar. Biri ihanet edip öteki sessiz kalırsa ihanet eden serbest kalır, sessiz kalan üç yıl yatar.

Oyun kuramında bir oyunun sonuçları **kazanç** (*payoff*) olarak yazılır ve her oyuncu kendi kazancını büyütmek ister. Hapis yılında ise az olan iyidir. İki bakışı uzlaştırmak için kazanç, hapis yılının eksi işaretlisi alınır: 1 yıl hapis −1 kazanç, 3 yıl hapis −3 kazanç demektir. Böylece "büyük sayı iyidir" kuralı her yerde geçerli kalır; −1, −3'ten büyüktür.

Oyun genellikle bir **kazanç matrisi** (*payoff matrix*) ile yazılır. Satırlar bir oyuncunun (burada Sam'in), sütunlar ötekinin (John'un) seçenekleridir. Her hücrede iki sayı vardır: önce satır oyuncusunun, sonra sütun oyuncusunun kazancı. Bu sıralama bütün not boyunca değişmez.

Sam \ John	Sessiz	İhanet
Sessiz	(−1, −1)	(−3, 0)
İhanet	(0, −3)	(−2, −2)

Tablo 3.1: Mahkûmun İkilemi. Hücrede (Sam, John) kazancı; kazanç = −(hapis yılı).

Tabloyu okumak bu bölümün en temel becerisidir. Örneğin sağ üst hücre (Sessiz, İhanet) = (−3, 0) şunu söyler: Sam sessiz kaldı, John ihanet etti; Sam 3 yıl yatar (kazancı −3), John serbest kalır (kazancı 0). Sam yalnızca satırı seçebilir; hangi hücreye düşüleceğini John'un sütunu belirler. Oyunu tek kişilik karar probleminden ayıran da budur.

Sam'in yerine geçip düşünelim. John'un ne yapacağı bilinmiyor. John sessiz kalırsa Sam'in seçeneği −1 (sessiz) ile 0 (ihanet) arasındadır; ihanet daha iyidir. John ihanet ederse seçenek −3 (sessiz) ile −2 (ihanet) arasındadır; yine ihanet daha iyidir. Demek ki John ne yaparsa yapsın Sam için ihanet daha iyidir. Aynı akıl yürütme John için de geçerlidir, çünkü oyun simetriktir. Sonuçta ikisi de ihanet eder ve ikişer yıl yatarlar. Oysa ikisi de sessiz kalsaydı yalnız birer yıl yatacaklardı. Oyunun adındaki "ikilem" buradan gelir: her birinin tek başına akıllıca seçimi, ikisi için de kötü bir sonuç doğurur.

Oyun tekrarlandığında tablo değişir. Bilgisayara karşı oynanan sürümde bilgisayar ***Tit-for-Tat*** (kısasa kısas) stratejisini kullanabilir: ilk turda sessiz kalır, sonraki her turda karşısındakinin bir önceki seçimini kopyalar. Böyle bir rakibe karşı on turda en az kaç yıl birikir? Sorunun cevabı, ihanetin bir sonraki turda cezalandırılmasına dayanır ve 3.3.2'de hesaplanır.

3.1.2 John Nash ve kavramın kökeni

Kavram adını Amerikalı matematikçi John Forbes Nash Jr.'dan (1928–2015) alır. Nash, 1950'de Princeton'da *Non-Cooperative Games* (İşbirliksiz Oyunlar) adlı kısa bir doktora tezi yazdı. Aynı yıl *PNAS* dergisinde iki sayfalık bir not yayımladı; bu notta her sonlu oyunun bir denge noktası olduğunu ispatladı (Nash, 1950). Tezin genişletilmiş hâli ertesi yıl *Annals of Mathematics*'te çıktı (Nash, 1951).

Nash'ten önce oyun kuramının en güçlü sonucu, John von Neumann'ın 1928'de iki kişilik **sıfır toplamlı oyunlar** (*zero-sum game*) için kurduğu minimaks kuramıydı (von Neumann, 1928). Sıfır toplamlı oyunda birinin kazancı ötekinin kaybıdır; çıkarlar tam olarak zıttır. Gerçek hayattaki etkileşimlerin çoğu böyle değildir. Mahkûmun İkilemi'nde iki oyuncu da (Sessiz, Sessiz) sonucunu (İhanet, İhanet) sonucuna tercih eder; çıkarlar kısmen ortaktır. Nash, denge fikrini n kişilik genel (sıfır toplamlı olmayan) oyunlara taşıdı.

Nash bu çalışmalarıyla 1994'te John Harsanyi ve Reinhard Selten'le birlikte Nobel Ekonomi Ödülü'nü aldı. 2015'te kısmi diferansiyel denklemlere katkıları için Abel Ödülü'ne layık görüldü. Hayatı *A Beautiful Mind* (2001) filmine konu oldu.

3.1.3 Stratejik davranış ve uygulama alanları

Oyun kuramının temel varsayımı, oyuncuların kendi kazançlarını **maksimize etmeye** çalıştığıdır. Tek kişilik karar kuramı da aynı varsayımla çalışır. Oyun kuramında yeni olan, herkesin kazancının ötekilerin seçimine de bağlı olmasıdır. Oyuncu en iyi seçimini yapmak için ötekilerin ne yapacağını tahmin etmelidir; ötekiler de onunkini tahmin eder. Buna **stratejik davranış** denir.

Stratejik bağımlılık çok farklı alanlarda karşımıza çıkar. Aşağıdaki tablo bu bölümde ve sonraki konularda sık geçen örnekleri ve yapılarını özetler.

Durum	Ne olur?	Yapı
Fiyat rekabeti	İki firma fiyat indirir; ikisi de kâr kaybeder.	Mahkûmun İkilemi yapısı
Reklam / silahlanma	Herkes harcar, göreli konum değişmez.	Baskın strateji: harcamak
Penaltı atışı	Atıcı ve kaleci tahmin edilemez olmalı.	Karışık strateji, sıfır toplamlı
Trafikte şerit seçimi	Herkes hızlı şeride geçerse o şerit yavaşlar.	Tıkanıklık oyunu
Standart seçimi	Aynı teknolojiyi seçmek herkese kazandırır.	Koordinasyon, çoklu denge
Kaynak paylaşımı	Ortak bant genişliği, sunucu, yol.	Anarşinin Bedeli

Tablo 3.2: Stratejik etkileşim örnekleri ve oyun kuramındaki yapıları.

Bu örneklerin hiçbirinde “en iyi seçim” rakipten bağımsız söylenemez. Penaltıda kaleci hep sola atlarsa atıcı sağa vurur; atıcı hep sağa vurursa kaleci sağa atlar. Standart seçiminde doğru teknoloji, ötekilerin seçtiği teknolojidir. Bilgisayar ağlarında bir cihazın o anda paket göndermesi, ötekilerin de gönderip göndermediğine göre iyi ya da kötüdür. Bu yüzden her oyuncunun ayrı ayrı en iyi seçimi değil, seçimlerin birbiriyle tutarlı olduğu durumlar aranır.

3.1.4 Kimsenin pişman olmadığı durum

Biçimsel tanımdan önce kavramı gündelik bir örnekle düşünelim. Yemekhanede iki kasa vardır ve öğrenciler kuyruklara dağılmıştır. Herkes daha kısa beklemek ister ve kuyruk değiştirmek serbesttir. Şekil 3.1'de iki durum görülür.



Şekil 3.1: Yemekhanede iki kasa. Solda 7 ve 3 kişilik kuyruklar: uzun kuyruğun sonundaki öğrenci geçerse 7. sıra yerine 4. sıraya düşer, yani geçmek kârlıdır. Sağda 5 ve 5 kişilik kuyruklar: geçen öğrenci 6. sıraya düşer, kimse geçmez.

Kuyruklar 7 ve 3 kişiyse uzun kuyruğun sonundaki öğrenci öteki kasaya geçerek 7. sıra yerine 4. sıraya gelir. Geçmek kârlı olduğu için bu durum kalıcı değildir; biri mutlaka yer değiştirir. Kuyruklar 5 ve 5 kişiyse geçen öğrenci öteki kuyruğun 6. sırasına, yani daha geriye düşer. Kimsenin yer değiştirmek için nedeni yoktur; durum kendi kendini sürdürür.

Bu gözlemden bir test çıkar. Ona **pişmanlık testi** diyelim: "Ötekilerin ne yaptığını görseydim, kendi seçimimi değiştirir miydim?" Herkesin cevabı "hayır" ise durum bir Nash dengesidir. Testin iki önemli özelliği vardır. Birincisi, her oyuncu yalnız *kendi* seçimini değiştirmeyi düşünür; ötekilerin seçimi sabit kabul edilir. İkincisi, test her oyuncu için ayrı ayrı yapılır; bir kişinin bile cevabı "evet" ise durum denge değildir.

Denge kendi kendini sürdüren bir durumdur: herkes ötekilerin davranışını doğru bekler ve buna göre elinden gelenin en iyisini yapar. Ancak denge "en iyi sonuç" ya da "adil sonuç" anlamına gelmez. Mahkûmun İkilemi'nde (İhanet, İhanet) pişmanlık testini geçer, ama iki oyuncu da (Sessiz, Sessiz) sonucunu tercih ederdi. Denge bir kararlılık kavramıdır, bir değerlendirme kavramı değildir.

3.1.5 Biçimsel tanım

Pişmanlık testini kesin bir dille yazmak için önce oyunun kendisi kesin bir dille yazılmalıdır. Aşağıdaki gösterim bütün bölümde kullanılır.

Tanım 3.1

Bir **normal form oyun** (*normal form, strategic form*) $G = (N, (S_i), (u_i))$ üç bileşenden oluşur:

- $N = \{1, \ldots, n\}$: oyuncular kümesi.
- S_i: oyuncu i'nin **strateji kümesi** (*strategy set*), yani seçebileceği seçenekler.
- u_i: oyuncu i'nin **kazanç fonksiyonu** (*payoff function*). Her oyuncunun bir seçim yaptığı her duruma bir sayı atar.

Her oyuncunun bir strateji seçtiği listeye **strateji profili** (*strategy profile*) denir: $s = (s_1, \ldots, s_n)$. Oyuncu i dışındakilerin stratejilerine s_{-i} yazılır; böylece profil $s = (s_i, s_{-i})$ biçiminde de gösterilir. $u_i(s)$, profil s oynandığında oyuncu i'nin kazancıdır.

Mahkûmun İkilemi'nde $N = \{Sam, John\}$, her iki oyuncu için $S_i = \{Sessiz, İhanet\}$'tir. Dört strateji profili vardır ve her biri tablonun bir hücresidir. Örneğin $s = (Sessiz, İhanet)$ için $u_{Sam}(s) = -3$ ve $u_{John}(s) = 0$'dır. Sam açısından bakınca s_{-Sam} yalnızca John'un stratejisidir.

s_{-i} gösterimi ilk bakışta garip gelebilir. Oysa mantığı basittir: oyuncu i bir karar verirken ötekilerin seçimini tek bir paket olarak düşünür. İki kişilik oyunda bu paket tek bir stratejidir; üç kişilik oyunda iki stratejiden oluşan bir listedir. Gösterim, oyuncu sayısı ne olursa olsun aynı kalır.

Tanım 3.2

$s^* = (s_1^*, \ldots, s_n^*)$ profili bir **Nash dengesidir** (*Nash equilibrium, NE*) ancak ve ancak her oyuncu i ve her $s_i \in S_i$ için

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

eşitsizliği sağlanır. Yani ötekiler s_{-i}^*'ı oynarken hiçbir oyuncu stratejisini değiştirerek kazancını artıramaz.

Eşitsizliğin sol tarafı, oyuncunun dengedeki kazancıdır. Sağ taraf, oyuncu kendi stratejisini s_i'ye değiştirdiğinde ve ötekiler yerinde kaldığında alacağı kazançtır. Bu tür bir değişikliğe **tek taraflı sapma** (*unilateral deviation*) denir. Tanım, hiçbir tek taraflı sapmanın kârlı olmadığını söyler. "≥" işaretine dikkat edilmelidir: sapma kazancı eşit bırakabilir, yalnız artıramaz.

Tanım üç açıdan okunabilir:

- **Tek taraflı sapma:** Diğerleri sabitken hiçbir oyuncu stratejisini değiştirerek kazancını artıramaz.
- **Kendini doğrulayan anlaşma:** Oyuncular önceden bir profilde anlaştıysa ve herkes ötekilerin anlaşmaya uyacağına inanıyorsa, kendisi de uyar. Anlaşmanın uygulanması için dışarıdan bir zorlama gerekmez.

- **Kararlılık verimlilik değildir:** Denge, adalet ya da verimlilik garantisi vermez. Özellikle bir denge **Pareto optimal** olmak zorunda değildir. Bir sonuç, herkesi en az onun kadar iyi ve en az birini kesin daha iyi yapan başka bir sonuç yoksa Pareto optimaldir.

Örnek 3.1

Mahkûmun İkilemi'nin dört hücresi tanımla tek tek denetlenir. Her hücre için iki soru sorulur: satır oyuncusu (Sam) satırını değiştirince kazanır mı, sütun oyuncusu (John) sütununu değiştirince kazanır mı?

1. (Sessiz, Sessiz) = (−1, −1). Sam İhanet'e geçerse hücre (İhanet, Sessiz) olur ve Sam 0 alır. 0 > −1 olduğundan sapma kârlıdır. Hücre denge değildir; John'a bakmaya gerek kalmaz.
2. (Sessiz, İhanet) = (−3, 0). Sam İhanet'e geçerse −2 alır; −2 > −3. Denge değildir.
3. (İhanet, Sessiz) = (0, −3). John İhanet'e geçerse −2 alır; −2 > −3. Denge değildir.
4. (İhanet, İhanet) = (−2, −2). Sam Sessiz'e geçerse −3 alır; −3 < −2, sapma kârsızdır. John Sessiz'e geçerse o da −3 alır; yine kârsızdır. İki oyuncu için de hiçbir sapma kazancı artırmaz. Hücre bir Nash dengesidir.

Sonuç: oyunun tek saf Nash dengesi (İhanet, İhanet)'tir ve kazançları (−2, −2)'dir, yani ikişer yıl hapis. (Sessiz, Sessiz) ise ikisine de daha iyi olduğu hâlde denge değildir. Bu yüzden denge Pareto optimal değildir.

Dört hücreyi tek tek denetlemek küçük bir oyunda kolaydır, ama oyun büyüdükçe yorucu olur. 3.2'deki en iyi tepki işaretlemesi aynı denetimi düzenli biçimde yapar.

3.1.6 Tanımla ilgili yanılgılar

Nash dengesinin tanımı kısa olduğu için yanlış okunmaya açıktır. Aşağıda en sık görülen yanılgılar sıralanmıştır. Her birinin yanlışlığı tanımdaki bir sözcükten anlaşılır.

Sık yapılan hata

- **"Denge en iyi sonuçtur."** Hayır. Mahkûmun İkilemi'nde denge (İ, İ) iki oyuncuya da (S, S)'den kötüdür. Denge kararlılık söyler, verimlilik söylemez.
- **"İkisi birlikte değişirse kazanır, o hâlde denge değildir."** Tanım yalnızca tek taraflı sapmaya bakar. İki oyuncunun birlikte sapması (koalisyon) Nash dengesini bozmaz. (İ, İ)'den birlikte (S, S)'ye geçmek ikisine de kazandırır, ama bu tanımın dışındadır.
- **"Bir oyuncu memnunsa yeter."** Test her oyuncu için ayrı yapılır; tek bir oyuncu bile sapmak isterse hücre denge değildir.
- **"Satır oyuncusu sütunu da seçer."** Her oyuncu yalnız kendi stratejisini değiştirir. Satır oyuncusu aynı sütunda yukarı-aşağı, sütun oyuncusu aynı satırda sağa-sola gider.

- **"Her oyunda tek bir saf denge vardır."** Yazı-Tura'da saf denge yoktur; Geyik Avı'nda iki saf ve bir karışık denge vardır. Bunlar 3.4 ve 3.5'te hesaplanır.
- **Ceza mı, kazanç mı?** Hapis yılı gibi maliyetlerde küçük sayı iyidir. Kazanca çevirmek için eksi işaret konur: 2 yıl → −2. İşaret konmadan "büyük olanı" seçmek, tam ters cevabı verir.

3.2 En iyi tepki

Nash dengesinin tanımı her profil için "sapma kârlı mı?" sorusunu sorar. Soruyu düzenli biçimde cevaplamak için her oyuncunun, ötekilerin her olası seçimine karşı en iyi cevabı önceden bulunur. Bu cevaba en iyi tepki denir. Dengeyi bulmak, en iyi tepkilerin çakıştığı yeri bulmaya dönüşür.

3.2.1 En iyi tepki ve karşılıklı en iyi tepki

En iyi tepkinin arkasındaki soru basittir: "Rakibin ne yapacağını bilseydim ne yapardım?" Rakibin seçimi sabitlenir, oyuncu yalnız kendi kazançlarına bakar ve en büyüğünü veren seçeneği alır.

Bunu yazmak için **argmax** gösterimi kullanılır. *max* bir listedeki en büyük *değeri* verir; *argmax* ise bu en büyük değeri veren *seçeneği* (ya da seçenekleri) verir. Örneğin dört seçeneğin kazançları A: 3, B: 7, C: 2, D: 7 ise *max = 7*'dir, ama *argmax = {B, D}*'dir. En büyük değer birden çok seçenekte görülüyorsa argmax bir kümedir; bu ayrıntı eşitlikli oyunlarda önem kazanır.

Tanım 3.3

Oyuncu i'nin, ötekilerin s_{-i} seçimine **en iyi tepkisi** (*best response*), kazancını en yüksek yapan stratejilerin kümesidir:

$$BR_i(s_{-i}) = \operatorname{argmax}_{s_i \in S_i} u_i(s_i, s_{-i})$$

En iyi tepki bir sabit değildir; rakibin seçimine bağlı bir kuraldır. Rakip değişince en iyi tepki de değişebilir. BR_i'yi "rakibin her seçimine karşı ne yapacağımı söyleyen tablo" olarak düşünmek yararlıdır. Bu kurala **en iyi tepki eşlemesi** (*best-response correspondence*) de denir. "Eşleme" sözcüğü, bir girdiye birden çok çıktı (eşitlik durumunda bir küme) atanabildiğini vurgular.

Nash dengesinin tanımındaki eşitsizlik, s_i^*'ın s_{-i}^*'a karşı en büyük kazancı verdiğini söyler. Bu, s_i^*'ın bir en iyi tepki olması demektir. Aşağıdaki teorem bu gözlemi kesinleştirir.

Teorem 3.1 (Denge ve en iyi tepki)

s^* profili bir Nash dengesidir ancak ve ancak her oyuncu i için $s_i^* \in BR_i(s_{-i}^*)$'dır. Kısaca: denge, karşılıklı en iyi tepkidir.

İspat. İki yön ayrı ayrı gösterilir. (⇒) s^* bir Nash dengesi olsun. Tanıma göre her s_i için $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$'dır. Bu, s_i^*'ın verdiği kazancın s_{-i}^*'a karşı ulaşılabilecek en büyük kazanç olduğu anlamına gelir. En büyük kazancı veren strateji argmax kümesindedir; demek ki $s_i^* \in BR_i(s_{-i}^*)$.

(⇐) Her i için $s_i^* \in BR_i(s_{-i}^*)$ olsun. Argmax kümesindeki bir strateji, en büyük kazancı verir; bu yüzden başka hiçbir s_i daha büyük kazanç veremez. Bu, tanımdaki eşitsizliğin ta kendisidir. Eşitsizlik her oyuncu için sağlandığından s^* bir Nash dengesidir. □

Teorem, dengeyi bulmak için pratik bir yöntem verir. İki kişilik bir matris oyununda üç adım yeterlidir:

1. **Satır oyuncusu:** Her sütunda, satır oyuncusunun en büyük kazancı (hücredeki ilk sayı, u_1) işaretlenir. Sütun sabit tutulduğu için bu, sütun oyuncusunun o seçimine karşı satır oyuncusunun en iyi tepkisidir.
2. **Sütun oyuncusu:** Her satırda, sütun oyuncusunun en büyük kazancı (ikinci sayı, u_2) işaretlenir.
3. **Kesişim:** İki kazancı da işaretli hücreler saf strateji Nash dengeleridir.

Yöntemin maliyeti küçüktür. $m \times n$ bir matriste her hücre bir sütun maksimumu ve bir satır maksimumuyla karşılaştırılır. Sütun maksimumları ve satır maksimumları bir kez hesaplanırsa toplam iş $O(mn)$ olur; yani matristeki hücre sayısıyla orantılıdır. Matris zaten verildiyse bundan daha hızlı bir yöntem olamaz, çünkü her hücreye en az bir kez bakmak gerekir.

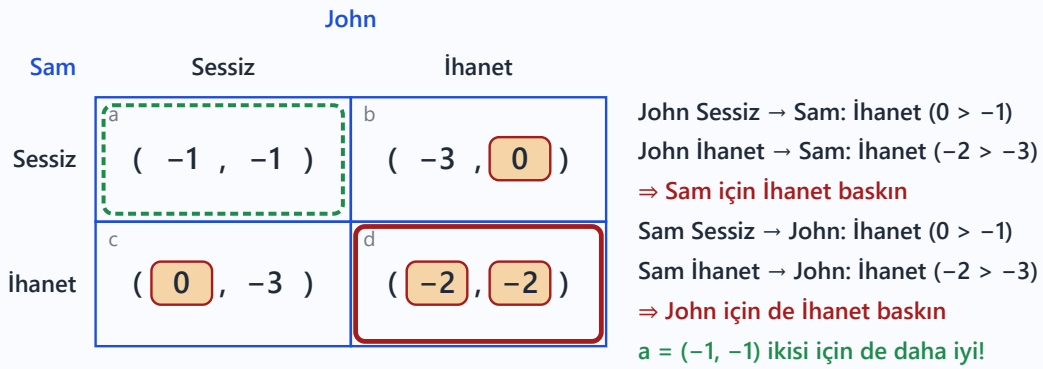
3.2.2 İşaretleme yöntemi: Mahkûmun İkilemi

Yöntemi önce tanıdık oyunda deneyelim. Kazançlar hapis yılının eksi işaretlisidir (−1 = 1 yıl) ve her oyuncu kendi sayısını büyütmek ister. Hücrelere a, b, c, d harfleri verelim: a = (Sessiz, Sessiz), b = (Sessiz, İhanet), c = (İhanet, Sessiz), d = (İhanet, İhanet).

Örnek 3.2

1. **John Sessiz kalırsa** Sam bu sütunda yalnızca ilk sayılara bakar: Sessiz −1, İhanet 0. 0 > −1 olduğundan Sam'in en iyi tepkisi İhanet'tir; c hücresindeki 0 işaretlenir.
2. **John İhanet ederse:** Sessiz −3, İhanet −2. −2 > −3, en iyi tepki yine İhanet; d hücresindeki ilk −2 işaretlenir.
3. İki sütunda da işaret aynı satırdadır. John ne yaparsa yapsın İhanet Sam için daha iyidir. Böyle bir stratejiye **baskın strateji** denir (3.3.1).
4. **Sam Sessiz kalırsa** John ikinci sayılara bakar: Sessiz −1, İhanet 0; en iyi tepki İhanet, b hücresindeki 0 işaretlenir.
5. **Sam İhanet ederse:** Sessiz −3, İhanet −2; John için de İhanet. d hücresindeki ikinci −2 işaretlenir. Oyun simetrik olduğu için bu sonuç şaşırtıcı değildir.
6. İki işaretin buluştuğu tek hücre d = (İhanet, İhanet)'tir. Kimse tek başına sapmak istemez: Nash dengesi (−2, −2).

Oysa a = (Sessiz, Sessiz) ikisine de yalnızca birer yıl verirdi. Denge Pareto optimal değildir; ikilemin özü budur.



Şekil 3.2: Mahkûmun İkilemi'nde en iyi tepkiler. Turuncu kutular her oyuncunun en iyi tepkisini gösterir; iki işaretli tek hücre d = (İhanet, İhanet) Nash dengesidir (kırmızı çerçeve). Yeşil kesikli çerçeve a = (Sessiz, Sessiz) ikisine de daha iyidir ama denge değildir.

Şekil 3.2'de işaretlemenin son hâli görülür. Her oyuncunun iki işareti de İhanet satırında ya da sütunundadır. Bu, İhanet'in iki oyuncu için de baskın strateji olduğunu gösterir. Baskın stratejisi olan oyuncunun rakibini tahmin etmesine gerek yoktur; bu da oyunun çözümünü çok kolaylaştırır.

3.2.3 Üçe üç bir örnek

Baskın strateji her oyunda bulunmaz. Aşağıdaki oyunda satır oyuncusu A, B, C; sütun oyuncusu X, Y, Z stratejilerinden birini seçer. Hücrede (u_1, u_2) yazılıdır: önce satır, sonra sütun oyuncusunun kazancı.

	X	Y	Z
A	**(4, 3)**	(0, 2)	(1, 1)
B	(2, 1)	**(3, 3)**	(0, 2)
C	(1, 4)	(2, 0)	(3, 2)

Tablo 3.3: Üçe üç örnek oyun. Yeşil hücreler iki saf Nash dengesidir.

Örnek 3.3

Satır oyuncusunun en iyi tepkileri (her sütunda ilk sayılar karşılaştırılır):

- Sütun oyuncusu X oynarsa: A: 4, B: 2, C: 1 → en iyisi A (4).
- Sütun oyuncusu Y oynarsa: A: 0, B: 3, C: 2 → en iyisi B (3).
- Sütun oyuncusu Z oynarsa: A: 1, B: 0, C: 3 → en iyisi C (3).

Sütun oyuncusunun en iyi tepkileri (her satırda ikinci sayılar karşılaştırılır):

- Satır oyuncusu A oynarsa: X: 3, Y: 2, Z: 1 → en iyisi X (3).
- Satır oyuncusu B oynarsa: X: 1, Y: 3, Z: 2 → en iyisi Y (3).
- Satır oyuncusu C oynarsa: X: 4, Y: 0, Z: 2 → en iyisi X (4).

Kesişim: İki kazancı da işaretli hücreler (A, X) = (4, 3) ve (B, Y) = (3, 3)'tür. Oyunun iki saf Nash dengesi vardır.

Sapma testi (A, X): Satır oyuncusu B'ye geçerse 2, C'ye geçerse 1 alır; ikisi de 4'ten küçüktür. Sütun oyuncusu Y'ye geçerse 2, Z'ye geçerse 1 alır; ikisi de 3'ten küçüktür. Kimse sapmayla kazanamaz ✓. (B, Y) için de aynı denetim yapılır: satır oyuncusu 2 ya da 0'a düşer, sütun oyuncusu 1 ya da 2'ye düşer ✓.



Şekil 3.3: Üçe üç oyunda işaretlemenin sonucu. Satır oyuncusunun işaretleri her sütunda bir tanedir (4, 3, 3), sütun oyuncusununkiler her satırda bir tanedir (3, 3, 4). İki işaretli hücreler (A, X) ve (B, Y) dengedir.

Bu örnekte iki önemli gözlem vardır. Birincisi, (C, Z) hücresinde satır oyuncusunun 3'ü işaretlidir ama sütun oyuncusu Z yerine X'e geçerek 2 yerine 4 alır. Tek işaret yetmez. İkincisi, "en büyük sayılar" dengeyi belirlemez: (C, X)'teki 4 de sütun oyuncusu için tablodaki en büyük kazançlardan biridir, ama satır oyuncusu orada A'ya geçmek ister. Denge, sayıları büyük olan hücre değil, karşılıklı en iyi tepki olan hücredir.

Oyunun iki dengesi olması da dikkat çekicidir. Kuram hangi dengenin oynanacağını tek başına söylemez. Satır oyuncusu (A, X)'i tercih eder (4 alır); sütun oyuncusu iki dengede de 3 alır, bu yüzden ikisi arasında kayıtsızdır. Birden çok denge olduğunda hangisinin seçileceği 3.5'in konusudur.

3.2.4 Eşitlik olduğunda: dil seçimi oyunu

İşaretleme yöntemi eşitlik olduğunda bir incelik gerektirir. Ayşe ve Bora ortak bir proje için programlama dili seçiyor: Python, Java ya da C. Hücrede (Ayşe, Bora) kazancı vardır; Ayşe satır, Bora sütun oyuncusudur. Kural aynıdır: Ayşe için her sütunda ilk sayıların, Bora için her satırda ikinci sayıların en büyüğünün altı çizilir. Bu yüzden yönteme **altını çizme yöntemi** de denir.

Ayşe \ Bora	Python	Java	C
Python	**(4, 2)**	(1, 1)	(1, 0)
Java	(0, 1)	**(2, 4)**	(1, 2)
C	(1, 0)	(0, 2)	(1, 1)

Tablo 3.4: Dil seçimi oyunu.
Hücrede (Ayşe, Bora).

Örnek 3.4

1. Bora Python seçerse Ayşe yalnız ilk sayılara bakar: 4, 0, 1 → en büyük 4 (Python) çizilir.
2. Bora Java seçerse: 1, 2, 0 → en büyük 2 (Java).
3. Bora C seçerse: 1, 1, 1 → üçü eşittir ve üçünün de altı çizilir. Eşitlikte hepsi en iyi tepkidir; argmax kümesi {Python, Java, C}'dir.
4. Ayşe Python seçerse Bora ikinci sayılara bakar: 2, 1, 0 → 2 (Python).
5. Ayşe Java seçerse: 1, 4, 2 → 4 (Java).
6. Ayşe C seçerse: 0, 2, 1 → 2 (Java).
7. İki sayının da altı çizili olduğu hücreler (Python, Python) = (4, 2) ve (Java, Java) = (2, 4)'tür. (C, C)'de yalnız ilk sayı çizilidir: Bora Java'ya geçip 1 yerine 2 alır, bu yüzden (C, C) denge değildir.



Şekil 3.4: Dil seçimi oyununda altını çizme. C sütununda Ayşe'nin üç kazancı eşit (1, 1, 1) olduğu için üçünün

de altı çizilir. İki çizgili hücreler (Python, Python) ve (Java, Java) saf dengelerdir.

Eşitlikte yalnız birini işaretlemek sık yapılan bir hatadır. Bu örnekte hata zararsız görünebilir. Başka bir oyunda ise eşit kazançlı hücrelerden biri tam da dengenin bulunduğu hücre olabilir; tek işaret o dengeyi gözden kaçırır. Kural nettir: en büyük değer birden çok kez görülüyorsa hepsi işaretlenir.

Aynı oyun 3.3'teki baskın strateji elemesiyle de incelenebilir. Bora için Java, C'yi her satırda kesin olarak yener (1 > 0, 4 > 2, 2 > 1); bu yüzden C sütunu silinebilir. Geriye kalan iki sütunda Ayşe için Python, C'yi yener (4 > 1, 1 > 0) ve C satırı da silinir. Kalan 2 × 2 oyunda hiçbir strateji baskılanmaz; iki denge birlikte kalır. Yani eleme oyunu küçültür ama çözmez; altını çizme ise iki dengeyi doğrudan bulur. Oyunun ayrıca bir karışık dengesi de vardır (Ayşe Python'u 3/4, Bora Python'u 1/5 olasılıkla seçer); bu tür dengeler 3.4'te hesaplanır.

3.2.5 İşaretlemede sık hatalar

İşaretleme mekanik bir yöntemdir, ama hangi sayının hangi yönde karşılaştırılacağı karıştırılabilir. Şekil 3.5'te doğru yönler gösterilmiştir: satır oyuncusu yalnız satırını değiştirebildiği için kendi seçeneklerini *aynı sütun içinde* karşılaştırır; sütun oyuncusu ise *aynı satır içinde* karşılaştırır.



Şekil 3.5: Karşılaştırma yönleri. Solda satır oyuncusu aynı sütunda ilk sayıları (3 ile 1) karşılaştırır; sağda sütun oyuncusu aynı satırda ikinci sayıları (0 ile 2) karşılaştırır.

Sık yapılan hata

- **Yanlış yön:** Satır oyuncusunun en iyisini satır boyunca aramak. O yalnız satırını değiştirir; aynı sütunda bakılır.
- **Yanlış sayı:** Sütun oyuncusu için ilk sayılara bakmak. Sütun oyuncusu ikinci sayıyı büyütmek ister.
- **Eşitlikte tek işaret:** En büyük değer iki kez geçiyorsa ikisi de en iyi tepkidir, ikisi de işaretlenir.

- **Tek işaret yeter sanmak:** Yalnız bir oyuncunun işareti olan hücre denge değildir; iki işaret gerekir.
- **En büyük hücreyi aramak:** Denge, sayıları büyük olan hücre değil, karşılıklı en iyi tepki olan hücredir.
- **Kontrol etmemek:** Her aday hücrede sapma testi yapılmalıdır: oyuncu başka seçeneğe geçince daha çok alıyor mu?

3.2.6 Üç oyuncu: grup ödevi oyunu

Tanım iki oyuncuyla sınırlı değildir. Üç öğrenci bir grup ödevi için aynı anda *Katıl* ya da *Katılma* der. En az iki kişi katılırsa ödev başarılı olur ve herkes (katılan da katılmayan da) 3 puan alır. Katılmanın kişisel maliyeti 1'dir. Üç oyuncu ve ikişer strateji olduğu için $2 \times 2 \times 2 = 8$ profil vardır. Oyun simetrik olduğu için profilleri katılan sayısına (k) göre gruplamak yeterlidir.

Katılan sayısı	Kazançlar	Tek taraflı sapma	Denge mi?
k = 3 (herkes)	3 − 1 = 2'şer	Biri bırakırsa ödev yine başarılı (k = 2), bırakan 3 alır: 3 > 2	Hayır
k = 2	katılanlar 2, katılmayan 3	Katılan bırakırsa ödev batar: 0 < 2. Katılmayan katılırsa 2 < 3	Evet (3 profil)
k = 1	katılan −1, ötekiler 0	Katılan bırakırsa 0 > −1	Hayır
k = 0 (kimse)	0'ar	Tek başına katılan −1 alır: −1 < 0	Evet

Tablo 3.5: Grup ödevi oyununda her profilin sapma testi.

Oyunun dört saf dengesi vardır: tam iki kişinin katıldığı üç profil ve kimsenin katılmadığı profil. Test n oyunculu oyunda da aynıdır: ötekiler sabitken her oyuncu ayrı ayrı denetlenir. Üç oyuncuda tek bir "matris" çizilemez, ama tanım hiç değişmez.

"Kimse katılmaz" dengesi ilginçtir. Herkes 0 alır, oysa iki kişi katılsa katılanlar 2, katılmayan 3 alırdı. Kimse bu dengeyi tek başına bozamaz, çünkü tek başına katılan ödevi kurtaramaz ve −1 alır. Bu duruma **koordinasyon başarısızlığı** denir. İki öğrenci birlikte katılmaya karar verse durum düzelirdi, ama birlikte sapmak Nash tanımının dışındadır. Nash dengesi böylece kötü bir durumda "sıkışıp kalmayı" da açıklar.

3.2.7 Kod izleme: saf_nash()

İşaretleme yöntemi kısa bir Python fonksiyonuyla yazılabilir. Fonksiyon, satır oyuncusunun kazanç matrisi `U1` ile sütun oyuncusunun kazanç matrisi `U2`'yi alır ve saf dengelerin (satır, sütun) indislerini döndürür.

```
def saf_nash(U1, U2):
    m, n, denge = len(U1), len(U1[0]), []
    for i in range(m):
        for j in range(n):
            br1 = max(U1[k][j] for k in range(m))
            br2 = max(U2[i][l] for l in range(n))
            if U1[i][j] == br1 and U2[i][j] == br2:
                denge.append((i, j))
    return denge
```

Kod satır satır şöyle çalışır:

- `m` satır sayısı, `n` sütun sayısıdır; `denge` boş bir listedir.
- İki iç içe döngü bütün hücrelerde gezinir: `i` satır, `j` sütun indisidir.
- `br1`, `j` sütunundaki en büyük `U1` değeridir. Satır oyuncusunun bu sütuna karşı alabileceği en iyi kazanç budur.
- `br2`, `i` satırındaki en büyük `U2` değeridir. Sütun oyuncusunun bu satıra karşı alabileceği en iyi kazanç budur.
- Hücredeki iki kazanç da kendi maksimumuna eşitse (iki işaret de varsa) hücre listeye eklenir. Eşitlik (`==`) kullanıldığı için eşit en büyük değerlerin hepsi kabul edilir; eşitlik sorunu kendiliğinden çözülür.

Tablo 3.3'teki oyun için `U1 = [[4, 0, 1], [2, 3, 0], [1, 2, 3]]` ve `U2 = [[3, 2, 1], [1, 3, 2], [4, 0, 2]]` 'dir; indis 0, 1, 2 sırasıyla A, B, C (ya da X, Y, Z) demektir. Sütun maksimumları X: 4, Y: 3, Z: 3; satır maksimumları A: 3, B: 3, C: 4'tür. İzleme tablosu dokuz hücreyi sırayla gösterir.

(i, j)	Hücre	U1[i][j] / br1	U2[i][j] / br2	Koşul	denge
(0, 0)	(A, X)	4 / 4	3 / 3	doğru	[(0, 0)]
(0, 1)	(A, Y)	0 / 3	2 / 3	yanlış	aynı
(0, 2)	(A, Z)	1 / 3	1 / 3	yanlış	aynı
(1, 0)	(B, X)	2 / 4	1 / 3	yanlış	aynı
(1, 1)	(B, Y)	3 / 3	3 / 3	doğru	[(0, 0), (1, 1)]
(1, 2)	(B, Z)	0 / 3	2 / 3	yanlış	aynı
(2, 0)	(C, X)	1 / 4	4 / 4	yanlış	aynı
(2, 1)	(C, Y)	2 / 3	0 / 4	yanlış	aynı
(2, 2)	(C, Z)	3 / 3	2 / 4	yanlış	aynı

Tablo 3.6: `saf_nash(U1, U2)` izlemesi.

Dokuz hücre tarandıktan sonra fonksiyon `[(0, 0), (1, 1)]` döndürür, yani (A, X) ve (B, Y). Dikkat edilirse (C, X)'te yalnız ikinci koşul, (C, Z)'de yalnız birinci koşul doğrudur; `and` ikisini birden istediği için bu hücreler elenir. Bu, “tek işaret yetmez” kuralının koddaki karşılığıdır.

Karmaşıklık açısından bu sürüm her hücrede bir sütun maksimumu (m karşılaştırma) ve bir satır maksimumu (n karşılaştırma) hesaplar. Toplam iş $O(mn \cdot (m + n))$'dir. Maksimumlar döngüden önce bir kez hesaplanıp bir listede saklanırsa iş $O(mn)$'ye düşer. Küçük matrislerde fark önemsizdir; 1000 × 1000 bir matriste ise yaklaşık 2000 kat fark eder.

3.2.8 Sürekli stratejiler: Cournot duopolü

Şimdiye kadar her oyuncunun birkaç seçeneği vardı. Bazı oyunlarda strateji bir sayıdır ve sonsuz sayıda değer alabilir. En eski örnek Antoine Augustin Cournot'nun 1838'de incelediği iki firmalı piyasadır; iki firmalı piyasaya **duopol** denir.

İki firma aynı malı üretir ve aynı anda üretim miktarı $y_1, y_2 \geq 0$ seçer. Piyasa fiyatı toplam üretime bağlıdır: $P = 12 - (y_1 + y_2)$. Ne kadar çok üretilirse fiyat o kadar düşer. Üretim maliyeti sıfırdır. Firma 1'in kârı miktar çarpı fiyattır:

$$u_1(y_1, y_2) = y_1 \cdot (12 - y_1 - y_2)$$

Burada matris çizilemez, çünkü satır sayısı sonsuzdur. Ama en iyi tepki fikri aynen geçerlidir: firma 2'nin y_2 miktarı sabitken firma 1'in kârını en büyük yapan y_1 aranır. Seçim sürekli olduğunda argmax tablodan okunmaz; **türev** ile bulunur.

Türev, bir eğrinin her noktadaki eğimidir. Kâr eğrisinin tepesinde eğim sıfırdır: tepenin solunda eğim artı (kâr artıyor), sağında eksidir (kâr azalıyor). Burada üç kural yeter: y^2'nin türevi $2y$'dir; $c \cdot y$'nin türevi c'dir; sabitin türevi 0'dır. Toplamın türevi türevlerin toplamıdır.

Örnek 3.5

1. Kârı açık yaz. $u_1 = 12y_1 - y_1^2 - y_1y_2$. Firma 1 için y_2 bir sabittir.

2. Türevi al. $12y_1$'in türevi 12, y_1^2'nin türevi $2y_1$, y_1y_2'nin türevi y_2'dir. Sonuç: $12 - 2y_1 - y_2$.

3. Sıfıra eşitle. $12 - 2y_1 - y_2 = 0 \Rightarrow y_1 = (12 - y_2)/2$.

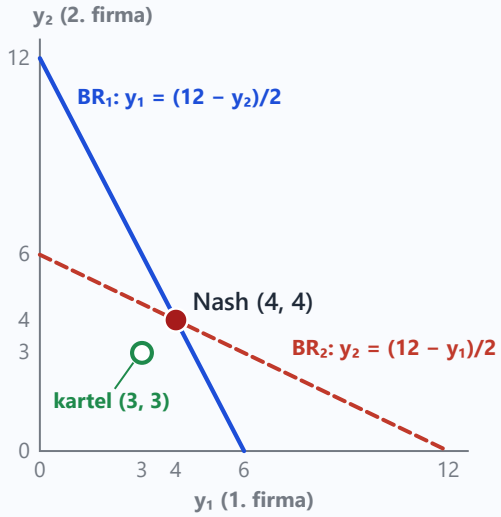
4. Tepe mi, çukur mu? Türevin türevi (ikinci türev) −2'dir ve negatiftir; eğim azalıyordur, nokta bir tepedir. Sayıyla da denetlenebilir: $y_2 = 4$ iken $u_1(4) = 4 \cdot 4 = 16$, komşularda $u_1(3) = 3 \cdot 5 = 15$ ve $u_1(5) = 5 \cdot 3 = 15$; ikisi de 16'dan küçüktür ✓.

5. Sınırı unutma. Miktar negatif olamaz. $y_2 \geq 12$ ise formül sıfır ya da eksi verir; bu durumda fiyat zaten sıfırdır ve en iyisi üretmemektir. Tam en iyi tepki:

$$BR_1(y_2) = \max\{0, (12 - y_2)/2\}$$

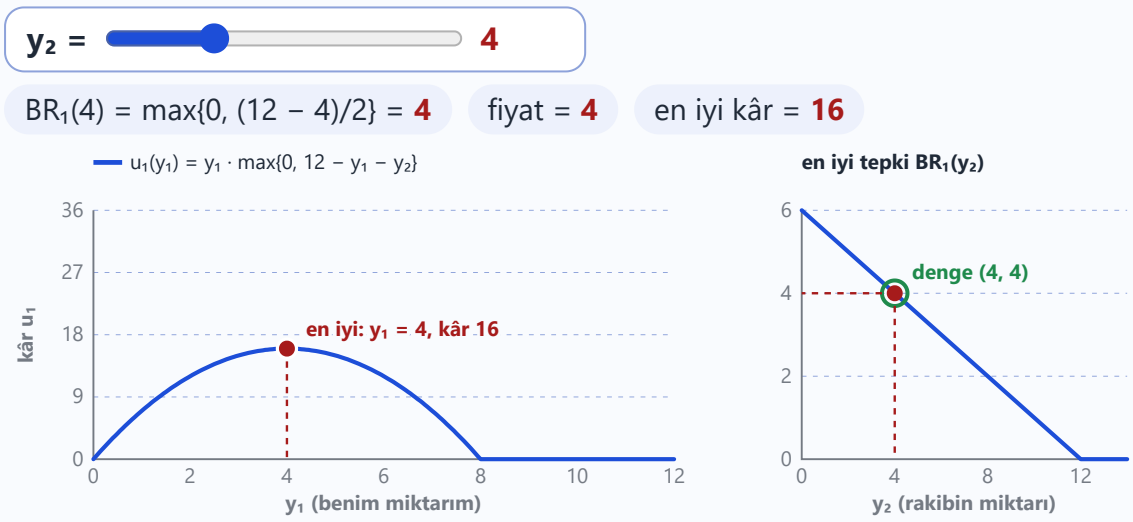
6. Dengeyi bul. Oyun simetrik olduğu için $BR_2(y_1) = \max\{0, (12 - y_1)/2\}$'dir. Denge, iki firmanın birbirine aynı anda en iyi tepki verdiği noktadır: $y_1 = (12 - y_2)/2$ ve $y_2 = (12 - y_1)/2$. Simetriden $y_1 = y_2 = y$ konursa $2y = 12 - y$, yani $y = 4$. Fiyat $12 - 8 = 4$, her firmanın kârı $4 \cdot 4 =$

16'dır.



Şekil 3.6: Cournot duopolünde en iyi tepki doğruları. Düz mavi doğru BR_1, kesikli kırmızı doğru BR_2'dir. Kesişim (4, 4) Nash dengesidir; yeşil (3, 3) noktası kartel üretimidir ve hiçbir doğrunun üzerinde değildir.

Şekil 3.6'da iki en iyi tepki doğrusu çizilmiştir. Yatay eksen firma 1'in, dikey eksen firma 2'nin miktarıdır. BR_1 doğrusu, firma 2'nin her miktarı için firma 1'in en iyi miktarını gösterir; $y_2 = 0$'da 6'dan başlar, $y_2 = 12$'de 0'a iner. Denge iki doğrunun kesiştiği tek noktadır. Doğrunun eğimi negatiftir: rakip daha çok ürettikçe firmanın en iyi miktarı azalır.



Şekil 3.7: Etkileşimli Cournot en iyi tepkisi. Kaydırıcı rakibin miktarı y_2'yi değiştirir; solda firma 1'in kâr eğrisi $u_1(y_1) = y_1 \cdot max\{0, 12 - y_1 - y_2\}$ ve tepe noktası, sağda en iyi tepki doğrusu üzerindeki nokta görünür. Varsayılan değer $y_2 = 4$'tür ve en iyi tepki 4'tür.

Kartel neden dengede değildir? İki firma anlaşıp toplam kârı en büyük yapmak isteseydi, toplam miktar Y için $Y(12 - Y)$'yi en büyük yapardı. Türev $12 - 2Y = 0$'dan $Y = 6$ çıkar; her firma 3 üretir, fiyat 6 olur ve her firma $3 \cdot 6 = 18$ kazanır. Bu, dengedeki 16'dan fazladır. Ancak anlaşma kendini doğrulamaz. Rakip 3 üretirken firma 1'in en iyi tepkisi $BR_1(3) = (12 - 3)/2 = 4{,}5$'tir. Sapan firma $4{,}5 \cdot (12 - 4{,}5 - 3) = 4{,}5 \cdot 4{,}5 = 20{,}25$ kazanır; anlaşmaya sadık kalan firmanın kârı ise $3 \cdot 4{,}5 = 13{,}5$'e düşer. Her firmanın sapma isteği vardır. Bu, Mahkûmun İkilemi'nin sürekli stratejili bir sürümüdür.

Denge nasıl bulunabilir? Firmaların dengeyi hesapladığını varsaymak zorunda değiliz. Firmalar sırayla birbirine en iyi tepki verirse üretimler dengeye yaklaşır. Başlangıçta firma 1 tek başına 6 üretsin (rakip 0 üretirken en iyi tepki). Firma 2 buna $(12 - 6)/2 = 3$ ile cevap verir. Firma 1 de 3'e $4{,}5$ ile cevap verir ve süreç böyle sürer.

Adım	Güncelleyen	Hesap	(y_1, y_2)
0	başlangıç	firma 1 tek başına	(6, 0)
1	firma 2	(12 − 6)/2 = 3	(6, 3)
2	firma 1	(12 − 3)/2 = 4,5	(4,5, 3)
3	firma 2	(12 − 4,5)/2 = 3,75	(4,5, 3,75)
4	firma 1	(12 − 3,75)/2 = 4,125	(4,125, 3,75)
5	firma 2	(12 − 4,125)/2 = 3,9375	(4,125, 3,9375)

Tablo 3.7: Cournot'da sıralı en iyi tepkiler. Her satırda yalnız bir firma miktarını günceller.

Her adımda, güncellenen miktarın dengeye uzaklığı yarıya iner: 6 için 2, 3 için 1, 4,5 için 0,5, 3,75 için 0,25, 4,125 için 0,125. Nedeni, en iyi tepki doğrusunun eğiminin $-1/2$ olmasıdır. Böyle bir "öğrenme süreci" her oyunda dengeye varmaz. Burada vardığı için Cournot dengesi davranış açısından da makul bir tahmindir.

3.2.9 Bertrand: fiyatla rekabet

Firmalar miktar yerine fiyat seçerse sonuç şaşırtıcı biçimde değişir. Bu model Joseph Bertrand'ın adını taşır. İki firma aynı malı satar; birim maliyet $c = 2$ TL, pazar 100 birimdir. Firmalar aynı anda fiyat seçer. Ucuz olan pazarın hepsini alır; fiyatlar eşitse pazar yarı yarıya bölünür. Bir firmanın kârı (fiyat − maliyet) × satılan miktardır.

Örnek 3.6

1. **(5, 5) denge mi?** Her firma 50 birim satar ve $(5 - 2) \cdot 50 = 150$ kazanır. Firma 1 fiyatını 4,99'a indirirse pazarın hepsini alır: $2{,}99 \cdot 100 \approx 299$. Sapma kârın neredeyse iki katını getirir; (5, 5) denge değildir.
2. **c'nin üstündeki her ortak fiyat:** Ortak fiyat $p > 2$ ise her firma $(p - 2) \cdot 50$ kazanır; biraz aşağı kıran yaklaşık $(p - 2) \cdot 100$ kazanır. Hiçbiri denge değildir.

3. **Farklı fiyatlar:** Ucuz firmanın fiyatı c'nin üstündeyse, fiyatını rakibin fiyatının biraz altına kadar yükseltip yine pazarın hepsini alır ve daha çok kazanır. Ucuz firmanın fiyatı tam c ise kârı 0'dır; fiyatını c ile rakibin fiyatı arasına çıkarıp kâr eder. Farklı fiyatlı hiçbir profil denge değildir.
4. **c'nin altı:** Fiyatı 2'nin altına indiren her sattığı birimde zarar eder; bu hiçbir zaman en iyi tepki değildir.
5. **(2, 2):** İki firma da 0 kazanır. Aşağı inen zarar eder, yukarı çıkan hiç satamaz ve yine 0 alır. Hiçbir sapma kârı artırmaz. Tek Nash dengesi $p_1 = p_2 = c = 2$, kâr 0'dır.



Şekil 3.8: Bertrand'da fiyat kırma zinciri. c'nin üstündeki her ortak fiyattan biraz aşağı kırmak kârlı olduğu için fiyat adım adım $p = c = 2$'ye iner.

Rekabet aracı	Denge	Firma kârı
Cournot (miktar), P = 12 − Y, maliyet 0	y = 4'er, fiyat 4	16'şar
Bertrand (fiyat), maliyet c	p = c	0

Tablo 3.8: Cournot ve Bertrand'ın karşılaştırması.

İki model arasındaki fark, sapmanın ne getirdiğinden kaynaklanır. Cournot'da bir firma miktarını artırınca fiyat düşer ve bu düşüş kendi kârını da azaltır; sapma kendi kendini frenler. Bertrand'da ise küçük bir fiyat indirimi pazarın tamamını getirir; sapmanın ödülü çok büyüktür. Bu yüzden iki firma bile fiyatı maliyete kadar indirir. Bu sonuca **Bertrand paradoksu** denir: "iki firma varsa kâr yüksek kalır" sezgisi fiyat rekabetinde yanlıştır. Gerçek piyasalarda kâr, ancak ürün farkı, kapasite sınırı ya da tekrarlanan etkileşim gibi etkenlerle korunur.

Sık yapılan hata

"Bertrand'ın tek dengesi p = c'dir" sonucu, fiyatın sürekli olduğu varsayımına dayanır: 4,99 gibi istenen kadar küçük indirimler yapılabilir. Fiyatlar 1 TL'lik adımlarla seçilebiliyorsa c'nin hemen üstündeki fiyatlar da denge olabilir. Bu örnekte (3, 3)'te her firma $1 \cdot 50 = 50$ kazanır;

2'ye inen 0 kazanır, çıkan hiç satamaz, yani sapma kârsızdır. (4, 4)'te her firma 100 kazanır; 3'e inen $1 \cdot 100 = 100$ kazanır, yani sapma kazancı artırmaz. (5, 5)'te ise 4'e inen 200 kazanır ve sapma kârlıdır. Sonuç: 1 TL adımda dengeler (2, 2), (3, 3) ve (4, 4)'tür.

3.3 Baskın strateji eleme

En iyi tepki işaretlemesi dengeyi doğrudan arar. Baskın strateji elemesi ise ters yönden çalışır: "hiçbir rasyonel oyuncunun oynamayacağı" stratejileri bulup oyundan çıkarır ve oyunu küçültür. Bazen geriye tek bir hücre kalır ve oyun bu yolla tamamen çözülür.

3.3.1 Baskın ve baskılanan strateji

İki stratejiyi karşılaştırmanın en güçlü yolu, rakibin *her* olası seçimine karşı karşılaştırmaktır. Bir strateji ötekinden rakip ne yaparsa yapsın daha iyiyse, ikinci stratejiyi oynamak için hiçbir neden kalmaz.

Tanım 3.4

Oyuncu i'nin iki stratejisi s_i' ve s_i olsun.

- **Kesin baskılama** (*strict dominance*): Her s_{-i} için $u_i(s_i', s_{-i}) > u_i(s_i, s_{-i})$ ise s_i', s_i'yi kesin baskılar.
- **Zayıf baskılama** (*weak dominance*): Her s_{-i} için ≥ ve en az bir s_{-i} için > ise s_i', s_i'yi zayıf baskılar.

Baskılanan stratejiye **baskılanan strateji** (*dominated strategy*) denir. Oyuncunun diğer bütün stratejilerini kesin baskılayan stratejiye **baskın strateji** (*dominant strategy*) denir.

Tanımdaki "her" sözcüğü kritiktir. Kesin baskılama için karşılaştırma rakibin her seçimi için ayrı ayrı yapılır ve her birinde kesin büyüklük gerekir. "Ortalamada daha iyi" ya da "çoğu durumda daha iyi" yetmez.

Rasyonel oyuncu kesin baskılanan stratejiyi asla oynamaz. Nedeni açıktır: rakip hakkında ne düşünürse düşünsün, baskılayan strateji ona daha çok kazandırır. Kesin baskılanan bir strateji hiçbir rakip seçimine en iyi tepki değildir.

Baskın stratejisi olan oyuncu için oyun kolaydır: rakibini tahmin etmek zorunda değildir. Herkesin baskın stratejisi varsa, herkesin baskın stratejisini oynadığı profil bir Nash dengesidir; buna **baskın strateji dengesi** (*dominant-strategy equilibrium*) denir. Mahkûmun İkilemi'nde İhanet iki oyuncu için de baskın stratejidir ve (İhanet, İhanet) bir baskın strateji dengesidir. Ancak baskın strateji her oyunda bulunmaz. Taş-Kağıt-Makas'ta (TKM) her hamle bir başka hamleye yenilir; hiçbir hamle ötekileri her durumda yenmez.

Baskın strateji olmasa bile tek tek baskılanan stratejiler olabilir. Bunları silmek ve küçülen oyunda yeniden aramak bir yöntem verir.

Tanım 3.5

Yinelemeli eleme (*iterated elimination of strictly dominated strategies, IESDS*): Herhangi bir oyuncunun kesin baskılanan bir stratejisi silinir; küçülen oyunda yeniden kesin baskılanan strateji aranır. Silinecek strateji kalmayınca durulur.

Yöntem, oyuncuların rasyonel olmasından fazlasını varsayar. Herkesin rasyonel olduğu herkesçe bilinir, bu da herkesçe bilinir ve zincir sonsuza kadar sürer. Bu varsayıma **ortak bilgi** (*common knowledge*) denir. İlk turda silinen strateji, yalnız oyuncunun kendisinin rasyonel olmasından gelir. İkinci turda silinen strateji, oyuncunun rakibinin rasyonel olduğunu bilmesini gerektirir: "rakibim o baskılanan stratejiyi oynamaz, o hâlde o sütunu hesaba katmam". Her yeni tur, bir kat daha derin bilgiye dayanır.

Teorem 3.2 (Kesin elemenin güvenliği)

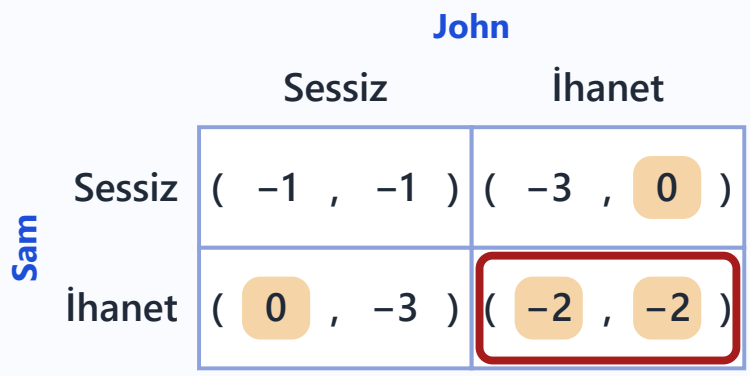
Kesin baskılanan stratejilerin yinelemeli elemesinde (a) hiçbir Nash dengesi silinmez, (b) sonuç silme sırasından bağımsızdır. Özellikle geriye tek hücre kalırsa o hücre oyunun tek Nash dengesidir.

İspat. (a) kısmının fikri şöyledir. Bir Nash dengesinde her oyuncunun stratejisi, ötekilerin denge stratejilerine karşı bir en iyi tepkidir. Kesin baskılanan bir strateji ise hiçbir rakip seçimine en iyi tepki olamaz, çünkü baskılayan strateji o seçime karşı da daha çok kazandırır. Demek ki kesin baskılanan bir strateji hiçbir dengede yer almaz ve onu silmek hiçbir dengeyi bozmaz. Silme sonrası küçülen oyunda da dengedeki stratejiler hâlâ en iyi tepkidir, çünkü onları yenen bir seçenek eklenmemiş, yalnızca seçenekler azalmıştır. Aynı akıl yürütme her turda tekrarlanır. (b) kısmının ispatı daha uzundur ve burada verilmez (Osborne, 2004, Bölüm 2). □

Zayıf baskılama için bu güvence yoktur. Zayıf baskılanan bir strateji, bazı durumlarda baskılayanla *eşit* kazanç verir; o durumlarda bir en iyi tepki olabilir ve bir dengede yer alabilir. Silindiğinde bazı dengeler kaybolabilir ve sonuç silme sırasına bağlı olabilir (3.3.7).

Tanıdık bir örnek bu farkı gösterir. **2/3 ortalaması oyununda** herkes 0 ile 90 arasında bir sayı yazar; ortalamanın 2/3'üne en yakın sayı kazanır. Ortalama en fazla 90 olabileceği için hedef en fazla 60'tır. 60'ın üstündeki bir tahmin, hiçbir durumda 60 yazmaktan iyi değildir; ama bazı durumlarda (örneğin iki tahmin de hedeften uzak kalıp ikisi de kaybettiğinde) eşit sonuç verir. Bu yüzden 60'ın üstündeki sayıları silmek zayıf baskılamaya dayanır, kesin baskılamaya değil.

3.3.2 Mahkûmun İkilemi'ne dönüş



Şekil 3.9: Mahkûmun İkilemi, kazanç = −yıl. Her oyuncunun en iyi tepkisi işaretlidir; iki işaret de İhanet satırı ve sütunundadır.

Sam sessiz mi kalmalı, ihanet mi etmeli? Kuramın cevabı nettir: İhanet baskın stratejidir, çünkü John sessizse *0 > −1*, ihanet ederse *−2 > −3*'tür. Tek Nash dengesi (İhanet, İhanet) = (−2, −2), yani ikişer yıldır (Şekil 3.9).

Gerçek davranış kuramdan farklıdır. Tek seferlik oyunda insanlar kuramın öngördüğünden çok daha sık işbirliği yapar. 1958–1992 arasındaki deneyleri birleştiren bir meta-analize göre katılımcıların kabaca yarısı işbirliği yapar ve oyun öncesinde konuşma izni verilmesi bu oranı belirgin biçimde artırır (Sally, 1995). Güven, adalet ve karşılıklılık duyguları da seçimi etkiler. Bu, kuramın yanlış olduğunu göstermez. Kazanç olarak yalnız hapis yılı yazılınca insanların önemsediği başka şeyler (suçluluk duygusu, itibar) tablonun dışında kalır. Kuram, bu etkenler olmasaydı ne olacağını gösteren açık bir karşılaştırma noktasıdır.

Tekrarlanan oyun. Oyun aynı rakiple tekrar tekrar oynanınca durum değişir. Bilgisayar Tit-for-Tat oynasın: ilk turda sessiz kalır, sonra karşısındakinin bir önceki seçimini kopyalar. Sam'in 10 turluk farklı planlarının toplam maliyeti şöyledir.

Sam'in planı	Tur tur ne olur?	Sam'in toplam yılı
Hep Sessiz	John da hep sessiz kalır; her tur 1 yıl.	10 × 1 = 10
Hep İhanet	1. tur John sessiz: Sam 0. Sonraki 9 turda John da ihanet eder: 2'şer yıl.	0 + 9 × 2 = 18
9 tur Sessiz, son tur İhanet	İlk 9 tur 1'er yıl; 10. turda John sessiz, Sam 0 alır ve ceza gelemez.	9 × 1 + 0 = 9 (en az)

Tablo 3.9: John Tit-for-Tat oynarken Sam'in 10 turluk planları.

Tek seferlik oyunda baskın olan İhanet, Tit-for-Tat'a karşı pahalıdır: her ihanet bir sonraki turda cezalandırılır. Yalnız son turda ceza gelemez, çünkü sonraki tur yoktur. Bu yüzden en iyi plan dokuz tur sessiz kalıp son turda ihanet etmektir. $2^{10} = 1024$ planın hepsini bilgisayarla denemek de aynı sonucu

verir: en küçük toplam 9 yıldır. Bu “son tur etkisi”, oyun ağacında sondan başa doğru akıl yürüten **geriye doğru çıkarım**ın (*backward induction*) konusudur.

3.3.3 Yinelemeli eleme: üçe üç örnek

Aşağıdaki oyunda hiçbir oyuncunun baskın stratejisi yoktur, ama yinelemeli eleme oyunu tek hücreye indirir.

Satır oyuncusu \ Sütun oyuncusu	X	Y	Z
A	(3, 2)	(3, 3)	(1, 4)
B	(3, 2)	(0, 4)	(5, 3)
C	(2, 1)	(2, 0)	(6, 6)

Şekil 3.10: Yinelemeli eleme örneği. Satır oyuncusu A, B, C; sütun oyuncusu X, Y, Z seçer; hücrede (u_1, u_2).

Örnek 3.7

1. **Başlangıç.** Satırlar ilk sayılarla karşılaştırılır. Hiçbir satır diğerini kesin baskılamaz; örneğin A ile B, X sütununda eşittir (3 = 3). Satırlarda ilerlenemediği için sütun oyuncusuna bakılır.
2. **X elenir.** Sütunlar ikinci sayılarla karşılaştırılır. Z, X'i her satırda yener: A'da 4 > 2, B'de 3 > 2, C'de 6 > 1. X kesin baskılanır ve silinir.
3. **B elenir.** Artık yalnız Y ve Z sütunları vardır. C, B'yi bu iki sütunda yener: Y'de 2 > 0, Z'de 6 > 5. (X dururken bu karşılaştırma geçersizdi: X'te B 3, C 2 alır.)
4. **Y elenir.** Kalan satırlar A ve C'dir. Z, Y'yi iki satırda da yener: A'da 4 > 3, C'de 6 > 0.
5. **A elenir.** Yalnız Z sütunu kaldı. C, A'yı yener: 6 > 1.
6. **Sonuç.** Tek hücre kalır: (C, Z) = (6, 6). Teorem 3.2'ye göre bu hücre oyunun tek Nash dengesidir. En iyi tepki işaretlemesi de aynı sonucu verir.

Adım	Kalan satırlar	Kalan sütunlar	Silinen	Gerekçe
1	A, B, C	X, Y, Z	X	Z > X: 4 > 2, 3 > 2, 6 > 1

Adım	Kalan satırlar	Kalan sütunlar	Silinen	Gerekçe
2	A, B, C	Y, Z	B	C > B: 2 > 0, 6 > 5
3	A, C	Y, Z	Y	Z > Y: 4 > 3, 6 > 0
4	A, C	Z	A	C > A: 6 > 1
5	C	Z	—	(C, Z) = (6, 6)

Tablo 3.10: Eleme sırası ve her adımdaki karşılaştırma.

3.3.4 Bir eleme ötekinin yolunu açar

Yinelemeli elemenin gücü, her silmenin yeni silmelere kapı açmasındadır. Aşağıdaki oyunda başta tek bir kesin baskılama vardır; ötekilerin hepsi ondan sonra ortaya çıkar. Satırlar yalnız ilk sayılarla, sütunlar yalnız ikinci sayılarla karşılaştırılır.

	Sol	Merkez	Sağ
Üst	(1, 5)	(5, 6)	(6, 2)
Orta	**(3, 5)**	(6, 0)	(2, 2)
Alt	(2, 2)	(0, 2)	(1, 5)

Tablo 3.11: Eleme zinciri oyunu. Hücrede (satır, sütun) kazancı.

Örnek 3.8

1. **Alt silinir.** Orta, Alt'ı her sütunda yener: 3 > 2, 6 > 0, 2 > 1. Başka kesin baskılama yoktur: örneğin Sol, Sağ'ı yenemez, çünkü Alt satırında Sağ 5, Sol 2 verir.
2. **Sağ silinir.** Alt gidince Sol, kalan iki satırda Sağ'ı yener: Üst'te 5 > 2, Orta'da 5 > 2. Alt dururken bu olmazdı.
3. **Üst silinir.** Sağ gidince Orta, Üst'ü kalan iki sütunda yener: Sol'da 3 > 1, Merkez'de 6 > 5. Sağ dururken Üst orada 6 > 2 ile kurtuluyordu.
4. **Merkez silinir.** Yalnız Orta satırı kaldı: Sol 5, Merkez 0. Merkez silinir.
5. **Sonuç.** Kalan (Orta, Sol) = (3, 5) tek Nash dengesidir.



Şekil 3.11: Eleme zincirinin son hâli. Önce Alt, sonra Sağ, sonra Üst satırı, en son Merkez sütunu silinmiştir (kırmızı çizgiler). Kalan (Orta, Sol) tek dengedir.

Örnekten iki ders çıkar. Birincisi, silme sırası mantıksal bir zincirdir: Üst'ün silinmesi Sağ'ın, Sağ'ın silinmesi de Alt'ın silinmesine bağlıdır. Her adım, bir öncekinin varsaydığı rasyonellik bilgisine bir kat daha ekler. İkincisi, denge herkes için en iyi sonuç değildir. (Üst, Merkez) = (5, 6) iki oyuncuya da (3, 5)'ten daha iyidir. Ama denge değildir: sütun oyuncusu Merkez oynarken satır oyuncusu Orta'ya geçip 5 yerine 6 alır.

3.3.5 Kod izleme: ele()

Yinelemeli eleme de kısa bir fonksiyonla yazılır. Fonksiyon, kalan satır indislerini `R`'de, kalan sütun indislerini `S`'de tutar.

```
def ele(U1, U2):
    R, S = [0, 1, 2], [0, 1, 2]   # kalanlar
    while True:
        x = [r for r in R for k in R if
             all(U1[k][c] > U1[r][c] for c in S)]
        y = [c for c in S for d in S if
             all(U2[r][d] > U2[r][c] for r in R)]
        if x: R.remove(x[0])      # satır elenir
        elif y: S.remove(y[0])    # sütun elenir
        else: return R, S
```

Kodun mantığı şöyledir:

- `x`, kalan satırlar içinde başka bir kalan satır `k` tarafından kesin baskılanan satırların listesidir. `all(U1[k][c] > U1[r][c] for c in S)`, `k` satırının `r` satırını kalan *her* sütunda kesin yendiğini denetler. Silinmiş sütunlar `S`'de olmadığı için karşılaştırmaya girmez.

- `y`, aynı denetimi sütunlar için `U2` ile yapar: `d` sütunu `c` sütununu kalan her satırda yeniyor mu?
- Önce satır silmeye bakılır: `x` boş değilse ilk baskılanan satır silinir. Yoksa sütuna bakılır. İkisi de boşsa kalanlar döndürülür.
- `while True` döngüsü her silmeden sonra baştan başlar; böylece bir silmenin açtığı yeni baskılamalar bir sonraki turda bulunur.

Üçe üç eleme örneğinin matrisleriyle çağrıldığında izleme şöyledir:

Tur	R	S	x	y	İşlem
1	[0, 1, 2]	[0, 1, 2]	[]	[0]	`S.remove(0)` : X silinir (Z baskılar)
2	[0, 1, 2]	[1, 2]	[1]	[]	`R.remove(1)` : B silinir (C baskılar)
3	[0, 2]	[1, 2]	[]	[1]	`S.remove(1)` : Y silinir (Z baskılar)
4	[0, 2]	[2]	[0]	[]	`R.remove(0)` : A silinir (C baskılar)
5	[2]	[2]	[]	[]	`return [2], [2]`

Tablo 3.12: `ele(U1, U2)` izlemesi (0, 1, 2 = A, B, C ve X, Y, Z).

Fonksiyon `([2], [2])` döndürür, yani yalnız (C, Z) kalır. Dört elemede sırayla X, B, Y, A silinmiştir. 2. turda `y`'nin boş çıkması önemlidir: o anda kalan üç satırda Z, Y'yi yenemez, çünkü B satırında Y 4, Z 3 verir. B silindikten sonra (3. tur) Z, Y'yi yener. Bu, 3.3.4'teki “bir eleme ötekinin yolunu açar” gözleminin koddaki karşılığıdır.

Her turda `x` en çok $m^2 \cdot n$, `y` en çok $n^2 \cdot m$ karşılaştırma yapar ve en çok $m + n$ tur vardır. Toplam iş polinomdur; küçük oyunlarda hesap anında biter.

3.3.6 Karışık stratejiyle baskılama

Bir stratejiyi yenen tek bir saf strateji olmayabilir, ama iki stratejinin bir karışımı onu yenebilir. Aşağıdaki tabloda yalnız satır oyuncusunun kazançları (u_1) verilmiştir, çünkü satırlar yalnız bu sayılarla karşılaştırılır.

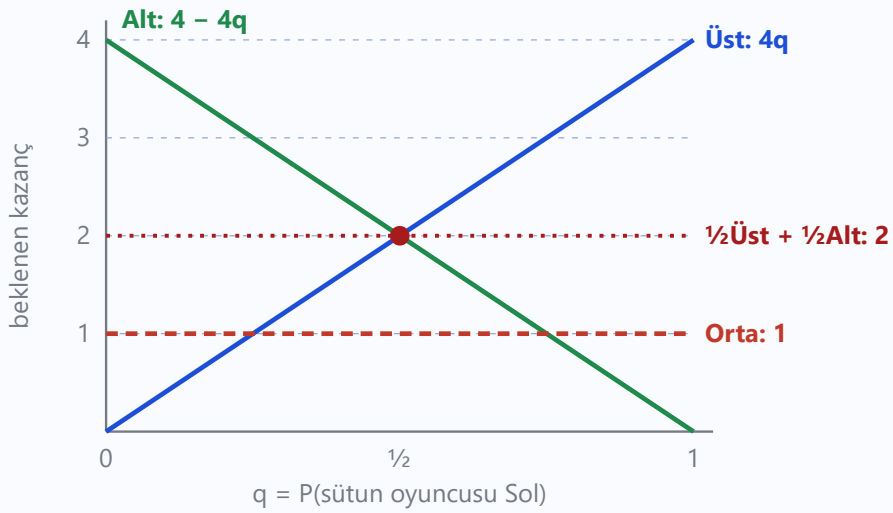
u_1	Sol	Sağ
Üst	4	0
Orta	1	1
Alt	0	4

Tablo 3.13: Satır oyuncusunun kazançları.

Orta'yı hiçbir saf satır baskılamaz: Sağ sütununda Üst'ten (1 > 0), Sol sütununda Alt'tan (1 > 0) iyidir. Ama satır oyuncusu yazı-tura atıp yarı olasılıkla Üst, yarı olasılıkla Alt oynarsa, her sütunda **beklenen kazancı** şöyle olur:

- Sol'a karşı: *½ · 4 + ½ · 0 = 2*.
- Sağ'a karşı: *½ · 0 + ½ · 4 = 2*.

İki sütunda da 2 > 1'dir. Demek ki ½–½ karışımı Orta'yı kesin baskılar ve Orta silinebilir. Beklenen kazancın ayrıntısı 3.4.1'dedir; burada "çok sayıda tekrarda tur başına ortalama kazanç" anlamı yeterlidir.



Şekil 3.12: Karışık stratejiyle baskılama. Yatay eksen sütun oyuncusunun Sol oynama olasılığı *q*'dur. Üst'ün beklenen kazancı *4q*, Alt'ınki *4 − 4q*, Orta'nınki her zaman 1'dir. ½Üst + ½Alt her *q* için 2 verir.

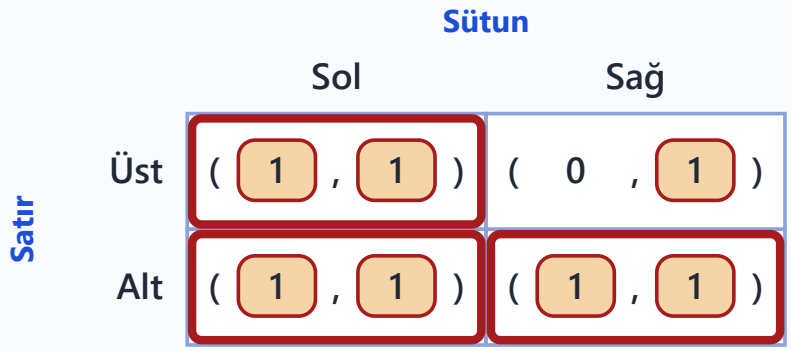
Şekil 3.12 sonucu bir adım ileri taşır. Satır oyuncusu, sütun oyuncusunun Sol'u *q* olasılıkla oynayacağına inansın. Üst'ün beklenen kazancı *4q*, Alt'ınki *4(1 − q)*'dur. Bu iki doğrunun büyük olanı (üst zarf) en az 2'dir; en düşük noktası *q = ½*'deki kesişimdir. Orta'nın doğrusu ise her yerde 1'dir ve üst zarfın tamamen altında kalır. Sonuç: Orta, sütun oyuncusu hakkındaki *hiçbir* inanca karşı en iyi tepki değildir. Karışık stratejiyle kesin baskılanmak ile "hiçbir inanca karşı en iyi tepki olmamak" iki kişilik oyunlarda aynı şeydir.

Sık yapılan hata

"Onu yenen tek bir saf strateji yok, o hâlde silinemez." Karışımlar da karşılaştırmaya girer. Eleme yaparken önce saf stratejiler, sonra ½–½ gibi basit karışımlar denenir. Özellikle iki uç stratejinin "ortasında" kalan ve hiçbir durumda en iyi olmayan stratejiler bu yolla silinir.

3.3.7 Zayıf baskılamanın tuzağı

Zayıf baskılananları silmenin neden tehlikeli olduğunu küçük bir oyun gösterir.



Şekil 3.13: Zayıf baskılama örneği. Sütun oyuncusu her hücrede 1 alır. İki işaretli üç hücre, yani üç saf denge vardır.

Örnek 3.9

1. Satır oyuncusu için Alt, Üst'ü zayıf baskılar: Sol'da 1 = 1, Sağ'da 1 > 0. Sol'daki eşitlik yüzünden baskılama kesin değildir.
2. Sütun oyuncusu her hücrede 1 aldığı için her durumda kayıtsızdır; her sütun her satıra karşı en iyi tepkidir.
3. Saf dengeler: (Üst, Sol), (Alt, Sol), (Alt, Sağ). (Üst, Sağ) denge değildir, çünkü satır oyuncusu Alt'a geçip 0 yerine 1 alır. Toplam üç denge.
4. Üst silinirse geriye yalnız Alt satırı kalır ve iki denge kalır: (Alt, Sol) ve (Alt, Sağ). (Üst, Sol) dengesi kaybolmuştur.

Kesin baskılama güvenlidir; zayıf baskılama ise denge kaybettirebilir. Birden çok zayıf baskılama olduğunda, hangisinin önce silindiğine göre farklı hücreler de kalabilir. Bu nedenle denge ararken yalnız kesin baskılananlar silinir.

Kesin eleme her oyunda tek hücreye de inmez. **Cinsiyetler Savaşı**'nda (*Battle of the Sexes*) iki kişi akşam Opera'ya mı Futbol'a mı gideceklerine ayrı ayrı karar verir. Biri Opera'yı, öteki Futbol'u daha çok sever; ama ikisi de ayrı yerlere gitmektense birlikte olmayı tercih eder. Bu oyunda hiçbir strateji baskılanmaz, eleme hiçbir şey silmez. Dengeler en iyi tepki işaretlemesiyle ve karışık stratejilerle bulunur.

3.3.8 Elemede sık hatalar

Sık yapılan hata

- **Yanlış sayılar:** İki satırı sütun oyuncusunun kazançlarıyla karşılaştırmak. Satır ↔ ilk sayı, sütun ↔ ikinci sayı.

- **“Çoğunlukla büyük” yeter sanmak:** Kesin baskılama için kalan her sütunda büyük olmalı; tek bir eşitlik ya da küçüklük elemeyi durdurur.
- **Silinenleri saymak:** Silinmiş sütun artık karşılaştırmaya girmez. Bir eleme zincirinde bir strateji (örneğin Üst) ancak başka bir strateji (Sağ) silindikten sonra elenebilir.
- **Zayıfı kesin sanmak:** Eşitlik varsa baskılama zayıftır; silmek denge kaybettirebilir ve sonuç sıraya bağlı olabilir.
- **Baskın ile baskılananı karıştırmak:** Baskın strateji diğer hepsini yener; baskılanan strateji en az birine yenilir. Baskın strateji olmadan da eleme yapılır.
- **Kalanların hepsini denge sanmak:** Eleme birden çok hücre bırakırsa (örneğin dil seçimi oyununda 2 × 2 kalır), dengeler kalan küçük oyunda altını çizerek bulunur.

3.4 Karışık strateji

Bazı oyunlarda hiçbir hücre karşılıklı en iyi tepki değildir. Yazı-Tura Eşleştirme'de iki oyuncu aynı anda bir madeni paranın yüzünü seçer; yüzler aynıysa satır oyuncusu, farklıysa sütun oyuncusu kazanır. Satır oyuncusu Yazı derse sütun oyuncusu Tura demek ister; o Tura derse satır oyuncusu Tura'ya geçmek ister ve döngü böyle sürer. Böyle oyunlarda oyuncular tahmin edilemez olmak ister. “Tahmin edilemez olmak” fikri olasılıkla kesinleştirilir.

3.4.1 Karışık strateji ve kayıtsızlık ilkesi

İki temel olasılık kavramı gerekir. Bir **olasılık dağılımı**, olası sonuçların her birine 0 ile 1 arasında bir sayı atar ve bu sayıların toplamı 1'dir. Adil bir zarda her yüzün olasılığı 1/6'dır. Değeri şansa bağlı bir X sayısının **beklenen değeri** (*expected value*), her olası değerin olasılığıyla çarpılıp toplanmasıyla bulunur:

$$E[X] = \Sigma_x \, x \cdot P(X = x)$$

Adil zarda $E[X] = (1 + 2 + 3 + 4 + 5 + 6) \cdot 1/6 = 21/6 = 3{,}5$'tir. Beklenen değer “en olası sonuç” değildir; zarda 3,5 hiç gelmez. Beklenen değer, aynı deney çok kez tekrarlansaydı tur başına ortalamanın ne olacağını söyler.

Tanım 3.6

Oyuncu i'nin **karışık stratejisi** (*mixed strategy*) σ_i, saf stratejileri üzerinde bir olasılık dağılımıdır: her saf stratejiye negatif olmayan bir olasılık verilir ve olasılıkların toplamı 1'dir. Tek bir saf stratejiye 1 olasılık veren karışık strateji, o saf stratejinin kendisidir.

Karışık stratejinin **desteği** (*support*), pozitif olasılık verilen saf stratejilerin kümesidir.

Oyuncular karışık strateji oynadığında kazanç, **beklenen kazançtır** (*expected payoff*): her hücrenin kazancı, o hücrenin gerçekleşme olasılığıyla çarpılıp toplanır. Oyuncuların zarları birbirinden bağımsızdır; bu yüzden (satır r, sütun c) hücresinin olasılığı $\sigma_1(r) \cdot \sigma_2(c)$'dir.

Örneğin Taş-Kağıt-Makas'ta (1/3, 1/3, 1/3), "bir zar at; 1–2 gelirse Taş, 3–4 gelirse Kağıt, 5–6 gelirse Makas oyna" demektir. 2 × 2 oyunlarda gösterim kısalır: p satır oyuncusunun ilk satırı, q sütun oyuncusunun ilk sütunu oynama olasılığıdır; öteki seçeneklerin olasılığı $1 - p$ ve $1 - q$'dur.

Bir saf stratejinin karışık bir rakibe karşı beklenen kazancı, o satırın kazançlarının rakibin olasılıklarıyla tartılmasıdır. Rakip Taş-Kağıt-Makas'ta (1/2 Taş, 1/4 Kağıt, 1/4 Makas) oynasın. Kazanınca +1, kaybedince −1, berabere 0 olsun:

- Taş: $\frac{1}{2} \cdot 0 + \frac{1}{4} \cdot (-1) + \frac{1}{4} \cdot (+1) = 0$.
- Kağıt: $\frac{1}{2} \cdot (+1) + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot (-1) = \frac{1}{4}$.
- Makas: $\frac{1}{2} \cdot (-1) + \frac{1}{4} \cdot (+1) + \frac{1}{4} \cdot 0 = -\frac{1}{4}$.

Bu rakibe karşı en iyi tepki Kağıt'tır ve tur başına ortalama ¼ puan kazandırır. Beklenen değer ile argmax birlikte, karışık bir rakibe karşı en iyi tepkiyi verir. Rakip (1/3, 1/3, 1/3) oynasaydı her hamle 0 getirir, hiçbir hamle ötekinden iyi olmazdı. TKM'nin dengesi tam olarak budur: iki oyuncu da (1/3, 1/3, 1/3) oynar ve her eylem 0 getirir. 1/3–1/3–1/3 dışındaki her karışımı rakip sömürebilir; az önceki hesap, Taş'a hafif yatkın bir oyuncunun Kağıt'la nasıl sömürüldüğünü gösterir.

Tanım 3.7

Karışık stratejilerden oluşan $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$ profili bir **karışık Nash dengesidir** (*mixed Nash equilibrium*) ancak ve ancak her oyuncu i ve her karışık strateji σ_i için $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*)$'dır. Burada u_i beklenen kazançtır.

Tanım saf durumdakiyle aynıdır; yalnızca stratejiler ve kazançlar genişlemiştir. Karışık dengeyi hesaplamanın anahtarı aşağıdaki ilkedir.

Teorem 3.3 (Kayıtsızlık ilkesi)

Bir karışık Nash dengesinde her oyuncu, desteğindeki tüm saf stratejilerden ötekilerin denge karışımına karşı aynı beklenen kazancı alır. Destek dışındaki saf stratejiler bundan daha fazla vermez.

İspat. Oyuncu i'nin karışık stratejisinin beklenen kazancı, desteğindeki saf stratejilerin beklenen kazançlarının olasılıklarla tartılmış ortalamasıdır. Desteğindeki iki strateji a ve b olsun ve a, b'den kesin daha çok versin. O zaman b'nin olasılığı a'ya kaydırılırsa ortalama artar; çünkü daha küçük bir değerin ağırlığı azalıp daha büyük bir değerinki artmıştır. Bu, dengede kârlı bir sapma olduğu anlamına gelir ve dengenin tanımıyla çelişir. Demek ki destekteki bütün stratejiler eşit kazanç verir. Destek dışındaki bir strateji bu ortak değerden fazla verseydi, oyuncu bütün olasılığı ona kaydırarak yine kazancını artırırdı; bu da çelişkidir. □

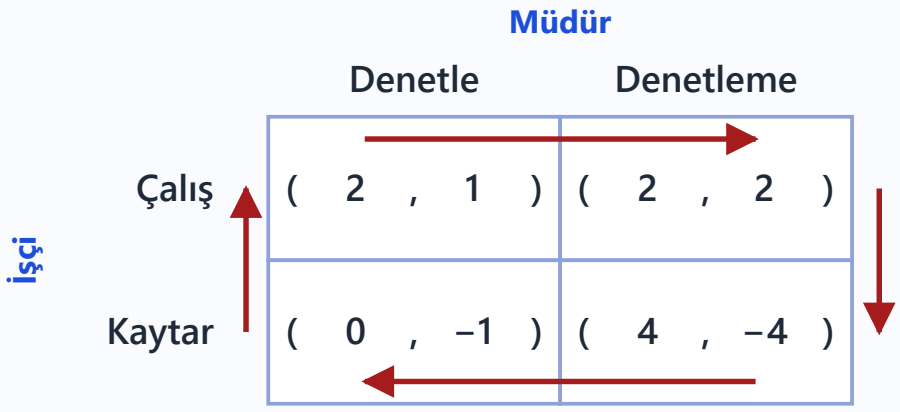
İlkenin şaşırtıcı bir sonucu vardır. Satır oyuncusunun karışık stratejisini oynamaya razı olması için iki seçeneği arasında kayıtsız olması gerekir; bu kayıtsızlık ise sütun oyuncusunun q'suna bağlıdır. Dolayısıyla satır oyuncusunun kayıtsızlık denklemi sütun oyuncusunun q'sunu belirler. Tersine, sütun oyuncusunun kayıtsızlık denklemi satır oyuncusunun p'sini belirler. Kural şudur: **satır oyuncusunun karışımı, sütun oyuncusunun kazançlarından hesaplanır (ve tersi).**

Karışık stratejinin iki yorumu vardır. Birincisi gerçekten zar atmaktır: penaltıda atıcı tarafını kafasından rastgele seçer. İkincisi nüfus yorumudur: kalabalık bir nüfusta olasılıklar davranış oranlarıdır, örneğin penaltıların yüzde kaçının sola atıldığı. Oyuncu zar atmasa bile rakibin gözünde onun hamlesi belirsizdir; olasılık bu belirsizliği ölçer.

3.4.2 Denetim oyunu adım adım

Denetim oyunu (*inspection game*), saf dengesi olmayan oyunların tipik örneğidir. Satır oyuncusu bir işçidir ve Çalış ya da Kaytar seçer. Sütun oyuncusu müdürdür ve Denetle ya da Denetleme seçer. Sayılar şöyledir: ücret 4, işçinin çaba maliyeti 2, çalışan işçinin üretimi 6, denetimin müdüre maliyeti 1. Kaytaran işçi denetimde yakalanırsa ücret alamaz.

Her hücre bu sayılardan türetilir. Çalış–Denetle hücresinde işçi ücretten çaba maliyetini çıkarır: $4 - 2 = 2$. Müdür üretimden ücreti ve denetim maliyetini çıkarır: $6 - 4 - 1 = 1$. Çalış–Denetleme'de işçi yine 2, müdür $6 - 4 = 2$ alır. Kaytar–Denetle'de işçi yakalanır: ücret yok, çaba yok, 0 alır; müdür üretim almaz, ücret ödemez, yalnız denetim maliyetini öder: −1. Kaytar–Denetleme'de işçi çalışmadan ücreti alır: 4; müdür üretim almadan ücret öder: −4.



Şekil 3.14: Denetim oyunu; hücrede (işçi, müdür). Oklar, her hücreden bir oyuncunun kârlı sapmasını gösterir. Dört ok bir döngü oluşturur: saf Nash dengesi yoktur.

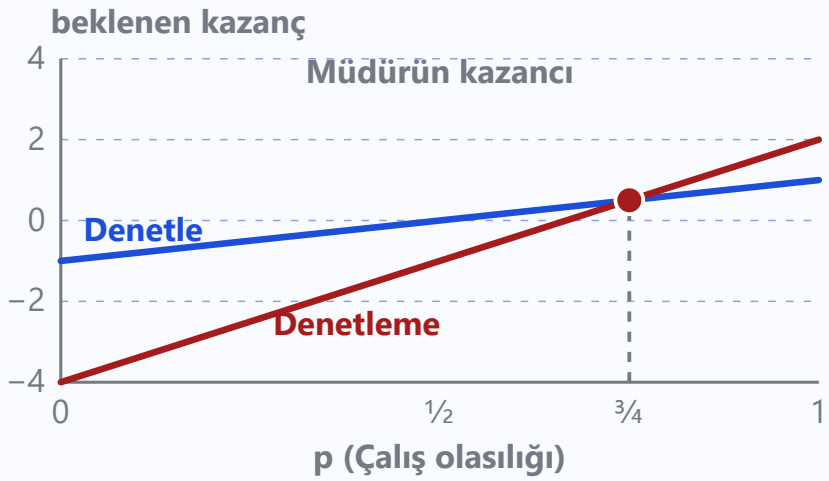
Şekil 3.14'teki okları takip edelim. İşçi çalışıyorsa müdür denetlemeyi bırakır ($2 > 1$). Müdür denetlemiyorsa işçi kaytarır ($4 > 2$). İşçi kaytarıyorsa müdür denetler ($-1 > -4$). Müdür denetliyorsa işçi çalışır ($2 > 0$). Her hücrede biri sapmak ister; en iyi tepkiler bir döngü oluşturur. Bu yüzden karışık denge aranır.

Örnek 3.10

İşçi p olasılıkla Çalış, müdür q olasılıkla Denetle oynasın.

1. **İşçinin iki seçeneği.** Çalış'ın beklenen kazancı $E[\text{Çalış}] = 2q + 2(1 - q) = 2$; müdür ne yaparsa yapsın 2'dir. Kaytar'ın beklenen kazancı $E[\text{Kaytar}] = 0 \cdot q + 4(1 - q) = 4 - 4q$'dur.
2. **İşçi kayıtsız olmalı.** $2 = 4 - 4q \Rightarrow 4q = 2 \Rightarrow q^* = 1/2$. Müdür yarı yarıya denetlemelidir. Daha az denetlerse işçi hep kaytarır; daha çok denetlerse işçi hep çalışır ve denetim boşa para olur.
3. **Müdürün iki seçeneği.** $E[\text{Denetle}] = 1 \cdot p + (-1)(1 - p) = 2p - 1$; $E[\text{Denetleme}] = 2p + (-4)(1 - p) = 6p - 4$.
4. **Müdür kayıtsız olmalı.** $2p - 1 = 6p - 4 \Rightarrow 3 = 4p \Rightarrow p^* = 3/4$.
5. **Denetim.** İki olasılık da 0 ile 1 arasındadır ✓. Beklenen kazançlar: işçi 2 (iki seçeneği de 2 verir), müdür $2 \cdot \tfrac{3}{4} - 1 = \tfrac{1}{2}$.

Denge: $(p^*, q^*) = (3/4, 1/2)$. İşçi zamanın dörtte üçünde çalışır, müdür yarı yarıya denetler.



Şekil 3.15: Müdürün beklenen kazancı. Mavi doğru Denetle ($2p - 1$), kırmızı doğru Denetleme ($6p - 4$). İki doğru $p = 3/4$'te kesişir; işçi bu olasılıkla çalışırsa müdür kayıtsızdır.

Şekil 3.15 müdürün durumunu gösterir. $p < 3/4$ iken mavi doğru üstte kalır: işçi sık kaytarıyorsa denetlemek iyidir. $p > 3/4$ iken kırmızı doğru üsttedir: işçi çoğunlukla çalışıyorsa denetim gereksizdir. Yalnız $p = 3/4$'te müdür iki seçenek arasında kayıtsızdır ve karıştırmaya razıdır.

3.4.3 Ceza değişince kimin olasılığı değişir?

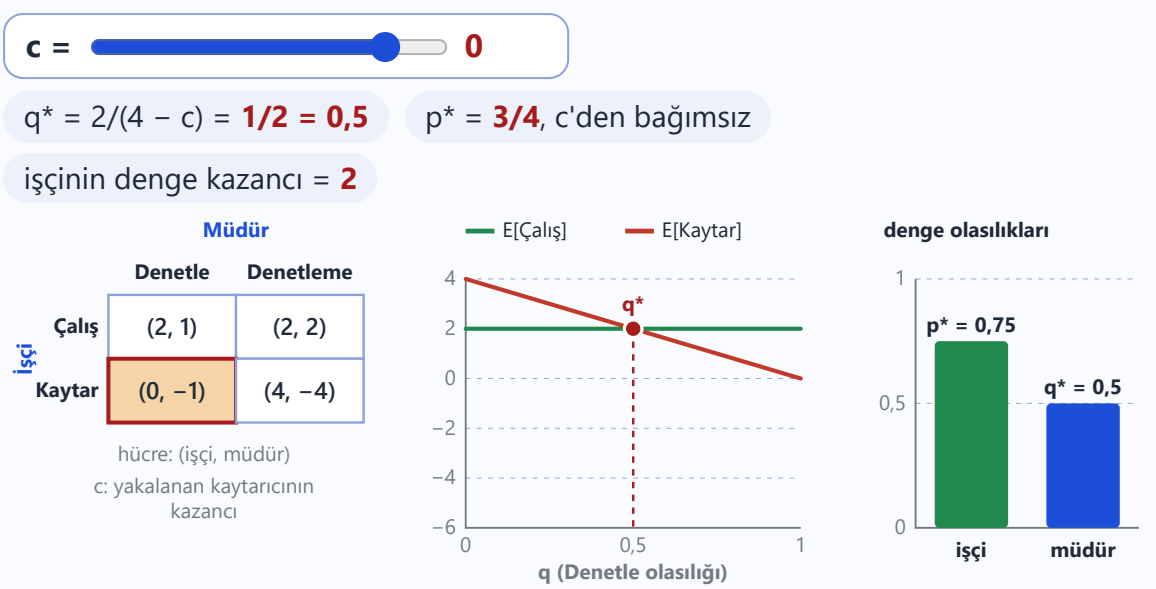
Kaytaran işçiye yakalandığında daha sert bir ceza verilsin: yakalanan kaytarıcının kazancı 0 yerine −2 olsun. Hangi olasılık değişir? Sezgisel cevap "işçi daha çok çalışır" olur. Kayıtsızlık ilkesi başka bir şey söyler.

Ceza yalnız işçinin kazancını değiştirir. İşçinin kazançları ise müdürün q'sunu belirler. Yeni hâliyle işçinin kayıtsızlığı $2 = -2q + 4(1 - q) = 4 - 6q$'dur ve $q^* = 1/3$ çıkar. Müdürün kazançları değişmediği için müdürün kayıtsızlık denklemi aynı kalır ve $p^* = 3/4$ değişmez. Daha sert ceza, işçiyi daha çalışkan yapmaz; müdürün daha az denetlemesine izin verir.

Genel olarak yakalanan kaytarıcının kazancı c ise işçinin kayıtsızlığı $2 = c \cdot q + 4(1 - q)$'dur. Bu denklem $(4 - c)\, q = 2$ biçimine getirilir ve

$$q^* = 2/(4 - c), \quad p^* = 3/4$$

bulunur. $c = 0$'da $q^* = 1/2$, $c = -2$'de $q^* = 1/3$, $c = -6$'da $q^* = 1/5$'tir.

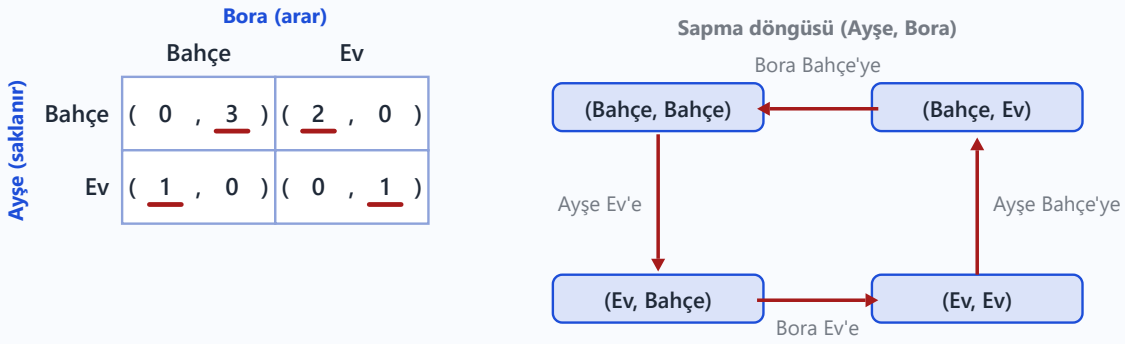


Şekil 3.16: Etkileşimli denetim oyunu. Kaydırıcı, yakalanan kaytarıcının kazancı *c*'yi değiştirir. Ortadaki grafikte işçinin iki seçeneğinin beklenen kazancı, sağda denge olasılıkları görünür: *q** değişir, *p* = 3/4* sabit kalır. Varsayılan değer *c = 0*'dır.

Bu sonuç, "her oyuncunun denge olasılığını rakibinin kazançları belirler" ilkesinin somut bir hâlidir. Sonucun politika açısından da anlamı vardır. Ceza artınca denetim olasılığı düşer ve denetim maliyetinden tasarruf edilir, ama kuralı çiğneme oranı değişmez. Bu oranı değiştirmek için denetleyenin (müdürün) kazançları değişmelidir.

3.4.4 Saklambaç: yöntemin tekrarı

Kayıtsızlık yöntemini bir kez daha, her adımı ayrı yazarak uygulayalım. Ayşe saklanır (satır), Bora arar (sütun). Seçenekler Bahçe ve Ev'dir. Hücrede keyif puanı (Ayşe, Bora) yazılıdır. Bora Ayşe'yi bahçede bulursa 3, evde bulursa 1 puan alır. Ayşe bahçede bulunmazsa 2, evde bulunmazsa 1 puan alır. Bulunan Ayşe 0, bulamayan Bora 0 alır.



Şekil 3.17: Saklambaç oyunu. Solda kazanç matrisi ve altı çizili en iyi tepkiler; hiçbir hücrede iki çizgi yoktur. Sağda en iyi tepkilerin oluşturduğu sapma döngüsü.

Örnek 3.11

1. **Saf denge var mı?** Altı çizilince hiçbir hücrede iki çizgi olmadığı görülür. (Bahçe, Bahçe)'de Ayşe Ev'e geçer; (Ev, Bahçe)'de Bora Ev'e geçer; (Ev, Ev)'de Ayşe Bahçe'ye geçer; (Bahçe, Ev)'de Bora Bahçe'ye geçer. Saf denge yoktur.
2. **Değişkenler.** Ayşe Bahçe'yi p, Bora Bahçe'yi q olasılıkla seçer. Ayşe karıştırıyorsa iki seçeneği ona eşit kazanç vermelidir; bu eşitlik q'yu verir.
3. **Ayşe Bahçe'de.** Bora q olasılıkla bahçeye bakar ve Ayşe 0 alır; $1 - q$ olasılıkla eve bakar ve Ayşe 2 alır. $E[\text{Bahçe}] = 0 \cdot q + 2(1 - q) = 2 - 2q$.
4. **Ayşe Ev'de.** Bora bahçeye bakarsa (q) Ayşe 1, eve bakarsa 0 alır. $E[\text{Ev}] = 1 \cdot q + 0 \cdot (1 - q) = q$.
5. **Eşitle ve çöz.** $2 - 2q = q \Rightarrow 3q = 2 \Rightarrow q^* = 2/3$. Bora bahçeye 2/3 olasılıkla bakmalıdır ki Ayşe kayıtsız kalsın.
6. **Bora için aynısı.** Bora Bahçe'ye bakarsa Ayşe orada iken (p) 3, değilken 0 alır: $E[\text{Bahçe}] = 3p$. Ev'e bakarsa: $E[\text{Ev}] = 0 \cdot p + 1 \cdot (1 - p) = 1 - p$.
7. **Eşitle.** $3p = 1 - p \Rightarrow 4p = 1 \Rightarrow p^* = 1/4$. p^*, Ayşe'nin değil Bora'nın sayılarından çıktı.
8. **Kontrol.** $0 < 1/4 < 1$ ve $0 < 2/3 < 1$ ✓. Beklenen kazançlar: Ayşe $2 - 2 \cdot (2/3) = 2/3$, Bora $3 \cdot (1/4) = 3/4$.

Sonuç ilk bakışta çelişkili görünür: bahçeyi daha çok seven Ayşe (orada bulunmazsa 2 puan) orada yalnız %25 olasılıkla saklanır. Nedeni, Bora'nın da bahçede bulmayı daha çok sevmesidir (3 puan). Bora'nın bahçeye bakma isteği, Ayşe'yi oradan uzaklaştırır. Ayşe'nin bahçe sevgisi ise Bora'nın davranışını belirler: Bora bahçeye 2/3 olasılıkla bakar.

3.4.5 Kod izleme: karisik_2x2()

Kayıtsızlık denklemleri her 2 × 2 oyun için genel olarak çözülebilir. Satır oyuncusunun kazançları A, sütun oyuncusununkiler B olsun (indis 0 ilk seçenek, 1 ikinci seçenek). Satır oyuncusunun kayıtsızlığı

$$A_{00}\,q + A_{01}(1-q) = A_{10}\,q + A_{11}(1-q)$$

eşitliğidir. q'lu terimler bir tarafa toplanınca $q\,(A_{00} - A_{01} - A_{10} + A_{11}) = A_{11} - A_{01}$ çıkar. Parantezdeki sayıya d_a denirse $q = (A_{11} - A_{01}) / d_a$'dır. Aynı işlem sütun oyuncusunun kayıtsızlığı $B_{00}\,p + B_{10}(1-p) = B_{01}\,p + B_{11}(1-p)$ için yapılınca $p = (B_{11} - B_{10}) / d_b$, $d_b = B_{00} - B_{01} - B_{10} + B_{11}$ bulunur. Aşağıdaki fonksiyon bu formülleri kesirlerle (`Fraction`) hesaplar; böylece 2/5 gibi sonuçlar ondalık yuvarlama hatası olmadan çıkar.

```
from fractions import Fraction as F
def karisik_2x2(A, B):
    da = A[0][0] - A[0][1] - A[1][0] + A[1][1]
    if da == 0: return None     # denklem yok
    q = F(A[1][1] - A[0][1], da)  # satır kayıtsız
    db = B[0][0] - B[0][1] - B[1][0] + B[1][1]
    if db == 0: return None
    p = F(B[1][1] - B[1][0], db)  # sütun kayıtsız
    if not (0 < p < 1 and 0 < q < 1):
        return None               # iç denge yok
    u1 = A[0][0] * q + A[0][1] * (1 - q)
    return p, q, u1
```

Fonksiyon üç durumda `None` döndürür: $d_a = 0$ ise satır oyuncusunun kayıtsızlık denkleminde q yok olur (iki seçeneğin farkı q'ya bağlı değildir); $d_b = 0$ ise aynısı sütun oyuncusu için olur; olasılıklardan biri 0 ile 1 arasında değilse iç karışık denge yoktur. Son satırdaki `u1`, satır oyuncusunun ilk satırı oynadığındaki beklenen kazancıdır. Kayıtsızlık ilkesi gereği ikinci satır da aynı değeri verir, bu yüzden bu sayı satır oyuncusunun denge kazancıdır.

Fonksiyonu Cinsiyetler Savaşı ile çalıştıralım. 0 = Opera, 1 = Futbol; `A = [[3, 0], [0, 2]]` satır, `B = [[2, 0], [0, 3]]` sütun oyuncusunun kazançlarıdır. Satır oyuncusu Opera'yı, sütun oyuncusu Futbol'u daha çok sever; ikisi de ayrı düşmekten (0) kaçınır.

Satır	Hesap	Değer	Anlamı
`da`	3 − 0 − 0 + 2	5	≠ 0, denklem var
`q`	(2 − 0) / 5	2/5	3q = 2(1 − q): sütun oyuncusu Opera'yı 2/5 oynar
`db`	2 − 0 − 0 + 3	5	≠ 0
`p`	(3 − 0) / 5	3/5	2p = 3(1 − p): satır oyuncusu Opera'yı 3/5 oynar
denetim	0 < 3/5 < 1, 0 < 2/5 < 1	doğru	iç karışık denge var
`u1`	3 · 2/5 + 0 · 3/5	6/5	satır oyuncusunun beklenen kazancı

Satır	Hesap	Değer	Anlamı
dönüş		(3/5, 2/5, 6/5)	

Tablo 3.14: `karisik_2x2(A, B)` izlemesi, Cinsiyetler Savaşı.

Sütun oyuncusunun beklenen kazancı da $u_2 = 2 \cdot (3/5) = 6/5$'tir. Bu sayı dikkat çekicidir: iki saf dengede (Opera, Opera) ve (Futbol, Futbol) oyuncular 2 ya da 3 alır. Karışık dengede ise ikisi de yalnız 6/5 alır, çünkü sık sık ayrı yerlere düşerler. Buluşamama, satır oyuncusu Opera'ya ve sütun oyuncusu Futbol'a gittiğinde ya da tersinde olur. Sütun oyuncusunun Futbol olasılığı $1 - 2/5 = 3/5$, satır oyuncusununki $1 - 3/5 = 2/5$ olduğundan buluşamama olasılığı $(3/5)(3/5) + (2/5)(2/5) = 9/25 + 4/25 = 13/25 = \%52$'dir.

Fonksiyonun `None` dalları da örneklerle anlaşılır. **Bedava Para** oyununda yalnız (Yazı, Yazı) hücresi (1, 1), öteki hücreler 0'dır. `A = [[1, 0], [0, 0]]` için $d_a = 1$, $q = (0 - 0)/1 = 0$ çıkar; q 0 ile 1 arasında olmadığı için fonksiyon `None` döndürür. **Mahkûmun İkilemi**'nde `A = [[-1, -3], [0, -2]]` için $d_a = -1 + 3 - 0 - 2 = 0$'dır ve ilk denetim `None` döndürür: İhanet, Sessiz'den her durumda tam 1 fazla verdiği için satır oyuncusu hiçbir q'da kayıtsız olmaz.

3.4.6 Altı klasik oyunun karışık dengeleri

Aşağıdaki tablo altı klasik 2 × 2 oyunun dengelerini karşılaştırır. p^* ve q^*, ilk stratejinin (Yazı, Kır, Opera, Geyik, Parti, Yazı) oynanma olasılığıdır.

Oyun	Saf NE	Karışık NE (p^*, q^*)	(u_1, u_2)	Karışık dengede ne olur?
Yazı-Tura Eşleştirme	0	1/2, 1/2	(0, 0)	Dört sonuç %25'er; tek denge budur, kimse sömürülemez.
Tavuk (kavşak)	2	9/10, 9/10	(−1/10, −1/10)	Çarpışma $(1/10)^2$ = %1; çarpışma −k olsaydı kırma olasılığı (k − 1)/k.
Cinsiyetler Savaşı	2	3/5, 2/5	(6/5, 6/5)	Buluşamama 0,36 + 0,16 = %52.
Geyik Avı (geyik 4, tavşan 3)	2	3/4, 3/4	(3, 3)	Ayrı düşme 3/8 = %37,5; birlikte geyik 9/16 ≈ %56.
Saf Koordinasyon	2	1/3, 1/3	(10/3, 10/3)	Farklı karar 4/9 ≈ %44; parti ne kadar cazipse o kadar az seçilir.
Bedava Para	2	yok (yalnız uç: 0, 0)	—	q > 0 ise Yazı kesin daha iyi; kayıtsızlık yalnız q = 0'da.

Tablo 3.15: Altı klasik oyunun karışık dengeleri. Tavuk: (Kır, Kır) = (0, 0), (Kır, Düz) = (−1, 1), (Düz, Düz) = (−10, −10). Saf Koordinasyon: (Parti, Parti) = (10, 10), (Ev, Ev) = (5, 5), öteki hücreler 0. Bedava Para: yalnız (Yazı, Yazı) = (1, 1), öteki hücreler 0.

Tablodaki sayıların ikisini ayrıntılı türetelim.

Örnek 3.12

Tavuk (kavşak). İki sürücü ışıksız bir kavşağa aynı anda gelir. Kır (yol ver) ya da Düz git seçer. Karşı sürücü x olasılıkla kırıyor olsun.

- $E[\text{Kır}] = 0 \cdot x + (-1)(1 - x) = x - 1$.
- $E[\text{Düz}] = 1 \cdot x + (-10)(1 - x) = 11x - 10$.
- Kayıtsızlık: $x - 1 = 11x - 10 \Rightarrow 9 = 10x \Rightarrow x = 9/10$.

Oyun simetrik olduğu için iki sürücü de 9/10 olasılıkla kırar. Beklenen kazanç $9/10 - 1 = -1/10$'dur. İki sürücünün birlikte düz gitme (çarpışma) olasılığı $(1/10)^2 = 1/100$'dür. Çarpışma bedeli $-k$ olsaydı kayıtsızlık $x - 1 = x - k(1 - x)$, yani $x = (k - 1)/k$ verirdi. Bedel büyüdükçe sürücüler daha sık kırar, ama çarpışma olasılığı $(1/k)^2$ hiçbir zaman sıfır olmaz.

Saf Koordinasyon. Karşı taraf q olasılıkla Parti seçiyorsa Parti'nin beklenen kazancı $10q$, Ev'inki $5(1 - q)$'dur. $10q = 5 - 5q \Rightarrow q = 1/3$. Parti daha cazip olduğu hâlde 1/3 olasılıkla seçilir. Nedeni kayıtsızlık ilkesidir: Parti'nin kazancı büyüdükçe, karşı tarafı kayıtsız bırakmak için Parti'nin daha *seyrek* seçilmesi gerekir. Farklı karar olasılığı $2 \cdot (1/3)(2/3) = 4/9$'dur.

Tablonun son satırı bir uyarıdır. Bedava Para'da sütun oyuncusu Yazı'yı $q > 0$ olasılıkla seçtiği sürece satır oyuncusu için Yazı (kazanç q) Tura'dan (kazanç 0) kesin daha iyidir. Kayıtsızlık yalnız $q = 0$'da olur; bu da bir "karışım" değil, saf (Tura, Tura) dengesidir. Kazançlarında eşitlik bulunan böyle oyunlarda iç karışık denge olmayabilir.

3.4.7 Karışık dengede sık hatalar

Sık yapılan hata

- **p'yi kendi sayılarından bulmak:** Satır oyuncusunun p'si sütun oyuncusunu kayıtsız bırakır; bu yüzden sütun oyuncusunun kazançlarından hesaplanır.
- **(1 − q)'yu unutmak:** $E = 0 \cdot q + 2 \cdot (1 - q) = 2 - 2q$'dur, "2q" değildir. Her terim kendi olasılığıyla çarpılır.
- **Sonucu sınamamak:** p ya da q 0 ile 1 arasında çıkmazsa iç karışık denge yoktur (genellikle bir strateji baskılanmıştır).
- **Saf dengeleri atlamak:** Karışık hesap saf dengelerin yerini tutmaz; önce altı çizilir, sonra karışık denge aranır.
- **"Kayıtsızsa her p olur":** Oyuncu kendi seçenekleri arasında kayıtsızdır, ama dengede rakibini kayıtsız bırakan tek p'yi oynar. Başka bir p oynarsa rakip saf bir en iyi tepkiye kayar ve denge bozulur.

- **Eşitliği yanlış kurmak:** Eşitlenen, aynı oyuncunun iki seçeneğinin beklenen kazancıdır; iki oyuncunun kazançları eşitlenmez.

3.5 Varlık ve denge seçimi

Şimdiye kadar incelenen oyunların bazılarında tek denge, bazılarında birden çok denge vardı; saf dengesi olmayan oyunlarda ise karışık bir denge bulundu. Buradan iki soru doğar. Her oyunun bir dengesi var mıdır? Birden çok denge varsa hangisi oynanır?

3.5.1 Nash'in varlık teoremi

Teorem 3.4 (Nash'in varlık teoremi, 1950)

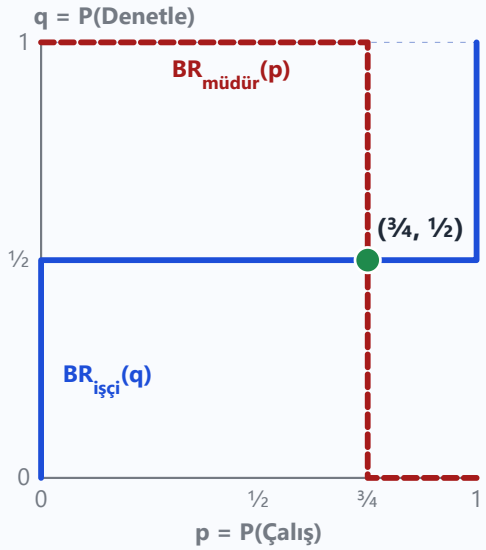
Sonlu sayıda oyuncusu olan ve her oyuncunun strateji kümesi sonlu olan her oyunun en az bir (karışık strateji) Nash dengesi vardır.

Teoremin tam ispatı bu dersin kapsamı dışındadır, ama fikri anlaşılabilir. Fikir **sabit nokta** kavramına dayanır. Bir fonksiyonun sabit noktası, fonksiyonun değiştirmediği noktadır: $f(x) = x$. En basit hâli şöyledir: [0, 1] aralığını yine [0, 1]'e götüren ve grafiği kopuksuz çizilebilen (sürekli) her fonksiyonun bir sabit noktası vardır. Grafik $x = 0$'da köşegenin üstünde ya da üzerinde başlar, $x = 1$'de köşegenin altında ya da üzerinde biter; kalemi kaldırmadan çizildiği için köşegeni bir yerde keser. O kesişim noktası sabit noktadır. Bu, Brouwer sabit nokta teoreminin bir boyutlu hâlidir.

Nash'in ispatında "nokta" bir karışık strateji profilidir. Fonksiyonun yerinde en iyi tepki eşlemesi vardır: her profile, herkesin ötekilere en iyi tepki verdiği profilleri atar. Eşlemenin sabit noktası, herkesin zaten ötekilere en iyi tepki verdiği bir profildir; Teorem 3.1'e göre bu bir Nash dengesidir. Kakutani'nin sabit nokta teoremi (Brouwer'in tek değer yerine küme döndüren eşlemelere genellemesi) bu eşlemenin bir sabit noktası olduğunu garanti eder.

Karışık stratejilerin burada neden gerekli olduğu da bu fikirden anlaşılır. Sabit nokta teoremleri "kopuksuzluk" ister: olasılıklar sürekli değişebilir (0,30'dan 0,31'e), saf stratejiler ise atlayarak değişir (Yazı'dan Tura'ya). Saf stratejilerle sınırlı kalınca kopukluk oluşur ve denge kaybolabilir. Nitekim Yazı-Tura'da saf denge yoktur. Bu yüzden teoremin garantisi karışık stratejiler içindir.

Sonluluk koşulu da gereklidir. Strateji kümesi sonsuzsa denge olmayabilir. Örneğin iki oyuncu aynı anda bir doğal sayı söylesin ve büyük sayıyı söyleyen kazansın. Her sayıdan daha büyüğü vardır; hangi profil seçilirse seçilsin, kaybeden daha büyük bir sayı söyleyerek kazanır. Hiçbir profil denge değildir. Cournot oyununda strateji kümesi sonsuz olduğu hâlde denge vardı; çünkü orada kazançlar düzgündü ve en iyi tepkiler sınırlı bir aralıkta kalıyordu. Sonsuz kümelerde varlık, ek koşullara bağlıdır.

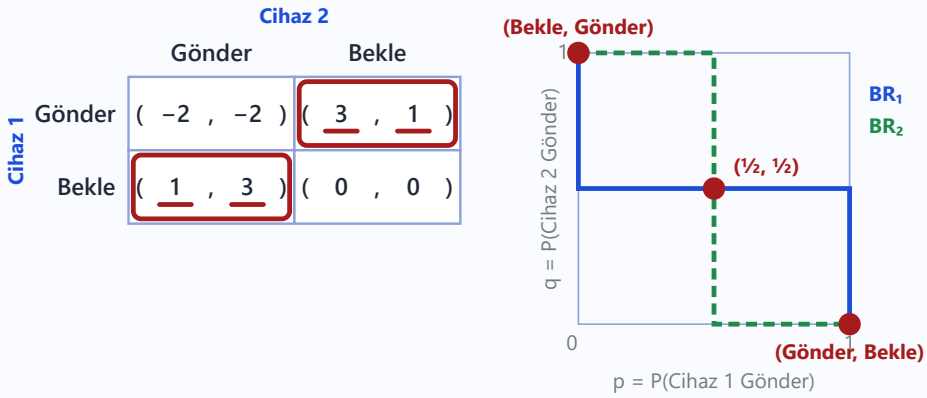


Şekil 3.18: Denetim oyununda en iyi tepki eğrileri. Mavi düz eğri işçinin en iyi tepkisi $BR_{işçi}(q)$, kırmızı kesikli eğri müdürün en iyi tepkisi $BR_{müdür}(p)$'dir. Tek kesişim noktası $(3/4, 1/2)$ dengedir.

Şekil 3.18 bu fikri denetim oyununda somutlaştırır. Yatay eksen işçinin Çalış olasılığı p, dikey eksen müdürün Denetle olasılığı q'dur. İşçinin en iyi tepkisi şöyledir: $q < 1/2$ iken Kaytar ($p = 0$), $q > 1/2$ iken Çalış ($p = 1$), $q = 1/2$'de ise kayıtsızdır ve her p en iyi tepkidir. Bu yüzden mavi eğri bir "basamak" biçimindedir ve $q = 1/2$'de yatay bir parça içerir. Müdürün en iyi tepkisi de benzer bir basamaktır: $p < 3/4$ iken Denetle ($q = 1$), $p > 3/4$ iken Denetleme ($q = 0$). İki basamak tek bir noktada kesişir. Kesişim noktasında iki oyuncu da birbirine en iyi tepki verir; bu nokta dengedir. Eğrilerin kesişmek zorunda olması, sabit nokta teoreminin bu basit oyundaki görüntüsüdür.

3.5.2 Kanal paylaşımında üç denge

İki kablosuz cihaz aynı kanala paket göndermek ister. İkisi aynı anda gönderirse paketler çarpışır ve her biri −2 alır. Yalnız biri gönderirse gönderen 3, bekleyen 1 alır (bekleyen de kanalın düzgün çalışmasından yararlanır). İkisi de beklerse kanal boşa gider ve ikisi de 0 alır. Bu yapı, Tavuk oyunu ile aynıdır: her oyuncu karşısındakinin tersini yapmak ister.



Şekil 3.19: Kanal paylaşımı. Solda kazanç matrisi; iki çizgili hücreler (Gönder, Bekle) ve (Bekle, Gönder) saf dengelerdir. Sağda en iyi tepki eğrileri: BR_1 (mavi) ve BR_2 (yeşil kesikli) üç noktada kesişir.

Örnek 3.13

1. **Cihaz 1'in en iyi tepkileri** (ilk sayılar). Karşısı Gönder derse: Gönder −2, Bekle 1; $-2 < 1$, en iyi tepki Bekle. Karşısı Bekle derse: Gönder 3, Bekle 0; en iyi tepki Gönder.
2. **Cihaz 2 için aynısı** (ikinci sayılar). İki çizgili hücreler (Gönder, Bekle) = (3, 1) ve (Bekle, Gönder) = (1, 3)'tür: iki saf denge.
3. **Karışık denge.** q = P(Cihaz 2 Gönder) olsun. Cihaz 1'in Gönder'den beklentisi $-2q + 3(1 - q) = 3 - 5q$'dur.
4. Bekle'den beklentisi: karşısı gönderirse 1, beklerse 0, yani $1 \cdot q + 0 \cdot (1 - q) = q$.
5. **Kayıtsızlık:** $3 - 5q = q \Rightarrow 6q = 3 \Rightarrow q^* = 1/2$. Oyun simetrik olduğu için Cihaz 2'nin kayıtsızlığı da $p^* = 1/2$ verir.
6. **Sonuçlar.** Karışık dengede dört hücrenin her biri 1/4 olasılıklıdır. Beklenen kazanç: $¼ \cdot (-2) + ¼ \cdot 3 + ¼ \cdot 1 + ¼ \cdot 0 = ½$.

En iyi tepki eğrileri (Şekil 3.19) üç noktada kesişir: iki köşe saf dengeler, orta nokta karışık dengedir. Karışık denge simetrik olduğu için "adil"dir, ama verimsizdir. Zamanın dörtte birinde paketler çarpışır, dörtte birinde kanal boş kalır ve her cihaz ortalama yalnız ½ alır. Saf dengelerde ise her cihaz 3 ya da 1 alır. Ağ protokollerindeki rastgele bekleme (*backoff*) bu rastgeleleştirmenin uygulamadaki karşılığıdır: çarpışan cihazlar rastgele bir süre bekleyip yeniden dener, böylece "kim önce gönderecek" sorusu koordinasyon olmadan çözülür.

3.5.3 Kaç denge olur? Tek sayı kuralı

Bölümdeki oyunların denge sayılarında ilginç bir örüntü vardır.

Oyun	Saf denge	Karışık denge	Toplam
Mahkûmun İkilemi	1	0	1
Saklambaç, Denetim, Yazı-Tura	0	1	1
Kanal paylaşımı, Tavuk, Cinsiyetler Savaşı, Geyik Avı, Saf Koordinasyon	2	1	3
Dil seçimi (3 × 3)	2	1 (Ayşe Python'u 3/4, Bora Python'u 1/5 olasılıkla seçer)	3
Zayıf Baskılamanın Tuzağı (eşitlikli)	3	sonsuz	—
Bedava Para (eşitlikli)	2	0	2 (çift)

Tablo 3.16: Bölümdeki oyunların denge sayıları.

Teorem 3.5 (Tek sayılılık, Wilson 1971)

Kazançlarında özel eşitlikler olmayan (**dejenere olmayan**) oyunlarda Nash dengesi sayısı sonludur ve tek sayıdır (1, 3, 5, ...).

Teoremin ispatı ileri düzeydedir (Wilson, 1971), ama pratik bir kural verir: iki saf denge bulunduysa büyük olasılıkla bir karışık denge daha vardır ve kayıtsızlık yöntemiyle aranmalıdır. Dil seçimi oyunu buna güzel bir örnektir. İki saf dengenin yanında, C seçeneklerinin hiç oynanmadığı bir karışık denge vardır. Ayşe Python'u 3/4, Bora Python'u 1/5 olasılıkla seçer. Bu karışıma karşı Ayşe'nin Python ve Java satırları 8/5, C satırı yalnız 1/5 verir; Bora'nın Python ve Java sütunları 7/4, C sütunu 1/2 verir. C'ye geçmek kimseye kazandırmaz, yani 2 × 2'de bulunan karışım 3 × 3 oyunda da dengedir.

Eşitlikli oyunlarda kural bozulabilir. Zayıf Baskılamanın Tuzağı'nda sütun oyuncusu her hücrede 1 aldığı için her karışıma razıdır; örneğin satır oyuncusu Alt oynarken sütun oyuncusunun her karışımı bir dengedir. Sonsuz sayıda denge vardır. Bedava Para'da ise denge sayısı 2'dir, yani çifttir. Bu oyunlar "dejenere"dir: kazançlardaki eşitlikler kuralın varsayımını bozar. Kazançlar biraz değiştirilirse (örneğin bir eşitlik 1 yerine 1,01 yapılırsa) tek sayı kuralı geri gelir.

3.5.4 Çoklu denge: Geyik Avı

Filozof Jean-Jacques Rousseau'nun anlattığı bir öyküye dayanan **Geyik Avı** (*Stag Hunt*), çoklu dengenin en bilinen örneğidir. İki avcı aynı anda Geyik ya da Tavşan avlamaya karar verir. Geyik ancak iki avcı birlikte avlanırsa yakalanır ve her biri 4 alır. Tavşan tek başına yakalanabilir ve ortak ne yaparsa yapsın 3 garanti eder. Geyiğe yalnız giden avcı eli boş döner (0).

	Geyik	Tavşan
Geyik	**(4, 4)**	(0, 3)
Tavşan	(3, 0)	**(3, 3)**

Tablo 3.17: Geyik Avı. Satırda Avcı 1, sütunda

Avcı 2; hücrede (Avcı 1,
Avcı 2).

En iyi tepkiler iki hücrede çakışır: (Geyik, Geyik) ve (Tavşan, Tavşan). Ortak geyiğe giderse geyiğe gitmek (4 > 3), ortak tavşana giderse tavşana gitmek (3 > 0) en iyisidir. Üçüncü denge karışıktır: karşı taraf Geyik'i q olasılıkla seçiyorsa Geyik'in beklenen kazancı $4q$, Tavşan'ınki her durumda 3'tür. $4q = 3 \Rightarrow q = 3/4$. Geyik, ancak ortağın en az %75 olasılıkla geleceğine inanılırsa en iyi tepkidir. Hangi denge oynanacak?

İki ölçüt farklı cevaplar verir.

Pareto baskınlık (*Pareto dominance*): Bir denge, öteki dengeye göre herkese en az onun kadar, en az birine daha çok veriyorsa onu Pareto baskılar. (Geyik, Geyik) iki avcıya da daha çok verir (4 > 3). Verimlilik ölçütü (Geyik, Geyik)'i seçer.

Risk baskınlık (*risk dominance*): Harsanyi ve Selten'in (1988) önerdiği bu ölçüt, ortağın ne yapacağından emin olunmadığında hangi dengenin daha "güvenli" olduğuna bakar. Her dengede, her oyuncunun tek taraflı sapma kaybı hesaplanır ve bu kayıplar çarpılır. Çarpımı büyük olan denge risk baskındır.

- (G, G)'de bir avcı Tavşan'a sapsa 4 yerine 3 alır: kaybı $4 - 3 = 1$. İki avcı için çarpım $1 \cdot 1 = 1$.
- (T, T)'de bir avcı Geyik'e sapsa 3 yerine 0 alır: kaybı $3 - 0 = 3$. Çarpım $3 \cdot 3 = 9$.

$9 > 1$ olduğu için (Tavşan, Tavşan) risk baskındır. Sezgisel anlamı şudur: (T, T)'de yanılmanın bedeli düşüktür (tavşan zaten garanti), (G, G)'de ise ortağa yanlış güvenmenin bedeli ağırdır (eli boş dönmek). 2 × 2 simetrik oyunlarda risk baskın denge, ortağın ne yapacağı hakkında yarı yarıya bir inançla en iyi tepki olan dengedir: ortak ½ olasılıkla geyiğe gelirse Geyik $4 \cdot ½ = 2$, Tavşan 3 verir.



Şekil 3.20: Geyik Avı'nda risk baskınlık. (Geyik, Geyik) Pareto baskın, (Tavşan, Tavşan) risk baskındır (kırmızı çerçeve): sapma kayıpları çarpımı 9, (Geyik, Geyik)'inkinden (1) büyüktür.

3.5.5 Denge seçimi ölçütleri

Kuram birden çok dengeden hangisinin oynanacağını tek başına söylemez. Bu soruya **denge seçimi** (*equilibrium selection*) sorunu denir. Literatürde öne çıkan ölçütler şunlardır:

Ölçüt	Ne söyler?	Örnek
Pareto baskınlık	Herkesi daha iyi yapan dengeyi seç. Güven gerektirir.	Geyik Avı'nda (G, G).
Risk baskınlık	Karşı tarafın ne yapacağından emin değilsen kayıp riski düşük olanı seç.	Geyik Avı'nda (T, T).
Odak noktası (Schelling, 1960)	Kültür, alışkanlık ya da belirginlik bir dengeyi öne çıkarır.	"İstanbul'da buluşacaksak saat 12'de Taksim."
Korelasyonlu denge (Aumann, 1974)	Ortak bir sinyal oyuncuları dengelerden birine yönlendirir.	Trafik ışığı, kavşaktaki Tavuk oyununu çözer.
Öğrenme ve evrim	Tekrarlanan oyunda oyuncular deneyerek öğrenir; varılan denge başlangıçtaki davranışa bağlıdır.	Herkes başta tavşana gidiyorsa geyik dengesine hiç ulaşılmayabilir.

Tablo 3.18: Denge seçimi ölçütleri.

Odak noktası (*focal point*) özellikle ilginçtir. İki arkadaş İstanbul'da buluşacak ama yer ve saati konuşamamışlar. Her ortak yer ve saat bir dengedir; kazançlar eşittir, bu yüzden Pareto ya da risk ölçütü yardımcı olmaz. Yine de pek çok kişi "saat 12'de Taksim" cevabını verir. Seçimi belirleyen sayılar değil, ortak kültür ve belirginliktir. Thomas Schelling bu gözlemi 1960'ta ortaya koydu.

Denge seçiminde bir de **adalet sorunu** vardır. Cinsiyetler Savaşı'nda saf dengelerin biri bir oyuncuyu (Opera: 3'e 2), diğeri öbürünü (Futbol: 2'ye 3) kayırır. "Adil" görünen simetrik karışık denge ise ikisine de yalnız 6/5 verir; iki oyuncu için de iki saf dengeden kötüdür. Adalet ile verimlilik burada çatışır. Gerçek hayatta bu tür çatışmalar sıra ile (bir hafta opera, bir hafta futbol) ya da yazı-tura gibi bir dış sinyalle çözülür. Dış sinyal fikri, 3.5.7'deki korelasyonlu dengenin çekirdeğidir.

3.5.6 Hata yapan rakip

Denge kavramı, herkesin ötekilerin davranışını doğru bildiğini varsayar. Bu varsayım biraz gevşetilsin: rakip küçük bir ε olasılıkla niyet etmediği hamleyi yapsın (eli kaysın, yanlış tuşa bassın). En iyi tepki değişir mi?

Mahkûmun İkilemi: değişmez. Rakip p olasılıkla Sessiz kalsın. Sam'in Sessiz kalmasının beklenen kazancı $-1 \cdot p - 3 \cdot (1 - p) = 2p - 3$, İhanet'inki $0 \cdot p - 2 \cdot (1 - p) = 2p - 2$'dir. Fark her p için $(2p - 2) - (2p - 3) = 1$'dir. İhanet her p için tam 1 yıl daha iyidir; rakibin hata payı ne olursa olsun karar değişmez.

Geyik Avı: hata payına bağlıdır. Rakip q olasılıkla Tavşan'a kaçsın. Geyik'in beklenen kazancı $4 \cdot (1 - q)$, Tavşan'ınki her durumda 3'tür. Geyik, ancak $4(1 - q) \geq 3$, yani $q \leq 1/4$ iken en iyi tepkidir. Ortağın tavşana kaçma olasılığı %25'i aşarsa güvenli seçim, risk baskın Tavşan'dır.

Genel ders şudur: baskın strateji dengesi hatalara dayanıklıdır, koordinasyon dengeleri değildir. Baskın strateji, rakibin ne yapacağına hiç bağlı olmadığı için rakibin hataları onu etkilemez. Koordinasyon dengesi ise karşı tarafın doğru oynayacağına duyulan güvene dayanır; güven azaldıkça denge sarsılır.

3.5.7 Korelasyonlu denge

Işıksız bir kavşağa iki sürücü aynı anda gelir. Her biri yol verir (Kır) ya da geçer (Düz git). Bu bir Tavuk oyunudur: (Kır, Kır) = (0, 0), (Kır, Düz) = (−1, 1), (Düz, Kır) = (1, −1), (Düz, Düz) = (−10, −10). İki saf denge birer sürücüyü kayırır; koordinasyon yoksa simetrik karışık dengede her sürücü 9/10 olasılıkla kırar, %1 olasılıkla çarpışma olur ve beklenen kazanç −1/10'dur (Tavuk örneği).

Şimdi kavşağa bir trafik ışığı konsun. Işık bir aracı gibi çalışır: önceden ilan edilmiş bir olasılık dağılımı μ'den bir hücre çeker ve her sürücüye *yalnız kendi önerisini* fısıldar. Burada μ, (Kır, Kır), (Kır, Düz) ve (Düz, Kır) hücrelerine 1/3'er olasılık, (Düz, Düz)'e 0 olasılık verir.

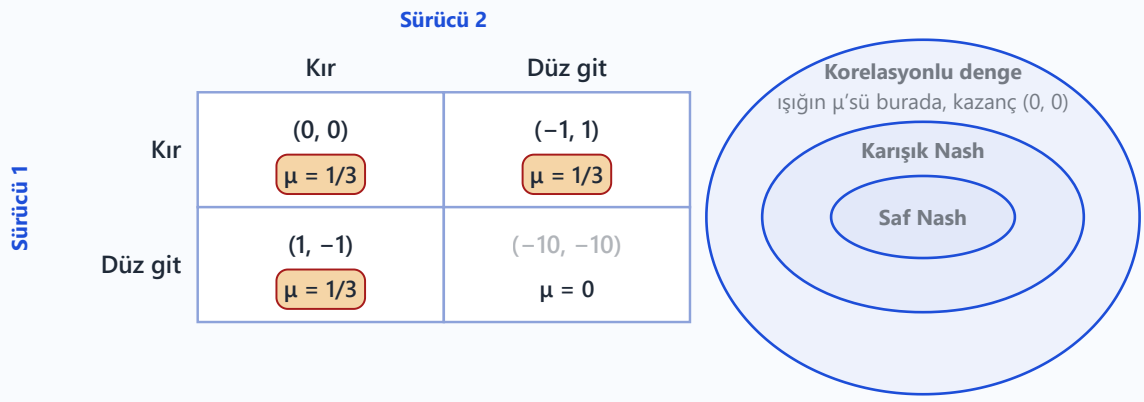
Tanım 3.8

Profiller üzerindeki bir olasılık dağılımı μ şu koşulu sağlıyorsa bir **korelasyonlu dengedir** (*correlated equilibrium, CE*): her oyuncu ve ona önerilebilecek her strateji için, oyuncu yalnız kendi önerisini bildiğinde öneriye uymanın beklenen kazancı, öneriden sapmanın beklenen kazancından az değildir. Bu koşullara **uyma kısıtları** denir.

Uyma kısıtlarını denetlemek için **koşullu olasılık** gerekir. Bir olayın, başka bir olayın gerçekleştiği bilindiğinde olasılığı $P(A \mid B) = P(A \text{ ve } B) / P(B)$'dir. **Bayes kuralı** bu tanıma dayanır; sürücü kendi önerisini görünce rakibin önerisi hakkındaki inancını bu yolla günceller.

Örnek 3.14

1. **Öneri "Kır" ise.** Sürücü 1'e Kır denen hücreler (Kır, Kır) ve (Kır, Düz)'dür; ikisi de 1/3 olasılıklıdır, toplam 2/3. Koşullu olasılıklar $(1/3)/(2/3) = 1/2$'şerdir: rakip ½ olasılıkla Kır, ½ olasılıkla Düz önerisi almıştır.
2. Uyarsa (Kır): $½ \cdot 0 + ½ \cdot (−1) = −½$. Saparsa (Düz): $½ \cdot 1 + ½ \cdot (−10) = −4{,}5$. $−½ > −4{,}5$; uymak daha iyidir.
3. **Öneri "Düz git" ise.** Sürücü 1'e Düz denen tek hücre (Düz, Kır)'dır; rakibe kesin "Kır" denmiştir. Uyarsa 1, saparsa (Kır) 0 alır. Uymak daha iyidir.
4. Oyun simetrik olduğu için Sürücü 2'nin kısıtları da aynı biçimde sağlanır. Hiçbir öneriden sapmak kârlı değildir; μ bir korelasyonlu dengedir.
5. **Kazanç.** Her sürücünün beklenen kazancı $⅓ \cdot 0 + ⅓ \cdot (−1) + ⅓ \cdot 1 = 0$'dır. Karışık dengedeki −1/10'dan iyidir ve çarpışma hiç olmaz.



Şekil 3.21: Kavşakta trafik ışığı. Solda ışığın dağılımı μ: üç hücreye 1/3, çarpışma hücresine 0. Sağda kümeler: her saf Nash dengesi bir karışık Nash dengesi, her Nash dengesi bir korelasyonlu dengedir.

Işığın gücü, sürücülerin kendi önerileri dışında bir şey bilmemesinden gelir. Kır önerisi alan sürücü, rakibin de Kır önerisi almış olabileceğini bilir; bu belirsizlik ona sapmayı pahalı kılar. Aracı hiçbir zorlama yapmaz; uymak herkesin kendi çıkarınadır.

Korelasyonlu denge iki özelliğiyle önemlidir. Birincisi, kapsamı geniştir: her Nash dengesi bir korelasyonlu dengedir (aracı, oyuncuların bağımsız karışımlarını aynen önerirse uyma kısıtları Nash koşuluna dönüşür), ama tersi doğru değildir. Bu örnekteki ışık hiçbir Nash dengesine karşılık gelmez. İkincisi, hesaplaması kolaydır: uyma kısıtlarının hepsi μ'nün olasılıklarında doğrusaldır. Bu yüzden bir korelasyonlu denge **doğrusal programlama** (*linear programming, LP*) ile polinom zamanda bulunur; Nash dengesi için böyle bir yöntem bilinmez (3.6.6). Işık sürelerini trafik yoğunluğuna göre ayarlamak ise ayrı bir tasarım kararıdır: hangi korelasyonlu dengenin seçileceği, kuralları koyanın elindedir.

3.6 Minimaks ve karmaşıklık

İki konu bir arada ele alınır: çıkarların tam olarak zıt olduğu sıfır toplamlı oyunlar ve genel oyunlarda dengeyi bulmanın bilgisayar için ne kadar zor olduğu. Sıfır toplamlı oyunlarda denge kavramı özellikle güçlüdür ve verimli biçimde hesaplanır.

3.6.1 Sıfır toplamlı oyun ve eyer noktası

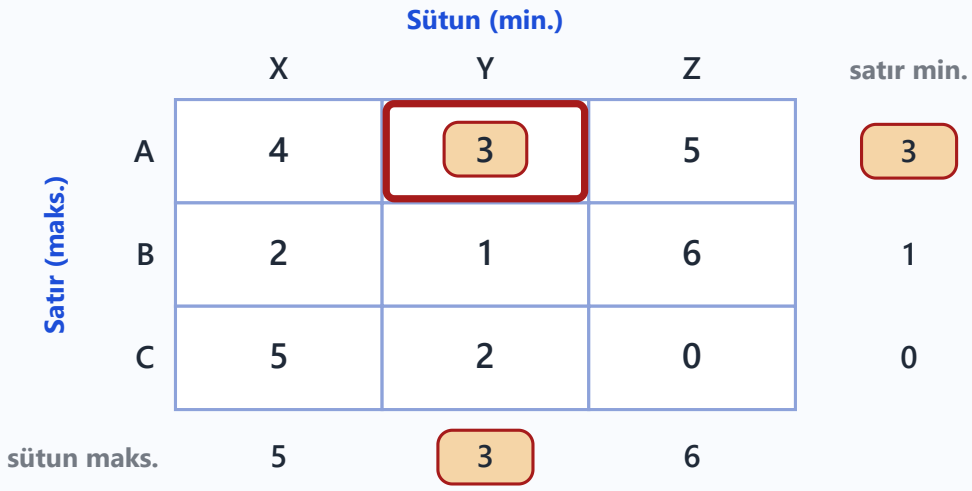
Sıfır toplamlı oyunda her hücrede iki oyuncunun kazançlarının toplamı sıfırdır: $u_2 = -u_1$. Bu yüzden hücreye yalnız satır oyuncusunun kazancı yazılır; sütun oyuncusu aynı miktarı kaybeder. Satır oyuncusu bu sayıyı büyütmek, sütun oyuncusu küçültmek ister.

Böyle bir oyunda doğal bir akıl yürütme kötümser düşünmektir. Satır oyuncusu der ki: “Hangi satırı seçersem seçeyim, rakibim bana en kötüsünü yapacak. O hâlde her satırın en kötü değerine bakıp en iyisini seçeyim.” Sütun oyuncusu da benzer düşünür.

Tanım 3.9

Satır oyuncusunun kazanç matrisi A olan iki kişilik sıfır toplamlı oyunda:

- **Maksimin** (*maximin*): $max_r\ min_c\ A[r][c]$. Satır oyuncusunun saf stratejilerle garanti edebileceği en büyük kazançtır.
- **Minimaks** (*minimax*): $min_c\ max_r\ A[r][c]$. Sütun oyuncusunun saf stratejilerle garanti edebileceği en küçük kayıptır.
- Maksimin ile minimaks eşitse, bu değere ulaşılan hücreye **eyer noktası** (*saddle point*) denir. Eyer noktası, satırının en küçüğü ve sütununun en büyüğü olan hücredir.



Şekil 3.22: Sıfır toplamlı oyunda maksimin ve minimaks. Sağ sütunda satır en küçükleri (3, 1, 0), alt satırda sütun en büyükleri (5, 3, 6). Maksimin = minimaks = 3; eyer noktası (A, Y).

Örnek 3.15

Matris: A = (4, 3, 5), B = (2, 1, 6), C = (5, 2, 0); sütunlar X, Y, Z.

1. **Satır oyuncusu kötümser düşünür.** Her satırda rakibin ona bırakacağı en kötü değer: A'da 3, B'de 1, C'de 0.
2. **Maksimin** = max(3, 1, 0) = 3 (satır A). Satır oyuncusu A'yı seçerek en az 3'ü garanti eder.
3. **Sütun oyuncusu da kötümser.** Her sütunda satır oyuncusunun alabileceği en yüksek değer: X'te 5, Y'de 3, Z'de 6.
4. **Minimaks** = min(5, 3, 6) = 3 (sütun Y). Sütun oyuncusu Y'yi seçerek en fazla 3 kaybetmeyi garanti eder.

5. **Sonuç.** Maksimin = minimaks = 3. (A, Y) hücresindeki 3, A satırının en küçüğü ve Y sütununun en büyüğüdür: eyer noktası. Bu hücre bir saf Nash dengesidir ve **oyunun değeri** (*value of the game*) $v = 3$'tür.

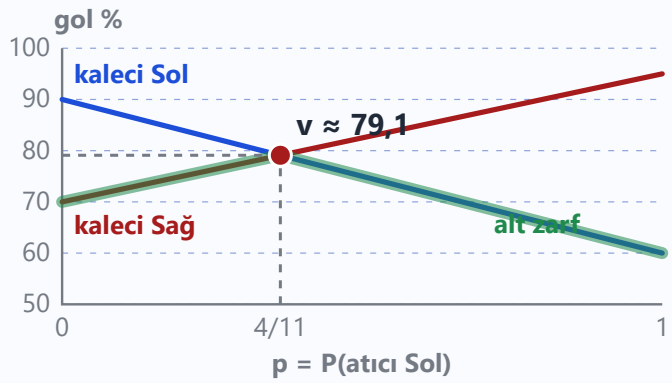
Eyer noktasının Nash dengesi olduğu kolayca görülür. Satır oyuncusu Y sütununda başka bir satıra geçerse 3'ten fazla alamaz, çünkü 3 o sütunun en büyüğüdür. Sütun oyuncusu A satırında başka bir sütuna geçerse 3'ten az veremez, çünkü 3 o satırın en küçüğüdür. Adı da buradan gelir: at eyeri bir yönde (sütun boyunca) tepe, öteki yönde (satır boyunca) çukurdur.

Maksimin ile minimaks her zaman eşit değildir; ama maksimin hiçbir zaman minimakstan büyük olamaz. Nedeni şudur: maksimin satırı r^*, minimaks sütunu c^* olsun. r^* satırının en küçüğü, bu satırdaki $A[r^*][c^*]$'dan büyük olamaz; c^* sütununun en büyüğü de bu sütundaki $A[r^*][c^*]$'dan küçük olamaz. Dolayısıyla maksimin $\le A[r^*][c^*] \le$ minimaks'tır. Eşitlik yoksa eyer noktası yoktur ve oyuncuların garantileri arasında bir boşluk kalır.

3.6.2 Eyer noktası yoksa: penaltı atışı

Penaltıda atıcı ve kaleci aynı anda Sol ya da Sağ seçer (Sol ve Sağ atıcıya göredir). Hücrede gol yüzdesi yazılıdır; atıcı bu sayıyı büyütmek, kaleci küçültmek ister. Sayılar örnek değerlerdir; gerçek verideki yapı benzerdir (Palacios-Huerta, 2003).

Atıcı \ Kaleci	Sol	Sağ
Sol	60	95
Sağ	90	70



Şekil 3.23: Penaltı atışı. Matriste gol yüzdeleri: atıcı Sol–kaleci Sol 60, atıcı Sol–kaleci Sağ 95, atıcı Sağ–kaleci Sol 90, atıcı Sağ–kaleci Sağ 70. Grafikte atıcının p = P(Sol) olasılığına göre iki doğru ve alt zarf (yeşil); zarfın tepesi $p^* = 4/11$, $v \approx 79{,}1$.

Örnek 3.16

1. **Saf garanti.** Satır en küçükleri 60 (Sol) ve 70 (Sağ): maksimin 70. Sütun en büyükleri 90 (kaleci Sol) ve 95 (kaleci Sağ): minimaks 90. $70 < 90$: eyer noktası yoktur.
2. **Atıcı karıştırır.** Atıcı p olasılıkla Sol'a vursun. Kaleci Sol'a atlarsa gol yüzdesi $60p + 90(1-p) = 90 - 30p$; Sağ'a atlarsa $95p + 70(1-p) = 70 + 25p$'dir.
3. **Atıcının garantisi.** Kaleci gol yüzdesini en aza indirmeye çalışır; bu yüzden her p için atıcının garantisi iki doğrudan küçük olanıdır. Bu küçük değerlerin oluşturduğu çizgiye **alt zarf** denir.
4. **Zarfın tepesi.** Alt zarf, iki doğrunun kesiştiği yerde en yüksektir: $90 - 30p = 70 + 25p \Rightarrow 20 = 55p \Rightarrow p^* = 4/11 \approx 0{,}36$. Değer $v = 90 - 30 \cdot 4/11 = 870/11 \approx 79{,}1$.
5. **Kaleci tarafı.** Kaleci q olasılıkla Sol'a atlasın. Atıcıyı kayıtsız bırakması gerekir: $60q + 95(1-q) = 90q + 70(1-q) \Rightarrow 95 - 35q = 70 + 20q \Rightarrow q^* = 5/11$. Bu durumda gol yüzdesi $95 - 35 \cdot 5/11 = 870/11$'dir.

Kalecinin garantisi de %79,1'dir. Karışık stratejilerle maks-min = min-maks olur. Atıcı her vuruşta %79,1'i garanti eder; kaleci de bundan fazla gol yememeyi garanti eder.

İlginç bir ayrıntı: atıcının güçlü tarafı Sol'dur (kaleci yanlış tarafa atlarsa %95), ama dengede Sol'a yalnız 4/11 olasılıkla vurur. Kaleci de Sol'a 5/11 olasılıkla, yani yarıdan seyrek atlar. Kayıtsızlık ilkesine göre her oyuncunun olasılığını rakibin sayıları belirler; bu yüzden "güçlü tarafa daha sık vur" sezgisi dengede geçerli değildir.

3.6.3 Güvenlik oyunları

Güvenlik oyunları (*security games*), sınırlı kaynakla çok sayıda hedefin korunduğu durumları modeller: havalimanları, limanlar, orman devriyeleri. Savunucu n hedeften k'sını koruyabilir. Hangi k hedefin seçileceği bir saf stratejidir ve bunların sayısı $C(n, k)$'dir. $C(n, k) = n! / (k!\,(n-k)!)$, n nesneden sırasız k tane seçme yollarının sayısıdır. $n = 20$, $k = 5$ için 15.504 saf strateji vardır.

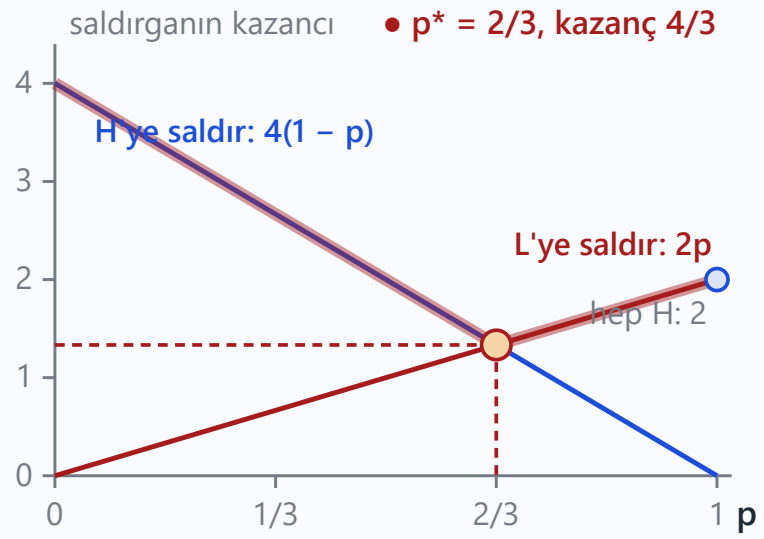
Saldırgan devriye düzenini gözlemleyebilir. Saldırgan sabit bir düzeni kolayca atlatır; bu yüzden savunucu rastgeleleştirir. Los Angeles Havalimanı'nda kullanılan ARMOR sistemi (2007), kontrol noktalarını ve köpekli devriyeleri bu modelle planladı (Tambe, 2011).

Örnek 3.17

Tek bekçi, iki hedef. Havalimanı H'nin değeri 4, liman L'nin değeri 2'dir. Bekçi H'yi p, L'yi $1 - p$ olasılıkla korur. Korunmayan hedefe saldıran değerini kazanır, korunana saldıran 0 alır.

1. Saldırgan H'ye saldırırsa beklenen kazancı $4 \cdot (1 - p)$'dir (H korunmuyorsa 4).
2. L'ye saldırırsa $2 \cdot p$'dir (bekçi H'deyse L korunmaz).
3. Bekçi saldırganı kayıtsız bırakır: $4(1 - p) = 2p \Rightarrow 4 = 6p \Rightarrow p = 2/3$.
4. Saldırganın kazancı $4 \cdot (1/3) = 4/3$'tür.

Bekçi hep H'yi korusaydı saldırgan L'ye saldırıp 2 alırdı. Rastgeleleştirme saldırganın kazancını 2'den 4/3'e indirir. Değerli hedef daha sık korunur, ama hiçbir hedef tamamen korumasız bırakılmaz.



Şekil 3.24: Güvenlik oyunu. Saldırganın iki seçeneğinin kazancı bekçinin H'yi koruma olasılığı p'ye göre çizilmiştir. Saldırgan büyük olanı seçer (üst zarf); bekçi zarfın en düşük noktasını, $p^* = 2/3$'ü seçer.

3.6.4 von Neumann'ın minimaks teoremi

Penaltı ve güvenlik örneklerinde karışık stratejilerle iki oyuncunun garantisi eşit çıktı. Bu bir rastlantı değildir. Teoremi yazmak için beklenen kazancın matris gösterimi gerekir. Satır oyuncusunun karışık stratejisi bir olasılık vektörü $x = (x_1, \dots, x_m)$, sütun oyuncusununki y olsun. Ay vektörünün her öğesi bir satırın y'ye karşı beklenen kazancıdır; $x^T A y$ ise bu satır beklentilerinin x'in olasılıklarıyla tartılmış toplamıdır, yani satır oyuncusunun beklenen kazancıdır. (x^T, x'in satır vektörü olarak yazıldığını gösterir.)

Teorem 3.6 (von Neumann'ın minimaks teoremi, 1928)

Her sonlu iki kişilik sıfır toplamlı oyunda

$$\max_x \min_y x^T A y = \min_y \max_x x^T A y = v$$

eşitliği sağlanır. v, oyunun değeridir.

Sol taraf, satır oyuncusunun karışık stratejilerle garanti edebileceği en büyük kazançtır; sağ taraf, sütun oyuncusunun garanti edebileceği en küçük kayıptır. Saf stratejilerde aralarında boşluk olabiliyordu (penaltıda 70 ile 90); karışık stratejilerde boşluk her zaman kapanır. Teoremin sonuçları şunlardır:

- **Denge = minimaks stratejileri.** (x^*, y^*) bir Nash dengesidir ancak ve ancak x^* bir maksimin, y^* bir minimaks stratejisidir. Tüm dengelerde satır oyuncusunun kazancı aynıdır: v.

- **Değiştirilebilirlik** (*interchangeability*). *(x, y)* ve *(x′, y′)* iki dengeyse *(x, y′)* de dengedir. Çünkü *x* bir maksimin, *y′* bir minimaks stratejisidir ve birinci sonuca göre böyle her çift bir dengedir. Oyuncuların hangi dengeyi oynayacaklarını koordine etmeleri gerekmez; sıfır toplamlı oyunda denge seçimi sorunu yoktur.
- **Doğrusal programlama.** Satır oyuncusu, her sütuna karşı en az *v* alacağı bir *x* karışımını ve olabilecek en büyük *v*'yi arar: *v*'yi en büyük yap; koşullar: her sütun *j* için $\Sigma_i\, x_i\, A[i][j] \geq v$, $\Sigma_i\, x_i = 1$, her $x_i \geq 0$. Bütün koşullar doğrusal olduğu için bu bir doğrusal programdır ve bilgisayar onu polinom zamanda çözer.
- **Öğrenme bağlantısı.** Oyuncular oyunu tekrar tekrar oynayıp geçmişte iyi giden stratejilerin ağırlığını artırarak öğrenirse, ortalama stratejileri minimaks stratejilerine yaklaşır.

Nash'in teoremi, minimaks teoreminin sıfır toplamlı olmayan *n* kişilik oyunlara genellemesi olarak görülebilir. Ancak genelleme bedelsiz değildir. Genel oyunda "değer" tek değildir (Cinsiyetler Savaşı'nda dengeler farklı kazançlar verir), değiştirilebilirlik bozulur (bir oyuncu bir dengenin, öteki başka bir dengenin stratejisini oynarsa sonuç denge olmayabilir) ve denge doğrusal programla bulunamaz.

3.6.5 Kod izleme: maksimin()

Satır oyuncusunun yalnız iki satırı varsa (*2 × n* oyun), maksimin stratejisi grafik yöntemiyle kolayca bulunur. Satır oyuncusu ilk satırı *p* olasılıkla oynasın. Her sütun *j*, *p*'ye bağlı bir doğru verir: $p \cdot A[0][j] + (1 - p) \cdot A[1][j]$. Sütun oyuncusu her *p*'ye en kötü sütunla cevap verir; bu yüzden satır oyuncusunun garantisi bu doğruların alt zarfıdır: $g(p) = \min_j [p \cdot A[0][j] + (1 - p) \cdot A[1][j]]$. Alt zarf parça parça doğrusal olduğu için en yüksek noktası ya uçlarda (*p = 0*, *p = 1*) ya da iki doğrunun kesişiminde bulunur. Fonksiyon bu adayları toplar ve en iyisini seçer.

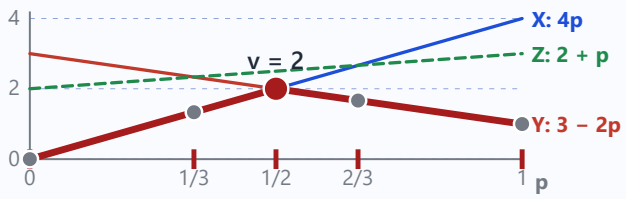
```
from fractions import Fraction as F
def maksimin(A):                  # A: 2 x n
    n, aday = len(A[0]), [F(0), F(1)]
    for j in range(n):
        for k in range(j + 1, n):
            d = A[0][j] - A[1][j] - A[0][k] + A[1][k]
            if d != 0:
                aday.append(F(A[1][k] - A[1][j], d))
    g = lambda p: min(p*a+(1-p)*b for a, b in zip(*A))
    p = max((c for c in aday if 0 <= c <= 1), key=g)
    return p, g(p)
```

İki doğrunun kesişimi şöyle bulunur. *j* ve *k* sütunlarının doğruları eşitlenir: $p\,A[0][j] + (1 - p)\,A[1][j] = p\,A[0][k] + (1 - p)\,A[1][k]$. Terimler düzenlenince $p \cdot d = A[1][k] - A[1][j]$ çıkar; burada $d = A[0][j] - A[1][j] - A[0][k] + A[1][k]$'dır. *d = 0* ise doğrular paraleldir ve kesişmez. `zip(*A)` ifadesi, matrisin sütunlarını (a, b) çiftleri olarak verir: a ilk satırdaki, b ikinci satırdaki sayıdır.

Örnek matris `A = [[4, 1, 3], [0, 3, 2]]` olsun; sütunlar X, Y, Z'dir. Doğrular X: *4p*, Y: *p + 3(1 − p) = 3 − 2p*, Z: *3p + 2(1 − p) = 2 + p*'dir. Saf maksimin 1, minimaks 3'tür; eyer noktası yoktur.

Adım	Hesap	Sonuç
Başlangıç	`n = 3`, uç noktalar	`aday = [0, 1]`
X ile Y	d = 4 − 0 − 1 + 3 = 6; p = (3 − 0)/6	1/2 eklenir
X ile Z	d = 4 − 0 − 3 + 2 = 3; p = (2 − 0)/3	2/3 eklenir
Y ile Z	d = 1 − 3 − 3 + 2 = −3; p = (2 − 3)/(−3)	1/3 eklenir
g(0)	min(0, 3, 2)	0
g(1)	min(4, 1, 3)	1
g(1/2)	min(2, 2, 5/2)	2 (en iyi)
g(2/3)	min(8/3, 5/3, 8/3)	5/3
g(1/3)	min(4/3, 7/3, 7/3)	4/3
dönüş	`max(..., key=g)`	(1/2, 2)

Tablo 3.19: `maksimin(A)` izlemesi.



aday p	0	1	1/2	2/3	1/3
g(p)	0	1	2	5/3	4/3

Şekil 3.25: `maksimin()` izlemesinin son hâli. Kırmızı kalın çizgi alt zarf *g(p)*'dir; aday noktalar işaretlidir. En büyük garanti *p = 1/2*'de *v = 2*'dir. Z'nin doğrusu (yeşil kesikli) zarfın üstünde kalır.

En büyük garanti *p = 1/2*'de *v = 2*'dir. Z sütununun doğrusu bu noktada *2,5* verir ve zarfın üstünde kalır; sütun oyuncusu Z'yi hiç oynamaz. Minimaks tarafı da denetlenir: sütun oyuncusu X'i 1/3, Y'yi 2/3 olasılıkla oynarsa satır oyuncusunun ilk satırı *4/3 + 2/3 = 2*, ikinci satırı *0 + 2 = 2* verir. Satır oyuncusu en fazla 2 alabilir. Maks-min = min-maks = 2'dir.

3.6.6 Dengeyi hesaplamak ne kadar zor?

Bilgisayar biliminde bir problemin “verimli çözülebilir” sayılması, girdi büyüdükçe çalışma süresinin girdi boyutunun bir polinomu (örneğin n^2 ya da n^3) kadar büyümesi demektir. Süre 2^n gibi üstel büyüyorsa küçük bir büyüme bile hesaplamayı imkânsız kılar. Nash dengesi için tablo şöyledir:

Problem	Karmaşıklık	Yöntem / kaynak
Saf Nash dengesi (matris verili)	Polinom	En iyi tepki işaretleme, O(mn)
İki kişilik sıfır toplamlı, karışık	Polinom	Doğrusal programlama (minimaks)
İki kişilik genel oyun, karışık	PPAD-tam	Chen–Deng (2006)

Problem	Karmaşıklık	Yöntem / kaynak
Üç ve daha çok oyuncu, tam denge	FIXP-tam	Etessami–Yannakakis (2007)
Üç ve daha çok oyuncu, ε-yaklaşık denge	PPAD-tam	Daskalakis–Goldberg–Papadimitriou (2006)
En yüksek refahlı denge / tek denge mi?	NP-zor	Gilboa–Zemel (1989); Conitzer–Sandholm (2008)

Tablo 3.20: Denge problemlerinin karmaşıklığı.

Tablonun ilk iki satırı bu bölümde gördüğümüz yöntemlerdir ve verimlidir. Genel oyunlarda karışık denge bulmak için pratikte iki yöntem kullanılır. **Destek sayımı** (*support enumeration*), her oyuncu için olası bir destek kümesi tahmin eder ve kayıtsızlık denklemlerini çözer; bu, 3.4.5'teki yöntemin bütün destek çiftleri üzerinde denenmesidir. **Lemke–Howson algoritması**, bir başlangıç noktasından bir dengeye giden bir yolu izler. İki yöntem de pratikte çoğu zaman iyi çalışır, ama en kötü durumda üstel süre alır: destek kümelerinin sayısı stratejilerle birlikte üstel büyür.

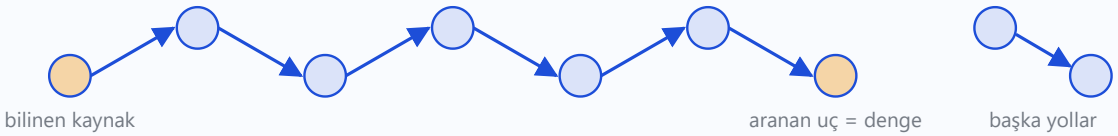
Üç ve daha çok oyunculu oyunlarda bir sorun daha vardır: denge olasılıkları irrasyonel sayılar (örneğin $\sqrt{2}$ içeren sayılar) olabilir ve bilgisayarda tam olarak yazılamaz. İki kişilik oyunlarda kayıtsızlık denklemleri doğrusal olduğu için dengeler rasyoneldir (kesirlerle yazılır); üç oyuncuda denklemlerde olasılık çarpımları görülür ve kökler irrasyonel çıkabilir. Bu yüzden çok oyunculu oyunlarda **ε-yaklaşık denge** (*ε-approximate equilibrium*), yani hiçbir sapmanın ε'dan fazla kazandırmadığı bir profil aranır.

Tablonun son satırı, "herhangi bir" denge yerine "özel bir" dengeyi aramanın zorluğudur: en yüksek sosyal refahlı dengeyi bulmak ya da oyunun tek dengesi olup olmadığına karar vermek NP-zordur. İleri okuma için: sabit bir ε (örneğin 0,1) için iki kişilik oyunlarda yarı-polinom süreli bir algoritma vardır (Lipton, Markakis ve Mehta, 2003); çok oyunculu oyunlarda ise sabit ε için bile problem PPAD-zor kalır (Rubinstein, 2015).

3.6.7 PPAD

Zor problemler denince akla genellikle NP-tamlık gelir. NP-tam problemler "çözüm var mı?" biçimindeki sorulardır (örneğin "bu çizgede tüm düğümleri bir kez dolaşan bir yol var mı?"). Nash dengesi için bu soru önemsizdir: Nash'in teoremi dengenin her zaman var olduğunu söyler, cevap her zaman "evet"tir. Bu yüzden NP-tamlık, çözümü garanti olan bu tür problemler için uygun bir ölçü değildir.

PPAD sınıfı (Papadimitriou, 1994) tam olarak bu durum için tanımlanmıştır. PPAD, çözümün var olduğu kesin bilinen, ama bulmak için çok uzun bir yolu adım adım izlemek gerekebilen problemlerin sınıfıdır. Adı da bu yapıdan gelir: *Polynomial Parity Arguments on Directed graphs* (yönlü çizgelerde polinom eşlik argümanları).



Şekil 3.26: PPAD'in yol benzetmesi. Bilinen bir kaynaktan başlayan yönlü yol bir uçta bitmek zorundadır; aranan uç dengedir. Yol üstel uzunlukta olabilir. Sağda başka yollar da vardır.

Şekil 3.26'daki benzetmeyi şöyle okuyabiliriz. Bir yolun başı bilinir. Her düğümden bir sonrakine nasıl gidileceği kolayca hesaplanır. Yol sonsuz olamaz, çünkü düğüm sayısı sonludur; demek ki yolun bir sonu kesin vardır. Ama yol üstel uzunlukta olabilir ve sonuna yürümek astronomik zaman alabilir. Lemke–Howson algoritması tam olarak böyle bir yolu izler. Nash dengesi bulmak bu sınıfta tamdır, yani sınıfın en zor problemlerindendir. Bu yüzden genel oyunlarda dengenin verimli biçimde bulunamayabileceği düşünülür. Bu durum kavramsal bir soru da doğurur: bir bilgisayar bile dengeyi hızlıca bulamıyorsa, gerçek oyuncular dengeyi "bulur" mu? Denge kavramının bir tahmin aracı olarak gücü, bu soruyla sınanır.

3.6.8 Yaklaşık denge ve hesaplama araçları

Tanım 3.10

Kazançlar [0, 1] aralığına ölçeklenmiş olsun. Bir profil bir **ε-Nash dengesidir** ancak ve ancak hiçbir oyuncu tek taraflı sapmayla kazancını ε'dan fazla artıramaz.

Ölçekleme önemlidir: kazançlar 1000 ile çarpılırsa her sapma kazancı da 1000 ile çarpılır ve aynı profil için ε anlamsızlaşır. [0, 1] aralığında $\varepsilon = 0{,}01$, "hiçbir oyuncu sapmakla en büyük olası kazanç farkının %1'inden fazlasını kazanamaz" demektir. Gerçek zamanlı sistemler (reklam açık artırmaları, piyasalar) çoğu zaman bu kadarıyla yetinir.

Tam dengeyi bulmak zor olabilir, ama "neredeyse kimsenin sapmak istemediği" bir profil çoğu zaman hızlıca bulunur. Az sayıda stratejiyi eşit olasılıkla karıştıran basit profiller bile iyi yaklaşıklık verir (Lipton, Markakis ve Mehta, 2003). İki kişilik oyunlarda çok basit bir profil ½-yaklaşık dengedir (Daskalakis, Mehta ve Papadimitriou, 2006): satır oyuncusunun rastgele bir i satırı seçilir; sütun oyuncusu i'ye en iyi tepki olan j sütununu oynar; k, j'ye en iyi tepki olan satırdır; satır oyuncusu i ile k'yi yarı yarıya karıştırır.

Pratikte denge hesaplamak için hazır araçlar kullanılır. Python kütüphanesi **Nashpy**, iki kişilik oyunlar için destek sayımı ve Lemke–Howson algoritmasını sunar. Denetim oyunu Nashpy ile şöyle çözülür:

```
import nashpy as nash
A = [[2, 2], [0, 4]]    # işçi: Çalış / Kaytar
B = [[1, 2], [-1, -4]]  # müdür: Denetle / Denetleme
oyun = nash.Game(A, B)
for x, y in oyun.support_enumeration():
    print(x, y)  # [0.75 0.25] [0.5 0.5]
```

`A` işçinin, `B` müdürün kazanç matrisidir; satırlar Çalış/Kaytar, sütunlar Denetle/Denetleme'dir. `nash.Game(A, B)` oyunu kurar. `support_enumeration()` bütün destek çiftlerini dener ve bulduğu her dengeyi bir (`x` , `y`) çifti olarak verir. Çıktıdaki `[0.75 0.25]` işçinin (Çalış, Kaytar) olasılıklarıdır, yani $p^* = 3/4$; `[0.5 0.5]` müdürün (Denetle, Denetleme) olasılıklarıdır, yani $q^* = 1/2$. Oyunun saf dengesi olmadığı için başka çıktı yoktur. Sonuç, elle yapılan hesapla aynıdır. **Gambit** adlı araç da aynı işi (destek sayımı, Lemke–Howson, doğrusal programlama) bir arayüzle ve çok oyunculu oyunlar için yapar.

Nash dengesinin genellemeleri ve daraltılmış biçimleri de vardır. Korelasyonlu denge (3.5.7) bir genellemedir. Hamlelerin sırayla yapıldığı oyunlarda ise daha dar bir kavram, **alt oyun mükemmel denge** (*subgame perfect equilibrium*) kullanılır: oyun ağacında geriye doğru çıkarımla bulunur ve inandırıcı olmayan tehditleri dışarıda bırakır. Her alt oyun mükemmel denge bir Nash dengesidir; tersi doğru değildir.

3.7 Tartışma ve kontrol listesi

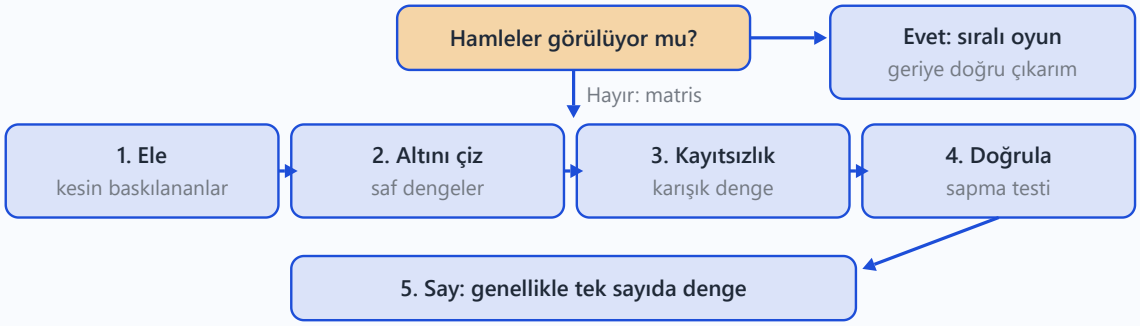
Kavramın sınırları ve bir oyunu çözerken izlenecek adımlar aşağıda bir araya getirilmiştir.

Denge adil mi? Olmak zorunda değildir. Denge yalnızca karşılıklı en iyi tepkidir. Cinsiyetler Savaşı'nda saf dengelerin kazançları eşit değildir; Mahkûmun İkilemi'nde ise herkes için daha iyi bir sonuç vardır. Adalet ile denge farklı kavramlardır. Bir sonucun denge olması, onun iyi ya da istenir olduğunu göstermez; yalnızca kendi kendini sürdürdüğünü gösterir.

İnsanlar rasyonel mi? Gerçek kararlar bilgi eksikliği, zaman baskısı, duygular ve sosyal normlardan etkilenir. Herbert Simon bu duruma **sınırlı rasyonellik** (*bounded rationality*) adını verdi. Tek seferlik Mahkûmun İkilemi deneylerinde katılımcıların yaklaşık yarısının işbirliği yapması bunun bir göstergesidir. Ancak model yine de değerlidir: stratejik bağımlılığı incelemek için açık ve kesin bir karşılaştırma noktası sunar. Gerçek davranış modelden saptığında, sapmanın nedeni (güven, itibar, adalet duygusu) ayrıca incelenebilir.

Etik sınırlar. Aldatma, manipülasyon, adil olmayan avantaj… "Kazanmak için her yol mubah mı?" sorusunun cevabını oyun kuramı vermez. Stratejik davranışın sınırlarını kurallar ve kurumlar çizer. Kuralları, istenen sonucun bir denge olacağı biçimde tasarlamak ise **mekanizma tasarımının** (*mechanism design*) konusudur. Bu bölümdeki örnekler bu fikrin ilk izlerini taşır: denetim oyununda cezayı artırmak denetim olasılığını değiştirdi, trafik ışığı ise çarpışmasız bir korelasyonlu denge kurdu.

Kontrol listesi. Bir oyunu adım adım çözmek için aşağıdaki sıra izlenir (Şekil 3.27).



Şekil 3.27: Bir oyunu çözmenin adımları. Hamleler görülüyorsa oyun sıralıdır ve geriye doğru çıkarım kullanılır; görülmüyorsa matris üzerinde eleme, altını çizme, kayıtsızlık, doğrulama ve sayma adımları izlenir.

1. **Önce okuma.** Kim satır, kim sütun? Hücrede ilk sayı kimin? Sayılar kazanç mı, maliyet mi (yıl, dakika)? Maliyetse eksi işaretle kazanca çevrilir.
2. **Ele.** Kesin baskılanan stratejiler yinelemeli olarak silinir. Karışımlar da denenir.
3. **Altını çiz.** Kalan oyunda en iyi tepkiler işaretlenir; iki işaretli hücreler saf dengelerdir.
4. **Kayıtsızlık.** Karışık denge kayıtsızlık denklemleriyle aranır; olasılıkların 0 ile 1 arasında olduğu denetlenir.
5. **Doğrula.** Her aday için sapma testi yapılır.
6. **Say.** Eşitliksiz oyunlarda denge sayısı genellikle tek sayıdır; çift sayıda denge bulunduysa bir denge atlanmış olabilir.
7. **Sonra yazma.** Her denge profil olarak yazılır: örneğin (Orta, Sol) = (3, 5). Karışık dengede iki oyuncunun olasılıkları ve beklenen kazançları verilir.

3.8 Özet

- **Nash dengesi**, hiçbir oyuncunun ötekiler sabitken tek başına sapmakla kazancını artıramadığı strateji profilidir: $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$. Denge kararlılık söyler; verimlilik ya da adalet söylemez (Mahkûmun İkilemi).
- **En iyi tepki**, ötekilerin seçimine karşı kazancı en büyük yapan stratejiler kümesidir. Denge, karşılıklı en iyi tepkidir. Matris oyunlarında saf dengeler, her sütunda satır oyuncusunun ve her satırda sütun oyuncusunun en büyük kazancı işaretlenerek $O(mn)$ sürede bulunur; eşitlikte bütün en büyükler işaretlenir.
- Sürekli stratejilerde en iyi tepki türevle bulunur. **Cournot** ($P = 12 - Y$) dengesi 4'er birim ve 16'şar kârdır; kartel (3'er, 18'er) dengede değildir. **Bertrand**'da fiyat maliyete iner ve kâr sıfırdır.

- **Kesin baskılanan** strateji hiçbir inanca karşı en iyi tepki değildir. Yinelemeli kesin eleme hiçbir dengeyi silmez ve sıradan bağımsızdır; zayıf baskılananları silmek denge kaybettirebilir. Karışımlar da baskılayabilir.
- **Karışık dengede** desteğindeki her saf strateji eşit beklenen kazanç verir (kayıtsızlık ilkesi). Bu yüzden bir oyuncunun denge olasılığı rakibinin kazançlarından hesaplanır. Denetim oyununda *(p*, q*) = (3/4, 1/2)*'dir ve ceza değişince yalnız *q** değişir.
- **Nash'in varlık teoremi**: sonlu her oyunun en az bir karışık dengesi vardır (Kakutani / Brouwer sabit noktası). Dejenere olmayan oyunlarda denge sayısı tek sayıdır (Wilson).
- **Çoklu dengede** kuram tek sonucu seçmeyebilir. Pareto baskınlık, risk baskınlık, odak noktası ve öğrenme farklı seçimler önerebilir (Geyik Avı'nda (G, G) Pareto baskın, (T, T) risk baskın). Baskın strateji dengeleri hatalara dayanıklıdır, koordinasyon dengeleri değildir.
- **Korelasyonlu dengede** bir aracının önerisine uymak herkesin çıkarınadır. Uyma kısıtları doğrusal olduğu için doğrusal programlamayla bulunur; trafik ışığı örneğinde kazanç 0'dır, karışık Nash dengesinde −1/10'dur.
- **Sıfır toplamlı** oyunlarda maks-min = min-maks = *v*'dir (von Neumann). Eyer noktası yoksa karışık stratejiler gerekir (penaltıda *p* = 4/11*, *v ≈ %79,1*); denge doğrusal programla bulunur ve dengeler değiştirilebilirdir.
- **Karmaşıklık**: iki kişilik genel oyunda karışık denge bulmak PPAD-tamdır; pratikte destek sayımı, Lemke–Howson ve ε-yaklaşık dengeler kullanılır.

Kavram	Ne söyler?	Nasıl bulunur?
Nash dengesi	Kimse tek başına sapmak istemez	Karşılıklı en iyi tepki
Baskın strateji eleme	Rasyonel oyuncu baskılananı oynamaz	Kesin baskılananları yinelemeli sil
Karışık denge	Destekteki stratejiler eşit kazandırır	Kayıtsızlık denklemleri
Varlık (Nash 1950)	Sonlu oyunda karışık denge hep var	Kakutani / Brouwer sabit noktası
Çoklu denge	Kuram tek sonucu seçmeyebilir	Pareto / risk baskınlık, odak noktası
Korelasyonlu denge	Aracının önerisine uymak herkesin çıkarına	Uyma kısıtları + doğrusal programlama
Minimaks	Sıfır toplamlıda maks-min = min-maks = v	Eyer noktası ya da LP
Karmaşıklık	Genel oyunda denge PPAD-tam	Lemke–Howson, destek sayımı

Tablo 3.21: Bölümün kavramları bir bakışta.

3.9 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

Nash dengesi *Nash equilibrium* · saf Nash dengesi *pure Nash equilibrium* · strateji profili *strategy profile* · strateji kümesi *strategy set* · kazanç fonksiyonu *payoff function* · normal form *normal form* · tek taraflı sapma *unilateral deviation* · Pareto optimal *Pareto optimal* · en iyi tepki *best response*

en iyi tepki eşlemesi *best-response correspondence* · Cournot duopolü *Cournot duopoly* · kartel *cartel* · Bertrand paradoksu *Bertrand paradox* · kesin baskılama *strict dominance* · zayıf baskılama *weak dominance* · baskın strateji *dominant strategy* · baskın strateji dengesi *dominant-strategy equilibrium* · yinelemeli eleme *IESDS* · ortak bilgi *common knowledge* · karışık strateji *mixed strategy* · destek *support* · beklenen kazanç *expected payoff* · kayıtsızlık ilkesi *indifference principle* · denetim oyunu *inspection game* · Nash'in varlık teoremi *Nash's existence theorem* · sabit nokta teoremi *fixed-point theorem* · tek sayılılık teoremi *oddness theorem* · dejenere oyun *degenerate game* · çoklu denge *multiple equilibria* · denge seçimi *equilibrium selection* · Pareto baskınlık *Pareto dominance* · risk baskınlık *risk dominance* · odak noktası *focal point* · korelasyonlu denge *correlated equilibrium* · sıfır toplamlı oyun *zero-sum game* · maksimin *maximin* · minimaks *minimax* · eyer noktası *saddle point* · oyunun değeri *value of the game* · minimaks teoremi *minimax theorem* · değiştirilebilirlik *interchangeability* · güvenlik oyunu *security game* · doğrusal programlama *linear programming* · destek sayımı *support enumeration* · Lemke–Howson algoritması *Lemke–Howson algorithm* · PPAD-tam *PPAD-complete* · ε-yaklaşık denge *ε-approximate equilibrium*

3.10 Kendini Dene

1. Hücrelerde (u_1, u_2): (Üst, Sol) = (3, 2), (Üst, Sağ) = (1, 1), (Alt, Sol) = (0, 0), (Alt, Sağ) = (2, 3). Oyunun bütün Nash dengeleri nelerdir? Karışık dengede iki oyuncunun beklenen kazancı kaçtır?

   Cevap

   Satır en iyileri: Sol → Üst (3 > 0), Sağ → Alt (2 > 1). Sütun en iyileri: Üst → Sol (2 > 1), Alt → Sağ (3 > 0). Saf dengeler (Üst, Sol) ve (Alt, Sağ)'dır. Karışık denge: satır oyuncusunun kayıtsızlığı $3q + 1(1 - q) = 0 \cdot q + 2(1 - q)$, yani $1 + 2q = 2 - 2q$ ve $q = P(Sol) = 1/4$. Sütun oyuncusunun kayıtsızlığı $2p + 0 = 1 \cdot p + 3(1 - p)$, yani $p = P(Üst) = 3/4$. Beklenen kazançlar: satır $3 \cdot ¼ + 1 \cdot ¾ = 3/2$, sütun $2 \cdot ¾ = 3/2$. Toplam üç denge vardır (tek sayı kuralına uygun).

2. Satırlar Üst, Orta, Alt; sütunlar Sol, Sağ. (Üst, Sol) = (2, 1), (Üst, Sağ) = (1, 3), (Orta, Sol) = (3, 0), (Orta, Sağ) = (2, 1), (Alt, Sol) = (1, 2), (Alt, Sağ) = (0, 0). Kesin baskılanan stratejiler yinelemeli ola-

rak silinince ne kalır?

Cevap

Orta, Üst'ü her sütunda yener (3 > 2, 2 > 1): Üst silinir. Orta, Alt'ı da yener (3 > 1, 2 > 0): Alt silinir. Yalnız Orta satırı kalınca sütun oyuncusu Sol'da 0, Sağ'da 1 alır: Sol silinir. Kalan (Orta, Sağ) = (2, 1) oyunun tek Nash dengesidir. Dikkat: başta sütunlar arasında baskılama yoktur (Üst satırında Sağ 3 > 1, ama Alt satırında Sol 2 > 0).

3. Satır oyuncusunun kazançları: Üst (Sol 5, Sağ 0), Orta (2, 2), Alt (0, 5). Orta'yı bir saf strateji baskılar mı? Üst'ü p, Alt'ı $1 - p$ olasılıkla karıştıran strateji hangi p değerlerinde Orta'yı kesin baskılar?

Cevap

Saf strateji baskılamaz: Orta, Sol'da Alt'tan (2 > 0), Sağ'da Üst'ten (2 > 0) iyidir. Karışım Sol'a karşı $5p$, Sağ'a karşı $5(1 - p)$ verir. Kesin baskılama için $5p > 2$ ve $5(1 - p) > 2$ gerekir, yani $2/5 < p < 3/5$. Örneğin ½–½ karışımı her sütunda 2,5 verir.

4. Cournot duopolünde fiyat $P = 18 - (y_1 + y_2)$, maliyet 0'dır. Nash dengesindeki miktarlar, fiyat ve kârlar nedir? Firmalar üretimi eşit bölen bir kartel kursa her biri ne kazanır ve kartelden sapan firma ne kazanır?

Cevap

Kâr $y_1(18 - y_1 - y_2)$; türev $18 - 2y_1 - y_2 = 0 \Rightarrow BR_1(y_2) = (18 - y_2)/2$. Simetrik denge: $2y = 18 - y \Rightarrow y^* = 6$, fiyat 6, kâr 36'şar. Kartel toplam $Y(18 - Y)$'yi $Y = 9$'da en büyük yapar: 4,5'er üretim, fiyat 9, kâr 40,5'er. Rakip 4,5 üretirken en iyi tepki $(18 - 4{,}5)/2 = 6{,}75$'tir ve sapan firma $6{,}75 \cdot 6{,}75 = 45{,}5625$ kazanır. Kartel dengede değildir.

5. Sıfır toplamlı oyunda satır oyuncusunun kazanç matrisi $A = [[5, 1], [2, 4]]$'tür. Saf maksimin ve minimaks nedir? Eyer noktası yoksa iki oyuncunun minimaks stratejileri ve oyunun değeri nedir?

Cevap

Satır en küçükleri 1, 2 ⇒ maksimin 2; sütun en büyükleri 5, 4 ⇒ minimaks 4. $2 < 4$, eyer noktası yoktur. Satır oyuncusu ilk satırı p ile oynarsa sütunlara karşı $5p + 2(1 - p) = 2 + 3p$ ve $p + 4(1 - p) = 4 - 3p$ alır; eşitlik $p = 1/3$, $v = 3$. Sütun oyuncusu ilk sütunu q ile oynarsa $5q + 1(1 - q) = 2q + 4(1 - q) \Rightarrow q = 1/2$ ve yine 3 verir.

6. Tek bekçi iki hedeften birini korur. Hedef H'nin değeri 5, hedef L'nin değeri 3'tür; korunmayan hedefe saldıran değerini kazanır, korunana saldıran 0 alır. Bekçi H'yi hangi olasılıkla korumalıdır ve

saldırgan ne kazanır? Bekçi hep H'yi korusaydı ne olurdu?

Cevap

Saldırganı kayıtsız bırakmak için $5(1 - p) = 3p \Rightarrow p = 5/8$. Saldırganın kazancı $3 \cdot 5/8 = 15/8 \approx 1{,}9$'dur. Bekçi hep H'yi korusaydı saldırgan L'ye saldırıp 3 alırdı.

7. Geyik Avı'nın bir sürümünde birlikte geyik 5'er kazandırır, tavşan her durumda 2 kazandırır, geyiğe yalnız giden 0 alır. Karışık dengede geyik olasılığı nedir? Hangi denge Pareto baskın, hangisi risk baskındır?

Cevap

Ortak Geyik'i q ile seçiyorsa Geyik $5q$, Tavşan 2 verir: $q = 2/5$. (G, G) Pareto baskındır (5 > 2). Sapma kayıpları çarpımı (G, G)'de $(5 - 2)^2 = 9$, (T, T)'de $(2 - 0)^2 = 4$'tür; (G, G) risk baskın da olur. Bu sürümde iki ölçüt aynı dengeyi seçer; ortağın yalnız %40 olasılıkla gelmesi geyiği seçmeye yeter.

8. Kavşaktaki Tavuk oyununda ((Kır, Kır) = (0, 0), (Kır, Düz) = (−1, 1), (Düz, Kır) = (1, −1), (Düz, Düz) = (−10, −10)) bir ışık yarı yarıya (Kır, Düz) ya da (Düz, Kır) önerir. Bu bir korelasyonlu denge midir? Işık bunun yerine yarı yarıya (Kır, Kır) ya da (Düz, Düz) önerseydi ne olurdu?

Cevap

İlk ışık bir korelasyonlu dengedir. “Kır” önerisi alan sürücü rakibe kesin “Düz” denildiğini bilir: uyarsa −1, saparsa −10 alır. “Düz” önerisi alan rakibe kesin “Kır” denildiğini bilir: uyarsa 1, saparsa 0 alır. Beklenen kazanç 0'dır. İkinci ışık korelasyonlu denge değildir: “Düz” önerisi alan sürücü rakibin de düz gideceğini bilir; uyarsa −10, Kır'a saparsa −1 alır ve sapar.

3.11 İleri okuma

- M. J. Osborne, *An Introduction to Game Theory* (2004), Bölüm 2–4: Nash dengesi, karışık strateji ve baskılama konularını çok sayıda çözülmüş örnekle, az matematikle anlatır.
- T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory* (2016), Ders 13, 18 ve 20: denge kavramlarının hiyerarşisini, minimaks teoreminin öğrenmeyle ispatını ve PPAD'i bilgisayar bilimi bakışıyla sunar.
- A. R. Karlin ve Y. Peres, *Game Theory, Alive* (2017): sıfır toplamlı oyunları ve Nash teoreminin sabit nokta ispatını öğrenci düzeyinde ayrıntılı verir.
- C. Daskalakis, P. Goldberg ve C. Papadimitriou, “The complexity of computing a Nash equilibrium” (2009): Nash dengesi bulmanın PPAD-tamlığını gösteren makalenin dergi sürümüdür; girişi teknik ayrıntıya girmeden okunabilir.

3.12 Kaynaklar

1. R. J. Aumann, “Subjectivity and correlation in randomized strategies”, *Journal of Mathematical Economics* 1(1), 67–96, 1974.
2. X. Chen, X. Deng, “Settling the complexity of two-player Nash equilibrium”, *IEEE FOCS*, 2006.
3. V. Conitzer, T. Sandholm, “New complexity results about Nash equilibria”, *Games and Economic Behavior* 63(2), 621–641, 2008.
4. C. Daskalakis, P. Goldberg, C. Papadimitriou, “The complexity of computing a Nash equilibrium”, *SIAM J. Computing* 39(1), 2009 (konferans sürümü *ACM STOC*, 2006).
5. C. Daskalakis, A. Mehta, C. Papadimitriou, “A note on approximate Nash equilibria”, *WINE*, LNCS 4286, 297–306, 2006.
6. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge, 2010 (Bölüm 6: Oyunlar).
7. K. Etessami, M. Yannakakis, “On the complexity of Nash equilibria and other fixed points”, *SIAM J. Computing* 39(6), 2010 (konferans sürümü *IEEE FOCS*, 2007).
8. I. Gilboa, E. Zemel, “Nash and correlated equilibria: Some complexity considerations”, *Games and Economic Behavior* 1(1), 80–93, 1989.
9. J. C. Harsanyi, R. Selten, *A General Theory of Equilibrium Selection in Games*, MIT Press, 1988.
10. A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017.
11. R. J. Lipton, E. Markakis, A. Mehta, “Playing large games using simple strategies”, *ACM EC*, 2003.
12. J. F. Nash, “Equilibrium points in n-person games”, *PNAS* 36(1), 1950.
13. J. F. Nash, “Non-cooperative games”, *Annals of Mathematics* 54(2), 1951.
14. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge, 2007 (Bölüm 1–3).
15. M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2004 (Bölüm 2–4).
16. I. Palacios-Huerta, “Professionals play minimax”, *Review of Economic Studies* 70(2), 395–415, 2003.
17. C. H. Papadimitriou, “On the complexity of the parity argument and other inefficient proofs of existence”, *Journal of Computer and System Sciences* 48(3), 498–532, 1994.
18. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge, 2016 (Ders 13, 18 ve 20).
19. A. Rubinstein, “Inapproximability of Nash equilibrium”, *ACM STOC*, 2015.
20. D. Sally, “Conversation and cooperation in social dilemmas: a meta-analysis of experiments from 1958 to 1992”, *Rationality and Society* 7(1), 1995.
21. T. C. Schelling, *The Strategy of Conflict*, Harvard University Press, 1960.
22. M. Tambe, *Security and Game Theory*, Cambridge University Press, 2011.
23. J. von Neumann, “Zur Theorie der Gesellschaftsspiele”, *Mathematische Annalen* 100, 1928.
24. R. Wilson, “Computing equilibria of n-person games”, *SIAM J. Applied Mathematics* 21, 1971.

KISIM II

Mekanizmalar

Bölüm 4–7

KISIM II · BÖLÜM 4

Mekanizma Tasarımı

Oyun kuramı genellikle kuralları verilmiş bir oyunu inceler ve oyuncuların dengede ne yapacağını tahmin eder. Mekanizma tasarımı soruyu tersine çevirir: istenen sonuç bellidir, bu sonucu doğuracak kurallar aranır. Zorluk, oyuncuların değer, tercih ya da maliyet gibi özel bilgilerini tasarımcıdan saklayabilmesidir. Not, doğrudan mekanizma ve açıklama ilkesiyle başlar; teşvik uyumluluğu, bireysel rasyonellik ve Myerson lemmasıyla sürer. Ardından paranın kullanıldığı en önemli genel mekanizma VCG, para kullanılamayan ortamlar için de Gale–Shapley, medyan, Boston ve TTC mekanizmaları gelir. Son konu eksik bilgili Bayesçi oyunlardır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum04`

ÖĞRENME ÇIKTILARI

Kavramlar
Sosyal seçim fonksiyonu, doğrudan mekanizma, açıklama ilkesi ve Bayes–Nash dengesini açıklamak.

Teşvik analizi
Bir mekanizmanın teşvik uyumlu (IC) ve bireysel rasyonel (IR) olup olmadığını sınamak.

VCG hesabı
Verimli sonucu ve her oyuncunun VCG ödemesini "o olmasaydı" refahıyla hesaplamak.

Parasız tasarım
Gale–Shapley ile kararlı eşleştirme, TTC ile takas bulmak; medyan mekanizmasının neden stratejiye dayanıklı olduğunu göstermek.

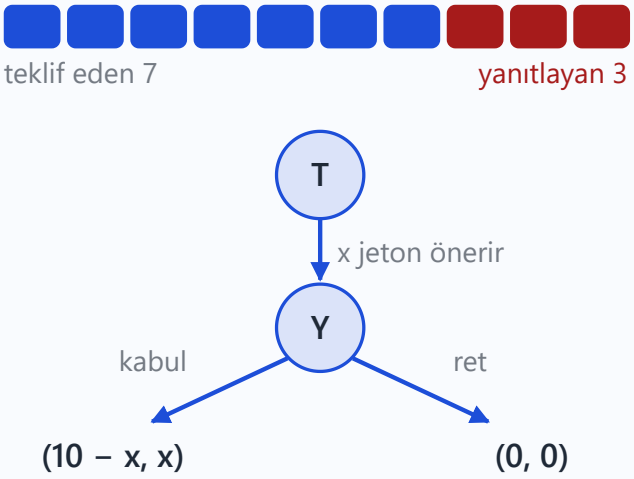
4.1 Temel kavramlar

Bu bölümün sorusu şudur: bir toplulukta herkes kendi çıkarını gözetiyorsa ve herkesin bildiği bir şeyi başkaları bilmiyorsa, topluluk için iyi bir kararı hangi kurallarla verebiliriz? Küçük bir pazarlık oyunu, kuralların davranışı nasıl biçimlendirdiğini gösterir.

4.1.1 Bir oyunla başlangıç: Ültimatom Oyunu

İki oyuncu 10 jetonu bölüşecektir. Birinci oyuncuya **teklif eden**, ikinci oyuncuya **yanıtlayan** denir. Teklif eden, yanıtlayana bir pay önerir: örneğin "sana 3 jeton, bana 7 jeton". Yanıtlayan bu teklifi ya kabul eder ya da reddeder. Kabul ederse bölüşüm uygulanır. Reddederse ikisi de 0 alır. Pazarlık yoktur; teklif tek seferliktir. Oyunun adı da buradan gelir: teklif bir "ültimatom"dur, al ya da bırak.

Oyunun yapısı Şekil 4.1'de görülür. Önce teklif eden (T) bir x sayısı seçer; bu, yanıtlayana önerilen jeton sayısıdır. Sonra yanıtlayan (Y) bu sayıyı görerek karar verir. Kabul edilirse kazançlar $(10 - x, x)$ olur; burada ilk sayı teklif edenin, ikinci sayı yanıtlayanın kazancıdır. Ret durumunda kazançlar $(0, 0)$'dır.



Şekil 4.1: Ültimatom Oyunu. Teklif eden (T) yanıtlayana x jeton önerir; yanıtlayan (Y) kabul ederse $(10 - x, x)$, reddederse $(0, 0)$ gerçekleşir. Üstteki çubuk 7–3 bölüşümünü gösterir.

Oyunda iki soru doğar. Yanıtlayan 1 jetonluk teklifi kabul etmeli midir? Teklif eden kaç jeton önermelidir? İki sorunun cevabı da oyuncunun neyi önemsediğine bağlıdır. Yalnız jeton sayısını önemseyen bir yanıtlayan için 1 jeton, 0 jetondan iyidir; bu yüzden reddetmek mantıksız görünür. Oysa birçok insan 1 jetonluk teklifi hakaret sayar ve reddeder. Bu gerilim, bölümün temel sorusunu öne çıkarır: kurallar oyuncuların davranışını nasıl etkiler ve kural koyucu bu etkiyi hesaba katarak kuralları nasıl seçmelidir?

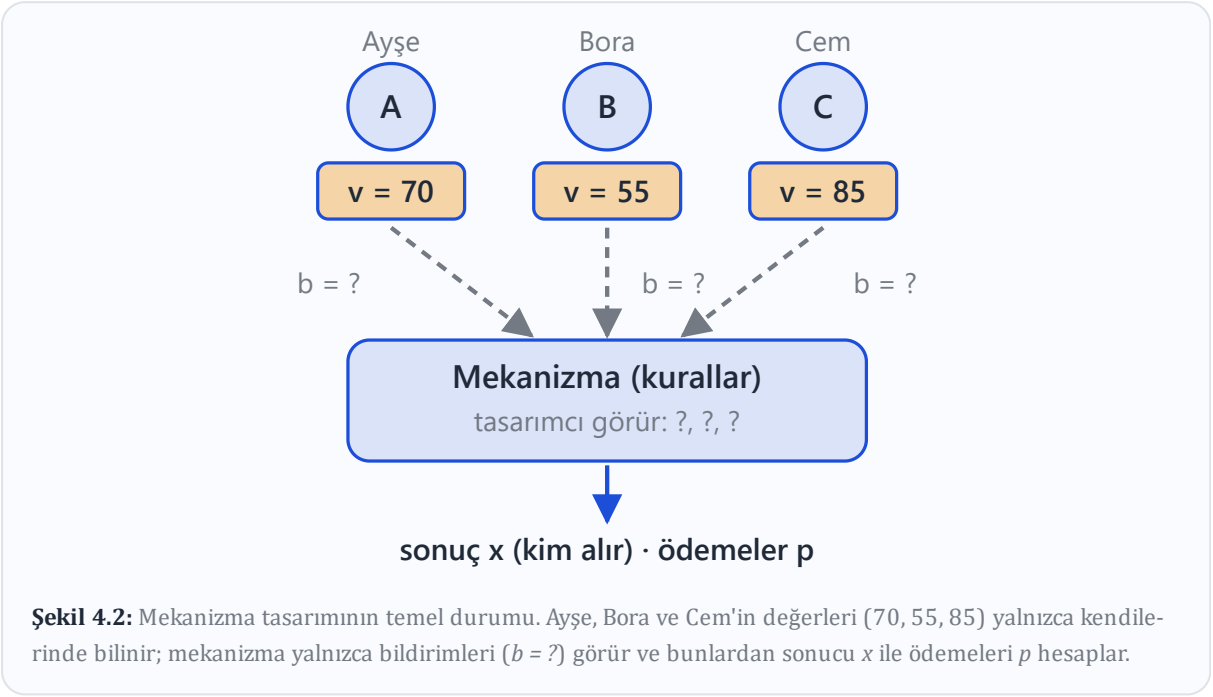
Ültimatom oyunu küçük bir **mekanizmadır**: teklif eden, yanıtlayana "şu bölüşüm ya da hiçbir şey" kuralını sunar. Oyunun kuramsal çözümü 4.2.5'te verilir.

4.1.2 Mekanizma tasarımı nedir?

Oyun kuramının alışılmış yaklaşımında oyun verilir: oyuncular, stratejileri ve kazançları bellidir. Soru, bu oyunda ne olacağıdır. **Mekanizma tasarımı** (*mechanism design*) bu yaklaşımı tersine çevirir. Oyun verilmez; oyunu tasarımcı kurar. Tasarımcının elinde istenen bir sonuç vardır ve bu sonucu doğuracak kuralları arar. Kurallar iki şeyi belirler: kim ne alır (tahsis) ve kim ne öder (ödeme).

Bu iş, ilk bakışta göründüğünden zordur. Tasarımcı oyuncuların özel bilgisini bilmez. Bir nesneyi en çok değer veren kişiye vermek isteyen bir satıcı, kimin ne kadar değer verdiğini bilmez. Bir belediye, sakinlerin bir parka ne kadar değer verdiğini bilmez. Bu bilgiler yalnızca oyuncuların kendisindedir. Tasarımcı bilgiyi oyunculara sormak zorundadır; oyuncular ise çıkarları gerektiriyorsa yanlış söyleyebilir.

Mekanizma tasarımının amacı bu yüzden şudur: oyuncular bencil ve stratejik davranırken, yani kendi çıkarlarını gözetirken bile istenen sonuca ulaşmak. Tasarımcı oyunculara "doğruyu söyleyin" diye emir veremez; yalnızca kuralları seçebilir. Kurallar, doğruyu söylemeyi ya da istenen davranışı oyuncunun kendi çıkarına uygun kılacak biçimde seçilmelidir.



Şekil 4.2: Mekanizma tasarımının temel durumu. Ayşe, Bora ve Cem'in değerleri (70, 55, 85) yalnızca kendilerinde bilinir; mekanizma yalnızca bildirimleri ($b = ?$) görür ve bunlardan sonucu x ile ödemeleri p hesaplar.

Şekil 4.2 bu durumu gösterir. Üç kişi bir nesneye sırasıyla 70, 55 ve 85 değer verir. Bu sayılar oyuncuların kafasındadır; tasarımcı bunları göremez ("tasarımcı görür: ?, ?, ?"). Mekanizma, oyuncuların bildirdiği b sayılarından sonucu ve ödemeleri hesaplar. Bildirimler gerçek değerlere eşit olmak zorunda değildir. İyi bir mekanizmada gerçek değeri bildirmek oyuncunun kendi çıkarınadır.

Alanın önemi bilimsel ödüllerle de tanındı. 2007 Nobel Ekonomi Ödülü, "mekanizma tasarımı kuramının temellerini attıkları için" Leonid Hurwicz, Eric Maskin ve Roger Myerson'a verildi. 2012'de Alvin Roth ve Lloyd Shapley kararlı eşleştirme ve piyasa tasarımı çalışmalarıyla ödüllendirildi.

4.1.3 Analiz ve sentez

Oyun kuramı ile mekanizma tasarımı aynı yolun iki ters yönüdür. Oyun kuramı bir **analiz** aracıdır: kurallar verilir, oyuncuların davranışı (denge stratejileri) bulunur ve bu davranıştan çıkan sonuç tahmin edilir. Mekanizma tasarımı ise bir **sentez** aracıdır: istenen sonuç (hedef) verilir, oyuncuların bu hedefe götürecek biçimde davranacağı kurallar aranır.



Şekil 4.3: Analiz ve sentez. Üst satırda oyun kuramı kurallardan davranışa, davranıştan sonuca ilerler. Alt satırda mekanizma tasarımı hedef $f(\theta)$'dan başlar, dengede ne yapılacağını sorar ve tahsis ile ödemeden oluşan kuralları bulur.

Şekil 4.3 iki yönü yan yana koyar. Üst satır şu soruyu sorar: "Bu kurallar altında oyuncular ne yapar ve kim kazanır?" Alt satır şu soruyu sorar: "Bu sonucu dengede elde etmek için hangi mekanizma gerekir?" Kısaca, oyun kuramında akış kurallar → sonuç; mekanizma tasarımında sonuç → kurallar yönündedir.

Sentez yönünde çalışmanın önemli bir sonucu vardır. Tasarımcı oyuncuları zorlayamaz; hangi stratejiyi oynayacaklarına oyuncular kendileri karar verir. Tasarımcının tek aracı kurallardır. Bu yüzden mekanizma tasarımında her öneri bir denge sorusuyla sınanır: önerilen kurallar altında oyuncular gerçekten istenen biçimde davranır mı? Davranmıyorlarsa kurallar ne kadar iyi niyetli olursa olsun hedefe ulaşılmaz.

4.1.4 Teşvikler

Mekanizma tasarımının dayandığı gözlem basittir: insanlar **teşviklere** (*incentives*) göre davranır. Kurallar belli bir davranışı kârlı yaparsa oyuncular o davranışa yönelir. Bir otopark ücreti yükselirse insanlar daha az araba kullanır. Bir sınavda yanlış cevaplar puan götürürse öğrenciler emin olmadıkları soruları boş bırakır. Kural koyucu her durumda "bu kural hangi davranışı ödüllendiriyor?" diye sormalıdır.

İyi bir mekanizma dürüst davranışı ödüllendirir ve stratejik yalanı caydırır. Doğruyu söylemenin oyuncunun kendi çıkarına olduğu mekanizmalara **teşvik uyumlu** (*incentive compatible*) mekanizma denir. Kavramın biçimsel tanımı 4.2.4'tedir. İki tanıdık örnek, teşvik uyumsuzluğunun neye yol açtığını gösterir.

Beleşçi sorunu. Bir mahallede park yapılacak ve maliyet, sakinlerin parka verdiği değere göre paylaştırılacak olsun. Her sakin parka ne kadar değer verdiğini kendisi söyler. Değerini düşük gösteren daha az öder; park yine de yapılırsa ondan aynı ölçüde yararlanır. Bu durumda herkes değerini düşük

göstermeye eğilimlidir. Başkalarının ödemesinden yararlanmak isteyen kişiye **beleşçi** (*free-rider*) denir. Herkes beleşçi gibi davranırsa toplam bildirim maliyeti karşılamayabilir ve herkesin istediği park hiç yapılmaz.

Stratejik oy. Bir seçimde seçmen en çok desteklediği adayın kazanma şansı yoksa oyunu kazanma şansı olan ikinci tercihine verebilir. Bu, seçmenin gerçek tercihini gizlemesidir. Kural (en çok oy alan kazanır) bu davranışı ödüllendirir. Böylece seçim sonucu, seçmenlerin gerçek tercihlerini yansıtmayabilir.

İki örnekte de aynı zincir işler: teşvik uyumsuzluğu oyuncuları yanlış bilgi vermeye iter; yanlış bilgi de verimsiz ya da adil olmayan sonuçlara yol açar. Mekanizma tasarımı bu zinciri baştan kırmaya çalışır: doğru bilgi vermek oyuncunun kendi çıkarına olmalıdır.

> **Sık yapılan hata**
>
> "Oyuncular dürüst olursa her kural iyi çalışır, o hâlde dürüstlüğü varsayalım." Mekanizma tasarımı tam olarak bu varsayımı yapmaz. Oyuncuların kendi çıkarlarını gözeteceği kabul edilir ve kurallar bu davranışa karşı sağlam olacak biçimde seçilir. Dürüst davranış bir varsayım değil, iyi bir mekanizmanın sonucudur.

4.1.5 Gerçek hayattan uygulamalar

Mekanizma tasarımının sorunları günlük hayatın birçok alanında görülür. Aşağıdaki örneklerin hepsinde bir kurum, bireylerin özel bilgisine dayanarak karar verir.

- **Vergi sistemleri.** Bir kişinin geliri onun özel bilgisidir. Devlet geliri doğrudan göremez; beyana ve denetime dayanır. Vergi oranları ve denetim kuralları dürüst beyanı teşvik etmelidir.
- **Kamu malı finansmanı.** Park, yol ya da kütüphane gibi herkesin yararlandığı mallarda iki soru vardır: proje yapılmalı mı ve kim ne kadar ödemeli? İki soru da sakinlerin gizli değerlerine bağlıdır. Sorunun mekanizmayla çözümü 4.3'teki VCG'dir.
- **İhale ve açık artırmalar.** Kamu alımları ihaleyle yapılır; frekans lisansları ve sanat eserleri açık artırmayla satılır. Satıcı, alıcıların ödemeye razı olduğu en yüksek fiyatı bilmez; kurallar bu bilgiyi açığa çıkarmaya çalışır.
- **Çevrim içi reklam.** Bir arama motorunda her sorguda, milisaniyeler içinde, reklam yerleri için açık artırma yapılır. Reklamverenlerin bir tıklamaya verdiği değer gizlidir.
- **Eşleştirme piyasaları.** Okul yerleştirme, tıp ihtisası ve yurt odası değişimi gibi durumlarda para kullanılmaz; ama kurallar yine de önemlidir. Öğrencilerin tercih listeleri özel bilgidir ve kurallar dürüst liste yazmayı ödüllendirmelidir (4.4).
- **Grup projesi notlama.** Bir grup projesinde öğrenciler katkılarını kendileri bildirirse, notlama kuralı abartmayı caydırmalıdır.

Bu örneklerin ortak yapısı biçimsel modelin temelidir: oyuncular, oyuncuların özel bilgisi (tip), olası sonuçlar ve tasarımcının hedefi. Bu dört öğe 4.2.1'de tanımlanır.

4.2 Doğrudan mekanizmalar

Mekanizma tasarımının biçimsel dili dört kavrama dayanır: oyuncunun özel bilgisi (tip), tasarımcının hedefi (sosyal seçim fonksiyonu), oyuncuların yalnızca tiplerini bildirdiği doğrudan mekanizma ve bu mekanizmalara odaklanmayı haklı çıkaran açıklama ilkesi. Bir mekanizmanın iyi olup olmadığı teşvik uyumluluğu, bireysel rasyonellik, verimlilik ve bütçe dengesi ölçütleriyle sınanır. Tek parametreli ortamlarda teşvik uyumlu mekanizmaları Myerson lemması tam olarak betimler.

4.2.1 Tip ve sosyal seçim fonksiyonu

Bir mekanizmada n oyuncu vardır ve oyuncular $1, 2, \ldots, n$ diye numaralanır. Her oyuncunun yalnızca kendisinin bildiği bir özel bilgisi vardır: bir nesneye verdiği değer, seçenekler arasındaki tercih sırası ya da bir işi yapmanın maliyeti. Bu özel bilgiye oyuncunun **tipi** (*type*) denir.

Tanım 4.1

Oyuncu i'nin **tipi** θ_i, olası tiplerin kümesi Θ_i'den bir elemandır: $\theta_i \in \Theta_i$. Tip özel bilgidir. Bütün oyuncuların tiplerinin listesi $\theta = (\theta_1, \ldots, \theta_n)$'ye **tip profili** denir. Olası sonuçların kümesi O ile gösterilir. Bir **sosyal seçim fonksiyonu** (*social choice function*)

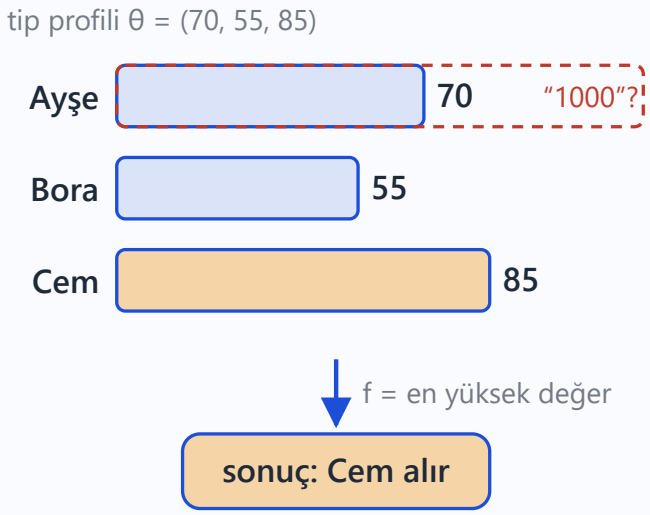
$$f : \Theta_1 \times \ldots \times \Theta_n \to O$$

her tip profiline "istenen" bir sonucu atayan kuraldır.

Tanımdaki semboller ilk okumada yabancı gelebilir. Θ Yunan alfabesinin "teta" harfinin büyüğüdür ve bir kümeyi gösterir; küçük θ o kümeden seçilmiş bir elemandır. $\Theta_1 \times \ldots \times \Theta_n$ ifadesindeki $\times$ işareti kartezyen çarpımdır: her oyuncudan bir tip seçilerek oluşturulabilecek bütün listelerin kümesi. Örneğin iki oyuncunun her biri "düşük" ya da "yüksek" tipte olabiliyorsa dört tip profili vardır: (düşük, düşük), (düşük, yüksek), (yüksek, düşük), (yüksek, yüksek). Sosyal seçim fonksiyonu bu dört profilin her birine bir sonuç atar.

Sonuç kümesi O probleme göre değişir. Tek bir nesne satılıyorsa sonuç "nesneyi kim alır?" sorusunun cevabıdır: $O = \{\text{Ayşe, Bora, Cem, kimse}\}$. Bir belediye birkaç projeden birini seçiyorsa sonuç seçilen projedir: $O = \{\text{Park, Kütüphane, Spor salonu}\}$. Bir okul yerleştirmesinde sonuç, her öğrencinin hangi okula gittiğini söyleyen bir atamadır.

Sosyal seçim fonksiyonu tasarımcının hedefini ifade eder. Tasarımcı "tipleri bilseydim ne yapmak isterdim?" sorusunu cevaplar. En sık kullanılan hedef **verimliliktir**: sonuç, oyuncuların değerlerinin toplamını en büyük yapmalıdır. Tek nesne satışında bu, nesneyi ona en çok değer verene vermek demektir.



Şekil 4.4: Verimli sosyal seçim fonksiyonu. Tip profili $\theta = (70, 55, 85)$'te en yüksek değer Cem'indir, bu yüzden $f(\theta)$ = "Cem alır". Kesikli kutu, Ayşe'nin değerini "1000" diye bildirebileceği durumu hatırlatır.

Örnek 4.1

Ayşe, Bora ve Cem bir nesneye sırasıyla 70, 55 ve 85 değer veriyor. Tip profili $\theta = (70, 55, 85)$'tir. Verimli sosyal seçim fonksiyonu "nesne en yüksek değerli oyuncuya gider" kuralıdır.

1. Değerler karşılaştırılır: $85 > 70 > 55$.
2. En büyük değer Cem'indir; $f(\theta)$ = "Cem alır".
3. Bu sonucun sosyal refahı, yani nesneyi alan kişinin değeri, 85'tir. Nesne Ayşe'ye gitseydi refah 70, Bora'ya gitseydi 55 olurdu. Verimli sonuç bu üçünün en büyüğünü seçer.

Sorun, tasarımcının bu üç sayıyı görememesidir; oyunculara sorması gerekir. Kural yalnızca "en yüksek değeri söyleyen alır" olursa ve hiçbir ödeme yoksa her oyuncu "benim değerim 1000" der. Bu durumda bildirimler hiçbir bilgi taşımaz ve f uygulanamaz.

Örnek, mekanizma tasarımının temel sorusunu açıkça gösterir: f'yi, oyuncular kendi çıkarlarını gözetirken de uygulayan kurallar var mıdır? Cevap bazen evet, bazen hayırdır. Tek nesne satışında cevap evettir: kazananın ödediği fiyatı doğru seçmek yeterlidir (4.2.4). Bazı ortamlarda ise hiçbir kural istenen her şeyi birden sağlayamaz (4.2.7).

4.2.2 Doğrudan mekanizma ve yarı doğrusal fayda

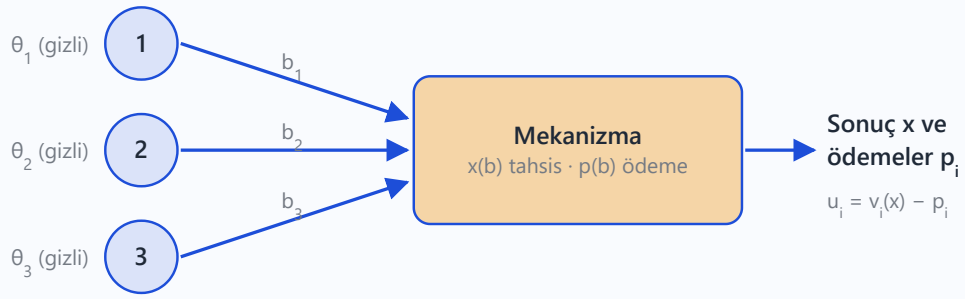
Tasarımcı bilgiyi oyunculardan almalıdır. En dolaysız yol, her oyuncudan tipini istemektir; bu tür mekanizmalara doğrudan mekanizma denir.

Tanım 4.2

Bir **doğrudan mekanizmada** (*direct mechanism*) her oyuncu i bir **bildirim** $b_i \in \Theta_i$ verir. Bildirim gerçek tipe eşit olabilir ya da olmayabilir. Mekanizma bildirim profili $b = (b_1, \ldots, b_n)$'den bir sonuç $x(b)$ ve her oyuncu için bir ödeme $p_i(b)$ hesaplar. Oyuncunun **yarı doğrusal faydası** (*quasi-linear utility*)

$$u_i = v_i(x) - p_i$$

dir. Burada $v_i(x)$, tipi θ_i olan oyuncunun x sonucuna verdiği değerdir. Tek nesnede, nesneyi alırsa $v_i = \theta_i$, almazsa 0'dır.



Şekil 4.5: Doğrudan mekanizma. Üç oyuncunun tipleri $\theta_1, \theta_2, \theta_3$ gizlidir; her oyuncu bir bildirim b_i gönderir. Mekanizma bildirimlerden tahsis $x(b)$ ve ödeme $p(b)$ hesaplar; oyuncunun faydası $u_i = v_i(x) - p_i$'dir.

Şekil 4.5 doğrudan mekanizmayı bir kutu olarak gösterir. Kutunun girdisi bildirimler, çıktısı sonuç ve ödemelerdir. Kutunun içindeki kuralları herkes bilir ve oyuncular ne bildireceklerine buna göre karar verir.

"Yarı doğrusal" ne demektir? Faydanın para kısmı doğrusaldır: 1 TL daha fazla ödemek faydayı tam 1 birim azaltır. Değer kısmı ise $v_i(x)$ herhangi bir fonksiyon olabilir. Bu varsayım, değerin ve paranın aynı birimle ölçülmesini sağlar. Değer 85 bin TL olan bir nesneyi 70 bin TL'ye alan oyuncunun faydası 15 bin TL'dir; nesneyi hiç almayan ve hiçbir şey ödemeyen oyuncunun faydası 0'dır. Yarı doğrusal fayda, paranın kişiden kişiye aktarılabildiği ortamların standart modelidir; bölümdeki paralı mekanizmaların hepsi bu varsayıma dayanır.

Örnek 4.2

Ayşe, Bora ve Cem'in değerleri 70, 55, 85 olsun. Bir doğrudan mekanizma şu kuralı kullansın: en yüksek bildirim kazanır, kazanan kendi bildirimini öder, kaybedenler ödemez.

1. Herkes dürüst bildirirse $b = (70, 55, 85)$'tir. Cem kazanır ve 85 öder. Cem'in faydası $85 - 85 = 0$'dır; Ayşe ve Bora'nın faydası 0'dır.

2. Cem 71 bildirirse yine en yüksek bildirim onundur. Cem 71 öder ve faydası *85 − 71 = 14* olur.
3. Cem yalan söyleyerek 14 birim kazandı. Bu mekanizmada dürüst bildirim Cem'e hiçbir şey kazandırmaz, yalan ise kazandırır.

Kural aynı kalıp yalnız ödeme "kazanan, ikinci en yüksek bildirimi öder" diye değiştirilirse durum farklıdır. Cem 85 de bildirse 71 de bildirse 70 öder; faydası 15'tir. Ödeme Cem'in kendi bildirimine bağlı olmadığı için bildirimini değiştirmesinin bir yararı yoktur. Bu farkın grafiği 4.2.4'tedir.

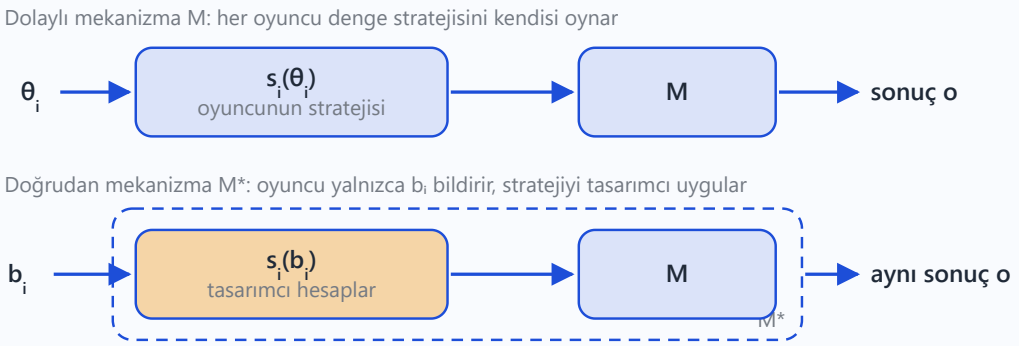
Dolaylı mekanizmalar. Her mekanizma oyunculardan tiplerini istemez. Bir İngiliz açık artırmasında fiyat yavaş yavaş yükselir ve oyuncular el kaldırarak oyunda kalıp kalmadıklarını gösterir. Bir pazarlıkta taraflar sırayla teklif verir. Bu tür mekanizmalara **dolaylı mekanizma** (*indirect mechanism*) denir: oyuncunun stratejisi tipini söylemek değil, başka bir eylemdir (el kaldırmak, teklif vermek, bir sıra yazmak). Dolaylı mekanizmalar uygulamada çok yaygındır, ama kuramsal olarak doğrudan mekanizmalara indirgenebilir. Bu indirgemeyi açıklama ilkesi sağlar.

4.2.3 Açıklama ilkesi

Mekanizma tasarımcısının önünde sonsuz sayıda olası kural vardır: açık artırmalar, sıralı teklifler, karmaşık oylama usulleri... Bunların hepsini incelemek imkânsızdır. **Açıklama ilkesi** (*revelation principle*) bu arama alanını büyük ölçüde daraltır: yalnızca doğrudan ve dürüst bildirimli mekanizmalara bakmak yeterlidir.

Teorem 4.1 (Açıklama ilkesi)

Bir sonuç, dolaylı bir mekanizma *M*'de bir denge olarak uygulanabiliyorsa, uygun koşullarda dürüst bildirimin denge olduğu bir doğrudan mekanizma *M** de aynı sonucu uygulayabilir.



Şekil 4.6: Açıklama ilkesinin fikri. Üstte dolaylı mekanizma *M*'de her oyuncu tipine göre denge stratejisi $s_i(\theta_i)$'yi kendisi oynar. Altta doğrudan mekanizma *M**'de oyuncu yalnızca b_i bildirir; $s_i(b_i)$'yi tasarımcı oyuncu adına hesaplayıp *M*'ye verir. Sonuç aynıdır.

İlkenin arkasındaki düşünce Şekil 4.6'da görülür. M bir dolaylı mekanizma olsun ve oyuncuların denge stratejileri $s_1, \ldots, s_n$ olsun. Her strateji bir kuraldır: tipi θ_i olan oyuncu $s_i(\theta_i)$ eylemini seçer. Tasarımcı şu yeni mekanizmayı kurar: "Bana tipini söyle; ben senin yerine, o tipte senin oynayacağın denge stratejisini oynarım." Bu yeni mekanizma M^* doğrudandır, çünkü oyuncudan yalnızca tipini ister.

İspat. M^*'de dürüst bildirimin bir denge olduğu şöyle gösterilir. Diğer oyuncular dürüst bildirsin. O zaman tasarımcı onlar adına $s_j(\theta_j)$ oynar; bu, onların M'deki denge davranışıdır. Oyuncu i dürüst θ_i bildirirse tasarımcı onun adına $s_i(\theta_i)$ oynar. Oyuncu i başka bir b_i bildirirse tasarımcı onun adına $s_i(b_i)$ oynar; bu, M'de tipi θ_i olan oyuncunun denge stratejisi yerine başka bir eylem seçmesiyle aynıdır. s bir denge olduğuna göre M'de böyle bir sapma kârlı değildir. Demek ki M^*'de yalan söylemek de kârlı değildir. Sonuç olarak dürüst bildirim M^*'de bir dengedir ve dürüst bildirimde M^*, M'nin denge sonucunu üretir. □

Bu basit gözlemin sonucu çok güçlüdür: "Bu hedefi uygulayan bir mekanizma var mı?" sorusu için yalnızca dürüst bildirimin denge olduğu doğrudan mekanizmalara bakmak yeterlidir. Böyle bir doğrudan mekanizma yoksa hiçbir dolaylı mekanizma da yoktur.

Örnek 4.3

İngiliz açık artırmasında fiyat sıfırdan başlayıp yavaş yavaş yükselir. Her teklifçi, fiyat kendi değerine ulaşana kadar oyunda kalır, sonra çekilir. Değerler 70, 55, 85 olsun.

1. Fiyat 55'e ulaşınca Bora çekilir; çünkü daha yüksek fiyatta kazanırsa faydası negatif olur.
2. Fiyat 70'e ulaşınca Ayşe çekilir. Oyunda yalnız Cem kalır.
3. Cem nesneyi yaklaşık 70'e, yani ikinci en yüksek değere alır. Faydası $85 - 70 = 15$'tir.

Şimdi tasarımcı oyunculara şöyle desin: "Değerini bir kâğıda yaz; ben senin yerine fiyat değerine gelene kadar oyunda kalırım." Bu bir doğrudan mekanizmadır. Kâğıtları açınca en yüksek değer kazanır ve ikinci en yüksek değeri öder. Bu, ikinci fiyat (Vickrey) açık artırmasıdır. Açıklama ilkesine göre bu doğrudan mekanizmada dürüst yazmak en iyisidir; çünkü yanlış bir değer yazmak, İngiliz açık artırmasında yanlış anda çekilmekle aynıdır.

İlkenin sınırları da vardır. Açıklama ilkesi "her mekanizmada dürüst bildirim baskın stratejidir" demez. İlke yalnızca, kurulan M^* mekanizmasında dürüst bildirimin *bir denge* olduğunu söyler. Hangi tür denge olduğu ise M'de kullanılan denge kavramına bağlıdır. M'de stratejiler baskın stratejiyse M^*'de dürüstlük baskın stratejidir. M'de stratejiler yalnızca bir Bayes–Nash dengesi oluşturuyorsa (4.5.1) M^*'de dürüstlük de yalnızca bir Bayes–Nash dengesidir. Bu iki durum aşağıda iki ayrı teşvik koşuluyla tanımlanır.

Sık yapılan hata

"Açıklama ilkesine göre doğrudan mekanizmalar daha iyidir, uygulamada da hep onlar kullanılmalıdır." İlke bir varlık ve indirgeme sonucudur; doğrudan mekanizmanın pratikte daha iyi olduğunu söylemez. Dolaylı mekanizmalar daha anlaşılır olabilir, daha az bilgi isteyebilir ya da oyuncuların mahremiyetini koruyabilir. İngiliz açık artırmasında kazanan, kendi değerini hiç açıklamaz; ikinci fiyat açık artırmasında ise yazmak zorundadır.

4.2.4 Teşvik uyumluluğu ve bireysel rasyonellik

Bir doğrudan mekanizmanın iyi işlemesi için iki koşul aranır. Birincisi, doğruyu söylemek oyuncunun kendi çıkarına olmalıdır. İkincisi, oyuncu mekanizmaya katılmak istemelidir. Birinci koşula teşvik uyumluluğu, ikincisine bireysel rasyonellik denir.

Gösterimi kolaylaştırmak için $u_i(\theta_i; b_i, b_{-i})$ yazılır. Bu ifade şunu anlatır: gerçek tipi θ_i olan oyuncu b_i bildirir, diğerleri b_{-i} bildirir; oyuncunun gerçek faydası budur. b_{-i}, "i dışındaki herkesin bildirimleri" demektir. Noktalı virgülden önceki θ_i faydanın hangi gerçek değere göre ölçüldüğünü, sonrası ise bildirimleri gösterir.

Tanım 4.3

Her oyuncu i, her gerçek tip θ_i, her yalan b_i ve diğerlerinin her bildirimi b_{-i} için aşağıdaki eşitsizlik sağlanıyorsa doğrudan mekanizma **baskın stratejide teşvik uyumludur** (*dominant-strategy incentive compatible*, DSIC):

$$u_i(\theta_i; \theta_i, b_{-i}) \geq u_i(\theta_i; b_i, b_{-i})$$

Yani diğerleri ne bildirirse bildirsin, doğruyu söylemek en az yalan söylemek kadar iyidir.

DSIC çok güçlü bir koşuldur. Oyuncunun diğerleri hakkında hiçbir şey bilmesine ya da tahmin etmesine gerek yoktur. Diğerleri dürüst de olsa yalancı da olsa, akıllı da olsa hatalı da olsa, dürüst bildirim oyuncu için en iyisidir; yani baskın stratejidir. Bu yüzden DSIC mekanizmalar oyunculara basit bir öğüt verir: "Ne düşündüğünü söyle, gerisini düşünme."

Tanım 4.4

Tipler herkesçe bilinen bir olasılık dağılımından gelsin. Diğer oyuncular dürüst bildirirken doğruyu söylemek her oyuncunun beklenen faydasını en büyük yapıyorsa doğrudan mekanizma **Bayesçi teşvik uyumludur** (*Bayesian incentive compatible*, BIC). Beklenen fayda, diğerlerinin bilinmeyen tipleri üzerinden alınır.

BIC koşulu DSIC'ten iki yönden zayıftır. Birincisi, yalnızca diğerleri *dürüstken* doğruyu söylemenin en iyi olduğunu ister; diğerleri yalan söylüyorsa bir şey söylemez. İkincisi, faydanın her durumda değil, *ortalamada* (beklenen değer olarak) en büyük olmasını ister. Beklenen değer, her olası durumun faydasının o durumun olasılığıyla çarpılıp toplanmasıdır. Örneğin rakibin değeri yarı yarıya 4 ya da 8 ise ve oyuncu bir bildirimle ilk durumda 6, ikinci durumda 0 alıyorsa beklenen faydası $\tfrac{1}{2}\cdot 6 + \tfrac{1}{2}\cdot 0 = 3$'tür.

Teorem 4.2 (DSIC ⇒ BIC)

Her DSIC mekanizma BIC'tir. Tersi genelde doğru değildir.

İspat. Mekanizma DSIC olsun. Diğerlerinin bildirimi ne olursa olsun, dürüst bildirim oyuncu i'ye yalan söylemekten az fayda vermez. Diğerleri dürüstken bildirimleri onların tipleridir, yani bu da "herhangi bir b_{-i}" durumlarından biridir. Her tip profili için dürüst faydanın yalan faydasından büyük ya da eşit olduğu biliniyor. Bu eşitsizlikler aynı olasılıklarla çarpılıp toplanırsa eşitsizlik korunur. Dolayısıyla beklenen faydada da dürüst bildirim en iyisidir. □

Tersinin neden doğru olmadığını birinci fiyat açık artırması gösterir (4.5.4).

Tanım 4.5

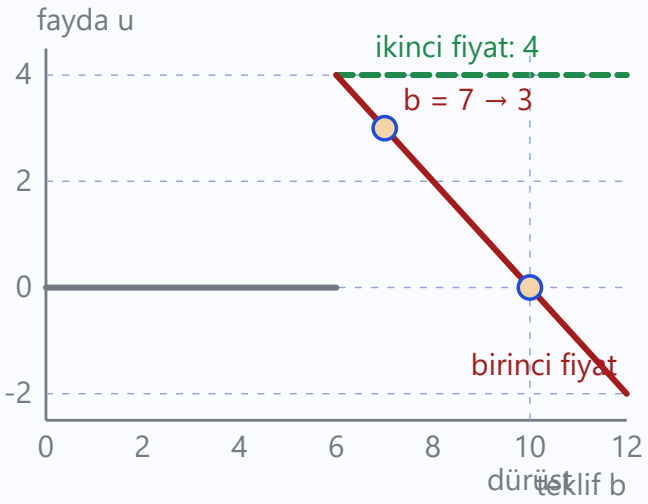
Katılım hiçbir oyuncuya sistemin dışında kalmaktan daha düşük fayda vermiyorsa mekanizma **bireysel rasyoneldir** (*individually rational*, IR). Dışarıda kalmanın faydası 0 kabul edilirse, dürüst bildirimde her oyuncu için

$$u_i(\theta_i;\, \theta_i,\, b_{-i}) \geq 0$$

sağlanmalıdır.

IR koşulu katılımla ilgilidir. Mekanizma insanları katılmaya zorlayamıyorsa katılanlar zarar etmemelidir. Katılmak zarar ettiriyorsa oyuncu dışarıda kalır ve mekanizma onun bilgisini hiç almaz. Örneğin bir açık artırmada kazanan, değerinden fazla ödemek zorunda kalıyorsa IR bozulur.

Teşvik uyumluluğu en kolay sayısal bir örnekle görülür. Bir nesne iki teklifçi arasında satılsın. Teklifçinin değeri 10, rakibin teklifi 6 olsun. İki ödeme kuralı karşılaştırılır. **Birinci fiyat** kuralında kazanan kendi teklifini öder. **İkinci fiyat** kuralında kazanan rakibin teklifini öder. İki kuralda da en yüksek teklif kazanır.



Şekil 4.7: Değer 10, rakip teklif 6. Yatay eksen teklif *b*, dikey eksen fayda *u*'dur. Birinci fiyatta (kırmızı) fayda *10 − b*'dir ve dürüst teklif *b = 10*'da 0'a düşer; 7 teklifi 3 verir. İkinci fiyatta (yeşil kesikli) 6'nın üstündeki her teklif 4 verir.

Örnek 4.4

1. **6'nın altında teklif.** İki kuralda da teklifçi kaybeder; fayda 0'dır.
2. **Birinci fiyat, 6'nın üstünde teklif *b*.** Teklifçi kazanır ve *b* öder. Fayda *10 − b*'dir. Dürüst teklif *b = 10* faydayı *10 − 10 = 0* yapar. *b = 7* teklifi *10 − 7 = 3* verir. Değerinin altında teklif vermek kazandırır; kural IC değildir.
3. **İkinci fiyat, 6'nın üstünde teklif.** Teklifçi kazanır ve rakibin teklifi olan 6'yı öder. Fayda *10 − 6 = 4*'tür. Teklif 7 de olsa 10 da olsa 50 de olsa fayda aynıdır: 4.
4. **İkinci fiyatta daha düşük teklif.** Teklif 6'nın altına inerse teklifçi kaybeder ve 4 yerine 0 alır. Dürüst teklif (10) hiçbir zaman daha kötü değildir.

Grafikteki kırmızı doğru, birinci fiyat kuralında teklifçinin dürüstlükten uzaklaşmak istediğini gösterir: teklifini 6'ya ne kadar yaklaştırırsa o kadar kazanır. Yeşil çizgi ise ikinci fiyatta teklifin kazanıp kazanmamayı belirlediğini, ama ödenen fiyatı değiştirmediğini gösterir.

Örnekteki rakip teklif 6 keyfî bir seçimdir. Rakip teklif *r* ne olursa olsun ikinci fiyat kuralında aynı mantık işler. *r < 10* ise dürüst teklif kazanır ve *10 − r > 0* fayda verir; daha düşük bir teklif ya yine aynı faydayı verir ya da kaybettirir. *r > 10* ise dürüst teklif kaybeder ve 0 verir; *r*'nin üstüne çıkan bir teklif kazandırır ama *10 − r < 0* fayda verir. Her durumda dürüst teklif en iyisidir; ikinci fiyat kuralı DSIC'tir. Ayrıca kazanan hiçbir zaman değerinden fazla ödemez ve kaybeden ödemez; kural IR'dir.

Sık yapılan hata

IC ile IR'yi karıştırmak. IC, oyuncunun *ne bildireceğiyle* ilgilidir: doğruyu söylemek yalandan iyi mi? IR, oyuncunun *katılıp katılmayacağıyla* ilgilidir: katılmak dışarıda kalmaktan iyi mi? Bir mekanizma birini sağlayıp ötekini sağlamayabilir. Örneğin ikinci fiyat açık artırmasına herkesin önce 3 TL giriş ücreti ödediği bir kural eklensin. Ücret teklife bağlı olmadığı için dürüst teklif yine en iyisidir (IC korunur). Ama değeri 5, rakip teklifi 4 olan bir kazananın faydası $5 - 4 - 3 = -2$'dir ve kaybedenlerin faydası −3'tür: IR bozulur.

4.2.5 Ültimatom oyununa dönüş

Teşvik uyumluluğu ve bireysel rasyonellik tanımlandıktan sonra Ültimatom Oyunu yeniden ele alınabilir. Oyunda teklif eden, yanıtlayana bir "al ya da bırak" mekanizması sunar. Yanıtlayanın IR koşulu, kabul ettiğinde aldığının reddettiğinde aldığından az olmamasıdır.

Kuramın cevabı. Yanıtlayan yalnızca jeton sayısını önemsiyorsa, x jetonluk teklifi kabul etmek x, reddetmek 0 verir. $x \geq 0$ olduğu sürece kabul etmek en iyi tepkidir; bu yanıtlayanın IR koşuludur. $x = 0$'da kabul de ret de 0 verdiği için yanıtlayan kayıtsızdır; iki seçenek de en iyi tepkidir.

Teklif eden, yanıtlayanın bu davranışını önceden görür ve teklifini buna göre seçer. Oyun sıralı olduğu için çözüm geriye doğru çıkarımla bulunur: önce yanıtlayanın her teklife ne diyeceği belirlenir, sonra teklif eden bunu bilerek kendi seçimini yapar. Bu yolla bulunan dengelere **alt oyun mükemmel denge** (*subgame perfect equilibrium*) denir. Teklifler tam sayıysa iki alt oyun mükemmel denge vardır.

Örnek 4.5

1. **Yanıtlayan 0'ı da kabul ediyor.** Bu durumda yanıtlayan her teklifi kabul eder. Teklif eden kendi payı $10 - x$'i en büyük yapmak için $x = 0$ önerir. Sonuç (10, 0)'dır.
2. **Yanıtlayan 0'ı reddediyor.** Yanıtlayan kayıtsız olduğu için 0'ı reddetmesi de bir en iyi tepkidir. Bu durumda $x \geq 1$ teklifleri kabul edilir, 0 reddedilir. Teklif eden 0 önerirse 0 alır; 1 önerirse 9 alır. En iyisi $x = 1$'dir ve sonuç (9, 1)'dir.
3. Başka denge yoktur: yanıtlayan 1 ve üstünü her zaman kabul ettiği için teklif eden 1'den fazlasını önermek istemez.

Gerçekte ne olur? Oyunun ilk deneyinden (Güth, Schmittberger ve Schwarze, 1982) bu yana yüzlerce deneyde benzer bir tablo ortaya çıkmıştır. En sık teklif pastanın %40–50'sidir. %20'nin altındaki tekliflerin yaklaşık yarısı reddedilir (Camerer, 2003). Yani insanlar adaletsiz buldukları teklifi kendi kazançları pahasına cezalandırır. Bu davranış, "yalnız jetonu önemseyen" oyuncu modeline uymaz.

Etkileşimli Ültimatom oyununda bilgisayar yanıtlayan üç farklı düzeyde oynar. Her düzeyde yanıtlayanın kabul ettiği en küçük teklif farklıdır. Teklif edenin en iyi tepkisi her seferinde yanıtlayanın kabul ettiği en küçük teklifi önermektir. Tablo 4.1 bu durumları karşılaştırır.

Oyundaki yanıtlayan	Kabul ettiği en küçük teklif	Teklif edenin kazancı
Kuram (yalnızca jeton)	1	9
Seviye 1: hoşgörülü	1	9
Seviye 2: stratejik	3	7
Seviye 3: adalete duyarlı	5	5

Tablo 4.1: Ültimatom oyununda yanıtlayan türleri ve teklif edenin en iyi tepkisi (10 jeton).

Tablonun "Kuram" satırı, yanıtlayanın 0'ı reddettiği (9, 1) dengesine karşılık gelir; yanıtlayan 0'ı da kabul ederse teklif eden 0 önerip 10 alır. Adalete duyarlı yanıtlayana karşı teklif eden 5'ten azını öneremez; önerirse ret gelir ve 0 alır. Teklif edenin kazancı, yanıtlayanın kabul eşiği yükseldikçe düşer.

Mekanizma tasarımı için çıkan ders şudur: teşvik uyumlu bir mekanizma tasarlarken oyuncunun fayda fonksiyonu doğru seçilmelidir. Oyuncu adaleti de önemsiyorsa IR koşulu "$x \geq 0$" değil, "x, oyuncunun adil bulduğu paya ulaşıyor" olur. Yanlış bir fayda modeliyle kurulan mekanizma kâğıt üstünde IR görünür ama gerçekte reddedilir. Davranış ve etik tartışması (4.6.1) bu konuya geri döner.

4.2.6 Orantılı maliyet paylaşımı adım adım

Beleşçi sorunu (4.1.4) sayılarla da incelenebilir. Ayşe, Bora ve Cem adlı üç sakin ortak bir park istiyor. Parkın maliyeti 60 bin TL'dir. Sakinlerin parka verdiği gerçek değerler sırasıyla 30, 25 ve 20 bin TL'dir. Değerlerin toplamı 75'tir. 75 > 60 olduğu için parkın yapılması verimlidir: park, maliyetinden fazla değer yaratır.

Belediye şu kuralı uygular. Her sakin parka verdiği değeri bildirir. Bildirimlerin toplamı 60 ya da daha fazlaysa park yapılır ve maliyet bildirimlerle orantılı paylaşılır: sakin i'nin ödemesi $60 \cdot b_i / (b_1 + b_2 + b_3)$'tür. Toplam 60'tan azsa park yapılmaz ve kimse ödemez. Kural adil görünür: parka çok değer veren çok öder. Soru, kuralın teşvik uyumlu olup olmadığıdır.

Durum	Bildirimler (A, B, C)	Toplam	Park	Ödemeler	Faydalar $u = v - ödeme$
Herkes dürüst	30, 25, 20	75	Evet	24, 20, 16	6, 5, 4
Cem 5 der	30, 25, 5	60	Evet	30, 25, 5	0, 0, 15
Herkes yarısını der	15, 12,5, 10	37,5	Hayır	0, 0, 0	0, 0, 0

Tablo 4.2: Orantılı maliyet paylaşımı. Park 60 bin TL; değerler, ödemeler ve faydalar bin TL'dir.

Örnek 4.6

1. **Herkes dürüst.** Bildirimler 30, 25, 20; toplam $75 \geq 60$, park yapılır. Ödemeler $60 \cdot 30/75 = 24$, $60 \cdot 25/75 = 20$ ve $60 \cdot 20/75 = 16$'dır; toplamları 60'tır. Faydalar $30 - 24 = 6$, $25 - 20 = 5$, $20 - 16 = 4$'tür. Herkesin faydası en az 0 olduğu için IR sağlanır. Toplam fayda 15'tir; bu, değerlerin toplamı 75 ile maliyet 60 arasındaki farka eşittir.

2. **Cem yalan söyler.** Ayşe ve Bora dürüst kalırken Cem değerini 5 gösterir. Toplam $30 + 25 + 5 = 60$ olur; park yine yapılır. Ödemeler bildirimlere eşit çıkar: $60 \cdot 30/60 = 30$, 25 ve 5. Cem 5 öder ama parktan yine 20 değerinde yararlanır. Faydası $20 - 5 = 15$'tir. Dürüstken 4'tü; yalan Cem'e 11 birim kazandırdı. Kural IC değildir.
3. **Cem'in en kârlı yalanı.** Cem'in bildirimi c olsun. Park yapılması için $55 + c \geq 60$, yani $c \geq 5$ gerekir. Park yapıldığında Cem'in ödemesi $60c/(55 + c)$'dir ve c büyüdükçe artar. Cem parkın yapılmasını isteyip en az ödemek istediği için $c = 5$ seçer: toplam tam 60 olur.
4. **Herkes aynı mantıkla davranır.** Her sakin, diğerlerinin bildirimine güvenerek kendi bildirimini düşürmek ister. Herkes değerinin yarısını bildirirse toplam $15 + 12{,}5 + 10 = 37{,}5 < 60$ olur. Park yapılmaz; toplam fayda 15'ten 0'a düşer.

Beleşçilik bu yüzden yalnızca bir adalet sorunu değil, bir verimlilik sorunudur. Herkesin istediği ve toplam değeri maliyetinden yüksek olan park, kuralın yarattığı teşvikler yüzünden yapılmayabilir.

Sorunun kaynağı açıktır: bir sakinin bildirimi, sonuç değişmese bile kendi ödemesini doğrudan etkiler. Cem 20 yerine 5 dediğinde park yine yapılır; yani Cem'in yalanı sonucu değiştirmez, yalnızca kendi ödemesini düşürür. Teşvik uyumlu bir kural için çözüm fikri şudur: oyuncunun bildirimi ödemesini *yalnızca seçilen sonucu değiştirerek* etkileyebilmelidir. Sonuç aynı kalıyorsa ödeme de aynı kalmalıdır. Bu fikir VCG mekanizmasının (4.3) temelidir.

Sık yapılan hata

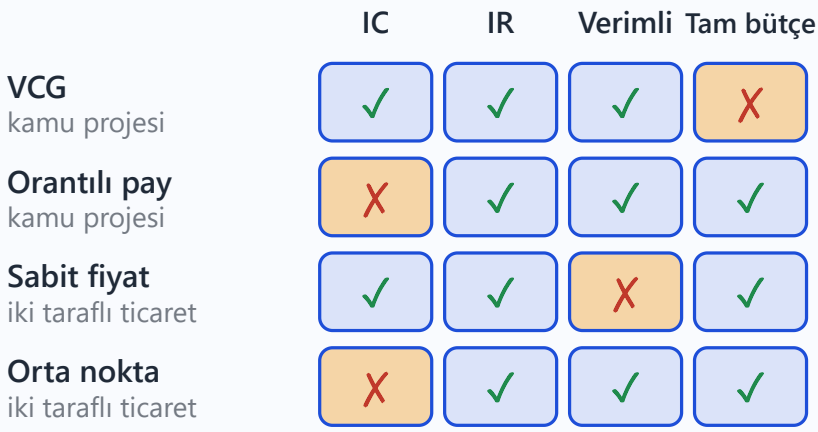
"Dürüst bildirimde herkes kâr ediyor, o hâlde kural iyidir." Dürüst bildirimde IR'nin sağlanması kuralın IC olduğunu göstermez. IC'yi sınamak için her oyuncunun, diğerleri sabitken, başka bir bildirimle daha fazla fayda alıp alamayacağına bakılmalıdır. Orantılı paylaşımda Cem tek başına sapınca 4 yerine 15 alır.

4.2.7 Tasarım ölçütleri ve ödünleşimler

Bir mekanizmayı değerlendirmek için dört temel ölçüt kullanılır. İlk ikisi teşviklerle, son ikisi sonucun niteliğiyle ilgilidir.

- **Teşvik uyumluluğu (IC):** gerçek bilgiyi açıklamak en iyi seçenektir. Baskın strateji düzeyinde (DSIC) ya da Bayesçi düzeyde (BIC) istenebilir.
- **Bireysel rasyonellik (IR):** katılmak, dışarıda kalmaktan kötü değildir.
- **Verimlilik** (*efficiency*): dürüst bildirimde seçilen sonuç sosyal refahı en yüksek sonuçtur. Bir x sonucunun **sosyal refahı** $SW(x) = \Sigma_i v_i(x)$, yani oyuncuların o sonuca verdiği değerlerin toplamıdır.
- **Bütçe dengesi:** ödemelerin toplamıyla ilgilidir. **Zayıf bütçe dengesi** ödemelerin toplamının en az 0 olmasıdır: mekanizma dışarıdan para gerektirmez. **Tam bütçe dengesi** ödemelerin toplamının tam 0 olmasıdır: ne dışarıdan para gerekir ne de para artar.

Σ sembolü "topla" demektir. $\Sigma_i\, v_i(x)$ ifadesi, i'nin bütün oyuncular üzerinde gezindiği ve her birinin $v_i(x)$ değerinin toplandığı anlamına gelir. Üç oyuncu için $v_1(x) + v_2(x) + v_3(x)$ yazmakla aynıdır.



Şekil 4.8: Dört örnek mekanizma ve dört ölçüt. Her biri en az bir ölçütten vazgeçer: VCG tam bütçe dengesinden, orantılı pay ve orta nokta kuralı teşvik uyumluluğundan, sabit fiyat verimlilikten.

Şekil 4.8 dört mekanizmayı bu ölçütlerle karşılaştırır. VCG ve orantılı pay kamu projesi örneğinden, sabit fiyat ve orta nokta ise **iki taraflı ticaretten** (*bilateral trade*) alınmıştır. İki taraflı ticarette bir satıcı ve bir alıcı vardır. Satıcının malı elinde tutmanın kendisine maliyeti c, alıcının mala verdiği değer v'dir. İkisi de özel bilgidir. Ticaret yalnızca $v \geq c$ olduğunda verimlidir: mal, ona daha çok değer verenin eline geçmelidir.

Örnek 4.7

Sabit fiyat. Aracı fiyatı önceden 50 olarak ilan eder. Alıcı "alırım" ya da "almam", satıcı "satarım" ya da "satmam" der; ikisi de evet derse ticaret 50'den gerçekleşir.

1. Alıcı için en iyi cevap, $v \geq 50$ ise "alırım"dır. Satıcı için en iyi cevap, $c \leq 50$ ise "satarım"dır. Bu cevaplar diğer tarafın ne dediğine bağlı değildir: kural DSIC'tir.
2. Ticaret gerçekleştiğinde alıcı $v - 50 \geq 0$, satıcı $50 - c \geq 0$ kazanır: IR sağlanır. Alıcının ödediği satıcıya gider: tam bütçe dengesi vardır.
3. Ama $v = 45$, $c = 40$ olsun. Ticaret verimlidir ($45 > 40$), ama alıcı 50'yi ödemeye razı olmadığı için gerçekleşmez. Kural verimli değildir.

Orta nokta kuralı. Alıcı bir değer, satıcı bir maliyet bildirir. Alıcının bildirimi satıcınınkinden büyük ya da eşitse ticaret iki bildirimin ortasındaki fiyattan yapılır.

1. $v = 60$, $c = 40$ olsun ve ikisi de dürüst bildirsin. Fiyat $(60 + 40)/2 = 50$'dir. Alıcı 10, satıcı 10 kazanır. Dürüst bildirimde ticaret tam olarak verimli olduğu durumlarda gerçekleşir ve kimse zarar etmez.

2. Alıcı değerini 52 bildirirse ticaret yine gerçekleşir, ama fiyat $(52 + 40)/2 = 46$ olur. Alıcının faydası $60 - 46 = 14 > 10$'dur. Değerini küçültmek kazandırdı: kural IC değildir.

Örneklerdeki ödünleşim tesadüf değildir. Myerson ve Satterthwaite (1983) bunun genel bir sonuç olduğunu gösterdi.

Teorem 4.3 (Myerson–Satterthwaite)

Alıcının değeri ile satıcının maliyetinin örtüşen aralıklardan geldiği iki taraflı ticarette verimlilik, (Bayesçi) teşvik uyumluluğu, bireysel rasyonellik ve bütçe dengesini birlikte sağlayan hiçbir mekanizma yoktur.

Teoremin ispatı bu notun kapsamı dışındadır. Önemli olan sonucun anlamıdır: dört ölçütün hepsini birden isteyen bir tasarımcı imkânsızı ister. Mekanizma tasarımı bu yüzden çoğu zaman bir ödünleşim (*trade-off*) problemidir. Tasarımcı hangi ölçütten ne kadar vazgeçeceğine karar verir. Örneğin VCG verimliliği ve DSIC'i korur, karşılığında bütçe fazlası verir; sabit fiyat bütçeyi ve teşvikleri korur, karşılığında bazı verimli ticaretleri kaçırır.

Sık yapılan hata

Myerson–Satterthwaite teoremini "ticaret imkânsızdır" diye okumak. Teorem ticaretin gerçekleşemeyeceğini söylemez; yalnızca dört özelliğin *hepsini birden* sağlayan kuralın olmadığını söyler. Sabit fiyat kuralı her gün kullanılır ve birçok verimli ticareti mümkün kılar; yalnızca bazılarını kaçırır.

4.2.8 Uygulama kuramı: dürüstlük tek denge mi?

Açıklama ilkesi, dürüst bildirimin bir denge olduğunu söyler. Ama aynı mekanizmada istenmeyen başka dengeler de bulunabilir. Kurallar iyi bir dengeyi mümkün kılar, oyuncuların o dengede buluşacağını ise garanti etmez.

Bir mekanizmanın *bütün* dengelerinin istenen sonuca götürmesi daha güçlü bir istektir. Bu tür mekanizmaları inceleyen alana **uygulama kuramı** (*implementation theory*) denir. Alanın temel sonuçlarından biri, Nash dengesiyle uygulanabilen sosyal seçim fonksiyonlarını karakterize eden Maskin'in çalışmasıdır (Maskin, 1999). Sorunun neden önemli olduğunu bir örnek gösterir.

Örnek 4.8

İkinci fiyat (Vickrey) açık artırmasında Ayşe'nin değeri 10, Bora'nın değeri 5 olsun. Ayşe 0, Bora 100 teklif eder.

1. **Sonuç.** En yüksek teklif Bora'nındır; Bora kazanır ve ikinci en yüksek teklifi, yani 0'ı öder. Bora'nın faydası $5 - 0 = 5$, Ayşe'nin faydası 0'dır.
2. **Ayşe sapabilir mi?** Ayşe 100'ün altındaki her teklifle kaybeder ve 0 alır. Kazanmak için 100'ü geçmesi gerekir; o zaman Bora'nın teklifi olan 100'ü öder ve faydası $10 - 100 = -90$ olur. Ayşe'nin sapması kârlı değildir.
3. **Bora sapabilir mi?** Bora şu anda bedavaya kazanıyor ve 5 alıyor. Hiçbir teklif ona 5'ten fazlasını veremez, çünkü nesneye verdiği değer 5'tir ve ödeme 0'ın altına inemez.
4. **Sonuç.** Kimse tek başına sapmak istemez: (0, 100) teklifleri bir Nash dengesidir. Ama nesne, değeri düşük olan Bora'ya gider; sonuç verimli değildir.

Bu kötü denge, mekanizmanın DSIC olmasıyla çelişmez. Dürüst teklif hâlâ her iki oyuncu için baskın stratejidir. Kötü denge, **zayıf baskılanan** tekliflerle kurulmuştur: Bora'nın 100 teklifi, dürüst teklif 5'e göre hiçbir durumda daha iyi değildir ve bazı durumlarda (örneğin Ayşe 50 teklif ederse) daha kötüdür. Ayşe'nin 0 teklifi de dürüst teklif 10'a göre zayıf baskılanır.

Çıkan ders iki yönlüdür. DSIC mekanizmalarda oyuncunun rakibini tahmin etmesi gerekmez; dürüstlük her durumda güvenli bir seçimdir. Bu yüzden DSIC tasarımlar, yalnızca Nash dengesine dayanan tasarımlardan daha sağlamdır. Yine de DSIC tek denge garantisi vermez. Oyuncular baskın stratejilerini oynamazsa garip dengeler ortaya çıkabilir.

4.2.9 Tek parametreli ortamlar: Myerson lemması

Birçok önemli problemde her oyuncunun özel bilgisi tek bir sayıdır. Tek nesne satışında bu sayı nesneye verilen değerdir. Birden çok özdeş nesnenin satıldığı ve her oyuncunun en fazla bir nesne istediği durumda yine tek bir sayı yeterlidir. Bu tür ortamlara **tek parametreli ortam** (*single-parameter environment*) denir.

Tek parametreli ortamda her oyuncunun bir **birim değeri** v_i vardır. Mekanizma bildirimler b'den her oyuncu için bir **tahsis** $x_i(b)$ ve bir ödeme $p_i(b)$ seçer. Tahsis, oyuncunun ne kadar "mal" aldığını gösterir: tek nesnede 0 ya da 1'dir; nesne kurayla veriliyorsa 0 ile 1 arasında bir olasılık olabilir. Oyuncunun faydası $v_i \cdot x_i - p_i$'dir.

Myerson (1981), tek parametreli ortamlarda DSIC mekanizmaları tam olarak betimleyen bir sonuç kanıtladı. Sonuç iki soruya cevap verir: hangi tahsis kuralları DSIC bir mekanizmada kullanılabilir ve bu durumda ödemeler ne olmalıdır?

Tanım 4.6

Her oyuncu i ve diğerlerinin her bildirimi b_{-i} için $x_i(z, b_{-i})$, z arttıkça azalmıyorsa tahsis kuralı x **monotondur**. Yani bir oyuncu teklifini artırınca daha az mal almamalıdır.

Teorem 4.4 (Myerson lemması)

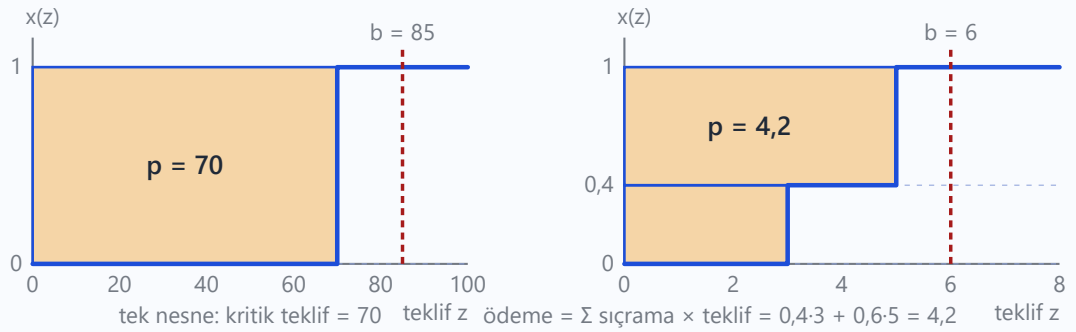
Tek parametreli bir ortamda:

1. Bir tahsis kuralı x, bir DSIC mekanizmada kullanılabilir ancak ve ancak monotonsa.
2. x monotonsa, $p_i(0, b_{-i}) = 0$ koşuluyla mekanizmayı DSIC yapan ödeme kuralı tektir:

$$p_i(b) = b_i \cdot x_i(b) - \int_0^{b_i} x_i(z, b_{-i})\, dz$$

Formüldeki integral işareti ($\int$) korkutucu görünebilir, ama burada yalnızca bir alanı ifade eder. $\int_0^{b_i} x_i(z, b_{-i})\, dz$, yatay eksende 0'dan b_i'ye kadar, x_i eğrisinin altında kalan alandır. Bu bölümdeki bütün örneklerde eğri basamak biçimindedir. Basamaklı bir eğrinin altındaki alan, dikdörtgenlerin alanları toplanarak bulunur: her dikdörtgen için genişlik × yükseklik.

Formülün geometrik okuması şöyledir. $b_i \cdot x_i(b)$, genişliği b_i, yüksekliği $x_i(b)$ olan dikdörtgenin alanıdır. Bu dikdörtgenden eğrinin altındaki alan çıkarılınca, eğrinin solunda, 0 ile $x(b)$ yükseklikleri arasında kalan alan elde edilir. Ödeme bu boyalı alandır. Eğrinin altındaki alan ise oyuncunun faydasıdır: dürüst bildirimde $v_i\, x_i - p_i = \int_0^{v_i} x_i(z)\, dz$.



Şekil 4.9: Myerson ödemesi boyalı alandır. Solda tek nesne: tahsis teklif 70'te 0'dan 1'e sıçrar; $b = 85$ teklifinde ödeme $p = 70$, yani kritik tekliftir. Sağda basamaklı tahsis: teklif 3'te 0,4'e, 5'te 1'e sıçrar; $b = 6$'da ödeme $0{,}4 \cdot 3 + 0{,}6 \cdot 5 = 4{,}2$'dir.

Örnek 4.9

Tek nesne (Şekil solu). Diğerlerinin en yüksek teklifi 70 olsun. Oyuncu 70'in üstünde teklif verirse nesneyi alır ($x = 1$), altında verirse almaz ($x = 0$). Oyuncunun teklifi $b = 85$'tir.

1. $b \cdot x(b) = 85 \cdot 1 = 85$.
2. Eğrinin altındaki alan: 0 ile 70 arasında yükseklik 0, alan 0; 70 ile 85 arasında yükseklik 1, alan $15 \cdot 1 = 15$. Toplam 15.
3. Ödeme $85 - 15 = 70$'tir. Fayda $85 - 70 = 15$, yani eğrinin altındaki alandır.

Basamaklı tahsis (Şekil sağı). Tahsis teklif 3'ün altında 0, 3 ile 5 arasında 0,4 ve 5'ten itibaren 1 olsun. Teklif $b = 6$'dır.

1. $b \cdot x(b) = 6 \cdot 1 = 6$.
2. Eğrinin altındaki alan: [0, 3) aralığında 0; [3, 5) aralığında $2 \cdot 0{,}4 = 0{,}8$; [5, 6] aralığında $1 \cdot 1 = 1$. Toplam 1,8.
3. Ödeme $6 - 1{,}8 = 4{,}2$'dir.
4. Aynı sonuç daha kısa bir yoldan da bulunur. Basamaklı tahsiste ödeme, her sıçramanın büyüklüğü ile sıçramanın olduğu teklifin çarpımlarının toplamıdır: $0{,}4 \cdot 3 + 0{,}6 \cdot 5 = 1{,}2 + 3 = 4{,}2$.

Basamak formülünün nedeni şudur: tahsisteki her sıçrama, oyuncunun o eşiği geçerek fazladan bir "mal parçası" almasıdır. Oyuncu bu parçanın her birimi için tam eşik kadar öder. 3'ü geçerek 0,4 birim mal alır ve her birim için 3 öder; 5'i geçerek 0,6 birim daha alır ve her birim için 5 öder.

Kritik teklif. Tahsis yalnızca 0 ya da 1 olabiliyorsa tek bir sıçrama vardır. Bu durumda ödeme, oyuncunun kazanması için gereken en düşük tekliftir. Bu teklife **kritik teklif** (*critical bid*) denir. Tek nesne satışında kritik teklif, diğerlerinin en yüksek teklifidir. Myerson lemması bu yüzden şunu söyler: tek nesnede nesneyi en yüksek teklifi verene veren ve DSIC olan tek ödeme kuralı, kazananın ikinci en yüksek teklifi ödediği Vickrey kuralıdır (kaybedenler 0 öder).

Ödeme kritik teklif olmazsa ne olur? Eşik 50, kazananın ödemesi ise sabit 20 olsun. Değeri 35 olan bir oyuncu dürüst teklif verirse eşiğin altında kalır ve 0 alır. 50'yi aşan bir teklif verirse nesneyi alır ve 20 öder; faydası $35 - 20 = 15$'tir. Değerini abartmak kazandırdı. Genel olarak, değeri ödeme ile eşik arasında kalan (burada 20 ile 50 arasında) her oyuncu abartmak ister; IC bozulur. Ödeme eşikten büyük olsaydı bu kez değeri eşik ile ödeme arasında kalan kazananlar zarar eder ve tekliflerini eşiğin altına indirmek isterdi.

Monotonluk neden gereklidir? Kısa bir akıl yürütme bunu gösterir. Değeri y olan bir oyuncunun dürüst bildirimi, z bildirmekten kötü olmamalıdır; değeri z olan bir oyuncunun dürüst bildirimi de y bildirmekten kötü olmamalıdır:

$$y \cdot x(y) - p(y) \geq y \cdot x(z) - p(z) \text{ ve } z \cdot x(z) - p(z) \geq z \cdot x(y) - p(y)$$

İki eşitsizlik taraf tarafa toplanınca ödemeler sadeleşir ve $y \cdot x(y) + z \cdot x(z) \geq y \cdot x(z) + z \cdot x(y)$ kalır. Bu, $(z - y)(x(z) - x(y)) \geq 0$ biçiminde yazılabilir. $z > y$ ise $x(z) \geq x(y)$ olmalıdır: daha yüksek teklif daha az tahsis veremez. Yani DSIC, monotonluğu zorunlu kılar. Lemmanın öteki yönü, yani her monoton tahsisin formüldeki ödemeyle DSIC olduğu, Roughgarden'ın (2016) ders notlarında adım adım ispatlanır.

Sık yapılan hata

Myerson ödemesini eğrinin *altındaki* alan sanmak. Eğrinin altındaki alan oyuncunun faydasıdır. Ödeme, $b \cdot x(b)$ dikdörtgeninden bu alan çıkarılınca kalan, eğrinin solundaki alandır. Tek nesne örneğinde eğrinin altındaki alan 15 (fayda), soldaki alan 70'tir (ödeme).

4.3 VCG mekanizması

Orantılı maliyet paylaşımının başarısızlığı (4.2.6) bir çözüm fikri bırakmıştı: oyuncunun bildirimi ödemesini yalnızca seçilen sonucu değiştirerek etkilemelidir. Vickrey–Clarke–Groves (VCG) mekanizması bu fikri genel bir kurala dönüştürür. Mekanizma, William Vickrey'nin (1961) tek nesne açık artırmasını, Edward Clarke'ın (1971) kamu malı çalışmasını ve Theodore Groves'un (1973) genel çerçevesini birleştirir.

4.3.1 Verimli seçim ve Clarke pivot ödemesi

VCG iki adımda çalışır. Birinci adımda sonuç seçilir, ikinci adımda ödemeler hesaplanır. Sonuç seçimi verimlidir: bildirilen değerlere göre sosyal refahı en yüksek sonuç seçilir. Ödeme ise her oyuncunun varlığının diğerlerine verdiği zarara eşittir.

Tanım 4.7

VCG mekanizması (*Vickrey–Clarke–Groves mechanism*) her oyuncudan her olası sonuca verdiği değeri ister; $b_i(x)$, oyuncu i'nin x sonucu için bildirdiği değerdir. Mekanizma

$$x^* = \operatorname{argmax}_x \Sigma_i \, b_i(x)$$

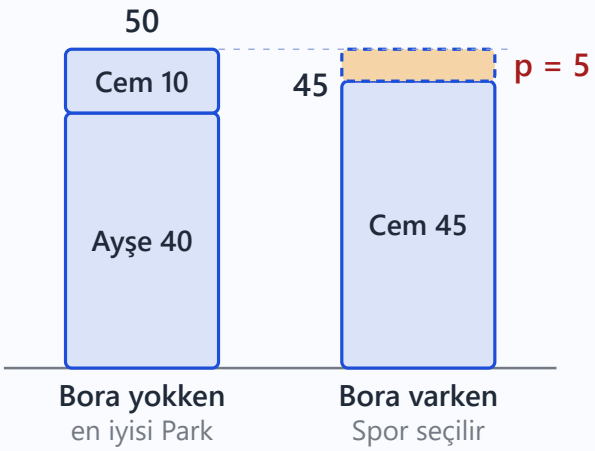
sonucunu seçer ve her oyuncuya **Clarke pivot kuralıyla** (*Clarke pivot rule*) ödeme yaptırır:

$$p_i = \max_x \Sigma_{j \neq i} \, b_j(x) - \Sigma_{j \neq i} \, b_j(x^*)$$

İki formülü sözle okumak yararlıdır. *argmax*, en büyük değeri veren *seçeneği* döndürür; *max* ise en büyük *değeri* döndürür. x^*, bütün oyuncuların bildirdiği değerlerin toplamını en büyük yapan sonuçtur. Ödeme formülünde $\Sigma_{j \neq i}$, "i dışındaki bütün oyuncular üzerinden topla" demektir. Formülün iki parçası şunlardır:

- **Birinci terim:** i hiç olmasaydı, diğer oyuncular kendi aralarında en iyi sonucu seçer ve bu sonuçtan şu kadar refah alırlardı. Kısaca, "i olmasaydı diğerlerinin en iyi refahı".
- **İkinci terim:** i varken seçilen x^* sonucunda diğerlerinin aldığı refah. Kısaca, "i varken diğerlerinin refahı".

Fark, *i*'nin katılımının diğerleri üzerinde yarattığı **dışsallıktır** (*externality*). Dışsallık, bir kişinin kararının başkalarına verdiği ama kendisinin ödemediği zarardır (ya da sağladığı yarardır). Bir fabrikanın havayı kirletmesi olumsuz bir dışsallıktır. VCG, her oyuncuya kendi yarattığı olumsuz dışsallığı ödetir. Oyuncu sonucu kendi lehine değiştirmiyorsa birinci ve ikinci terim eşittir ve oyuncu hiçbir şey ödemez. Sonucu değiştiren, yani "pivot" olan oyuncu, değişikliğin diğerlerine maliyetini öder. Kuralın adı buradan gelir.



Şekil 4.10: Kamu projesi örneğinde Bora'nın dışsallığı. Bora yokken diğerleri için en iyisi Park'tır ve refahları *40 + 10 = 50*'dir. Bora varken Spor salonu seçilir ve diğerlerinin refahı *0 + 45 = 45*'tir. Aradaki fark *p = 5*, Bora'nın ödemesidir.

Şekil 4.10 bu fikri bir sütun çiftiyle gösterir. Soldaki sütun Bora yokken diğerlerinin alacağı refahı, sağdaki sütun Bora varken aldıkları refahı gösterir. Bora'nın varlığı seçimi Park'tan Spor salonuna kaydırır ve diğerlerinin refahını 50'den 45'e düşürür. Bora bu 5 birimlik zararı öder. Bu örneğin bütün sayıları 4.3.2'dedir.

VCG neden DSIC'tir? Ödeme formülünün birinci terimi yalnızca diğerlerinin bildirimlerine bağlıdır; *i* ne bildirirse bildirsin değişmez. Bu terime h_i diyelim. Diğerleri b_{-i} bildirsin ve *i*'nin bildirimi sonucu x yapsın. *i*'nin gerçek faydası şöyledir:

$$u_i = v_i(x) - p_i = v_i(x) + \Sigma_{j \neq i}\, b_j(x) - h_i$$

Teorem 4.5 (VCG'nin teşvik uyumluluğu)

VCG mekanizmasında dürüst bildirim her oyuncu için baskın stratejidir. Değerler negatif değilse Clarke pivot ödemeleri negatif değildir ve mekanizma bireysel rasyoneldir.

İspat. **DSIC.** Yukarıdaki faydanın son terimi h_i, i'nin bildirimine bağlı değildir. Bu yüzden i, faydasını en büyük yapmak için yalnızca $v_i(x) + \Sigma_{j\neq i}\, b_j(x)$ toplamını en büyük yapan x'in seçilmesini ister. Bu toplam, i'nin *gerçek* değeri ile diğerlerinin bildirimlerinin toplamıdır. Mekanizma ise $b_i(x) + \Sigma_{j\neq i}\, b_j(x)$'i en büyük yapan sonucu seçer. i dürüst bildirirse ($b_i = v_i$) iki toplam aynı olur ve mekanizma tam olarak i'nin en çok istediği sonucu seçer. Yalan bildirim ise mekanizmayı ya aynı sonuca ya da i'nin istediği toplamı daha küçük yapan bir sonuca götürebilir. Hiçbir yalan dürüstlükten daha iyi sonuç veremez ve bu, diğerlerinin bildirimleri ne olursa olsun geçerlidir.

Ödemeler negatif değil. Birinci terim, diğerlerinin bütün sonuçlar arasında alabileceği *en büyük* refahtır. İkinci terim, diğerlerinin belirli bir sonuçta (x^*'te) aldığı refahtır. En büyük değer, herhangi bir değerden küçük olamaz; bu yüzden $p_i \geq 0$.

IR. Dürüst bildirimde $u_i = \Sigma_j\, v_j(x^*) - max_x\, \Sigma_{j\neq i}\, v_j(x)$'tir. x', i yokken diğerlerinin en iyi sonucu olsun. x^* toplam refahı en büyük yaptığı için $\Sigma_j\, v_j(x^*) \geq \Sigma_j\, v_j(x') = v_i(x') + \Sigma_{j\neq i}\, v_j(x')$'tür. Değerler negatif olmadığından $v_i(x') \geq 0$ ve böylece $u_i \geq 0$ çıkar. □

İspatın ana fikri tek bir cümleyle söylenebilir: VCG'de oyuncunun faydası, sosyal refahtan kendisinden bağımsız bir terimin çıkarılmasıyla elde edilir. Dolayısıyla her oyuncu sosyal refahın en büyük olmasını ister ve mekanizma da tam olarak bunu yapar. Oyuncunun çıkarı ile toplumun çıkarı aynı hizaya getirilmiştir.

4.3.2 Kamu projesi seçimi adım adım

Ayşe, Bora ve Cem adlı üç sakin, üç projeden birini seçecek: Park, Kütüphane ve Spor salonu. Yalnızca bir proje yapılacak. Her sakin her projeye verdiği değeri (bin TL) bildirir; herkesin dürüst bildirdiği varsayılır.

	Park	Kütüphane	Spor salonu	Ödeme p_i	Fayda u_i
Ayşe	40	5	0	0	0
Bora	0	30	20	5	15
Cem	10	20	45	20	25
(a) Σ herkes	50	55	**65 ✓**	25	40
(b) Σ Ayşe olmadan	10	50	65		
(b) Σ Bora olmadan	50	25	45		
(b) Σ Cem olmadan	40	35	20		

Tablo 4.3: VCG ile kamu projesi seçimi. Değerler bin TL; ✓ seçilen proje. Son iki sütun hesap bitince doldurulur.

Örnek 4.10

Hesap dört adımlık bir kalıbı izler: (a) herkesle en iyi seçim, (b) i olmasaydı en iyi seçim, (c) i varken seçilen projede diğerlerinin refahı, (d) ödeme = (b) − (c).

1. **(a) Herkesle en iyi seçim.** Her projede üç değer toplanır. Park: *40 + 0 + 10 = 50*. Kütüphane: *5 + 30 + 20 = 55*. Spor salonu: *0 + 20 + 45 = 65*. En büyük toplam Spor salonundadır: x^* = Spor salonu, sosyal refah 65.
2. **Ayşe (b).** Ayşe olmasaydı yalnız Bora ve Cem'in değerleri toplanır: Park *0 + 10 = 10*, Kütüphane *30 + 20 = 50*, Spor *20 + 45 = 65*. En iyisi Spor'dur; diğerlerinin refahı 65.
3. **Ayşe (c).** Ayşe varken seçilen proje Spor salonudur. Orada diğerlerinin refahı Bora 20 + Cem 45 = 65'tir.
4. **Ayşe (d).** $p_{Ayşe} = 65 - 65 = 0$. Ayşe olsa da olmasa da Spor seçiliyor; Ayşe başkalarına zarar vermiyor.
5. **Bora (b).** Bora olmasaydı Ayşe ve Cem'in toplamları: Park *40 + 10 = 50*, Kütüphane *5 + 20 = 25*, Spor *0 + 45 = 45*. En iyisi Park; diğerlerinin refahı 50.
6. **Bora (c) ve (d).** Bora varken Spor seçilir ve diğerlerinin refahı Ayşe 0 + Cem 45 = 45'tir. $p_{Bora} = 50 - 45 = 5$. Bora olmasaydı Park seçilirdi ve diğerleri 50 alırdı; Bora yüzünden Spor seçiliyor ve diğerleri 45 alıyor. Bora'nın başkalarına verdiği zarar 5'tir.
7. **Cem (b).** Cem olmasaydı Ayşe ve Bora'nın toplamları: Park *40 + 0 = 40*, Kütüphane *5 + 30 = 35*, Spor *0 + 20 = 20*. En iyisi Park; diğerlerinin refahı 40.
8. **Cem (c) ve (d).** Cem varken Spor seçilir ve diğerlerinin refahı Ayşe 0 + Bora 20 = 20'dir. $p_{Cem} = 40 - 20 = 20$. Cem'in başkalarına verdiği zarar 20'dir.
9. **Faydalar.** Fayda = seçilen projeye verilen değer − ödeme. Ayşe *0 − 0 = 0*, Bora *20 − 5 = 15*, Cem *45 − 20 = 25*. Herkesin faydası en az 0'dır (IR). Toplanan ödeme *0 + 5 + 20 = 25*'tir.

Sonuçları yorumlamak, mekanizmanın mantığını pekiştirir. Ayşe en çok Park'ı istiyordu ama onun tercihi sonucu değiştirmedi; Ayşe ödemez. Bora ve Cem'in her biri tek başına pivottur: biri olmasa Park seçilirdi. Bu yüzden ikisi de öder. Cem'in ödemesi daha büyüktür, çünkü Cem'in varlığı seçimi daha pahalı bir biçimde değiştirir: Cem olmadan Ayşe ve Bora'nın Park'tan alacağı 40 yerine Spor'da yalnızca 20 alırlar.

Toplanan 25 birim kimseye geri dağıtılmaz. Bu, VCG'nin bir **bütçe fazlası** verdiği anlamına gelir. Para sakinlere geri dağıtılsaydı, dağıtılan pay oyuncuların bildirimine bağlı olacağı için teşvikler bozulabilirdi. Ödemeler negatif olmadığı için mekanizma dışarıdan para gerektirmez: zayıf bütçe dengesi sağlanır, tam bütçe dengesi sağlanmaz.

Sık yapılan hata

(c) adımında "i varken diğerlerinin en iyi refahını" almak. (c) her zaman x^*'te, yani herkesle seçilen projede hesaplanır; diğerlerinin o anda en sevdiği projede değil. Bora için (c) Spor salonundaki 45'tir; Park'taki 50 değil. Bir başka sık hata da (b) adımında i'nin kendi değerini toplamda bırakmaktır.

4.3.3 Kod izleme: vcg()

VCG'nin bütün hesabı kısa bir Python fonksiyonuyla yazılabilir. Fonksiyon iki girdi alır. `v` bir sözlüktür: `v[i][a]`, oyuncu `i`'nin `a` sonucuna bildirdiği değerdir. `A` olası sonuçların listesidir. Fonksiyon seçilen sonucu ve ödemeleri döndürür.

```
def vcg(v, A):
  N = list(v)            # oyuncular
  def w(a, S):           # S'nin a'daki refahı
    return sum(v[i][a] for i in S)
  x = max(A, key=lambda a: w(a, N))
  p = {}
  for i in N:
    D = [j for j in N if j != i]
    yok = max(w(a, D) for a in A)  # i yokken
    var = w(x, D)                  # i varken
    p[i] = yok - var
  return x, p
```

Kod satır satır şöyle çalışır:

- `N = list(v)` sözlüğün anahtarlarından oyuncu listesini çıkarır: `['Ayşe', 'Bora', 'Cem']`.
- İç fonksiyon `w(a, S)`, `S` kümesindeki oyuncuların `a` sonucuna verdiği değerlerin toplamıdır. `w(a, N)` herkesin refahı, `w(a, D)` ise yalnız `D` kümesinin refahıdır.
- `x = max(A, key=lambda a: w(a, N))`, `A`'daki sonuçlardan `w(a, N)`'yi en büyük yapanı seçer. Bu, argmax işleminin Python karşılığıdır.
- `p = {}` boş bir ödeme sözlüğüdür.
- Döngüde her oyuncu `i` için `D`, `i` dışındaki oyuncuların listesidir. `yok`, `D`'nin bütün sonuçlar arasındaki en büyük refahıdır: ödeme formülünün birinci terimi. `var`, `D`'nin seçilen `x` sonucundaki refahıdır: ikinci terim. Ödeme `yok − var`'dır.

Kamu projesi örneğinde girdi `v = {'Ayşe': {'P': 40, 'K': 5, 'S': 0}, 'Bora': {'P': 0, 'K': 30, 'S': 20}, 'Cem': {'P': 10, 'K': 20, 'S': 45}}` ve `A = ['P', 'K', 'S']`'dir. Tablo 4.4 döngünün her adımını gösterir.

Adım	w(·, N) ya da w(·, D): P, K, S	yok	var	p
x seçimi	w(·, N) = 50, 55, 65 → x = S			{}
i = Ayşe	D = [Bora, Cem]: 10, 50, 65	65	65	{Ayşe: 0}
i = Bora	D = [Ayşe, Cem]: 50, 25, 45	50	45	{…, Bora: 5}
i = Cem	D = [Ayşe, Bora]: 40, 35, 20	40	20	{…, Cem: 20}
dönüş	(S, {Ayşe: 0, Bora: 5, Cem: 20}); toplam 25			

Tablo 4.4: `vcg(v, A)` izlemesi. P = Park, K = Kütüphane, S = Spor salonu.

İzleme, elle yapılan hesapla birebir örtüşür. Kodun sadeliği VCG'nin genel bir mekanizma olmasından gelir: sonuçlar ne olursa olsun (projeler, nesnelerin dağılımları, yollar) aynı fonksiyon çalışır. Tek koşul, her oyuncunun her sonuca bir değer bildirebilmesidir.

Kodla ilgili iki ayrıntı önemlidir. Birincisi, `max` eşitlikte listede ilk gelen sonucu seçer; bu sabit eşitlik bozma kuralı DSIC'i etkilemez. İkincisi, kod $|A|$ sonuç ve n oyuncu için x'i bulmak üzere $|A| \cdot n$ toplama yapar; her oyuncunun ödemesi için de yaklaşık aynı işi tekrarlar. Toplam iş $O(n^2 \cdot |A|)$'dır. Sonuç kümesi küçükken bu çok hızlıdır. Ama sonuçlar kombinatoryal olarak çoksa, örneğin sonuçlar 30 nesnenin oyunculara bütün olası dağılımlarıysa, $|A|$ astronomik büyür. Bu sorun 4.3.6'da ele alınır.

4.3.4 VCG'de yalan kazandırır mı?

Teoreme göre VCG'de yalan kazandırmaz. Kamu projesi örneğinde birkaç yalanı doğrudan sınamak, ispatı somutlaştırır. Her satırda yalnızca bir oyuncu yalan söyler; diğerleri dürüsttür. Son sütun, yalancının *gerçek* değerine göre faydasıdır.

Durum	Bildirim (P, K, S)	Seçilen	Ödeme	Gerçek fayda
Cem dürüst	10, 20, 45	Spor salonu	20	45 − 20 = 25
Cem abartır	10, 20, 100	Spor salonu	20	45 − 20 = 25
Cem küçültür	10, 20, 25	Kütüphane	5	20 − 5 = 15
Bora abartır	0, 60, 20	Kütüphane	25	30 − 25 = 5

Tablo 4.5: VCG'de tek oyuncunun yalanları. Bildirim (P, K, S) sırasıyla Park, Kütüphane ve Spor salonuna verilen değerlerdir.

Örnek 4.11

1. **Cem Spor'u 100 gösterir.** Toplamlar 50, 55, 120 olur; yine Spor seçilir. Cem'in ödemesi (b) − (c) = 40 − 20 = 20'dir: iki terim de yalnız Ayşe ve Bora'nın bildirimlerine bağlıdır ve değişmez. Cem'in faydası yine $45 - 20 = 25$'tir. Abartma hiçbir şey değiştirmedi.
2. **Cem Spor'u 25 gösterir.** Toplamlar Park 50, Kütüphane 55, Spor $0 + 20 + 25 = 45$ olur; Kütüphane seçilir. Cem'in ödemesi: Cem olmasaydı Park seçilirdi ve diğerleri 40 alırdı; Kütüphane'de diğerleri $5 + 30 = 35$ alır. Ödeme $40 - 35 = 5$'tir. Cem'in Kütüphane'ye gerçek değeri 20'dir; faydası $20 - 5 = 15 < 25$. Küçültmek Cem'i en çok istediği projeden etti.
3. **Bora Kütüphane'yi 60 gösterir.** Toplamlar Park 50, Kütüphane $5 + 60 + 20 = 85$, Spor 65 olur; Kütüphane seçilir. Bora'nın ödemesi: Bora olmasaydı Park seçilirdi ve diğerleri 50 alırdı; Kütüphane'de diğerleri $5 + 20 = 25$ alır. Ödeme $50 - 25 = 25$'tir. Bora'nın Kütüphane'ye gerçek değeri 30'dur; faydası $30 - 25 = 5 < 15$.

Bora'nın yalanı ilginçtir. Bora gerçekten de Kütüphane'yi Spor'a tercih ediyordu (30 > 20). Yalanla en sevdiği projeyi yaptırdı. Ama bunun bedeli, diğerlerine verdiği zarar kadar arttı: Kütüphane diğerleri için Spor'dan 20 birim daha kötüdür. Bora'nın kazancı (Kütüphane'de 10 birim daha fazla değer) bu

bedeli karşılamaz. VCG'nin işleyişi tam olarak budur: sonucu değiştirmek serbesttir, ama değişikliğin başkalarına maliyeti ödenir. Değişiklik toplum için iyi değilse yalancı için de iyi değildir.

Bu dört durum yalnızca örnektir. Bilgisayarla daha geniş bir deneme de yapılabilir: bir oyuncu her projeye 0, 10, …, 100 değerlerinden birini bildirsin ve öteki ikisi dürüst kalsın. Bütün bu bildirimlerde hiçbir oyuncu dürüst bildirimden yüksek fayda elde edemez. Bu, Teorem 4.5'in sayısal bir doğrulamasıdır.

4.3.5 Tek nesnede VCG: Vickrey açık artırması

VCG genel bir mekanizmadır; tek nesne satışı onun özel bir durumudur. Bu özel durumda VCG'nin tam olarak ikinci fiyat (Vickrey) açık artırmasına dönüştüğü gösterilebilir. Bunun için aynı `vcg()` fonksiyonu tek nesneye uygulanır.

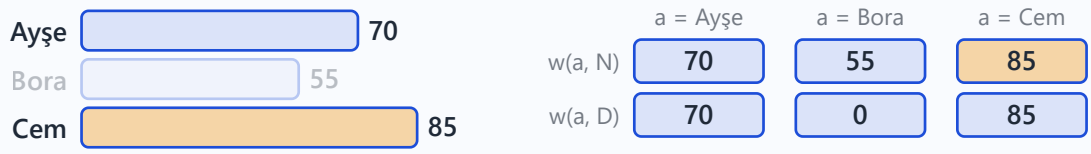
Tek nesnede bir sonuç "nesneyi kim alır?" sorusunun cevabıdır. Sonuçlar oyuncuların kendisidir: `A = ['Ayşe', 'Bora', 'Cem']`. Oyuncu `i`'nin `a` sonucuna değeri, `a = i` ise (nesne kendisine gidiyorsa) nesneye verdiği değer, değilse 0'dır. Değerler 70, 55, 85 olsun. Örneğin `v['Ayşe'] = {'Ayşe': 70, 'Bora': 0, 'Cem': 0}`'dır.

Adım	a = Ayşe	a = Bora	a = Cem	yok	var	p
w(a, N)	70	55	**85 ✓**			x = Cem
i = Ayşe, w(a, D)	0	55	85	85	85	0
i = Bora, w(a, D)	70	0	85	85	85	0
i = Cem, w(a, D)	70	55	0	70	0	70

Tablo 4.6: Tek nesnede `vcg()` izlemesi. Değerler Ayşe 70, Bora 55, Cem 85. Sütunlar "nesne kime gider?" sonuçlarıdır.

Örnek 4.12

1. **Sonuç seçimi.** Her sonucun refahı yalnızca nesneyi alan kişinin değeridir, çünkü diğerleri 0 alır: w(Ayşe, N) = 70, w(Bora, N) = 55, w(Cem, N) = 85. En büyüğü Cem'dir: nesne en yüksek değerli oyuncuya gider.
2. **i = Ayşe.** D = [Bora, Cem]. Ayşe yokken nesne Cem'e giderdi: `yok` = 85. Nesne zaten Cem'de; D'nin oradaki refahı `var` = 85. p[Ayşe] = 85 − 85 = 0: kaybeden ödemez.
3. **i = Bora.** D = [Ayşe, Cem]. Bora yokken nesne yine Cem'e giderdi: `yok` = 85; `var` = 85. p[Bora] = 0.
4. **i = Cem.** D = [Ayşe, Bora]. Cem yokken nesne Ayşe'ye giderdi: `yok` = 70. Nesne Cem'de olduğu için D'nin refahı `var` = 0'dır. p[Cem] = 70 − 0 = 70.
5. **Sonuç.** Cem kazanır ve 70, yani ikinci en yüksek değeri öder. Kaybedenler ödemez. Bu tam olarak Vickrey (ikinci fiyat) açık artırmasıdır.



Şekil 4.11: Tek nesnede VCG izlemesinin bir adımı. i = Bora iken Bora'nın satırı silikleşir; D = {Ayşe, Cem} için $w(a, D) = 70, 0, 85$'tir. Bora yokken de nesne Cem'e giderdi; Bora ödemez.

Sonucun genel nedeni açıktır. Kazanan olmasaydı nesne ikinci en yüksek değerli oyuncuya giderdi ve diğerlerinin refahı bu ikinci en yüksek değer olurdu. Kazanan varken diğerleri hiçbir şey almaz. Fark, ikinci en yüksek değerdir. Kaybeden bir oyuncu olmasaydı ise nesne yine aynı kişiye giderdi; kaybedenin dışsallığı sıfırdır. Bu sonuç, Myerson lemmasıyla (4.2.9) da tutarlıdır: tek nesnede DSIC ödeme kritik tekliftir ve kritik teklif ikinci en yüksek tekliftir.

4.3.6 VCG'nin özellikleri ve sınırları

VCG kuramsal olarak çok çekici bir mekanizmadır. Güçlü yönleri şöyle sıralanır:

- **DSIC:** dürüst bildirim baskın stratejidir; oyuncuların birbirini tahmin etmesi gerekmez.
- **Verimlilik:** sosyal refahı en yüksek sonuç seçilir.
- **IR ve negatif olmayan ödemeler:** Clarke pivotu kullanıldığında ve değerler negatif olmadığında katılan kimse zarar etmez, mekanizma kimseye para ödemez.
- **Genellik:** tek nesnede Vickrey açık artırmasına, birden çok özdeş nesnede "kaybedenlerin en yüksek değerini öde" kuralına dönüşür.

Buna karşın VCG uygulamada az kullanılır. Bunun birkaç nedeni vardır:

- **Tam bütçe dengesi yoktur.** Kamu projesi örneğinde 25 birim fazla kalır. Mekanizma açık vermez (zayıf bütçe dengesi), ama fazla parayı oyunculara geri dağıtamaz. Kamu malı örneklerinde bu para "yakılmış" gibidir.
- **Hesaplama zor olabilir.** *argmax* hesabı, sonuçların çok olduğu kombinatoryal problemlerde NP-zor olabilir. Örneğin birçok nesnenin, her biri farklı nesne demetleri isteyen teklifçilere dağıtıldığı bir açık artırmada en iyi dağılımı bulmak, bilinen en zor optimizasyon problemleri sınıfındadır. Hesap tam yapılamaz da yaklaşık yapılırsa VCG'nin teşvik özellikleri bozulabilir.
- **Danışıklılık ve sahte kimlik.** Oyuncular gizlice anlaşırsa ya da bir oyuncu birden çok sahte kimlikle katılırsa yalan kârlı hâle gelebilir. VCG'nin teşvik garantisi yalnızca tek tek sapmalar içindir.
- **Gelir düşük kalabilir.** Ödemeler dışsallıklara eşit olduğu için, rekabetin az olduğu durumlarda satıcının geliri çok düşük, hatta sıfır olabilir.
- **Mahremiyet.** Oyuncular her sonuca verdikleri değeri açıklamak zorundadır. Bir şirket, bir ürüne ödemeye razı olduğu en yüksek fiyatı rakiplerine ve satıcıya açıklamak istemeyebilir.

Örnek 4.13

Tek nesne Vickrey açık artırmasında değerler Ayşe 70, Bora 55, Cem 85 olsun. Dürüst tekliflerde Cem kazanır ve 70 öder.

1. Ayşe ve Cem gizlice anlaşır: Ayşe 0 teklif eder, Cem dürüst kalır.
2. Teklifler 0, 55, 85 olur. Cem yine kazanır ama ikinci en yüksek teklif artık Bora'nın 55'idir. Cem 55 öder.
3. Cem'in faydası $85 - 70 = 15$'ten $85 - 55 = 30$'a çıkar. Ayşe dürüstken de 0 alıyordu; anlaşmadan kaybı yoktur. Kazanılan 15 birim aralarında paylaşılabilir.

Ayşe tek başına teklifini düşürseydi kendi faydası değişmezdi (0). Bu yüzden DSIC tanımı bu sapmayı yasaklamaz. Ama iki oyuncunun birlikte sapması grup olarak kazandırır. Satıcının geliri 70'ten 55'e düşer.

Bu sınırlar nedeniyle gerçek uygulamalarda VCG yerine daha basit kurallar tercih edilir. Örneğin sponsorlu arama reklamlarında çoğunlukla **genelleştirilmiş ikinci fiyat** (*generalized second price*, GSP) kuralı kullanılır: reklamlar tekliflere göre sıralanır ve her reklamveren kendisinin bir altındaki teklifi öder. GSP DSIC değildir, ama basittir ve gelir açısından avantajlıdır. VCG ise kuramsal bir referans noktası olarak önemini korur: bir problemde verimli ve teşvik uyumlu bir sonucun mümkün olup olmadığını ve bunun bedelini gösterir.

4.4 Parasız mekanizmalar

Önceki mekanizmalarda teşvikler ödemelerle düzeltildi. VCG, oyuncunun yarattığı dışsallığı ona ödeterek dürüstlüğü kârlı kıldı. Ancak birçok önemli problemde para kullanmak ya yasaktır ya da uygunsuzdur. Para olmadan çalışan başlıca mekanizmalar şunlardır: kararlı eşleştirme için Gale–Shapley, tesis yerleşimi için medyan kuralı, okul yerleştirmede seri diktatörlük ve Boston mekanizması, oda takası için TTC.

4.4.1 Para olmadan tasarım

Okul yerleştirmede öğrencilerden okula girmek için para istenmez. Yurt odası değişiminde öğrenciler birbirine para ödemez. Bir seçimde oy satın alınamaz. Organ naklinde organ alım satımı yasaktır. Bu ortamlarda tasarımcının elinde yalnızca bir araç kalır: sonuç kuralı. Teşvik ödemeyle düzeltilemez; kural, doğruyu söylemenin sonucu oyuncu için kötüleştirmeyeceği biçimde seçilmelidir.

Para olmadan teşvik uyumluluğu için kullanılan terim **stratejiye dayanıklılıktır** (*strategy-proofness*). Bir kural stratejiye dayanıklıysa hiçbir oyuncu, diğerleri ne bildirirse bildirsin, gerçek tercihleri yerine başka bir şey bildirerek daha iyi bir sonuç elde edemez. Bu, parasız ortamda DSIC'in karşılığıdır.

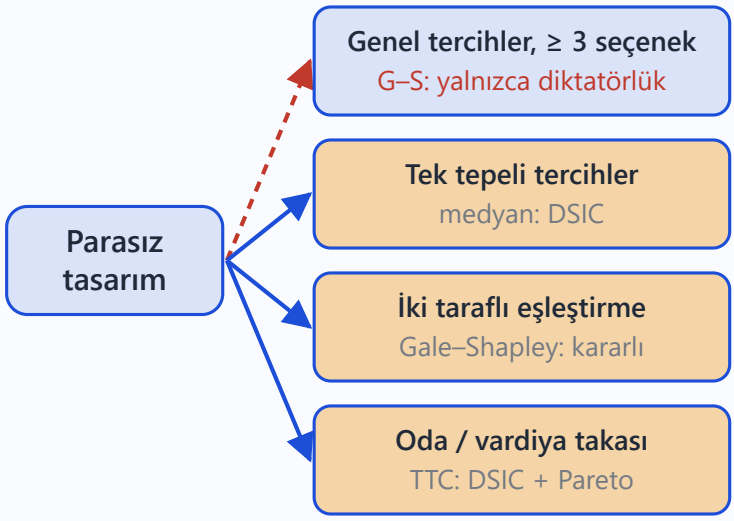
Genel tercihler için kötü bir haber vardır. Gibbard (1973) ve Satterthwaite (1975) birbirinden bağımsız olarak şunu gösterdi:

Teorem 4.6 (Gibbard–Satterthwaite)

En az üç seçeneğin bulunduğu, her seçeneğin bazı tercih profillerinde sonuç olabildiği ve oyuncuların seçenekleri her türlü sırayla sıralayabildiği bir ortamda, stratejiye dayanıklı her kural diktatörlüktür: sonucu her zaman aynı tek kişinin en sevdiği seçenek belirler.

Diktatörlük elbette kabul edilebilir bir kural değildir. Teorem, genel tercihlerde dürüstlüğü garanti eden makul bir oylama kuralının bulunmadığını söyler. Bu yüzden parasız mekanizma tasarımı teoremin varsayımlarından birini gevşeterek ilerler. İki çıkış yolu vardır:

- **Tercihlerin yapısını kullanmak.** Tercihler her türlü sıralama olamıyorsa teorem geçerli olmaz. Örneğin seçenekler bir doğru üzerinde diziliyse ve her oyuncunun tek bir en sevdiği nokta varsa (tek tepeli tercihler), medyan kuralı stratejiye dayanıklıdır (4.4.7).
- **Hedefi değiştirmek.** "En iyi" seçeneği bulmak yerine başka bir ölçüt aranabilir. İki taraflı piyasalarda hedef kararlılıktır (Gale–Shapley). Herkesin elinde bir mal varken hedef takastır (TTC).



Şekil 4.12: Parasız tasarımda yollar. Genel tercihlerde ve en az üç seçenekte Gibbard–Satterthwaite yalnızca diktatörlüğe izin verir. Tek tepeli tercihlerde medyan DSIC'tir; iki taraflı eşleştirmede Gale–Shapley kararlıdır; oda ya da vardiya takasında TTC DSIC ve Pareto optimaldir.

4.4.2 Kararlı eşleştirme problemi

Bir bölümde üç öğrenci (A, B, C) ve üç bitirme projesi (X, Y, Z) vardır. Her proje tek öğrenci alır. Her öğrenci projeleri, her proje danışmanı da öğrencileri bir tercih sırasıyla sıralar. Tercih sıraları Tablo 4.7'de verilmiştir. "$X \succ Y \succ Z$" yazımı, "X'i Y'ye, Y'yi de Z'ye tercih eder" demektir.

Öğrenci	Tercih	Proje	Tercih
A	$X \succ Y \succ Z$	**X**	$B \succ A \succ C$
B	$X \succ Z \succ Y$	**Y**	$A \succ C \succ B$
C	$Y \succ X \succ Z$	**Z**	$A \succ B \succ C$

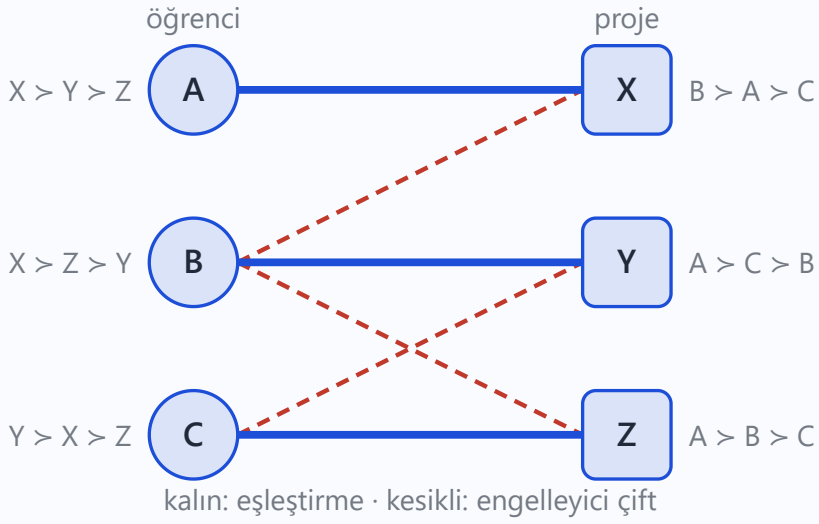
Tablo 4.7: Öğrencilerin ve projelerin tercih sıraları.

Bir **eşleştirme**, her öğrenciyi farklı bir projeye atayan bir listedir. Üç öğrenci ve üç proje için $3! = 3 \cdot 2 \cdot 1 = 6$ farklı eşleştirme vardır. Bunların hangisinin "iyi" olduğu sorusu, bir eşleştirmenin dağılıp dağılmayacağı sorusuyla yanıtlanır.

Tanım 4.8

Bir eşleştirmede, birbirlerini şu anki eşlerine tercih eden bir öğrenci–proje çiftine **engelleyici çift** (*blocking pair*) denir. Engelleyici çifti olmayan eşleştirmeye **kararlı eşleştirme** (*stable matching*) denir.

Engelleyici çift eşleştirmeyi bozabilir. Öğrenci de danışman da mevcut durumdan memnun değildir ve birbirlerini ister; kuralı bir yana bırakıp kendi aralarında anlaşabilirler. Kararlı bir eşleştirmede ise kimse anlaşmayı bozup kaçmak istemez. Bir çiftin engelleyici olması için iki koşul birlikte sağlanmalıdır: öğrenci projeyi şu anki projesine tercih etmeli *ve* proje öğrenciyi şu anki öğrencisine tercih etmelidir. Yalnızca biri istiyorsa çift engelleyici değildir.



Şekil 4.13: Kararsız bir eşleştirme. Kalın çizgiler {A–X, B–Y, C–Z} eşleştirmesini, kırmızı kesikli çizgiler üç engelleyici çifti gösterir: (B, X), (B, Z) ve (C, Y).

Örnek 4.14

{A–X, B–Y, C–Z} eşleştirmesi kararlı mıdır? Dokuz öğrenci–proje çiftinden mevcut eşleştirmede olmayan altısı denetlenir.

1. **A.** A'nın ilk tercihi X'tir ve zaten X'tedir. A hiçbir projeyi X'e tercih etmez; A'yı içeren engelleyici çift yoktur.
2. **(B, X).** B, X'i şu anki projesi Y'ye tercih eder ($X \succ Z \succ Y$). X de B'yi şu anki öğrencisi A'ya tercih eder ($B \succ A \succ C$). Engelleyici.
3. **(B, Z).** B, Z'yi Y'ye tercih eder. Z de B'yi şu anki öğrencisi C'ye tercih eder ($A \succ B \succ C$). Engelleyici.
4. **(C, Y).** C, Y'yi şu anki projesi Z'ye tercih eder ($Y \succ X \succ Z$). Y de C'yi şu anki öğrencisi B'ye tercih eder ($A \succ C \succ B$). Engelleyici.
5. **(C, X).** C, X'i Z'ye tercih eder; ama X, şu anki öğrencisi A'yı C'ye tercih eder. Engelleyici değil.

Eşleştirmenin üç engelleyici çifti vardır; kararlı değildir. Dokuz çiftin hepsi denetlendiğinde başka engelleyici çift çıkmaz.

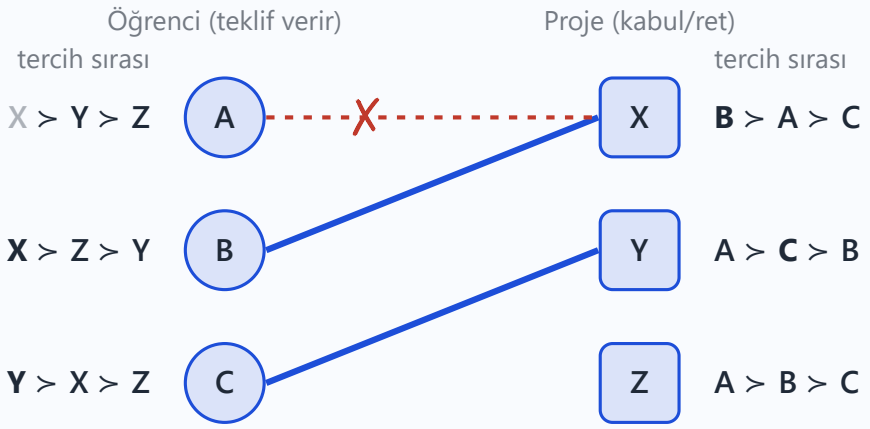
Kararlı bir eşleştirme her zaman var mıdır? Varsa nasıl bulunur? Altı eşleştirmenin hepsini denetlemek küçük örnekte mümkündür. Ama n öğrenci için $n!$ eşleştirme vardır; 20 öğrenci için bu sayı $2 \cdot 10^{18}$'i aşar. Daha akıllıca bir yöntem gerekir. Gale ve Shapley (1962) iki soruyu birden cevaplayan bir algoritma verdi.

4.4.3 Gale–Shapley: ertelenmiş kabul

Gale–Shapley algoritmasında (*Gale–Shapley algorithm*) bir taraf teklif verir, öteki taraf teklifleri değerlendirir. Burada öğrenciler teklif verir, projeler kabul ya da ret eder. Algoritma turlar hâlinde ilerler:

1. Her serbest öğrenci, henüz kendisini reddetmemiş en iyi projesine teklif verir.
2. Her proje, gelen teklifler ve elinde tuttuğu aday arasından en iyisini *geçici olarak* tutar, diğerlerini reddeder.
3. Reddedilen öğrenciler serbest kalır. Serbest öğrenci kalmayınca algoritma durur ve geçici eşleşmeler kesinleşir.

Algoritmanın öteki adı **ertelenmiş kabul**dür (*deferred acceptance*). Ad, ikinci adımdaki "geçici" sözcüğünden gelir. Proje bir teklifi kabul ettiğinde kesin söz vermez; daha iyi bir aday gelirse eskisini bırakabilir. Kesin kabul algoritmanın sonuna ertelenir. Bu ayrıntı, algoritmanın kararlı bir sonuç vermesinin anahtarıdır.



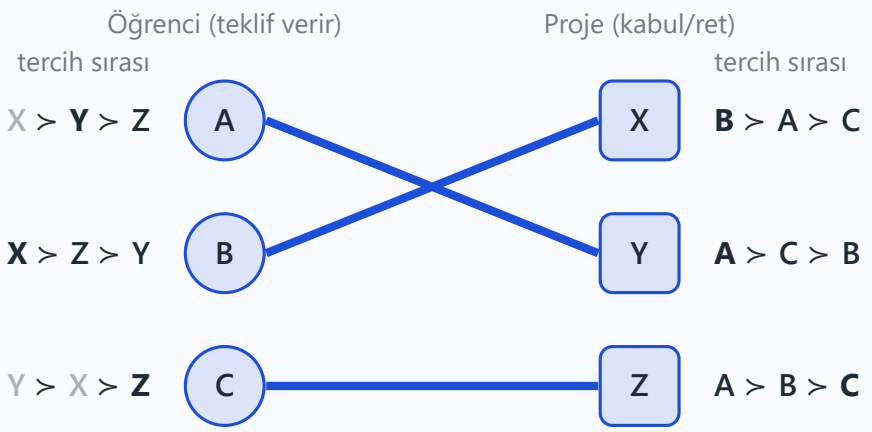
Şekil 4.14: Gale–Shapley, tur 1 kararı. A ve B ikisi de X'e teklif verir; X, 1. sıradaki B'yi geçici tutar ve A'yı reddeder (kırmızı çarpı). C'nin Y'ye teklifi geçici olarak kabul edilir. Gri harfler, öğrencinin artık teklif veremeyeceği projelerdir.

Örnek 4.15

Tablo 4.7'deki tercihlerle algoritma şöyle çalışır.

1. **Başlangıç.** Herkes serbesttir, projeler boştur.
2. **Tur 1 teklifleri.** Her öğrenci ilk tercihine teklif verir: A→X, B→X, C→Y.
3. **Tur 1 kararı.** X'e iki teklif gelir. X'in sırası B ≻ A ≻ C'dir; B tutulur, A reddedilir. Y'ye yalnız C teklif verir; C geçici olarak kabul edilir.
4. **Tur 2 teklifleri.** Serbest olan yalnız A'dır. A'yı X reddetti; sıradaki tercihi Y'dir: A→Y.
5. **Tur 2 kararı.** Y'nin elinde C vardır, şimdi A da gelir. Y'nin sırası A ≻ C ≻ B'dir; A daha iyidir. Y, A'yı tutar ve C'yi reddeder. Daha önce "kabul edilmiş" olan C artık serbesttir: kabul geçiciydi.
6. **Tur 3 teklifleri.** C'yi Y reddetti; sıradaki tercihi X'tir: C→X.
7. **Tur 3 kararı.** X'in elinde B vardır. X, B'yi C'ye tercih eder; C reddedilir.
8. **Tur 4.** C'nin sıradaki tercihi Z'dir: C→Z. Z boştur; C kabul edilir.
9. **Son.** Herkes eşleşti, algoritma durur. Sonuç: A–Y, B–X, C–Z.

Kararlılık denetimi. B ilk tercihi X'tedir; şikâyeti yoktur. A, Y'dedir ve yalnız X'i Y'ye tercih eder; ama X, B'yi A'ya tercih eder. C, Z'dedir ve X ile Y'yi Z'ye tercih eder; ama X, B'yi C'ye, Y de A'yı C'ye tercih eder. Hiçbir engelleyici çift yoktur: eşleştirme kararlıdır.



Şekil 4.15: Gale–Shapley'in sonucu: A–Y, B–X, C–Z. Kalın harfler her öğrencinin ve projenin eşini gösterir; engelleyici çift yoktur.

Kararlılık "herkes en sevdiğini alır" demek değildir. C son tercihi Z'ye kaldı, çünkü C'nin daha çok istediği X ve Y, C'den daha çok sevdikleri öğrencilerle eşleşmiştir.

4.4.4 Kod izleme: gale_shapley()

Algoritmanın Python sürümü turlar yerine bir kuyrukla çalışır. Her adımda kuyruktan bir serbest öğrenci çıkar ve bir teklif verir. Girdi olarak `ogr` sözlüğü her öğrencinin tercih listesini (en iyiden kötüye), `proj` sözlüğü ise her projenin öğrencilere verdiği sırayı tutar: `proj[p][s]`, `p` projesinin `s` öğrencisine verdiği sıradır ve 0 en iyi demektir. Örneğin `proj['X'] = {'B': 0, 'A': 1, 'C': 2}` 'dir.

```
def gale_shapley(ogr, proj):
  serbest, es = list(ogr), {}
  sira = {s: 0 for s in ogr}
  while serbest:
    s = serbest.pop(0)
    p = ogr[s][sira[s]]        # sıradaki
    sira[s] += 1
    r = es.get(p)              # p'deki aday
    if r is None or proj[p][s] < proj[p][r]:
      es[p] = s                # kabul
      if r: serbest.append(r)
    else: serbest.append(s)    # red
  return es
```

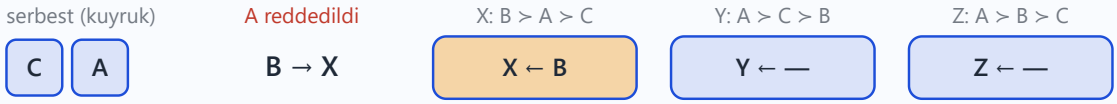
Kodun parçaları şunlardır:

- `serbest` serbest öğrencilerin kuyruğudur; başta herkes buradadır. `es` sözlüğü her projenin geçici olarak tuttuğu öğrenciyi saklar.
- `sira[s]`, öğrenci `s` 'nin tercih listesinde sıradaki indistir. Başta 0'dır, yani en sevilen proje.

- Döngüde kuyruğun başındaki öğrenci `s` çıkarılır, sıradaki tercihi `p`'ye teklif verir ve `sira[s]` bir artırılır. Böylece bir öğrenci aynı projeye iki kez teklif vermez.
- `r`, `p`'nin şu anda tuttuğu adaydır; proje boşsa `None`'dır.
- Proje boşsa ya da `p`, `s`'yi `r`'ye tercih ediyorsa (sıra sayısı küçükse) `s` kabul edilir. Eski aday `r` varsa kuyruğa döner.
- Aksi hâlde `s` reddedilir ve kuyruğa geri döner.

Teklif	s → p	r (p'deki aday)	Karar	es (X, Y, Z)	serbest (sonra)
1	A → X	None	kabul	A, —, —	[B, C]
2	B → X	A	B kabul (0 < 1), A kuyruğa	B, —, —	[C, A]
3	C → Y	None	kabul	B, C, —	[A]
4	A → Y	C	A kabul (0 < 1), C kuyruğa	B, A, —	[C]
5	C → X	B	red (B'nin sırası 0, C'nin 2)	B, A, —	[C]
6	C → Z	None	kabul	B, A, C	[]

Tablo 4.8: `gale_shapley(ogr, proj)` izlemesi. Her satır bir tekliftir.



Şekil 4.16: İzlemenin 2. tekliften sonraki durumu. B, X'e teklif verdi; X için B daha iyi olduğundan B kabul edildi ve A reddedilip kuyruğa döndü. Kuyrukta C ve A vardır; X ← B, Y ve Z boştur.

Kuyruk boşalınca algoritma 6 teklifle biter. Sonuç `es = {'X': 'B', 'Y': 'A', 'Z': 'C'}`'dir; bu, eşzamanlı turlarla bulunan A–Y, B–X, C–Z eşleştirmesiyle aynıdır. Tekliflerin işlenme sırası farklı olsa da (turlarda A ile B aynı anda teklif verirken kodda sırayla verir) sonuç değişmez. Gale–Shapley'in sonucu tekliflerin hangi sırayla işlendiğine bağlı değildir; bu, algoritmanın bilinen bir özelliğidir.

Kaç teklif? Her öğrenci her projeye en fazla bir kez teklif verir, çünkü `sira[s]` yalnızca artar. n öğrenci ve n proje için toplam teklif sayısı en fazla $n \cdot n = n^2$'dir. Her teklif sabit sürede işlendiği için algoritmanın çalışma süresi $O(n^2)$'dir. 20 öğrencilik bir örnekte en fazla 400 teklif yeterlidir; bu, *20!* eşleştirmeyi tek tek denetlemekle kıyaslanamayacak kadar hızlıdır.

4.4.5 Gale–Shapley'in özellikleri

Teorem 4.7 (Gale ve Shapley, 1962)

Gale–Shapley algoritması her girdide sonlanır ve kararlı bir eşleştirme döndürür. Dolayısıyla eşit sayıda öğrenci ve proje varken kararlı eşleştirme her zaman vardır.

İspat. **Sonlanma.** Her teklif bir öğrenci–proje çiftini bir kez kullanır ve en fazla n^2 çift vardır. Bu yüzden algoritma en fazla n^2 teklifte durur. Bir öğrencinin bütün projelerden reddedilmesi de imkânsızdır: bir proje bir kez teklif aldıktan sonra hep birini tutar; bir öğrenci bütün projelerden reddedilmişse bütün projeler dolu demektir, ama proje sayısı öğrenci sayısına eşit olduğundan bu öğrencinin kendisi de bir projede olmalıdır.

Kararlılık. Sonuçta öğrenci s'nin p projesini kendi eşine tercih ettiğini varsayalım. s tercih sırasını izleyerek teklif verdiği için, eşine ulaşmadan önce p'ye teklif vermiş ve p tarafından reddedilmiştir. Bir proje bir öğrenciyi yalnızca daha iyi bir aday tuttuğunda reddeder. Projenin tuttuğu aday da zamanla yalnızca iyileşir. Demek ki sonunda p'nin eşi, p'nin gözünde s'den iyidir ve p, s'yi istemez. (s, p) engelleyici çift olamaz. Bu her çift için geçerli olduğundan eşleştirme kararlıdır. □

Algoritmanın öteki önemli özellikleri şunlardır:

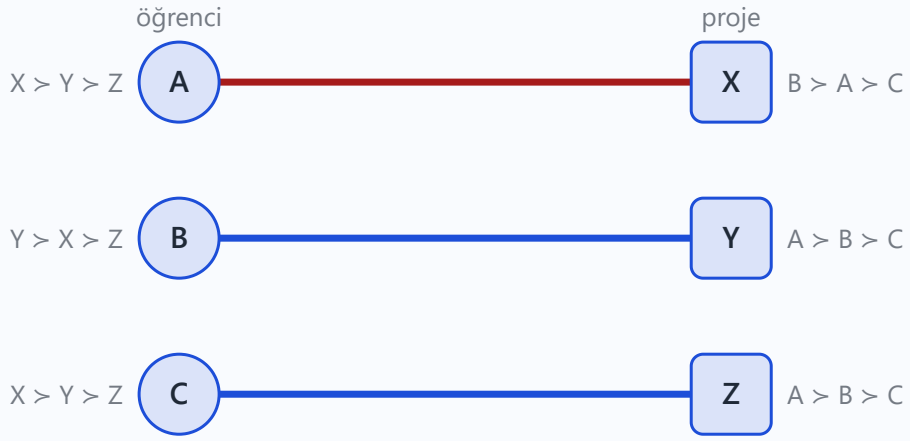
- **Teklif veren tarafa en iyi.** Bir problemde birden çok kararlı eşleştirme olabilir. Gale–Shapley'de teklif veren her oyuncu, bütün kararlı eşleştirmeler içinde alabileceği en iyi eşi alır. Teklif alan taraf ise bütün kararlı eşleştirmeler içinde alabileceği en kötü eşi alır. Hangi tarafın teklif verdiği bu yüzden önemli bir tasarım kararıdır.
- **Teklif verenler için DSIC.** Teklif veren taraf için gerçek tercih sırasını bildirmek baskın stratejidir (Roth, 1982). Teklif alan taraf ise bazen yalan söyleyerek kazanabilir (4.4.6).
- **Uygulamalar.** ABD'de tıp fakültesi mezunlarının ihtisas yerlerine yerleştirilmesi (NRMP) bu algoritmanın bir sürümüyle yapılır. Boston ve New York okul seçimi sistemleri de ertelenmiş kabule dayanır. Roth ve Shapley 2012 Nobel Ekonomi Ödülü'nü bu çalışmalarla aldı.

Örnekteki tercihlerde altı eşleştirmenin hepsi denendiğinde yalnızca bir kararlı eşleştirme bulunur: A–Y, B–X, C–Z. Bu yüzden projeler teklif verseydi de sonuç aynı olurdu. Projeler teklif verdiğinde X'in ilk tercihi B'dir, Y'nin ve Z'nin ilk tercihi A'dır. A, Y ile Z arasında Y'yi tutar; Z reddedilir ve sıradaki B'ye teklif verir. B, X'i Z'ye tercih ettiği için Z'yi reddeder; Z son olarak C'ye teklif verir. Sonuç yine A–Y, B–X, C–Z'dir.

> **Sık yapılan hata**
>
> Ertelenmiş kabulü “ilk gelen alır” diye uygulamak. Tur 1'de Y'nin C'yi kabul etmesi kesin değildir; tur 2'de A gelince C bırakılır. Kabulü kesin sayan bir uygulama (örneğin Y'nin C'yi tutup A'yı reddetmesi) kararsız bir eşleştirmeye götürebilir: A ve Y birbirini ister ama ayrı kalır.

4.4.6 Teklif alan taraf yalanla kazanabilir

Gale–Shapley teklif verenler için DSIC'tir, teklif alan taraf için değildir. Bunu yeni bir örnek gösterir. Öğrenciler yine teklif verir. Tercihler şöyledir: A: X ≻ Y ≻ Z, B: Y ≻ X ≻ Z, C: X ≻ Y ≻ Z. Projelerin gerçek tercihleri X: B ≻ A ≻ C; Y ve Z: A ≻ B ≻ C'dir.



Şekil 4.17: Dürüst bildirimde sonuç A–X, B–Y, C–Z. X, gerçek 2. tercihi A ile eşleşir (kırmızı çizgi).

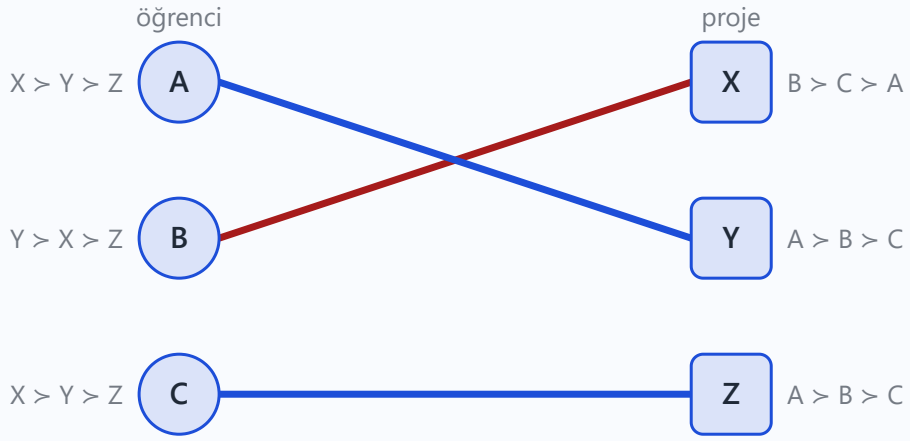
Örnek 4.16

Dürüst bildirim.

1. A→X: X boş, A tutulur. B→Y: Y boş, B tutulur.
2. C→X: X, A'yı C'ye tercih eder; C reddedilir. C→Y: Y, B'yi C'ye tercih eder; C reddedilir. C→Z: kabul.
3. Sonuç A–X, B–Y, C–Z. X, gerçek sırasında 2. olan A'yı alır.

X yalan söyler ve sırasını B ≻ C ≻ A diye bildirir; A'yı en sona atar.

1. A→X: X boş, A tutulur. B→Y: B tutulur.
2. C→X: X'in bildirdiği sırada C, A'dan iyidir. X, C'yi tutar ve A'yı reddeder.
3. A→Y: Y, A'yı B'ye tercih eder. A tutulur, B reddedilir.
4. B→X: X, B'yi C'ye tercih eder. B tutulur, C reddedilir.
5. C→Y: Y, A'yı C'ye tercih eder; C reddedilir. C→Z: kabul.
6. Sonuç A–Y, B–X, C–Z. X artık gerçek 1. tercihi B'yi almıştır. Yalan kazandırdı.



Şekil 4.18: X'in yalanından sonra sonuç A–Y, B–X, C–Z. A'nın X'ten reddi zincirleme B'yi X'e getirir; X gerçek 1. tercihi B ile eşleşir (kırmızı çizgi).

Yalanın nasıl işlediği Şekil 4.17 ile Şekil 4.18 karşılaştırılınca görülür. X, A'yı reddederek bir *ret zinciri* başlatır. Reddedilen A, Y'ye gider ve Y'de B'nin yerini alır. Yerinden olan B ise sıradaki tercihi X'e gelir. X'in yalanı, istediği öğrenciyi kendisine geri getiren bir zincir kurmuştur.

Altı eşleştirme denetlendiğinde bu tercihlerde iki kararlı eşleştirme olduğu görülür: öğrencilere en iyi gelen A–X, B–Y, C–Z ve projelere en iyi gelen A–Y, B–X, C–Z. Öğrenci önerili Gale–Shapley birincisini bulur. X'in yalanı sonucu ikincisine, yani projelere en iyi gelene taşır. Bu, genel bir sonucun özel bir hâlidir: hiçbir kararlı mekanizma iki taraf için birden DSIC olamaz (Roth, 1982). Tasarımcı, dürüstlüğü hangi tarafa garanti edeceğini seçmek zorundadır. Okul yerleştirmede genellikle öğrenciler teklif veren taraf yapılır; okulların tercihleri ise çoğunlukla yasal öncelik kurallarından gelir ve stratejik olarak değiştirilemez.

4.4.7 Tesis yerleşimi: ortalama mı, medyan mı?

A, B, C, D ve E adlı beş köy, düz bir yol üzerinde sırasıyla 2, 5, 9, 14 ve 20. kilometrelerde bulunur. Yol üzerinde tek bir tesis, örneğin bir hastane, kurulacaktır. Her köy tesisin kendisine yakın olmasını ister; bir köyün maliyeti tesise uzaklığıdır. Köylerin konumu özel bilgi gibi düşünülür: her köy kendi konumunu (ya da en çok istediği noktayı) bildirir ve tesis bildirimlere göre yerleştirilir.

Bu ortamda tercihlerin özel bir yapısı vardır. Her köyün yol üzerinde tek bir en sevdiği noktası (kendi konumu) vardır ve bu noktadan uzaklaştıkça memnuniyeti azalır. Bu tür tercihlere **tek tepeli tercihler** (*single-peaked preferences*) denir: memnuniyet grafiğinin tek bir tepesi vardır. Gibbard–Satterthwaite teoremi bu yapıyı varsaymaz; bu yüzden burada stratejiye dayanıklı ve diktatörlük olmayan bir kural bulunabilir.

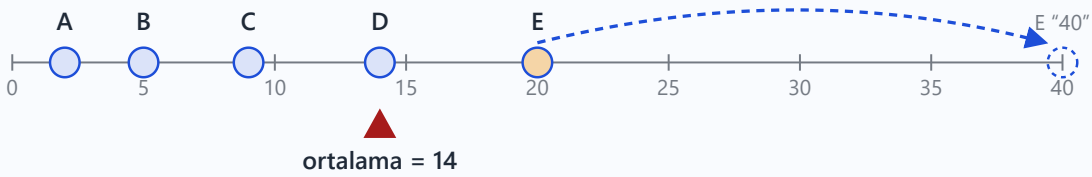
İki kural karşılaştırılır. **Ortalama mekanizması** tesisi bildirimlerin ortalamasına kurar. **Medyan mekanizması** bildirimleri sıralar ve ortadakini seçer. Beş bildirimde medyan, sıralamadaki üçüncü sayıdır.

Durum	Bildirimler	Tesis	E'nin uzaklığı
Ortalama, herkes dürüst	2, 5, 9, 14, 20	10	10
Ortalama, E "40" der	2, 5, 9, 14, 40	14	6
Medyan, herkes dürüst	2, 5, 9, 14, 20	9	11
Medyan, E "40" der	2, 5, 9, 14, 40	9	11
Medyan, E "8" der	2, 5, 8, 9, 14	8	12

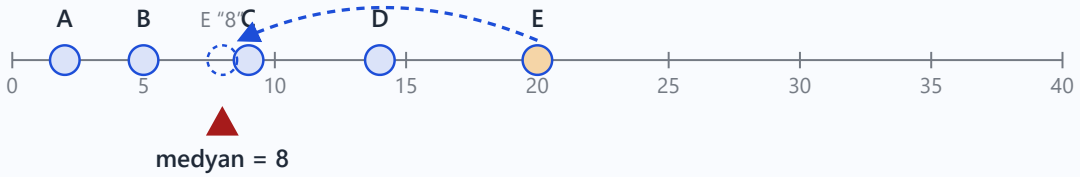
Tablo 4.9: Ortalama ve medyan kuralında E köyünün (20. km) bildirimleri.

Örnek 4.17

1. **Ortalama, herkes dürüst.** *(2 + 5 + 9 + 14 + 20)/5 = 50/5 = 10*. Tesis 10. km'ye kurulur; E'nin uzaklığı *20 − 10 = 10* km'dir.
2. **Ortalama, E yalan söyler.** E konumunu 40 bildirir: *(2 + 5 + 9 + 14 + 40)/5 = 70/5 = 14*. Tesis 14. km'ye kayar; E'nin uzaklığı *20 − 14 = 6* km'ye iner. Yalan kazandırdı: ortalama manipüle edilebilir.
3. **E'nin en kârlı yalanı.** E, tesisi tam kendi konumuna çekmek isterse ortalamanın 20 olması gerekir: *(2 + 5 + 9 + 14 + x)/5 = 20*, yani *30 + x = 100* ve *x = 70*. E, 70 bildirerek uzaklığını 0'a indirir.
4. **Medyan, herkes dürüst.** Sıralı bildirimler 2, 5, 9, 14, 20'dir; ortadaki 9'dur. E'nin uzaklığı *20 − 9 = 11* km'dir.
5. **Medyan, E 40 der.** Sıralama 2, 5, 9, 14, 40 olur; medyan yine 9'dur. Uç değer medyanı çekemez; E'nin uzaklığı 11'de kalır.
6. **Medyan, E 8 der.** E medyanın öbür yanına geçerse sıralama 2, 5, 8, 9, 14 olur ve medyan 8'e iner. E'nin uzaklığı *20 − 8 = 12* km'ye çıkar. Yalan zarar verdi.



Şekil 4.19: Ortalama kuralında E, konumunu 20 yerine 40 bildirir (kesikli ok); ortalama 10'dan 14'e kayar ve tesis E'ye yaklaşır.



Şekil 4.20: Medyan kuralında E, medyanın öbür yanına geçip 8 bildirir; medyan 9'dan 8'e iner ve tesis E'den uzaklaşır.

Ortalamanın zayıflığı, her bildirimin sonucu doğrudan etkilemesinden gelir. Bir köy bildirimini 5 birim artırırsa ortalama 1 birim artar. Uçtaki bir köy, konumunu abartarak ortalamayı istediği yere kadar çekebilir. Medyan ise yalnızca sıralamaya bakar. Medyanın sağındaki bir köy, bildirimini ne kadar büyütürse büyütsün sağda kalır ve sıralamadaki orta eleman değişmez.

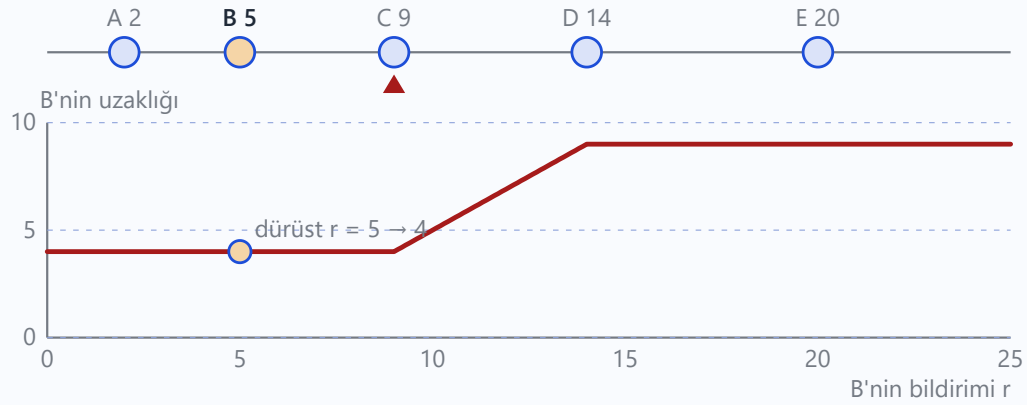
Medyanın bir üstünlüğü daha vardır. Toplam uzaklık medyanda $7 + 4 + 0 + 5 + 11 = 27$, ortalamada $8 + 5 + 1 + 4 + 10 = 28$'dir. Medyan, toplam uzaklığı en aza indiren noktadır. Nedeni şudur: tesis medyanın sağındaki bir noktadaysa, onu bir birim sola kaydırmak soldaki köylerin her birine bir birim yaklaştırır ve sağdakilerin her birinden bir birim uzaklaştırır. Medyanın sağında, solda kalan köy sayısı sağdakinden fazladır; kaydırma toplamı azaltır. Aynı akıl yürütme soldan da yapılır. Toplam ancak medyanda daha fazla azaltılamaz. Yani medyan hem stratejiye dayanıklıdır hem de toplam maliyet açısından en iyi sonucu verir. Her köy için 0 ile 60 arasındaki bütün bildirimler bilgisayarla denendiğinde medyanda hiçbir yalanın kazandırmadığı görülür.

4.4.8 Medyan neden stratejiye dayanıklı?

Teorem 4.8 (Medyan mekanizması)

Tek tepeli tercihlerde (burada: her köy tesise uzaklığını en aza indirmek ister) medyan mekanizması stratejiye dayanıklıdır: her köy için gerçek konumunu bildirmek baskın stratejidir (Moulin, 1980).

İspat, köyün medyana göre nerede olduğuna bakan üç durumdan oluşur. Köy i'nin gerçek konumu x_i, diğerleri dürüstken medyan m, köyün yalan bildirimi r olsun. Akıl yürütme diğerlerinin bildirimleri ne olursa olsun geçerlidir; somutlaştırmak için örnekteki köyler kullanılır.



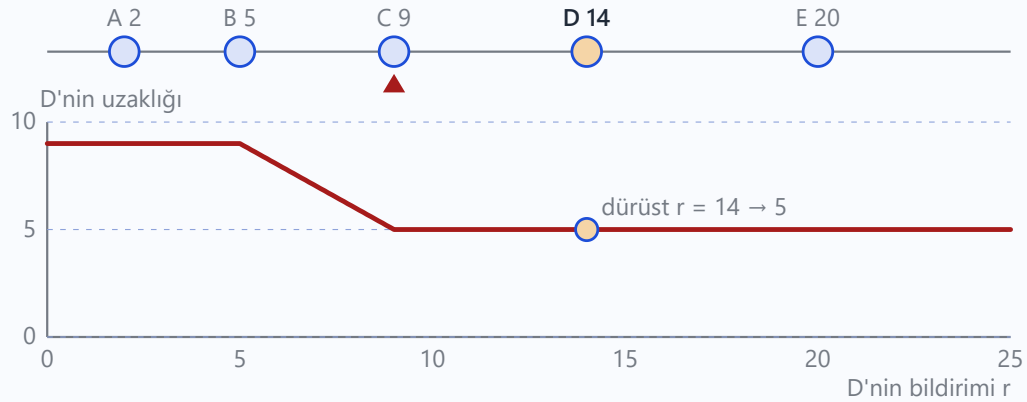
Şekil 4.21: Durum 1: B (5. km) medyanın solunda. Grafik B'nin bildirimi r'ye göre tesise uzaklığını gösterir. $r \le 9$ iken uzaklık 4'te sabittir; $r > 9$ medyanı sağa iter ve uzaklık 9'a kadar artar.

İspat. **Durum 1: $x_i < m$.** Örnekte B = 5, medyan 9'dur. B dürüstken uzaklığı $9 - 5 = 4$'tür. B'nin 9'dan küçük ya da eşit her bildirimi B'yi sıralamada medyanın solunda (ya da üstünde) tutar; medyan 9'da kalır ve uzaklık 4 olur. B'nin 9'dan büyük bir bildirimi B'yi medyanın sağına taşır; o zaman medyan sağa kayar (B'nin bildirimine ya da en fazla 14'e) ve B'den uzaklaşır. B'nin hiçbir bildirimi medyanı sola, yani B'ye doğru çekemez; çünkü medyanın solunda zaten B ile A vardır ve B'nin oradaki yerini değiştirmesi sıralamanın ortasını değiştirmez. Hiçbir r uzaklığı 4'ün altına indiremez.

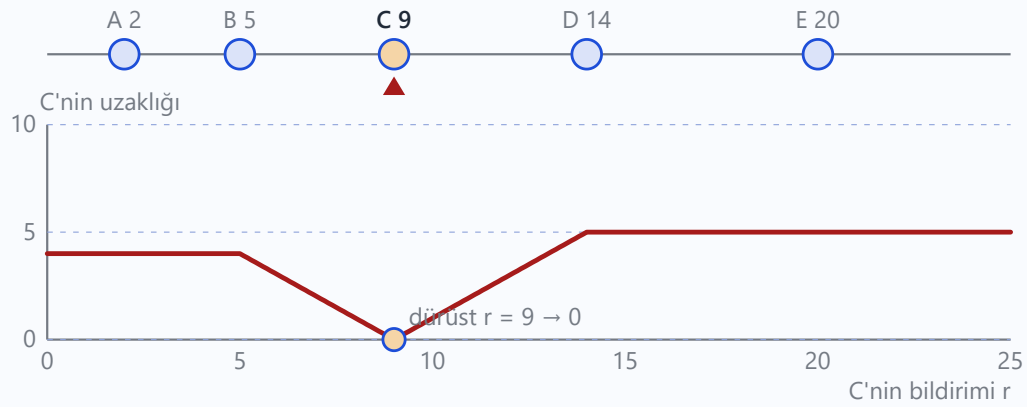
Durum 2: $x_i > m$. Örnekte D = 14. Bu durum birincinin aynadaki görüntüsüdür. D'nin 9'dan büyük ya da eşit her bildirimi medyanı değiştirmez; uzaklık $14 - 9 = 5$'te kalır. D'nin 9'dan küçük bir bildirimi medyanı sola, yani D'den uzağa çeker.

Durum 3: $x_i = m$. Örnekte C = 9. Tesis tam C'nin konumundadır ve uzaklık 0'dır. Bu, mümkün olan en küçük uzaklıktır; hiçbir yalan daha iyisini yapamaz. Her yalan medyanı ancak C'den uzaklaştırabilir ya da yerinde bırakabilir.

Üç durumda da dürüst bildirim, diğerleri ne bildirirse bildirsin en iyi sonucu verir. Medyan mekanizması stratejiye dayanıklıdır. □



Şekil 4.22: Durum 2: D (14. km) medyanın sağında. $r \ge 9$ iken uzaklık 5'te sabittir; $r < 9$ medyanı sola çeker ve uzaklık 9'a kadar artar.



Şekil 4.23: Durum 3: C (9. km) medyanın üstünde. Dürüst bildirimde uzaklık 0'dır; 9'dan uzaklaşan her bildirim medyanı da C'den uzaklaştırır (en fazla 5. km'ye ya da 14. km'ye kadar).

Üç grafikte de uzaklık eğrisi dürüst bildirimin altına hiçbir yerde inmez: düz kısımlar yalanın hiçbir şey değiştirmediği, yükselen kısımlar yalanın zarar verdiği bildirimlerdir.

Aynı fikir oylamaya da taşınır. Seçenekler bir doğru üzerinde sıralanabiliyorsa (örneğin solcu–sağcı ekseninde adaylar ya da bütçe büyüklükleri) ve her seçmenin tek tepeli bir tercihi varsa, medyan seçmenin en sevdiği seçenek her başka seçeneği ikili oylamada yener. Bu sonuca Duncan Black'in medyan seçmen teoremi denir. Gibbard–Satterthwaite'in kötü haberinden kaçmanın bir yolu, tercihlerin bu yapıda olduğu durumları tanımaktır.

Sık yapılan hata

Köy sayısı çift olduğunda medyanı “iki ortanın ortalaması” diye almak. Bu, ortalamanın manipüle edilebilirliğini geri getirir: iki orta köyden biri bildirimini değiştirerek tesisi kendine çekebilir. Stratejiye dayanıklılık için çift sayıda bildirimde iki ortadakinden biri (örneğin soldaki) kuralla seçilmelidir.

4.4.9 Seri diktatörlük ve Boston mekanizması

Okul yerleştirmede öğrenciler okulları sıralar ve her okulun sınırlı kontenjanı vardır. Okulların öğrenciler üzerinde **öncelik** sıraları da olabilir: örneğin okula yakın oturanlar ya da kardeşi okulda olanlar öncelikli olabilir. İki basit mekanizma bu ortamda çok farklı teşvikler yaratır.

Seri diktatörlük (*serial dictatorship*). Öğrenciler bir sıraya dizilir. Sırası gelen öğrenci, kalan okullar içinden en sevdiğini alır. Öğrenci, kendinden öncekilerin seçimini etkileyemez; kendi seçimi ise yalnızca kendi tercih listesine bağlıdır. Bu yüzden gerçek tercihleri bildirmek baskın stratejidir: mekanizma DSIC'tir. Sonuç ayrıca **Pareto optimaldir**: başka hiçbir atama, kimseyi kötüleştirmeden birini daha iyi duruma getiremez. Sıra kurayla belirlenirse mekanizmaya **rastgele seri diktatörlük** (*random serial dictatorship*, RSD) denir. Gibbard–Satterthwaite'in diktatörlüğünden farkı, “diktatörün” tek kişi olmaması ve sıranın herkese sırayla seçim hakkı vermesidir.

Boston mekanizması ya da **doğrudan kabul** (*immediate acceptance*). Her turda yerleşmemiş öğrenciler tercih listelerindeki sıradaki okula başvurur. Her okul, o turda kendisine başvuran öğrencilere boş koltuklarını öncelik sırasına göre verir. Verilen koltuk sonraki turlarda geri alınmaz. Ertelenmiş kabulle farkı buradadır: kabul geçici değil, kesindir. Bu yüzden ilk tercihini kaybeden bir öğrenci, ikinci tercihine başvurduğunda onu dolmuş bulabilir.

Örnekte üç öğrenci (A, B, C) ve her biri tek koltuklu üç okul (X, Y, Z) vardır. Tercihler A: X ≻ Y ≻ Z, B: X ≻ Y ≻ Z, C: Y ≻ X ≻ Z'dir. Öncelikler X ve Z için A, B, C; Y için B, A, C'dir.

Tur	B dürüst (X ≻ Y ≻ Z)	B "Y ≻ X ≻ Z" der
1	X'e A, B başvurur → A alır; Y'ye C başvurur → C alır	X'e A başvurur → A alır; Y'ye B, C başvurur → B alır
2	B → Y: dolu	C → X: dolu
3	B → Z: B alır	C → Z: C alır
Sonuç	A–X, B–Z, C–Y	A–X, B–Y, C–Z

Tablo 4.10: Boston mekanizmasında B'nin dürüst ve stratejik bildirimi.

Örnek 4.18

1. **B dürüst, tur 1.** A ve B ilk tercihleri X'e başvurur. X'in önceliğinde A, B'den öndedir; X, A'ya verilir. C ilk tercihi Y'ye başvurur ve tek başvuran olduğu için Y'yi alır. Bu koltuklar artık kesindir.
2. **Tur 2.** Yerleşmemiş tek öğrenci B'dir. B ikinci tercihi Y'ye başvurur; ama Y, tur 1'de C'ye verilmiştir.
3. **Tur 3.** B üçüncü tercihi Z'ye başvurur ve Z'yi alır. Sonuç A–X, B–Z, C–Y. B son tercihine yerleşti.
4. **B yalan söyler.** B listesini "Y ≻ X ≻ Z" diye yazar. Tur 1'de X'e yalnız A başvurur ve X'i alır. Y'ye B ve C başvurur; Y'nin önceliğinde B, C'den öndedir; Y, B'ye verilir.
5. **Tur 2 ve 3.** C ikinci tercihi X'e başvurur; X doludur. C, Z'ye yerleşir. Sonuç A–X, B–Y, C–Z.
6. **Karşılaştırma.** B'nin gerçek sırası X ≻ Y ≻ Z'dir. Dürüstken Z'yi, yalanla Y'yi aldı. İkinci tercihini listenin başına yazmak kazandırdı: Boston mekanizması DSIC değildir.

Yalanın nedeni kesin kabuldür. B, X'i zaten alamayacağını tahmin ederse ilk turu "boşa harcamak" istemez; çünkü ilk turda Y'nin koltuğu dolacaktır. Boston mekanizmasında öğrenciler, kazanma şanslarını tahmin edip tercih listelerini buna göre düzenlemek zorunda kalır. Bu, bilgiye ve danışmanlığa erişimi olan ailelere avantaj sağlar; tercihlerini olduğu gibi yazan aileler ise zarar görebilir.

Aynı profilde öğrenci önerili Gale–Shapley dürüst bildirimle A–X, B–Y, C–Z verir. Tur 1'de A ve B X'e, C Y'ye teklif verir; X, A'yı tutar ve B'yi reddeder. B, Y'ye teklif verir; Y'nin önceliğinde B, C'den önde olduğu için B tutulur ve C reddedilir. C, X'te A'ya yenilir ve Z'ye yerleşir. Ertelenmiş kabulde C'nin tur 1'deki Y koltuğu geçici olduğu için B, dürüst kalarak Y'yi alabilir. Abdulkadiroğlu ve Sönmez (2003),

Boston mekanizmasında tercihleri olduğu gibi yazmanın öğrenciye zarar verebildiğini gösterdi ve okul seçimine mekanizma tasarımı bakışını getirdi. Boston şehri 2005'te okul yerleştirmesinde öğrenci önerili ertelenmiş kabule geçti.

4.4.10 Yurt odası takası: TTC

Bu kez her öğrencinin elinde bir yurt odası vardır ve para kullanılmaz. Öğrenciler başkalarının odalarını kendi odalarıyla birlikte sıralar. Amaç, gönüllü takaslarla herkesin elindekinden daha iyi ya da aynı durumda olduğu bir yeniden dağıtım bulmaktır. Bu probleme **ev takası** (*housing market*) problemi denir. Problemi Shapley ve Scarf (1974) tanımladı ve aşağıdaki algoritmayı David Gale'e atfetti.

En İyi Takas Döngüleri (*Top Trading Cycles*, TTC) algoritması şöyle çalışır:

1. Kalan her kişi, kalan odalar içinde en sevdiği odanın sahibini gösterir (bir ok çizer). Kendi odasını en çok seviyorsa kendini gösterir.
2. Her kişiden tam bir ok çıktığı için bu çizgede en az bir döngü vardır. Döngüdeki herkes gösterdiği odayı alır ve odasıyla birlikte oyundan çıkar.
3. Kimse kalmayana kadar tekrarlanır.

İkinci adımdaki döngü iddiası basit bir gözleme dayanır. Herhangi bir kişiden başlanıp okları izlenirse sonlu sayıda kişi olduğundan bir noktada daha önce ziyaret edilmiş birine dönülür. Bu dönüş bir döngü oluşturur. Döngü tek kişiden de oluşabilir: kendi odasını gösteren kişi kendi odasında kalır.

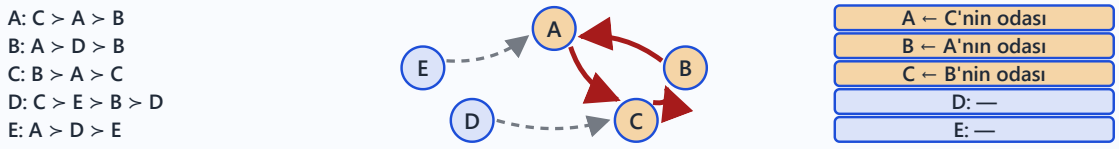
Kişi	Tercih (oda sahibi)
A	C ≻ A ≻ B
B	A ≻ D ≻ B
C	B ≻ A ≻ C
D	C ≻ E ≻ B ≻ D
E	A ≻ D ≻ E

Tablo 4.11: Beş öğrencinin oda tercihleri. Odalar sahiplerinin adıyla anılır; listede olmayan odalar kişinin kendi odasından kötüdür.

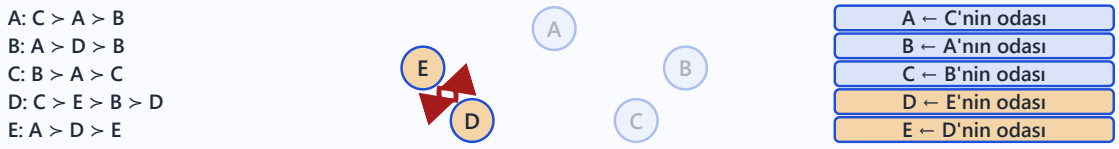
Örnek 4.19

1. **Tur 1, oklar.** Herkes kalanlar içinde en sevdiği odanın sahibini gösterir: A→C, B→A, C→B, D→C, E→A.
2. **Tur 1, döngü.** A'dan başlanıp oklar izlenir: A→C, C→B, B→A. A'ya dönüldü; döngü A → C → B → A'dır.

3. **Tur 1, takas.** Döngüdeki herkes gösterdiği odayı alır: A, C'nin odasını; C, B'nin odasını; B, A'nın odasını. Üçü de ilk tercihini aldı ve odalarıyla birlikte çıkar. D ve E'nin okları (C'ye ve A'ya) döngüde değildi; onlar kalır.
4. **Tur 2, oklar.** Kalanlar D ve E'dir. D'nin ilk tercihi C gitti; sıradaki E'dir: D→E. E'nin ilk tercihi A gitti; sıradaki D'dir: E→D.
5. **Tur 2, döngü ve takas.** D → E → D bir döngüdür. D, E'nin odasını; E, D'nin odasını alır. Kimse kalmadı.
6. **Sonuç.** A ← C'nin odası, B ← A'nın odası, C ← B'nin odası, D ← E'nin odası, E ← D'nin odası. A, B, C üçlü döngüyle; D ve E ikili döngüyle takas etti.



Şekil 4.24: TTC, tur 1. Kalın kırmızı oklar A → C → B → A döngüsüdür; döngüdekiler gösterdikleri odaları alır (sağdaki liste). D ve E'nin kesikli okları döngüde olmadığı için ikisi bir sonraki tura kalır.



Şekil 4.25: TTC, tur 2. Kalan D ve E birbirini gösterir; D → E → D döngüsüyle takas ederler. Sağdaki liste son atamayı gösterir.

TTC'nin üç önemli özelliği vardır:

- **DSIC.** Dürüst tercih bildirmek her kişi için baskın stratejidir (Roth, 1982). Bir kişi, döngüye girdiği turda kalan odalar içinde en sevdiğini alır. Daha önce çıkanların odaları onun bildirimiyle değiştirilemez; çünkü o döngüler kişinin okundan bağımsız oluştu.
- **Pareto optimallik.** Hiçbir başka atama, kimseyi kötüleştirmeden birini daha iyi duruma getiremez. İlk turda çıkanlar en sevdikleri odayı aldı; onları iyileştirmek imkânsızdır. Sonraki turda çıkanlar, kalanlar içinde en sevdiklerini aldı; onları iyileştirmek için daha önce çıkan birinin odasını almak gerekir, bu da o kişiyi kötüleştirir.
- **Çekirdek.** Sonuç **çekirdektedir** (*core*): hiçbir grup, kendi odalarını yalnız aralarında takas ederek her üyesi için en az aynı derecede iyi ve en az bir üyesi için daha iyi bir sonuç elde edemez (Roth ve Postlewaite, 1977). Özellikle her kişi en az kendi odası kadar iyi bir oda alır; IR sağlanır.

Örnekte her kişinin olası bütün yalan tercih listeleri bilgisayarla denendiğinde hiçbirinin daha iyi bir oda getirmediği görülür. Aynı kural nöbet ya da vardiya takasına da uyar. Gerçek dünyada en etkileyici uygulaması böbrek nakli eşleştirmeleridir: uyumsuz bir vericisi olan hastalar, vericilerini döngüler

hâlinde takas eder (Roth, Sönmez ve Ünver, 2004).

> **Sık yapılan hata**
>
> Bir turda çıkan kişilerin odalarını sonraki turlarda okların hedefi olarak bırakmak. Tur 2'de D'nin ilk tercihi C'nin odasıdır, ama C'nin odası tur 1'de A'ya geçti ve oyundan çıktı. D artık kalanlar içinden, yani E'nin ve kendi odası arasından seçer. Oklar her turda yalnız kalanlar arasında yeniden çizilir.

4.4.11 Kod izleme: ttc()

TTC'nin Python sürümü her turda okları bir sözlükte toplar, bir kişiden başlayıp okları izleyerek döngüyü bulur ve döngüdekileri çıkarır. Girdi `pref` sözlüğüdür: `pref[i]`, `i`'nin oda sahiplerini en iyiden kötüye sıralayan listesidir ve kendi adı da listededir.

```
def ttc(pref):
  kalan, atama = list(pref), {}
  while kalan:
    g = {i: next(j for j in pref[i]
             if j in kalan) for i in kalan}
    i, yol = kalan[0], []
    while i not in yol:          # okları izle
      yol.append(i); i = g[i]
    dongu = yol[yol.index(i):]
    for k in dongu:              # takas
      atama[k] = g[k]; kalan.remove(k)
  return atama
```

Kodun parçaları şunlardır:

- `kalan` henüz takas etmemiş kişilerin listesi, `atama` ise sonuç sözlüğüdür: `atama[k]`, `k`'nin aldığı odanın sahibidir.
- `g` sözlüğü bu turun oklarıdır: her kalan kişi `i` için, `pref[i]` listesinde kalanlar arasında bulunan ilk kişi. `next(...)`, koşulu sağlayan ilk elemanı döndürür.
- `i = kalan[0]` ile ilk kalan kişiden başlanır. İç döngü, `i` daha önce yola eklenmemişken onu yola ekler ve okunu izler (`i = g[i]`). Daha önce görülmüş birine ulaşınca durur.
- `dongu = yol[yol.index(i):]`, yolun döngü kısmıdır: tekrar edilen kişinin ilk göründüğü yerden sona kadar. Yolun başındaki kişiler döngüye bağlanan ama döngüde olmayan kişiler olabilir; bu dilimleme onları dışarıda bırakır.
- Döngüdeki her kişi gösterdiği odayı alır ve `kalan`'dan çıkarılır.

Tur	kalan	g (oklar)	yol	dongu	atama (eklenen)
1	A, B, C, D, E	A→C, B→A, C→B, D→C, E→A	[A, C, B], sonra i = A	[A, C, B]	A: C, C: B, B: A

Tur	kalan	g (oklar)	yol	dongu	atama (eklenen)
2	D, E	D→E, E→D	[D, E], sonra i = D	[D, E]	D: E, E: D
son	boş	dönüş: {A: C, C: B, B: A, D: E, E: D}			

Tablo 4.12: `ttc(pref)` izlemesi.

Bu örnekte yol her turda doğrudan döngüyle başladı. Başka bir girdide yol, döngüde olmayan birinden başlayabilir. Örneğin tur 1'de izlemeye D'den başlansaydı yol D → C → B → A → C olurdu; C ikinci kez görülünce durulur ve `yol.index('C')` ile döngü [C, B, A] olarak ayrılır. D döngüde değildir ve kalır. Her turda en az bir kişi çıktığı için en fazla n tur yapılır; her turda okları kurmak en fazla n^2 iş gerektirir. Algoritma bu basit sürümüyle $O(n^3)$ sürede biter.

4.5 Bayesçi oyunlar

Şimdiye kadar özel bilgi tasarımcı ile oyuncular arasındaydı: tasarımcı tipleri bilmiyordu. Oysa oyuncular da çoğu zaman birbirlerinin tipini bilmez. Bir açık artırmada teklifçi rakiplerinin değerlerini, bir pazarlıkta alıcı satıcının maliyetini bilmez. Bu tür eksik bilgili durumları incelemek için oyun kuramı Bayesçi oyunları kullanır. Bayesçi oyunlar BIC kavramının (4.2.4) da dayandığı çerçevedir.

4.5.1 Eksik bilgi ve Bayesçi oyun

Eksik bilgili bir oyunu çözmek ilk bakışta imkânsız görünür: oyuncu, rakibinin hangi oyunu oynadığını bile bilmez. John Harsanyi (1967–68) bu sorunu zarif bir fikirle çözdü. Oyuna, oyuncuların dışında bir "şans" ya da "Doğa" hamlesi eklenir. Oyunun başında Doğa her oyuncunun tipini bir olasılık dağılımına göre seçer. Her oyuncu yalnızca kendi tipini görür. Dağılımın kendisi ise herkesçe bilinir. Böylece "bilinmeyen oyun" sorunu, "bilinen ama içinde şans olan bir oyun" sorununa dönüşür. Bu yönteme **Harsanyi dönüşümü** denir.

Tanım 4.9

Bir **Bayesçi oyun** (*Bayesian game*) şu öğelerden oluşur: oyuncular; her oyuncu için eylem kümesi ve tip kümesi Θ_i; tip profilleri üzerinde herkesçe bilinen bir **ortak ön olasılık** (*common prior*) π; ve her oyuncu için, eylem profili a ile tip profili θ'ya bağlı bir kazanç fonksiyonu $u_i(a, \theta)$.

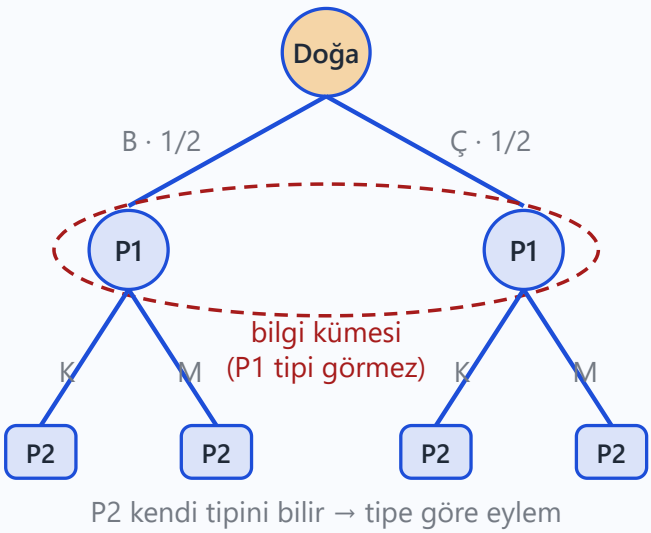
Bayesçi oyunda strateji kavramı değişir. Oyuncu kendi tipini öğrendikten sonra karar verir; ama analizi yapan kişi (ve rakipler) oyuncunun tipini bilmez. Bu yüzden strateji, her tip için hangi eylemin seçileceğini söyleyen bir **kuraldır**. Örneğin iki tipli bir oyuncunun stratejisi "B tipindeysem K, Ç tipindeysem M" gibi bir eşlemedir. Rakip bu kuralı bilir, ama hangi satırın uygulanacağını bilmez.

Tanım 4.10

Her oyuncu, her tipinde, diğerlerinin stratejilerine karşı beklenen kazancını maksimize ediyorsa strateji profili bir **Bayes–Nash dengesidir** (*Bayes–Nash equilibrium*). Beklenen kazanç, oyuncunun bilmediği tipler üzerinden ortak ön olasılıkla alınır.

Tanım, Nash dengesinin doğal bir genişlemesidir. Nash dengesinde "kimse tek başına sapmak istemez" denir. Bayes–Nash dengesinde de aynısı söylenir, yalnızca sapmanın faydası beklenen değerle ölçülür ve her tip ayrı ayrı denetlenir. Bir oyuncunun iki tipi varsa, iki tipin her biri kendi en iyi tepkisini oynamalıdır.

Örnek olarak bir buluşma oyunu ele alınsın. P1 akşam konsere (K) ya da maça (M) gidecektir ve konseri tercih eder. P2 iki tipten biridir: **B tipi** P1 ile buluşmak ister, **Ç tipi** P1'den kaçınmak ister. P1, P2'nin tipini bilmez; iki tipin olasılığı da 1/2'dir. P2 kendi tipini bilir.



Şekil 4.26: Harsanyi dönüşümüyle buluşma oyunu. Önce Doğa P2'nin tipini seçer: B (1/2) ya da Ç (1/2). P1 hangi daldadır bilmez; iki düğümü birleştiren kesikli elips P1'in bilgi kümesidir. P2 kendi tipini bilir ve tipe göre eylem seçer.

Şekil 4.26 oyunun ağacını gösterir. Kesikli elips, P1'in iki düğümü birbirinden ayıramadığını anlatır: P1, hangi tipteki P2 ile oynadığını bilmeden aynı eylemi seçmek zorundadır. Buna P1'in **bilgi kümesi** denir. P2 ise her dalda ayrı karar verir; bu yüzden tipe bağlı bir stratejisi olabilir.

Mekanizma tasarımıyla bağlantıyı açıklama ilkesinin Bayesçi sürümü kurar. Bayes–Nash dengesinde uygulanabilen her sonuç, dürüst bildirimin bir Bayes–Nash dengesi olduğu, yani BIC olan bir doğrudan mekanizmayla da uygulanabilir. İspat 4.2.3'tekiyle aynıdır; tek fark, "sapma kârlı değil" cümlesinin beklenen kazanç için kurulmasıdır.

4.5.2 Bayes–Nash dengesini bulmak

Oyunun kazançları iki matrisle verilir: biri P2 B tipindeyken, öteki Ç tipindeyken. Hücrelerde (P1 kazancı, P2 kazancı) yazılıdır. P1'in kazançları iki matriste aynıdır: P1 her durumda konserde ya da maçta P2 ile buluşmak ister ve konseri tercih eder (K'de buluşma 2, M'de buluşma 1, buluşamama 0). Değişen, P2'nin kazançlarıdır.

P1 \ P2	K	M
K	(2, 1)	(0, 0)
M	(0, 0)	(1, 2)

Tablo 4.13: P2 B tipinde (buluşmak ister), olasılık 1/2.

P1 \ P2	K	M
K	(2, 0)	(0, 2)
M	(0, 1)	(1, 0)

Tablo 4.14: P2 Ç tipinde (kaçınmak ister), olasılık 1/2.

Saf stratejileri saymak, aramanın ne kadar büyük olduğunu gösterir. P1'in iki saf stratejisi vardır: K ya da M. P2'nin her tipi K ya da M seçebildiği için P2'nin $2 \times 2 = 4$ saf stratejisi vardır: (B→K, Ç→K), (B→K, Ç→M), (B→M, Ç→K), (B→M, Ç→M). Toplam $2 \times 4 = 8$ saf strateji profili vardır. Hepsini tek tek denetlemek yerine daha kısa bir yol izlenir: P1'in her seçimi için P2'nin her tipinin en iyi tepkisi bulunur, sonra P1'in bu tepkilere karşı sapıp sapmadığına bakılır.

Örnek 4.20

1. **Aday 1: P1 = K.** P2'nin en iyi tepkisi tipe göre değişir. B tipi, P1 K'deyken K'de 1, M'de 0 alır; K seçer. Ç tipi, P1 K'deyken K'de 0, M'de 2 alır; M seçer. P2'nin stratejisi s_2 = (B→K, Ç→M) olur.
2. **P1 sapar mı?** P2 bu stratejiyi oynarken P1'in beklenen kazancı: K seçerse yarı olasılıkla B tipiyle K'de buluşur (2), yarı olasılıkla Ç tipi M'ye kaçar (0): $½ \cdot 2 + ½ \cdot 0 = 1$. M seçerse B tipi K'de olduğu için buluşamaz (0), Ç tipi M'de olduğu için buluşur (1): $½ \cdot 0 + ½ \cdot 1 = 0{,}5$. $1 > 0{,}5$; P1 için K en iyisidir.
3. **Sonuç.** (K; B→K, Ç→M) bir Bayes–Nash dengesidir.
4. **Aday 2: P1 = M.** B tipi, P1 M'deyken K'de 0, M'de 2 alır; M seçer. Ç tipi, P1 M'deyken K'de 1, M'de 0 alır; K seçer. s_2 = (B→M, Ç→K).
5. **P1 sapar mı?** K seçerse B tipi M'de (0), Ç tipi K'de (2): $½ \cdot 0 + ½ \cdot 2 = 1$. M seçerse B tipi M'de (1), Ç tipi K'de (0): $½ \cdot 1 + ½ \cdot 0 = 0{,}5$. P1 için K daha iyidir; M'den K'ye sapar. Denge değildir.

6. **Kalan profiller.** P1'in bir seçimi sabitken P2'nin her tipinin en iyi tepkisi tektir (eşitlik yok). Bu yüzden P2'nin başka bir stratejisi en az bir tipte en iyi tepki olmaz ve o profiller denge olamaz. 8 profilin hepsi denendiğinde tek saf Bayes–Nash dengesinin (K; B→K, Ç→M) olduğu görülür.

Aday profil	P1: K beklenen	P1: M beklenen	Denge?
P1 = K; B→K, Ç→M	½·2 + ½·0 = 1	½·0 + ½·1 = 0,5	✓ evet
P1 = M; B→M, Ç→K	½·0 + ½·2 = 1	½·1 + ½·0 = 0,5	✗ K'ye sapar

Tablo 4.15: Aday profiller ve P1'in beklenen kazançları (P(B) = 1/2).

Yorum ilginçtir. P1 M'ye giderse B tipi onu izler ve Ç tipi konsere kaçar. Ama Ç tipinin kaçtığı konser, P1'in daha çok sevdiği yerdir. P1, buluşma şansı aynıyken (iki adayda da yarı yarıya) konseri tercih ettiği için K'de kalmak ister. Dengede P2'nin davranışı tipini ele verir: konserde görülen P2 B tipindedir, maçta görülen Ç tipindedir.

Sık yapılan hata

P2'nin stratejisini tek bir eylem sanmak. "P2 K oynar" bir Bayesçi strateji değildir; her tipin ne yapacağı ayrı ayrı söylenmelidir. Ayrıca P1'in beklenen kazancı hesaplanırken her tipin *kendi* seçimi kullanılmalıdır: P1 = K adayında B tipi K'de, Ç tipi M'dedir; iki tipi aynı yere koymak yanlış sonuç verir.

4.5.3 Olasılık değişince denge

Ön olasılık değişirse denge de değişebilir. P2'nin B tipinde olma olasılığı genel bir π sayısı olsun; Ç tipinin olasılığı $1 - \pi$'dir. P2'nin en iyi tepkileri π'ye bağlı değildir, çünkü P2 kendi tipini bilir. Değişen yalnızca P1'in beklenen kazançlarıdır.

- **Aday 1 (P1 = K; B→K, Ç→M).** P1 K'de kalırsa $\pi \cdot 2 + (1-\pi) \cdot 0 = 2\pi$, M'ye geçerse $\pi \cdot 0 + (1-\pi) \cdot 1 = 1 - \pi$ alır. Denge için $2\pi \geq 1 - \pi$, yani $\pi \geq 1/3$ gerekir.
- **Aday 2 (P1 = M; B→M, Ç→K).** P1 M'de kalırsa $\pi \cdot 1 = \pi$, K'ye geçerse $(1-\pi) \cdot 2$ alır. Denge için $\pi \geq 2(1-\pi)$, yani $\pi \geq 2/3$ gerekir.

π aralığı	Aday 1	Aday 2	Saf denge sayısı
π < 1/3	✗	✗	0 (karışık denge gerekir)
1/3 ≤ π < 2/3	✓	✗	1
π ≥ 2/3	✓	✓	2

Tablo 4.16: B tipinin olasılığı π'ye göre saf Bayes–Nash dengeleri.

$\pi = 1/2$, ikinci satıra düşer; bu yüzden örnekte tek bir saf denge bulundu. B tipi çok az olasıysa ($\pi < 1/3$), P1 nereye giderse gitsin P2 büyük olasılıkla Ç tipindedir ve ondan kaçar ve P1 her seferinde öbür yere geçmek ister. Bu durumda saf denge yoktur; Nash'in varlık teoremi gereği karışık stratejili bir denge vardır. B tipi çok olasıysa ($\pi \geq 2/3$) oyun neredeyse bir koordinasyon oyununa döner ve iki saf denge ortaya çıkar.

Eşik değerleri $\pi = 1/3$ ve $\pi = 2/3$'te P1 iki eylemi arasında kayıtsızdır. Örneğin $\pi = 1/3$'te aday 1'de K de M de $2/3$ verir. Eşitlik durumunda P1'in K'de kalması en iyi tepkidir; bu yüzden eşitsizlikler "≥" ile yazılmıştır.

4.5.4 Bayesçi bakışla mekanizmalar

Bayesçi oyunlar mekanizma tasarımında DSIC'in yetmediği ya da fazla katı kaldığı yerlerde kullanılır. Dört örnek bu bakışın neden gerekli olduğunu gösterir.

Birinci fiyat açık artırması. Birinci fiyatta dürüst teklif, kazanınca faydayı 0 bırakır (4.2.4). Bu yüzden dengede herkes değerinin altında teklif verir. Ne kadar altında? Cevap rakiplerin değerleri hakkındaki inanca bağlıdır; denge bu yüzden Bayesçidir. Standart bir örnekte iki teklifçinin değerleri 0 ile 1 arasında tekdüze (her değer eşit olasılıkla) dağılır. Rakip değerinin yarısını teklif ederse, değeri v olan teklifçinin b teklifiyle kazanma olasılığı, rakibin değerinin $2b$'den küçük olma olasılığıdır, yani $2b$'dir ($b \leq 1/2$ için). Beklenen fayda $(v - b) \cdot 2b$'dir. Bu ifade $b = v/2$'de en büyük olur. Yani "değerinin yarısını teklif et" stratejisi iki oyuncu için birbirine en iyi tepkidir: bir Bayes–Nash dengesidir.

Örnek 4.21

Açıklama ilkesi birinci fiyat dengesinden bir doğrudan mekanizma kurar: "Değerini bildir; en yüksek bildirim kazanır ve bildiriminin yarısını öder." Değerler 0 ile 1 arasında tekdüzeyse bu mekanizma BIC'tir. DSIC olmadığını bir örnek gösterir.

1. Teklifçinin değeri 0,8 olsun ve rakip 0,9 bildirsin. Dürüst bildirim (0,8) kaybeder; fayda 0'dır.
2. Teklifçi 0,95 bildirirse kazanır ve $0{,}95/2 = 0{,}475$ öder. Faydası $0{,}8 - 0{,}475 = 0{,}325 > 0$'dır.
3. Bu rakip bildiriminde yalan kazandırdı. Dürüstlük, rakibin bildirimi ne olursa olsun en iyi değildir; mekanizma DSIC değildir. Ama rakip dürüstken ve değerler tekdüzeyken, ortalamada dürüstlük en iyisidir.

Bu örnek, DSIC ⇒ BIC ilişkisinin tersinin neden genelde doğru olmadığını gösterir.

DSIC mi, BIC mi? DSIC, oyuncuların inançlarından bağımsızdır ve sağlamdır: tip dağılımı yanlış tahmin edilse bile dürüstlük en iyisi kalır. BIC daha zayıftır, ama daha fazla hedefe izin verir. Örneğin VCG kamu malı probleminde tam bütçe dengesini sağlayamaz; oysa BIC düzeyinde hem verimli hem tam

bütçe dengeli bir kamu malı mekanizması vardır (d'Aspremont ve Gérard-Varet, 1979; Arrow, 1979). Bu mekanizmada ödemeler, her oyuncunun bildiriminin diğerlerine *beklenen* etkisine göre ayarlanır ve toplanan para oyunculara geri dağıtılır.

En yüksek gelir. Satıcının beklenen gelirini maksimize eden açık artırma da Bayesçi bir sorudur; çünkü "beklenen gelir" alıcıların değer dağılımına bağlıdır. Myerson (1981) bu soruyu çözdü. Standart koşullarda cevap, **rezerv fiyatlı** bir açık artırmadır: satıcı bir en düşük fiyat belirler; bu fiyatın altında satış yapılmaz. Rezerv fiyat bazen hiç satış olmamasına yol açar, yani verimliliği bozar; ama karşılığında beklenen geliri artırır. Verimlilik ile gelir arasındaki bu ödünleşim, açık artırma tasarımının temel konularından biridir.

Poker ve blöf. Pokerde rakibin eli (tipi) gizlidir. Blöf, yani zayıf elle büyük bahis, karışık bir Bayesçi stratejinin parçasıdır: güçlü eller hep büyük bahisle oynanırsa bahis büyüklüğü eli ele verir.

4.6 Değerlendirme

Mekanizma tasarımının varsayımları ve sınırları, bölümdeki mekanizmaların karşılaştırması ve birkaç kısa problem aşağıda bir araya getirilmiştir.

4.6.1 Davranış, adalet ve etik

İnsanlar her zaman teşviklere göre mi davranır? Kuram, oyuncuların faydalarını maksimize ettiğini varsayar. Gerçekte duygular, ahlak, alışkanlık ve adalet duygusu da kararları etkiler. Ültimatom oyununda düşük tekliflerin reddedilmesi (4.2.5) bunun en bilinen örneğidir. Teşvikler güçlüdür ama tek belirleyici değildir. Bu gözlem mekanizma tasarımını geçersiz kılmaz, yalnızca iki uyarı getirir. Birincisi, fayda fonksiyonu insanların gerçekten neyi önemsediğini yansıtmalıdır. İkincisi, kurallar basit ve anlaşılır olmalıdır. Örneğin İngiliz açık artırması Vickrey ile aynı sonucu verir, ama dürüst davranışın neden en iyisi olduğu orada çok daha kolay görülür.

Mekanizma tasarımında etik sorunlar. İyi kurallar bile kötü sonuçlara yol açabilir:

- **Manipülasyon ve bilgi saklama.** DSIC olmayan kurallarda bilgiye ve danışmanlığa erişimi olanlar avantaj sağlar. Boston mekanizmasında stratejik liste yazmayı bilen aileler, tercihlerini olduğu gibi yazan ailelerin önüne geçebilir.
- **Güçlü oyuncuların avantajı.** Danışıklılık ve sahte kimlikler, organize gruplara kuralın öngörmediği güç verir.
- **Adil olmayan gelir dağılımı.** Verimlilik, değerlerin toplamını en büyük yapar; ama değer, ödeme gücüne bağlıdır. Zengin bir oyuncunun "yüksek değeri" çoğu zaman yalnızca daha çok parası olduğunu gösterir.
- **Ödeme gücü olmayanların dışlanması.** Para kullanan mekanizmalar, ödeme yapamayanları baştan dışarıda bırakır. Okul yerleştirme ve organ nakli gibi alanlarda paranın kullanılmamasının nedeni budur.

Bu yüzden etik sınırlar toplum ve kurumlarca belirlenmelidir. İyi bir mekanizma yalnızca verimli değil, adil ve etik de olmalıdır. Mekanizma tasarımı bu tartışmada bir araçtır: hangi hedefin hangi kurallarla ve hangi bedelle sağlanabileceğini açıkça gösterir; hangi hedefin seçileceğine ise toplum karar verir.

4.6.2 Bölümün mekanizmaları bir arada

Bölümde görülen mekanizmalar Tablo 4.17'de karşılaştırılır. Tablo iki ayrımı birlikte gösterir: paranın kullanılıp kullanılmadığı ve teşvik özelliğinin hangi düzeyde sağlandığı.

Mekanizma	Para	Teşvik özelliği	Ne sağlar?
Orantılı maliyet paylaşımı	var	IC değil (beleşçilik)	—
VCG (Clarke pivotu)	var	DSIC, IR	Verimli sonuç; bütçe fazlası olabilir
Vickrey (tek nesne VCG)	var	DSIC, IR	En yüksek değerli kazanır
Gale–Shapley	yok	Teklif verenler için DSIC	Kararlı eşleştirme, $O(n^2)$
Medyan (tesis yerleşimi)	yok	DSIC (stratejiye dayanıklı)	Toplam uzaklık en az
Seri diktatörlük	yok	DSIC	Pareto optimal atama
Boston (doğrudan kabul)	yok	DSIC değil	İlk tercihlere öncelik; stratejik liste ödüllenir
TTC (ev takası)	yok	DSIC	Pareto optimal, çekirdekte
Bayesçi mekanizmalar	var	BIC (Bayes–Nash)	Beklenen hedefler (ör. gelir)

Tablo 4.17: Bu bölümün mekanizmaları.

Tablonun ana dersi şudur: açıklama ilkesi sayesinde doğrudan ve dürüst bildirimli mekanizmalara odaklanmak yeterlidir; ama IC, IR, verimlilik ve bütçe dengesi her zaman birlikte sağlanamaz. Para varsa VCG verimliliği ve DSIC'i sağlar, karşılığında bütçe dengesinden vazgeçer. Para yoksa Gibbard–Satterthwaite genel bir çözüme izin vermez; çözüm, tercihlerin yapısından (medyan) ya da hedefin değiştirilmesinden (kararlılık, takas) gelir.

4.6.3 Örnek problemler

Aşağıdaki beş problem, bölümün hesap yöntemlerini kısa durumlarda bir araya getirir.

Örnek 4.22

İki özdeş nesne, üç teklifçi. Her teklifçi en fazla bir nesne ister; değerler 10, 8 ve 5'tir. VCG'de kim kazanır, kim ne öder?

1. **Verimli sonuç.** İki nesne en yüksek iki değere verilir: 10 ve 8 kazanır; refah 18'dir.
2. **10'luk teklifçinin ödemesi.** O olmasaydı nesneler 8 ve 5'e giderdi: diğerlerinin en iyi refahı $8 + 5 = 13$. O varken diğerlerinden yalnız 8'lik nesne alır: refah 8. Ödeme $13 - 8 = 5$.

3. **8'lik teklifçinin ödemesi.** O olmasaydı diğerlerinin refahı $10 + 5 = 15$; o varken 10. Ödeme $15 - 10 = 5$.
4. **5'lik teklifçi.** O olmasa da sonuç aynıdır; ödemesi 0'dır.

Her kazanan, kaybedenlerin en yüksek değerini (5) öder. Bu, tek nesnedeki ikinci fiyat kuralının birden çok özdeş nesneye genellemesidir.

Örnek 4.23

Bora'nın en kârlı bildirimi. Orantılı maliyet paylaşımı örneğinde (park 60; değerler 30, 25, 20) Ayşe ve Cem dürüstse Bora ne bildirmelidir?

1. Ayşe ve Cem'in bildirimleri toplamı $30 + 20 = 50$'dir. Parkın yapılması için Bora en az 10 bildirmelidir.
2. Bora 10 bildirirse toplam tam 60 olur ve Bora $60 \cdot 10/60 = 10$ öder. Faydası $25 - 10 = 15$'tir.
3. Dürüstken faydası 5'ti. Yalan 10 birim kazandırır; kural IC değildir.

Örnek 4.24

Köyler 3, 7, 8, 12, 30. Medyan ve ortalama nedir? Ortalama kuralında 30. km'deki köy tesisi tam kendine çekmek için ne bildirmelidir?

1. Sıralı listede ortadaki sayı 8'dir: medyan 8.
2. Ortalama $(3 + 7 + 8 + 12 + 30)/5 = 60/5 = 12$.
3. Ortalamanın 30 olması için $(3 + 7 + 8 + 12 + x)/5 = 30$ gerekir: $30 + x = 150$, $x = 120$. Köy 120 bildirerek tesisi kendi konumuna çeker. Medyanda ise 120 bildirimi medyanı 8'de bırakır.

Örnek 4.25

B tipinin olasılığı 1/4 olsaydı (K; B→K, Ç→M) yine Bayes–Nash dengesi olur muydu?

1. P2'nin stratejisi değişmez: B tipi K'ye, Ç tipi M'ye gider.
2. P1 K'de kalırsa $¼ \cdot 2 + ¾ \cdot 0 = 0{,}5$; M'ye geçerse $¼ \cdot 0 + ¾ \cdot 1 = 0{,}75$ alır.
3. $0{,}75 > 0{,}5$: P1 M'ye sapar. Profil denge değildir. $1/4 < 1/3$ olduğu için Tablo 4.16 gereği saf denge yoktur; karışık denge gerekir.

Örnek 4.26

Üç kişilik TTC. Tercihler A: B ≻ C ≻ A, B: A ≻ C ≻ B, C: A ≻ B ≻ C. Kim hangi odayı alır?

1. Tur 1 okları: A→B, B→A, C→A. A ile B birbirini gösterir: A ↔ B bir döngüdür. A, B'nin odasını; B, A'nın odasını alır ve çıkar.
2. Tur 2: yalnız C kalır. A ve B'nin odaları gittiği için C kendini gösterir ve kendi odasında kalır.

C en çok A'nın odasını isterdi, ama A, C'nin odasını B'ninkine tercih etmez. C'nin odası kimsenin ilk tercihi değildir; C takas edecek kimse bulamaz.

4.7 Özet

- **Mekanizma tasarımı**, oyun kuramının ters yönüdür: istenen sonuç verilir, bu sonucu dengede doğuracak kurallar (tahsis ve ödeme) aranır. Zorluk, oyuncuların tiplerinin (değer, tercih, maliyet) tasarımcıdan gizli olmasıdır.
- **Sosyal seçim fonksiyonu** f, her tip profiline istenen sonucu atar. **Doğrudan mekanizmada** oyuncular tiplerini bildirir; mekanizma sonucu $x(b)$ ve ödemeleri $p(b)$ hesaplar; fayda yarı doğrusaldır: $u_i = v_i(x) - p_i$.
- **Açıklama ilkesi**: dolaylı bir mekanizmada dengede uygulanan sonuç, dürüst bildirimin denge olduğu bir doğrudan mekanizmayla da uygulanabilir. Denge kavramı baskın stratejiyse DSIC, Bayes–Nash ise BIC elde edilir; DSIC ⇒ BIC.
- **IR**, katılımın dışarıda kalmaktan kötü olmamasıdır. IC ile IR farklı koşullardır. Ültimatom oyununda kuram (10, 0) ve (9, 1) dengelerini verir; deneylerde ise düşük teklifler reddedilir: fayda modeli adaleti de içermelidir.
- **Orantılı maliyet paylaşımı** IC değildir: park örneğinde Cem 5 bildirerek faydasını 4'ten 15'e çıkarır; herkes yarısını bildirirse park yapılmaz. **Myerson–Satterthwaite**: iki taraflı ticarette verimlilik, BIC, IR ve bütçe dengesi birlikte sağlanamaz.
- **Uygulama kuramı**: dürüstlük tek denge olmayabilir; Vickrey'de (0, 100) teklifleri, nesneyi değeri düşük olana veren bir Nash dengesidir.
- **Myerson lemması**: tek parametreli ortamda tahsis DSIC'te kullanılabilir ⇔ monoton. Ödeme tek biçimde belirlenir: $p = b \cdot x(b) - \int_0^b x(z)\,dz$; 0/1 tahsiste kritik teklif.
- **VCG** sosyal refahı en yüksek sonucu seçer ve her oyuncuya dışsallığını ödetir: p_i = (i olmasaydı diğerlerinin en iyi refahı) − (i varken diğerlerinin refahı). DSIC, verimli ve (Clarke pivotuyla) IR'dir; tam bütçe dengesi yoktur. Kamu projesi örneğinde Spor salonu seçilir, ödemeler 0, 5, 20'dir. Tek nesnede VCG = Vickrey.
- **Parasız tasarımda** Gibbard–Satterthwaite genel tercihlerde yalnızca diktatörlüğe izin verir. **Gale–Shapley** her zaman kararlı eşleştirme bulur ($O(n^2)$), teklif verenler için en iyi ve DSIC'tir; teklif alan taraf yalanla kazanabilir.
- **Medyan** tek tepeli tercihlerde stratejiye dayanıklıdır ve toplam uzaklığı en aza indirir; **ortalama** manipüle edilebilir. **Boston** mekanizması DSIC değildir; **seri diktatörlük** ve **TTC** DSIC'tir. TTC Pareto optimal ve çekirdektedir.

- **Bayesçi oyunda** Doğa tipleri ortak ön olasılıkla seçer; strateji tipten eyleme bir kuraldır; **Bayes–Nash dengesinde** her tip beklenen kazancını maksimize eder. Buluşma oyununda *P(B) = 1/2* için tek saf denge (K; B→K, Ç→M)'dir.

4.8 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

mekanizma tasarımı *mechanism design* · tip *type* · sosyal seçim fonksiyonu *social choice function* · doğrudan mekanizma *direct mechanism* · yarı doğrusal fayda *quasi-linear utility* · açıklama ilkesi *revelation principle* · teşvik uyumluluğu *incentive compatibility* · baskın stratejide teşvik uyumlu (DSIC) *dominant-strategy IC* · Bayesçi teşvik uyumlu (BIC) *Bayesian IC* · bireysel rasyonellik *individual rationality* · beleşçi *free-rider* · sosyal refah *social welfare* · bütçe dengesi *budget balance* · Myerson–Satterthwaite teoremi *Myerson–Satterthwaite theorem* · uygulama kuramı *implementation theory* · Myerson lemması *Myerson's lemma* · kritik teklif *critical bid* · VCG mekanizması *VCG mechanism* · Clarke pivot kuralı *Clarke pivot rule* · dışsallık *externality* · Vickrey açık artırması *Vickrey auction* · stratejiye dayanıklı *strategy-proof* · Gibbard–Satterthwaite teoremi *Gibbard–Satterthwaite theorem* · kararlı eşleştirme *stable matching* · engelleyici çift *blocking pair* · Gale–Shapley (ertelenmiş kabul) algoritması *deferred acceptance* · tek tepeli tercihler *single-peaked preferences* · medyan mekanizması *median mechanism* · seri diktatörlük *serial dictatorship* · Boston mekanizması *Boston mechanism* · En İyi Takas Döngüleri (TTC) *Top Trading Cycles* · çekirdek *core* · Pareto optimal *Pareto optimal* · Bayesçi oyun *Bayesian game* · ortak ön olasılık *common prior* · Bayes–Nash dengesi *Bayes–Nash equilibrium*

4.9 Kendini Dene

1. Bir oyun alanının maliyeti 40 bin TL'dir. Değerler Ayşe 20, Bora 16, Cem 14'tür. Kural orantılı maliyet paylaşımıdır: bildirimlerin toplamı en az 40 ise alan yapılır ve maliyet bildirimlerle orantılı paylaşılır. Herkes dürüstken ödemeler ve faydalar nedir? Bora ve Cem dürüstken Ayşe'nin en kârlı

bildirimi nedir?

Cevap

Toplam $50 \geq 40$. Ödemeler $40 \cdot 20/50 = 16$, $40 \cdot 16/50 = 12{,}8$, $40 \cdot 14/50 = 11{,}2$; faydalar 4; 3,2; 2,8 (IR sağlanır). Bora ve Cem'in toplamı 30'dur; alanın yapılması için Ayşe en az 10 bildirmelidir. Ayşe 10 bildirirse toplam 40 olur ve $40 \cdot 10/40 = 10$ öder; faydası $20 - 10 = 10 > 4$. Kural IC değildir.

2. Üç sakin, üç projeye (Park, Kütüphane, Spor salonu) şu değerleri bildiriyor: Ayşe (20, 5, 10), Bora (0, 25, 15), Cem (15, 10, 30). VCG hangi projeyi seçer? Her sakinin ödemesi ve faydası nedir?

Cevap

Toplamlar Park 35, Kütüphane 40, Spor 55: Spor salonu seçilir. Ayşe olmadan toplamlar 15, 35, 45; en iyisi 45, Spor'da diğerleri 45 alır: ödeme 0. Bora olmadan 35, 15, 40; en iyisi 40, Spor'da 40: ödeme 0. Cem olmadan 20, 30, 25; en iyisi Kütüphane (30), Spor'da diğerleri $10 + 15 = 25$ alır: ödeme $30 - 25 = 5$. Faydalar Ayşe $10 - 0 = 10$, Bora $15 - 0 = 15$, Cem $30 - 5 = 25$. Bütçe fazlası 5'tir.

3. Tek nesne için dört teklifçinin değerleri 40, 65, 50 ve 30'dur. Vickrey açık artırmasında kim kazanır, kaç öder? 65 ve 50 değerli teklifçiler anlaşır ve 50'lik teklifçi 0 teklif ederse sonuç nasıl değişir?

Cevap

Dürüst tekliflerde 65 kazanır ve ikinci en yüksek teklif olan 50'yi öder; faydası 15'tir. Anlaşmada teklifler 40, 65, 0, 30 olur. 65 yine kazanır ama ikinci en yüksek teklif 40'tır; 40 öder ve faydası 25'e çıkar. Grup 10 birim kazanır, satıcının geliri 50'den 40'a düşer. Tek başına sapma kazandırmasa da danışıklılık kazandırır.

4. Tek parametreli bir ortamda, diğerlerinin teklifleri sabitken bir oyuncunun tahsisi şöyledir: teklif 3'ün altında 0, 3 ile 6 arasında 1/4, 6 ve üstünde 1. Myerson ödemesine göre $p(4)$ ve $p(8)$ nedir? Değeri 8 olan oyuncu dürüst teklif verirse faydası kaçtır?

Cevap

Tahsis monotondur. $\int_0^4 x = ¼ \cdot 1 = 0{,}25$, $p(4) = 4 \cdot ¼ - 0{,}25 = 0{,}75$ (kontrol: sıçrama × eşik = $¼ \cdot 3$). $\int_0^8 x = ¼ \cdot 3 + 1 \cdot 2 = 2{,}75$, $p(8) = 8 - 2{,}75 = 5{,}25$ (kontrol: $¼ \cdot 3 + ¾ \cdot 6 = 0{,}75 + 4{,}5 = 5{,}25$). Fayda $8 - 5{,}25 = 2{,}75$, yani eğrinin altındaki alandır.

5. Öğrenci tercihleri A: Y ≻ X ≻ Z, B: X ≻ Y ≻ Z, C: Y ≻ Z ≻ X; proje tercihleri X: A ≻ B ≻ C, Y: B ≻ C ≻ A, Z: C ≻ A ≻ B. Öğrenciler teklif verdiğinde Gale–Shapley kaç teklif yapar ve sonuç nedir? Projeler

teklif verseydi?

Cevap

Öğrenciler teklif verir: A→Y, B→X, C→Y; Y, C'yi tutar ve A'yı reddeder (C ≻ A). A→X; X, A'yı tutar ve B'yi reddeder. B→Y; Y, B'yi tutar ve C'yi reddeder. C→Z: kabul. 6 teklifle sonuç A–X, B–Y, C–Z. Projeler teklif verirse X→A, Y→B, Z→C ve üçü de kabul edilir: 3 teklifle aynı sonuç. Bu eşleştirme tek kararlı eşleştirmedir. Her öğrenci ikinci tercihini, her proje ilk tercihini almıştır.

6. Beş köy 1, 4, 6, 11 ve 18. km'dedir. Ortalama ve medyan kuralında tesis nereye kurulur ve toplam uzaklık kaçtır? Ortalama kuralında 18. km'deki köy tesisi tam kendine çekmek için ne bildirmelidir?

Cevap

Medyan 6: toplam $5 + 2 + 0 + 5 + 12 = 24$. Ortalama $40/5 = 8$: toplam $7 + 4 + 2 + 3 + 10 = 26$. Ortalamanın 18 olması için $(1 + 4 + 6 + 11 + x)/5 = 18$, yani $x = 90 - 22 = 68$. Medyanda 68 bildirimi medyanı 6'da bırakır.

7. Dört öğrencinin oda tercihleri: A: B ≻ C ≻ A, B: C ≻ A ≻ B, C: B ≻ D ≻ C, D: A ≻ B ≻ D. TTC uygulanınca kim hangi odayı alır?

Cevap

Tur 1 okları: A→B, B→C, C→B, D→A. A'dan başlanınca yol A → B → C → B olur; döngü B ↔ C'dir. B, C'nin; C, B'nin odasını alır. Tur 2: A'nın ilk iki tercihi gitti, A kendini gösterir; D'nin ilk iki tercihinden A kalmıştır, D→A. Döngü A → A: A kendi odasında kalır. Tur 3: D kendini gösterir ve kendi odasında kalır. Sonuç B ← C, C ← B, A ← A, D ← D.

8. Buluşma oyununda B tipinin olasılığı 2/5 olsun. (K; B→K, Ç→M) ve (M; B→M, Ç→K) Bayes–Nash dengesi midir?

Cevap

Birinci aday: K → $2/5 \cdot 2 = 4/5$, M → $3/5 \cdot 1 = 3/5$; K daha iyi, denge ✓. İkinci aday: K → $3/5 \cdot 2 = 6/5$, M → $2/5 \cdot 1 = 2/5$; P1 K'ye sapar, denge değil. $1/3 \le 2/5 < 2/3$ olduğu için tek saf denge vardır.

4.10 İleri okuma

- T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory* (2016), Ders 2–4, 7 ve 10: Myerson lemmasının tam ispatını, VCG'yi, böbrek takasını ve kararlı eşleştirmeyi bilgisayar bilimi bakışıyla kısa derslerle anlatır.
- N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory* (2007), Bölüm 9 ve 10: paralı mekanizma tasarımını ve para olmadan tasarımı (medyan, ev takası, kararlı eşleştirme) ayrıntılı ve ispatlarıyla sunar.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 10 ve 15: eşleştirme piyasalarını ve VCG'nin sponsorlu arama reklamlarıyla ilişkisini az matematikle, sezgisel örneklerle açıklar.
- A. E. Roth, *Who Gets What — and Why* (2015): okul yerleştirme, tıp ihtisası ve böbrek takası gibi gerçek piyasaların nasıl tasarlandığını, bu tasarımları yapan araştırmacının kendi anlatımıyla aktarır.

4.11 Kaynaklar

1. A. Abdulkadiroğlu, T. Sönmez, “School choice: A mechanism design approach”, *American Economic Review* 93(3), 2003.
2. K. J. Arrow, “The property rights doctrine and demand revelation under incomplete information”, M. Boskin (ed.), *Economics and Human Welfare*, Academic Press, 1979.
3. D. Black, “On the rationale of group decision-making”, *Journal of Political Economy* 56(1), 1948.
4. C. F. Camerer, *Behavioral Game Theory*, Princeton University Press, 2003.
5. E. H. Clarke, “Multipart pricing of public goods”, *Public Choice* 11, 1971.
6. C. d'Aspremont, L.-A. Gérard-Varet, “Incentives and incomplete information”, *Journal of Public Economics* 11(1), 1979.
7. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge, 2010 (Bölüm 10 ve 15).
8. D. Gale, L. S. Shapley, “College admissions and the stability of marriage”, *American Mathematical Monthly* 69(1), 1962.
9. A. Gibbard, “Manipulation of voting schemes: A general result”, *Econometrica* 41(4), 1973.
10. T. Groves, “Incentives in teams”, *Econometrica* 41(4), 1973.
11. W. Güth, R. Schmittberger, B. Schwarze, “An experimental analysis of ultimatum bargaining”, *Journal of Economic Behavior and Organization* 3(4), 1982.
12. J. C. Harsanyi, “Games with incomplete information played by ‘Bayesian’ players, I–III”, *Management Science* 14, 1967–68.
13. A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017 (mekanizma tasarımı, kararlı eşleştirme).
14. E. Maskin, “Nash equilibrium and welfare optimality”, *Review of Economic Studies* 66(1), 1999.
15. H. Moulin, “On strategy-proofness and single peakedness”, *Public Choice* 35(4), 1980.

16. R. B. Myerson, “Optimal auction design”, *Mathematics of Operations Research* 6(1), 1981.

17. R. B. Myerson, M. A. Satterthwaite, “Efficient mechanisms for bilateral trading”, *Journal of Economic Theory* 29(2), 1983.

18. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007.

19. M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2004 (Bayesçi oyunlar).

20. A. E. Roth, “Incentive compatibility in a market with indivisible goods”, *Economics Letters* 9(2), 1982.

21. A. E. Roth, “The economics of matching: Stability and incentives”, *Mathematics of Operations Research* 7(4), 1982.

22. A. E. Roth, *Who Gets What — and Why*, Houghton Mifflin Harcourt, 2015.

23. A. E. Roth, A. Postlewaite, “Weak versus strong domination in a market with indivisible goods”, *Journal of Mathematical Economics* 4(2), 1977.

24. A. E. Roth, T. Sönmez, M. U. Ünver, “Kidney exchange”, *Quarterly Journal of Economics* 119(2), 2004.

25. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press, 2016.

26. M. A. Satterthwaite, “Strategy-proofness and Arrow's conditions”, *Journal of Economic Theory* 10(2), 1975.

27. L. Shapley, H. Scarf, “On cores and indivisibility”, *Journal of Mathematical Economics* 1(1), 1974.

28. Y. Shoham, K. Leyton-Brown, *Multiagent Systems*, Cambridge University Press, 2009.

29. W. Vickrey, “Counterspeculation, auctions, and competitive sealed tenders”, *Journal of Finance* 16(1), 1961.

KISIM II · BÖLÜM 5

Açık Artırmalar

Bir satıcı elindeki nesneyi kime ve kaça satacağını bilmez, çünkü alıcıların nesneye ne kadar değer verdiğini göremez. Açık artırma bu sorunu kurallarla çözer: alıcılar teklif verir, kurallar kazananı ve ödemeyi belirler. Ancak kurallar değişince akıllı bir teklifçinin davranışı da değişir. Not, dört temel açık artırma türünü küçük bir sayısal örnek üzerinde çalıştırarak başlar. Ardından ikinci fiyat kuralında dürüst teklifin neden en iyi olduğu ispatlanır, birinci fiyat kuralında teklifin ne kadar düşürüleceği türetilir ve farklı kuralların satıcıya ortalamada neden aynı geliri getirdiği gösterilir. Son konular rezerv fiyatı, kazananın laneti ve çift taraflı piyasalardır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum05`

ÖĞRENME ÇIKTILARI

Türleri çalıştırmak
İngiliz, Hollanda, birinci fiyat ve ikinci fiyat açık artırmalarında kazananı ve ödemeyi bulmak.

Dürüstlüğü ispatlamak
Vickrey açık artırmasında dürüst teklifin zayıf baskın strateji olduğunu durum analiziyle göstermek.

Denge teklifi
Birinci fiyatta $b = (n-1)v/n$ kırpmasını türetmek; gelir eşdeğerliğini (2 teklifçi → 1/3) ve rezerv fiyatının etkisini (5/12) hesaplamak.

Piyasa olguları
Kazananın lanetini açıklamak; çift taraflı açık artırmada eşleştirme ve fiyatı bulmak.

5.1 Temel kavramlar

Açık artırma (*auction*), bir ya da birden çok nesnenin, katılımcıların verdiği tekliflere göre önceden ilan edilmiş kurallarla dağıtıldığı bir satış yöntemidir. Bu bölümde nesneyi satın almak isteyen katılımcılara **teklifçi** (*bidder*) denir. Kurallar iki soruyu cevaplar: nesneyi kim alır ve ne öder? Bu iki soru, mekanizma tasarımındaki **tahsis kuralı** (*allocation rule*) ile **ödeme kuralı** (*payment rule*) kavramlarının ta kendisidir. Açık artırmalar bu yüzden mekanizma tasarımının en yaygın ve en eski uygulamasıdır.

Bölümün ana sorusunu küçük bir oyun ortaya koyar. Açık artırmaların günlük hayattaki yeri, temel terimler, örneklerde gereken olasılık araçları ve teklifçilerin nesneye nasıl değer biçtiğini anlatan üç model bu sorunun çerçevesini kurar.

5.1.1 Bir oyunla başlangıç: Vickrey açık artırması

Oyunun kuralı kısadır. Herkes teklifini kapalı bir zarfa yazar ve zarflar aynı anda açılır. En yüksek teklifi veren nesneyi alır. Ancak kazanan kendi teklifini değil, *ikinci en yüksek* teklifi öder. Bu kurala, onu 1961'de inceleyen ekonomist William Vickrey'nin adıyla **Vickrey açık artırması** ya da **ikinci fiyat kapalı zarf açık artırması** denir (Vickrey, 1961).

Oyunu oynayan biri için üç soru doğal olarak akla gelir:

- Nesne bir oyuncuya 70 TL değerindeyse oyuncu zarfa 70'ten azını mı, 70'i mi, yoksa fazlasını mı yazmalıdır?
- Rakiplerin teklifi bilinseydi en iyi teklif değişir miydi?
- Kazanan kendi teklifini ödeseydi (birinci fiyat kuralı), 70 yazmak yine iyi bir fikir olur muydu?

Bir oyun turunda oyuncu 70, birinci rakip 60, ikinci rakip 45 yazmış olsun. En yüksek teklif oyuncunundur; nesneyi o alır. Ödediği tutar ise ikinci en yüksek teklif, yani 60 TL'dir. Nesne oyuncuya 70 TL değerinde olduğuna göre bu alışverişten 70 − 60 = 10 TL kazançlı çıkar. Şekil 5.1'de bu tur gösterilir.



Şekil 5.1: Vickrey açık artırmasında bir tur. Üç zarf açılır: 70, 60 ve 45 TL. 70 yazan kazanır, ancak fiyatı rakibin 60'lık teklifi belirler. Değeri 70 TL olan kazanan için kazanç 70 − 60 = 10 TL'dir.

Şekildeki önemli ayrıntı şudur: kazananın ödediği 60 TL, kazananın kendi zarfında yazan sayı değildir. 70 yerine 65 ya da 100 yazılsaydı da kazanan değişmez, ödenen tutar yine 60 olurdu. Bölümün ilk büyük sonucu bu gözleme dayanır: ikinci fiyat kuralında teklif yalnızca kazanıp kazanmamayı belirler, ödenecek tutarı belirlemez. Bu gözlemin neden dürüst teklifi en iyi strateji yaptığı 5.3.2'de adım adım

ispatlanır. Üçüncü sorunun cevabı ise farklıdır: kazanan kendi teklifini öderse, 70 yazıp kazanan biri 70 öder ve kazancı sıfır olur. Birinci fiyat kuralında bu yüzden teklif değerin altına çekilir; ne kadar çekileceği 5.3.5'in konusudur.

5.1.2 Açık artırmalar neden önemli?

Açık artırmaların bu kadar yaygın olmasının üç nedeni vardır.

Birincisi, **kaynak tahsisinde verimlidir**. İyi tasarlanmış bir açık artırmada nesne, ona en çok değer veren kişiye gitme eğilimindedir. Bu, toplumun nesneden elde ettiği toplam faydanın büyük olması demektir. Bir tabloyu ona 85 TL değer biçen kişinin alması, 55 TL değer biçen kişinin almasından daha çok refah yaratır.

İkincisi, **fiyat keşfi** (*price discovery*) sağlar. Satıcı alıcıların değerlerini bilmez; bilseydi en yüksek değerli alıcıya doğrudan o fiyattan satardı. Açık artırma, alıcıların özel bilgisini tekliflere döker ve fiyat bu tekliflerden oluşur. Kurallar herkese aynı biçimde uygulandığı için piyasa şeffaflaşır.

Üçüncüsü, açık artırma **mekanizma tasarımının en yaygın uygulamasıdır**. Bir mekanizma, katılımcılardan bilgi (burada teklif) alıp bir sonuç (kim alır, kim ne öder) üreten kurallar bütünüdür. Açık artırma, tahsis kuralı ile ödeme kuralının en yalın birleşimidir.

Alan	Nasıl kullanılır?
Sanat eserleri	Müzayede evlerinde yükselen fiyatlı İngiliz açık artırması.
Çevrim içi reklam	Her arama sorgusunda reklam yerleri için çalışan açık artırmalar.
Telekom lisansları	Spektrum açık artırmaları; milyarlarca liralık çok nesneli tasarımlar.
Kamu ihaleleri, hazine	Kapalı zarf teklifler; hazine açık artırmaları.

Tablo 5.1: Açık artırmanın kullanıldığı başlıca alanlar.

Tablo 5.1'deki örnekler ölçek bakımından çok farklıdır. Bir müzayededе tek bir tablo satılır, bir arama motorunda ise saniyede binlerce küçük açık artırma çalışır. Spektrum açık artırmalarında aynı anda yüzlerce lisans satılır ve teklifçiler lisansları birlikte değerlendirir. Kamu ihalelerinde ise yön tersine döner: devlet alıcıdır, firmalar en düşük fiyatı vermek için yarışır. Bu bölümün çözümlemeleri tek nesneli ve tek satıcılı en basit durumda kurulur; ama ortaya çıkan fikirler (dürüst teklif, kırpma, gelir eşdeğerliği, rezerv fiyatı) büyük uygulamaların da temelidir.

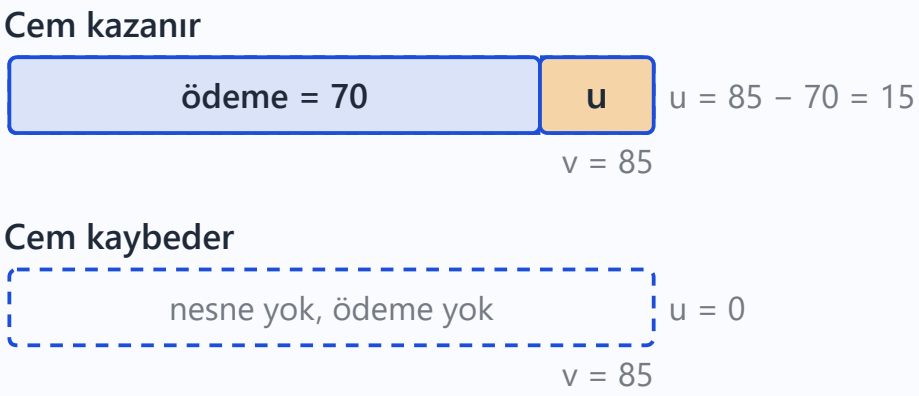
5.1.3 Temel terimler ve model

Açık artırma çözümlemesi dört kavrama dayanır. Bunları ayırt etmek, bölüm boyunca yapılan hataların çoğunu önler. En sık karıştırılan ikili *değer* ile *teklif*tir.

Tanım 5.1

- **Değer** (*value*) v_i: teklifçi i'nin nesneye biçtiği gerçek değerdir; nesne için ödemeye razı olduğu en yüksek tutardır. Değer *özel bilgidir*: yalnızca teklifçinin kendisi bilir.
- **Teklif** (*bid*) b_i: teklifçinin mekanizmaya bildirdiği sayıdır. Teklif değere eşit olmak zorunda değildir; teklifçi stratejik olarak daha az ya da daha çok yazabilir.
- **Fayda** (*utility*) u_i: teklifçi kazanırsa ve p_i öderse $u_i = v_i - p_i$; kaybederse $u_i = 0$'dır.
- **Bağımsız özel değerler** (*independent private values, IPV*): her teklifçinin değeri ötekilerden bağımsız olarak çekilir ve herkes yalnız kendi değerini bilir.

Fayda tanımına **yarı doğrusal fayda** (*quasi-linear utility*) denir. "Doğrusal" sözcüğü, ödemenin faydadan bire bir düşülmesini anlatır: 1 TL fazla ödemek faydayı tam 1 azaltır. Kaybeden teklifçi nesneyi almaz ve hiçbir şey ödemez; faydası sıfırdır. Yani kaybetmek bir "ceza" değildir; yalnızca hiçbir şey olmaz. Teklifçinin amacı faydasını maksimize etmektir.



Şekil 5.2: Yarı doğrusal fayda. Cem'in değeri $v = 85$'tir. İkinci fiyat kuralında kazanır ve ikinci en yüksek teklif olan 70'i öder: değer çubuğunun 70'lik kısmı ödemeye gider, kalan $u = 85 - 70 = 15$ faydadır. Kaybetseydi ne nesne ne ödeme olurdu: $u = 0$.

Şekil 5.2 faydayı bir çubuğun bölünmesi olarak gösterir. Değer çubuğun tamamıdır; ödeme bu çubuğun bir kısmını "yer", kalan kısım teklifçiye kalır. Ödeme değerden büyük olsaydı kalan kısım negatif olurdu; teklifçi nesneyi almakla zarar ederdi. "Fazla teklif" stratejisinin tehlikesi de buradan gelir.

IPV modelinin iki parçası vardır. *Özel* olması, herkesin kendi değerini bildiği ve bu değerin başkalarının bilgisinden etkilenmediği anlamına gelir. Kendi oturma odası için tablo alan biri, başkalarının tabloyu ne kadar beğendiğini öğrense de kendi değerini değiştirmez. *Bağımsız* olması ise değerlerin birbirinden bağımsız rastgele sayılar olarak üretildiği anlamına gelir: Ayşe'nin değerini bilmek, Bora'nın değeri hakkında hiçbir şey söylemez. IPV, açık artırma kuramının temel modelidir. Bu bölümdeki teoremlerin çoğu bu modelde geçerlidir; model değişince sonuçlar da değişebilir (5.1.5).

Örneklerde sık kullanılan iki varsayım daha vardır. Birincisi değerlerin *[0, 1]* aralığında **düzgün dağılımlı** olmasıdır; kısaca $v_i \sim U[0, 1]$ yazılır. İkincisi teklifçilerin **risk nötr** (*risk-neutral*) olmasıdır: risk nötr bir teklifçi belirsiz sonuçları beklenen faydalarıyla karşılaştırır. Örneğin yarı yarıya 10 ya da 0 fayda veren bir seçeneği kesin 5 fayda veren seçenekle aynı görür. Bu iki varsayım, hesapları basit tutmak için seçilmiştir. Riskten kaçınan teklifçilerin durumu 5.4.3'te ayrıca incelenir.

> **Sık yapılan hata**
>
> Değer ile teklifi aynı şey sanmak. Değer, teklifçinin kafasındaki gerçek sayıdır ve değişmez; teklif ise bu değere bakarak verilen bir karardır. "Dürüst teklif" $b_i = v_i$ demektir; ama birinci fiyat kuralında rasyonel teklifçi $b_i < v_i$ yazar. Fayda her zaman *değerden* hesaplanır: $u_i = v_i - p_i$; teklif faydanın formülüne doğrudan girmez, yalnızca kazanıp kazanmamayı ve bazı kurallarda ödemeyi etkiler.

5.1.4 Örneklerde kullanılan olasılık araçları

Bölümün hesapları birkaç basit olasılık aracına dayanır. Bu araçlar aşağıda kısaca toplanmıştır.

Düzgün dağılım. $v \sim U[0, 1]$ demek, v'nin 0 ile 1 arasındaki her değeri "eşit şansla" alabilmesi demektir. Daha kesin söylemek gerekirse, *[0, 1]* içindeki herhangi bir aralığa düşme olasılığı o aralığın uzunluğuna eşittir. Örneğin $P(v < 0{,}3) = 0{,}3$ ve $P(0{,}2 < v < 0{,}7) = 0{,}5$'tir. Bu olasılığı veren fonksiyona **dağılım fonksiyonu** denir ve $F(t) = P(v < t)$ diye yazılır; $U[0, 1]$ için $F(t) = t$'dir. Dağılım fonksiyonunun türevi **yoğunluk** $f(t)$'dir; $U[0, 1]$ için $f(t) = 1$'dir. Yoğunluk, olasılığın değerler üzerine ne kadar sık serpiştirildiğini gösterir; düzgün dağılımda her yerde aynıdır.

Bağımsız olayların çarpımı. İki olay bağımsızsa ikisinin birden gerçekleşme olasılığı olasılıkların çarpımıdır. İki teklifçinin değerleri bağımsız $U[0, 1]$ ise "ikisinin de değeri 0,5'ten küçük" olayının olasılığı $0{,}5 \cdot 0{,}5 = 0{,}25$'tir. Bölümdeki kazanma olasılığı formüllerinin hepsi bu çarpım kuralından çıkar: bir teklifçi kazanmak için *her* rakibini geçmelidir ve rakipler bağımsızsa bu olasılıklar çarpılır.

Beklenen değer. Rastgele bir büyüklüğün beklenen değeri, aldığı değerlerin olasılıklarıyla ağırlıklandırılmış ortalamasıdır. Bir deneyi çok kez tekrarlayınca elde edilen ortalama beklenen değere yaklaşır. $v \sim U[0, 1]$ için $E[v] = 1/2$'dir. Bölümde iki tür beklenen değer sık geçer: bir teklifçinin *beklenen faydası* (bir teklifin ortalamada ne kazandırdığı) ve satıcının *beklenen geliri* (bir kuralın satıcıya ortalamada ne getirdiği).

Kuyruk formülü. Değeri 0 ile 1 arasında olan rastgele bir X için beklenen değer şu formülle de hesaplanabilir:

$$E[X] = \int_0^1 P(X > t)\, dt$$

Formülün anlamı şudur: X, 0'dan başlayan bir çubuğun uzunluğu olsun. Her t noktasında çubuğun o noktayı geçme olasılığı $P(X > t)$'dir. Bu olasılıkları bütün t'ler boyunca toplamak (integral almak), çubuğun ortalama uzunluğunu verir. Örnek olarak $X \sim U[0, 1]$ alınırsa $P(X > t) = 1 - t$ ve $\int_0^1 (1 - t)\, dt = 1 -$

$1/2 = 1/2$ çıkar; bu, bildiğimiz $E[X] = 1/2$ sonucudur. Kuyruk formülü, birkaç sayının en büyüğünün ya da en küçüğünün beklentisini bulmak için çok kullanışlıdır, çünkü "en büyük t'den küçük" gibi olayların olasılığı çarpım kuralıyla kolayca yazılır (5.4.1).

Polinomların integrali. Bölümde yalnızca polinomların integrali gerekir: $\int_0^a t^k\,dt = a^{k+1}/(k+1)$. Örneğin $\int_0^1 t^2\,dt = 1/3$ ve $\int_0^r 2t^2\,dt = 2r^3/3$'tür.

Türev ve en büyük nokta. Bir teklifçinin beklenen faydası teklifin bir fonksiyonudur. Bu fonksiyon önce artıp sonra azalıyorsa en büyük değerini türevin sıfır olduğu noktada alır. Kullanılan iki kural şunlardır: t^k'nın türevi $k\,t^{k-1}$'dir ve iki fonksiyonun çarpımının türevi $(g\,h)' = g'\,h + g\,h'$'dir (çarpım kuralı).

Örnek 5.1

Üç teklifçinin değerleri bağımsız $U[0,\,1]$'dir. Değeri 0,6 olan bir teklifçinin değerinin öteki iki teklifçinin değerinden büyük olma olasılığı nedir?

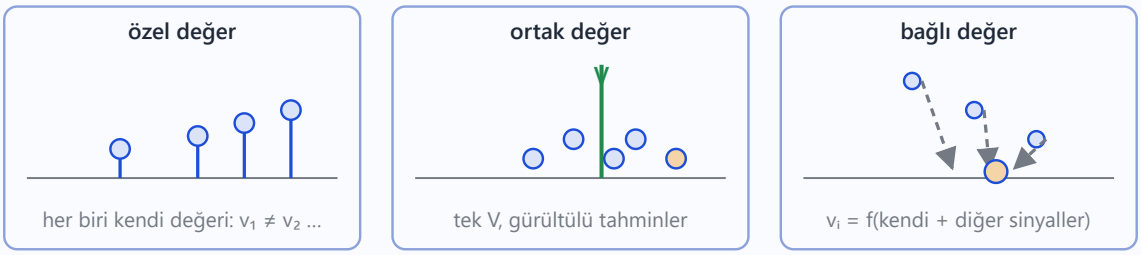
1. Tek bir rakip için $P(v_j < 0{,}6) = F(0{,}6) = 0{,}6$'dır (düzgün dağılımda olasılık aralığın uzunluğudur).
2. İki rakibin değeri bağımsızdır; ikisinin birden 0,6'dan küçük olma olasılığı çarpımdır: $0{,}6 \cdot 0{,}6 = 0{,}36$.
3. Sonuç: teklifçinin değeri %36 olasılıkla en yüksek değerdir. Genel olarak n teklifçi varsa bu olasılık v^{n-1}'dir.

5.1.5 Değerleme modelleri

Teklifçilerin nesneye nasıl değer biçtiği, açık artırmanın nasıl çözümleneceğini belirler. Literatürde üç temel model ayırt edilir.

Tanım 5.2

- **Özel değer** (*private value*): her teklifçi nesnenin değerini yalnızca kendi bilgisine ve zevkine göre belirler. Örneğin kendi evinde asacağı bir tablo. Değerler birbirinden farklıdır: $v_1 \neq v_2 \neq \dots$
- **Ortak değer** (*common value*): nesnenin gerçek değeri herkes için aynıdır, ama kimse bu değeri tam bilmez; katılımcıların tahminleri farklıdır. Örneğin bir petrol sahasının içindeki petrolün piyasa değeri.
- **Birbirine bağlı değerler** (*interdependent values*): bir teklifçinin değeri hem kendi özel bilgisine hem de ötekilerin bilgisine bağlıdır: $v_i = f(\text{kendi sinyali, diğer sinyaller})$.



Şekil 5.3: Üç değerleme modeli. Solda özel değer: her teklifçinin kendi değeri vardır ve değerler farklıdır. Ortada ortak değer: tek bir gerçek değer V (yeşil çizgi) etrafında gürültülü tahminler. Sağda birbirine bağlı değerler: bir teklifçinin değeri kendi sinyaliyle birlikte ötekilerin sinyallerinden de etkilenir.

Şekil 5.3 üç modelin farkını görselleştirir. Özel değer modelinde başkalarının ne düşündüğünü öğrenmek kimsenin değerini değiştirmez. Ortak değer modelinde ise durum tersinedir: başka bir firmanın sahayı düşük değerlendirdiğini öğrenen firma, kendi tahmininin fazla iyimser olabileceğini düşünür. Bilgi değerlidir, çünkü herkesin tahmini aynı gerçeğin gürültülü bir ölçümüdür.

Gerçek nesnelerin çoğu iki uç arasında bir yerdedir. Bir sanat eseri hem kişisel zevke (özel bileşen) hem de ileride satılabileceği fiyata (ortak bileşen) göre değerlendirilir. Bir şirket satın alımında alıcı firmanın kendi sinerjisi özel, hedef şirketin nakit akışları ortak bileşendir. Birbirine bağlı değerler modeli bu karışık durumu temsil eder.

Modeller arasındaki fark yalnızca kuramsal değildir. Ortak değer modeli, **kazananın laneti** (*winner's curse*) adı verilen olguya yol açar: açık artırmayı kazanan, büyük olasılıkla değeri en iyimser tahmin eden katılımcıdır ve bunu hesaba katmazsa ortalamada zarar eder. Bu olgu 5.4.8'de sayısal olarak incelenir. Bölümün geri kalanında aksi söylenmedikçe IPV modeli varsayılır.

5.2 Dört temel tür

Açık artırma türleri iki ölçüte göre sınıflanır. Birinci ölçüt tekliflerin nasıl verildiğidir: herkesin gözü önünde, fiyat değişirken (açık, "sesli" açık artırma) ya da bir kez ve gizlice (kapalı zarf). İkinci ölçüt kazananın ne ödediğidir: kendi teklifini mi, yoksa ikinci en yüksek teklifi mi? Bu iki ölçütün birleşimi dört temel türü verir. Dört türün de aynı dört kişilik örnek üzerinde adım adım çalıştırılması farklarını açıkça gösterir.

5.2.1 Dört kural

Tür	Nasıl işler?	Kazanan	Öder
İngiliz (*English*)	Fiyat açıkça yükselir; teklifçiler birer birer çekilir.	Son kalan	Son rakibin çekildiği fiyat
Hollanda (*Dutch*)	Fiyat yüksekten başlar, düşer; ilk "kabul" diyen alır.	İlk kabul eden	O anki fiyat

Tür	Nasıl işler?	Kazanan	Öder
Birinci fiyat kapalı zarf	Teklifler gizli verilir, bir kez.	En yüksek teklif	Kendi teklifi
İkinci fiyat (Vickrey)	Teklifler gizli verilir, bir kez.	En yüksek teklif	İkinci en yüksek teklif

Tablo 5.2: Dört temel açık artırma türü.

İngiliz açık artırması (*English (ascending) auction*) müzayede evlerinde görülen klasik biçimdir. Açık artırmacı düşük bir fiyattan başlar ve fiyatı adım adım yükseltir. Fiyat bir teklifçinin ödemeye razı olduğu tutarı geçince o teklifçi çekilir. Tek kişi kalınca açık artırma biter; son kalan, son rakibinin çekildiği fiyattan nesneyi alır. Kuramsal çözümlemede genellikle "saat" sürümü kullanılır: fiyat sürekli yükselir ve teklifçiler bir düğmeyi bırakarak çekilir. Böylece bir teklifçinin tek kararı "hangi fiyatta çekileceğim?" sorusuna indirgenir.

Hollanda açık artırması (*Dutch (descending) auction*) adını Hollanda'daki çiçek mezatlarından alır. Fiyat yüksek bir değerden başlar ve düşer. İlk "kabul" diyen nesneyi o anki fiyattan alır ve açık artırma hemen biter. Hızlı olduğu için çok sayıda küçük partinin satıldığı çiçek ve balık hallerinde kullanılır.

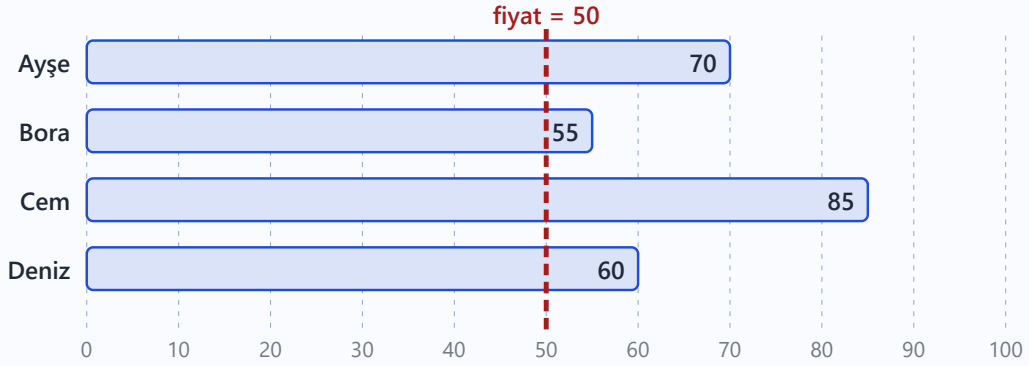
Birinci fiyat kapalı zarf açık artırması (*first-price sealed-bid auction*) kamu ihalelerinin bilinen biçimidir. Her teklifçi bir kez ve gizlice teklif verir. En yüksek teklif kazanır ve kazanan kendi teklifini öder.

İkinci fiyat (Vickrey) açık artırması (*second-price (Vickrey) auction*) da gizli ve tek seferliktir. Kazanan yine en yüksek teklifi verendir, ama ödediği tutar ikinci en yüksek tekliftir.

Dört türü karşılaştırmak için aynı dört kişi kullanılır. Her kişinin nesne için ödemeye razı olduğu en yüksek tutar, yani *sınırı* şöyledir: **Ayşe 70, Bora 55, Cem 85, Deniz 60**. Bu sınırlar özel değerlerdir. Şimdilik herkesin sınırını teklif ettiği, yani İngiliz açık artırmasında sınırına kadar kaldığı ve Hollanda açık artırmasında fiyat sınırına inince kabul ettiği varsayılır. Bu varsayımın ne zaman akıllıca olduğu 5.3'te incelenir.

5.2.2 İngiliz açık artırması adım adım

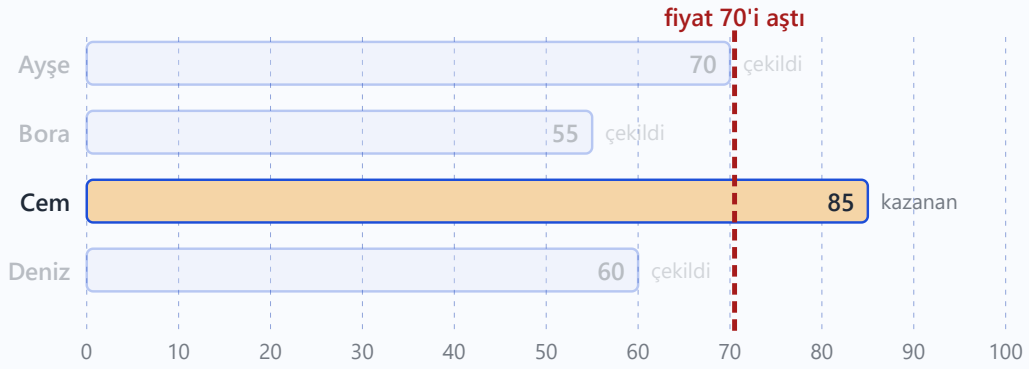
Fiyat 40'tan başlasın ve yavaşça yükselsin. Her teklifçi, fiyat kendi sınırının altında kaldığı sürece içeride kalır.



Şekil 5.4: İngiliz açık artırmasının başı. Çubuklar dört teklifçinin sınırlarıdır; kırmızı kesikli çizgi o anki fiyattır. Fiyat = 50 iken bütün sınırlar 50'nin üstündedir ve dört kişi de içeridedir.

Örnek 5.2

1. **Fiyat 50:** en küçük sınır 55'tir; herkes içeride (Şekil 5.4).
2. **Fiyat 55'i geçince** Bora çekilir, çünkü bu fiyattan almak onun için zararlıdır (sınırı 55). Kalanlar: Ayşe, Cem, Deniz.
3. **Fiyat 60'ı geçince** Deniz çekilir (sınırı 60). Kalanlar: Ayşe, Cem.
4. **Fiyat 70'i geçince** son rakip Ayşe çekilir. Cem tek başına kalır ve kazanır.
5. **Ödeme:** Cem, Ayşe'nin çekildiği fiyatı, yani yaklaşık 70'i öder. Fiyat 1 TL'lik adımlarla artıyorsa Cem 70 ya da 71 öder; artım küçüldükçe ödeme tam 70'e yaklaşır. Cem'in faydası yaklaşık *85 − 70 = 15*'tir.



Şekil 5.5: İngiliz açık artırmasının sonu. Fiyat 70'i aşınca Ayşe çekilir; soluk çubuklar çekilenlerdir. Cem kazanır ve ikinci en yüksek değer olan yaklaşık 70'i öder.

Şekil 5.5'te görülen sonuç İngiliz açık artırmasının genel bir özelliğidir: *kazanan en yüksek değerli kişidir ve ödediği fiyat ikinci en yüksek değerdir*. Bunun nedeni basittir. Açık artırma, ikinci en yüksek değerli kişi çekilene kadar sürer; o kişi çekilince başka rakip kalmaz. Cem'in kendi sınırı olan 85, ödediği

fiyatı hiç etkilemez. 85 yerine 75 ya da 100 olsaydı da Cem yine 70'te kazanırdı. Bu, 5.1.1'deki Vickrey gözlemiyle aynıdır.

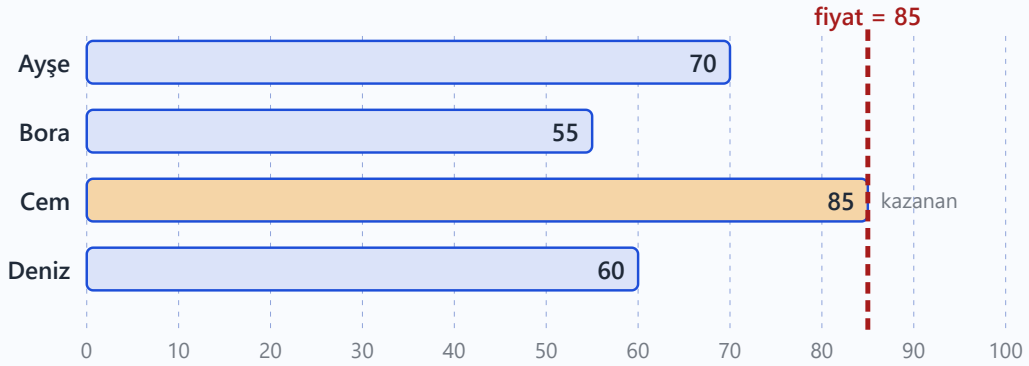
Özel değerlerde bir teklifçinin İngiliz açık artırmasındaki en iyi stratejisi sınırına kadar kalmaktır. Sınırdan önce çekilmek, kazanılabilecek bir kârı kaçırabilir: fiyat 65'teyken çekilen Ayşe, Cem olmasaydı 60'ı biraz geçen bir fiyattan nesneyi alabilirdi. Sınırdan sonra kalmak ise zararına kazanma riskini doğurur. Bu akıl yürütme Vickrey açık artırmasındaki dürüstlük ispatının İngiliz açık artırmasındaki karşılığıdır (5.3.2).

5.2.3 Hollanda açık artırması adım adım

Hollanda açık artırmasında fiyat 100'den başlar ve her adımda 5 düşer. Her teklifçi, fiyat kendi sınırına inince "kabul" der.

Örnek 5.3

1. **Fiyat 100:** bu fiyat herkesin sınırının üstündedir; kimse kabul etmez, fiyat 5 düşer.
2. **Fiyat 95:** en yüksek sınır 85'tir; hâlâ kimse kabul etmez.
3. **Fiyat 90:** yine kimse kabul etmez.
4. **Fiyat 85:** fiyat Cem'in sınırına iner. Cem "kabul" der ve 85 öder. Açık artırma bu anda biter; diğerleri hiç konuşamaz.
5. **Sonuç:** Cem kazanır, 85 öder; faydası $85 - 85 = 0$'dır.



Şekil 5.6: Hollanda açık artırmasının sonu. Fiyat 100'den 5'er düşer; 85'e inince fiyat Cem'in sınırına ulaşır, Cem kabul eder ve 85 öder.

Şekil 5.6 ile İngiliz açık artırmasının şekli karşılaştırıldığında önemli bir fark göze çarpar. İngiliz açık artırmasında süreç boyunca bilgi ortaya çıkar: kimin hangi fiyatta çekildiğini herkes görür. Hollanda açık artırmasında ise ilk kabul eden kazandığı için açık artırma biter; kimse başkasının sınırı hakkında bir şey öğrenmez. Bir teklifçi önceden yalnızca tek bir soruyu cevaplar: "fiyat kaça inince kabul ederim?" Bu sorunun cevabı tek bir sayıdır ve bu sayı fiilen gizli bir tekliftir. Bu gözlemin sonuçları 5.2.5'te tartışılır.

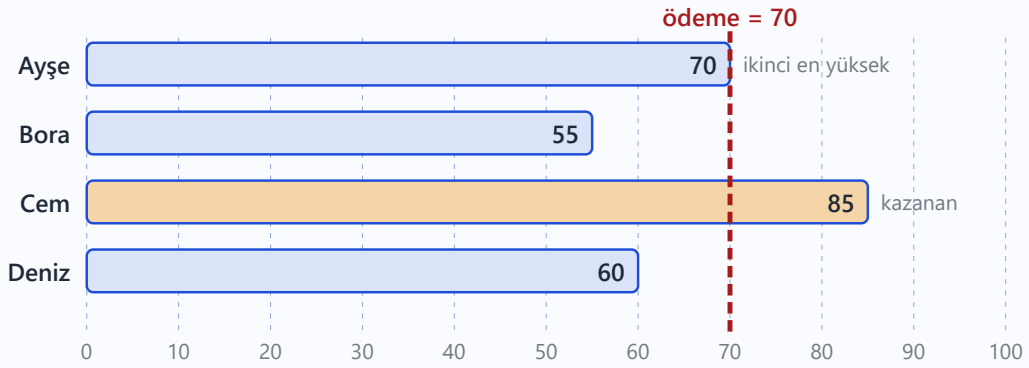
Örnekte Cem sınırında kabul ettiği için faydası sıfır çıktı. Bu, gerçekçi bir davranış değildir. Cem 85'te kabul etmek yerine biraz bekleyip örneğin 75'te kabul etseydi, rakiplerin sınırları 75'in altında kaldığı sürece yine kazanır ve 10 fayda elde ederdi. Beklemenin riski, fiyat bir rakibin sınırına inince o rakibin önce kabul etmesidir. Bu ödünleşim birinci fiyat açık artırmasındaki teklif kırpmanın aynısıdır.

5.2.4 Birinci ve ikinci fiyat kapalı zarf

Kapalı zarf açık artırmasında herkes teklifini bir kez ve gizlice verir. Kimse diğerlerinin teklifini görmez; zarflar aynı anda açılır. Dört kişi sınırlarını yazarsa zarflardan 70, 55, 85 ve 60 çıkar.

Örnek 5.4

1. **Zarflar açılır:** Ayşe 70, Bora 55, Cem 85, Deniz 60. Her iki kuralda da kazanan en yüksek teklifi veren Cem'dir.
2. **Birinci fiyat:** Cem kendi teklifini, 85'i öder. Değeri 85 olduğu için faydası $85 - 85 = 0$'dır.
3. **İkinci fiyat (Vickrey):** Cem ikinci en yüksek teklifi, Ayşe'nin 70'ini öder. Faydası $85 - 70 = 15$'tir.



Şekil 5.7: İkinci fiyat kapalı zarf. Zarflar 70, 55, 85, 60 olarak açılır; kazanan Cem'dir (turuncu) ve ödemeyi ikinci en yüksek teklif olan 70 belirler (kırmızı kesikli çizgi).

Birinci fiyat kuralında kazananın ödemesi kendi teklifine eşittir, ikinci fiyat kuralında ise kendi teklifinden bağımsızdır. Şekil 5.7'de Cem'in çubuğu ne kadar uzun olursa olsun ödeme çizgisi 70'te kalır. Bu tek fark, iki kuralda akıllı bir teklifçinin davranışını tamamen değiştirir: ikinci fiyatta değerini yazmak en iyisidir, birinci fiyatta ise değerin altına inmek gerekir.

5.2.5 Aynı teklifler, dört sonuç

Dört kuralın sonucu Tablo 5.3'te yan yana konur.

Tür	Kazanan	Ödeme	Ödemeyi kimin teklifi belirliyor?
İngiliz	Cem	≈ 70	Son çekilen rakip (Ayşe)

Tür	Kazanan	Ödeme	Ödemeyi kimin teklifi belirliyor?
İkinci fiyat	Cem	70	İkinci en yüksek teklif (Ayşe)
Hollanda	Cem	85	Kazananın kendisi
Birinci fiyat	Cem	85	Kazananın kendisi

Tablo 5.3: Sınırlar 70, 55, 85, 60 ve herkes sınırını teklif ederken dört kuralın sonucu.

Tablo iki ikiliyi ortaya çıkarır.

İngiliz ≈ ikinci fiyat. Özel değerlerde iki türde de kazanan en yüksek değerli kişidir ve ikinci en yüksek değeri öder. İngiliz açık artırmasında bunun nedeni sınırına kadar kalmanın baskın olmasıdır; ikinci fiyatta ise dürüst teklifin baskın olmasıdır. İki tür arasındaki "≈" işareti fiyat artımının yarattığı küçük farkı anlatır: İngiliz açık artırmasında ödeme ikinci en yüksek değere en fazla bir artım kadar yakındır. Ayrıca bu denklik yalnızca özel değerlerde tamdır. Ortak ya da birbirine bağlı değerlerde İngiliz açık artırmasında rakiplerin hangi fiyatta çekildiğini görmek bilgi verir; bu bilgi kapalı zarfta yoktur ve iki tür farklılaşabilir.

Hollanda ≡ birinci fiyat. Bu iki tür, değerleme modelinden bağımsız olarak *stratejik olarak özdeştir*. Hollanda açık artırmasında bir teklifçi açık artırma bitene kadar işe yarar hiçbir bilgi edinmez; çünkü açık artırma biri kabul ettiği anda biter. Bu yüzden önceden "fiyat şu değere inince kabul ederim" diye bir sayı belirlemek, bu sayıyı kapalı zarfa yazmakla aynıdır. Her iki kuralda da en yüksek sayıyı belirleyen kazanır ve kendi sayısını öder. Üçlü eşitlik işareti (≡) bu özdeşliğin yaklaşık değil tam olduğunu anlatır.

Sık yapılan hata

Tablo 5.3'e bakıp "birinci fiyat satıcıya daha çok kazandırır, çünkü 85 > 70" sonucunu çıkarmak. Tabloda herkes sınırını teklif etmiştir; bu varsayım birinci fiyat kuralında akılcı değildir. Rasyonel teklifçi birinci fiyatta teklifini kırpar, yani değerinin altında teklif verir. Kırpma hesaba katılınca iki kuralın satıcıya getirdiği *beklenen* gelir eşitlenir. Bu sonuca **gelir eşdeğerliği** denir ve 5.4.2'de ispatlanır.

Örnek 5.5

Sınırlar 40, 65, 80 ve 90 olsun; herkes sınırını teklif etsin. Dört kuralda kazanan kimdir ve ne öder?

1. Her dört kuralda kazanan, sınırı en yüksek olan 90'lık teklifçidir.
2. **İngiliz:** fiyat 40 ve 65'i geçince iki teklifçi çekilir; 80'i geçince son rakip çekilir. Ödeme ≈ 80, fayda ≈ 10.
3. **İkinci fiyat:** zarflar 40, 65, 80, 90; ödeme ikinci en yüksek teklif olan 80, fayda 10.
4. **Hollanda:** fiyat 90'a inince kazanan kabul eder; ödeme 90, fayda 0.

5. **Birinci fiyat:** kazanan kendi teklifini, 90'ı öder; fayda 0.

5.3 Kapalı zarf stratejileri

Önceki örneklerde herkesin sınırını teklif ettiği varsayıldı. Bu varsayım kaldırılınca soru şudur: akıllı bir teklifçi zarfa ne yazmalıdır? Cevap kurala göre çok farklıdır. İkinci fiyat kuralında cevap "değerini" olur ve rakiplerin ne yaptığına bağlı değildir. Birinci fiyat kuralında ise cevap rakiplerin davranışına bağlıdır ve bir denge hesabı gerektirir.

5.3.1 Vickrey açık artırmasının stratejik özelliği

Vickrey açık artırmasının kuralı şudur: en yüksek teklifi veren kazanır ve ikinci en yüksek teklif kadar öder. Bu kuralın en önemli sonucu, teklifçinin rakiplerini tahmin etmesine gerek bırakmamasıdır. Bunu ifade etmek için önce baskın strateji kavramı hatırlanmalıdır.

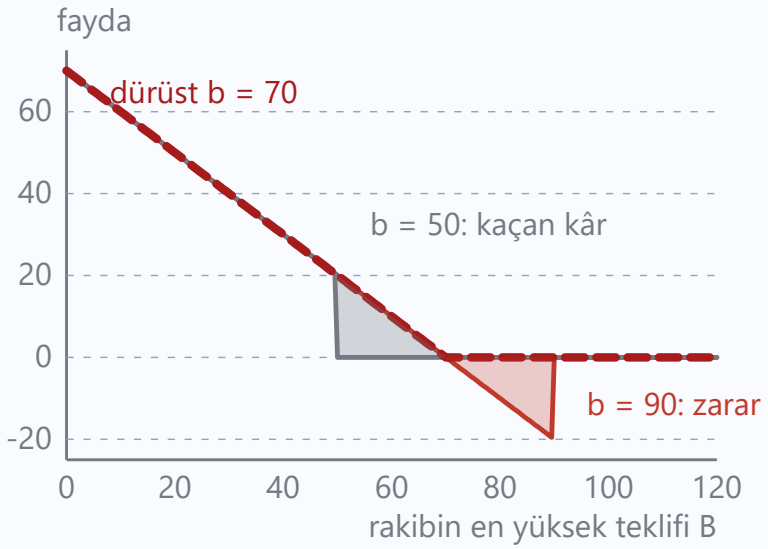
Tanım 5.3

Bir strateji, ötekilerin *her* olası seçimine karşı başka herhangi bir strateji kadar iyi ve en az bir seçime karşı kesin daha iyi sonuç veriyorsa **zayıf baskın strateji**dir (*weakly dominant strategy*). Açık artırmada bu, teklifçinin faydasının rakiplerin her teklif profili için en az öteki tekliflerinki kadar büyük olması ve bazı profillerde kesin büyük olması demektir.

Teorem 5.1 (Vickrey, 1961)

Bağımsız özel değerli tek nesne modelinde, ikinci fiyat kapalı zarf açık artırmasında gerçek değeri teklif etmek ($b_i = v_i$) her teklifçi için zayıf baskın stratejidir.

Teoremin arkasındaki fikir tek bir cümleyle söylenebilir: *teklif, ne ödeneceğini değil yalnızca kazanılıp kazanılmayacağını belirler*. Teklifçi kazanırsa rakiplerin en yüksek teklifini öder; bu tutar kendi zarfındaki sayıdan bağımsızdır. O hâlde zarfa yazılan sayının tek işi bir eşik koymaktır: rakiplerin en yüksek teklifi bu eşiğin altındaysa teklifçi kazanır, üstündeyse kaybeder. En iyi eşik, kazanmanın kârlı olduğu yerle zararlı olduğu yeri ayıran noktadır; bu nokta da teklifçinin değerinin kendisidir.



Şekil 5.8: Değer $v = 70$ iken faydanın rakiplerin en yüksek teklifi B'ye göre değişimi. Kesikli kırmızı eğri dürüst teklif $b = 70$'in faydasıdır: $B < 70$ iken $70 - B$, sonra 0. Gri bölge $b = 50$ teklifinin kaçırdığı kârı, kırmızı bölge $b = 90$ teklifinin uğradığı zararı gösterir. Hiçbir teklif, hiçbir B için dürüst teklifin eğrisini geçemez.

Şekil 5.8 ispatın görsel özetidir. Yatay eksen rakiplerin en yüksek teklifi B'dir. Dürüst teklifçi B 70'in altındayken kazanır ve $70 - B$ fayda alır; bu fayda B büyüdükçe doğrusal olarak azalır. B 70'i geçince dürüst teklifçi kaybeder ve faydası 0'dır. 50 teklif eden biri B 50 ile 70 arasındayken kaybeder; oysa bu aralıkta kazanmak kârlıydı (gri bölge). 90 teklif eden biri B 70 ile 90 arasındayken kazanır; ama bu aralıkta kazanmak zararlıdır (kırmızı bölge, faydası negatif). Her iki sapma bazı B'lerde dürüst teklifle aynı, bazılarında daha kötüdür; hiçbir yerde daha iyi değildir.

Teoremin geçerlilik koşulları önemlidir. Sonuç bağımsız özel değerli tek nesne modelinde kanıtlanır. Ortak değerlerde, bütçe kısıtları olduğunda ya da değerler birbirine bağlı olduğunda aynı biçimde geçerli olmayabilir. Örneğin bütçesi değerinden küçük olan bir teklifçi değerini teklif edemez; ortak değerde ise "değer" zaten bilinmez. Vickrey açık artırması, mekanizma tasarımındaki VCG (Vickrey–Clarke–Groves) mekanizmasının tek nesneli özel durumudur. VCG'de de her katılımcı, başkalarına yüklediği zarar kadar öder; tek nesnede bu zarar ikinci en yüksek değerdir.

5.3.2 Dürüstlüğün ispatı adım adım

İspat bir *durum analizidir*: rakiplerin en yüksek teklifi $B = max_{j \neq i}\, b_j$ için olası durumlar tek tek incelenir. Teklifçi B'yi bilmez; ama ispat her B için ayrı ayrı yapıldığından bu bilgisizlik sorun olmaz. Önce sayısal bir örnekle başlanır: değer $v = 70$'tir ve üç strateji karşılaştırılır: dürüst $b = 70$, fazla $b = 90$, eksik $b = 50$.

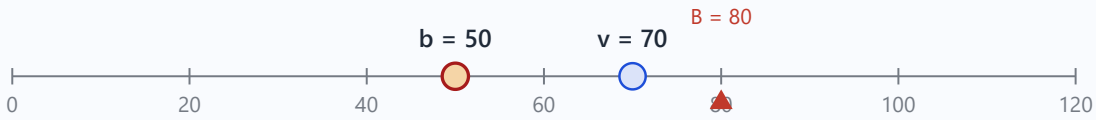
Örnek 5.6

1. **Dürüst, B = 60:** 70 > 60 olduğundan kazanır ve 60 öder; fayda $70 - 60 = 10$.

2. **Dürüst, B = 80:** 70 < 80 olduğundan kaybeder; fayda 0. Dürüst teklif hep doğru kararı verir: kâr varsa kazanır, zarar varsa kaybeder.
3. **Fazla b = 90, B = 60:** $B < v$'dir. Yine kazanır, yine 60 öder; fayda 10. Dürüstle fark yok.
4. **Fazla b = 90, B = 80:** B, v ile b arasındadır. Kazanır ama 80 öder; fayda $70 - 80 = -10$. Dürüst olsa 0 alırdı: zarar.
5. **Fazla b = 90, B = 95:** $B > b$'dir. Kaybeder; fayda 0, dürüstle aynı.
6. **Eksik b = 50, B = 40:** $B < b$'dir. Kazanır, 40 öder; fayda 30. Dürüst teklif de 40 öderdi; fark yok, çünkü ödemeyi b değil B belirler.
7. **Eksik b = 50, B = 60:** B, b ile v arasındadır. Kaybeder; fayda 0. Dürüst olsa $70 - 60 = 10$ kazanırdı: fırsat kaçtı.
8. **Eksik b = 50, B = 80:** kaybeder; dürüstle aynı.

Strateji	B (rakip)	Sonuç	Fayda	Dürüste göre
Dürüst b = 70	60	kazanır, 60 öder	10	—
Dürüst b = 70	80	kaybeder	0	—
Fazla b = 90	60	kazanır, 60 öder	10	aynı
Fazla b = 90	80	kazanır, 80 öder	−10	daha kötü
Fazla b = 90	95	kaybeder	0	aynı
Eksik b = 50	40	kazanır, 40 öder	30	aynı
Eksik b = 50	60	kaybeder	0	daha kötü
Eksik b = 50	80	kaybeder	0	aynı

Tablo 5.4: Değer $v = 70$ için Vickrey'de durum analizi. Son sütun sapmanın dürüst teklife göre durumunu verir.



Şekil 5.9: Son durumun sayı doğrusu: eksik teklif $b = 50$, değer $v = 70$ ve rakip teklifi $B = 80$. B ikisinin de sağında olduğundan iki strateji de kaybeder; sonuç aynıdır.

Tablo 5.4 bir örüntü gösterir. Sapmanın sonucu, B'nin b ile v arasına düşüp düşmediğine bağlıdır. B bu aralığın dışındaysa sapan teklifçi dürüst teklifçiyle aynı kararı verir (ikisi de kazanır ya da ikisi de kaybeder) ve kazanırsa aynı B'yi öder. B aralığın içindeyse karar değişir ve değişen karar hep yanlış yöndedir. Şekil 5.9 bu aralığı sayı doğrusu üzerinde gösterir: b ile v arasındaki bölge "tehlike bölgesi"dir. Sayısal kontrol olarak, $v = 70$ için bütün $b, B \in \{0, 1, \ldots, 120\}$ çiftleri denendiğinde hiçbir b dürüst teklifi geçemez.

Aynı akıl yürütme genel olarak şöyle yazılır.

İspat. Teklifçinin değeri v, rakiplerin en yüksek teklifi B olsun. Herhangi bir teklifle iki sonuç mümkündür: kazanmak (fayda $v - B$, çünkü kazanan B öder) ya da kaybetmek (fayda 0). Yani B sabitken hangi teklif verilirse verilsin fayda ya $v - B$ ya da 0'dır. Bu iki sayıdan büyük olanı $max(v - B, 0)$'dır ve hiçbir teklif bundan fazlasını veremez.

Dürüst teklif tam olarak bu en büyük değeri verir. $B < v$ ise dürüst teklifçi kazanır ve $v - B > 0$ alır. $B > v$ ise kaybeder ve 0 alır; bu durumda $v - B < 0$ olduğundan 0 daha iyidir. $B = v$ ise iki sonuç da 0 fayda verir ve eşitliğin nasıl bozulduğu önemsizdir. Demek ki her B için dürüst teklif ulaşılabilecek en büyük faydayı verir; başka hiçbir teklif ondan iyi olamaz.

Kesinlik için her sapmanın bir yerde kesin daha kötü olduğunu göstermek yeterlidir. $b > v$ ise $v < B < b$ olan bir B için sapan teklifçi kazanır ve $v - B < 0$ alır; dürüst teklif 0 verirdi. $b < v$ ise $b < B < v$ olan bir B için sapan teklifçi kaybeder ve 0 alır; dürüst teklif $v - B > 0$ verirdi. Böylece $b = v$ zayıf baskın stratejidir. □

İspatta rakiplerin nasıl teklif verdiği hiç kullanılmadı. Rakipler dürüst de olabilir, yalan da söyleyebilir, rastgele de davranabilir; sonuç değişmez. Bu, baskın stratejinin Nash dengesinden daha güçlü bir kavram olmasının nedenidir: Nash dengesinde bir oyuncunun stratejisi rakiplerin dengedeki stratejilerine göre en iyidir, baskın stratejide ise rakiplerin *her* stratejisine göre en iyidir. Bu özelliği taşıyan mekanizmalara **stratejiye dayanıklı** (*strategy-proof*) ya da baskın stratejide teşvik uyumlu (DSIC) denir.

> **Sık yapılan hata**
>
> "Dürüst teklif kazanmayı garanti eder" ya da "dürüst teklif her zaman pozitif fayda verir" sanmak. Dürüst teklif $B > v$ iken kaybeder ve 0 fayda alır. Zayıf baskınlık, dürüst teklifin her durumda *en az başka bir teklif kadar iyi* olduğunu söyler; her durumda kazandırdığını söylemez. Bir başka yaygın hata da baskınlığın "rakipler de dürüstse" geçerli olduğunu düşünmektir; ispat rakiplerin davranışından bağımsızdır.

5.3.3 Oyuna dönüş: kuram ve laboratuvar

5.1.1'deki açılış sorusunun kuramsal cevabı artık açıktır: zarfa değer yazılmalıdır. Açılıştaki zarflarda oyuncu 70, rakipler 60 ve 45 yazmıştı. Dürüst teklifle oyuncu kazanır ve 60 öder; kazancı 10'dur. 65 ya da 100 yazsaydı da kazanır ve aynı 60'ı öderdi. 55 yazsaydı nesne 60 yazan rakibe gider ve oyuncu 10'luk kazancı kaçırırdı. Teklif değerden yukarı ya da aşağı çekildiğinde kazanç hiçbir B için artmaz; yalnızca bazı B değerlerinde azalır.

Üçüncü açılış sorusunun cevabı da bellidir. Birinci fiyat kuralında 70 yazan oyuncu kazansa bile $70 - 70 = 0$ kazanç elde eder. Orada teklifi değerin altına çekmek gerekir; nasıl çekileceği 5.3.4'ün konusudur.

Gerçek davranış kuramdan farklı olabilir. Laboratuvar deneylerinde ikinci fiyat kapalı zarf açık artırmasında *değerin üstünde* teklif vermek yaygındır ve deney tekrarlansa da kolay kaybolmaz. Aynı denekler yükselen fiyatlı İngiliz açık artırmasında dengeye çok daha hızlı yaklaşır (Kagel, Harstad ve

Levin, 1987). Bunun olası bir nedeni şudur: fazla teklifin zararı yalnızca B belirli bir aralığa düştüğünde ortaya çıkar ve bu durum seyrek yaşanır; denekler zararı nadiren gördüğü için alışkanlıklarını değiştirmez. İngiliz açık artırmasında ise fiyatın değeri geçtiği an herkesin gözü önündedir ve çekilmek doğal bir karardır.

Gündelik hayatta ikinci fiyat kuralına çok yakın bir örnek eBay'in otomatik (vekil) teklif sistemidir. Kullanıcı ödemeye razı olduğu en yüksek tutarı sisteme bildirir; sistem onun adına, rakipleri geçecek en küçük tutarda teklif verir. Sonuçta kazanan, ikinci en yüksek bildirimi bir artış miktarı kadar geçen fiyatı öder. Bu, ikinci fiyat kuralının artımlı bir sürümüdür.

5.3.4 Birinci fiyatta teklif kırpma

Birinci fiyat kuralında dürüst teklif anlamsızdır: teklifçi kazansa bile $v - v = 0$ fayda elde eder. Teklifi değerin altına çekmeye **teklif kırpma** (*bid shading*) denir. Kırpma bir ödünleşim yaratır. Teklifi düşürmek kazanınca kalan kârı ($v - b$) artırır, ama kazanma olasılığını azaltır. Bir teklifin beklenen faydası bu iki etkinin çarpımıdır:

$$E[u(b)] = (v - b) \cdot P(b \text{ ile kazanmak})$$

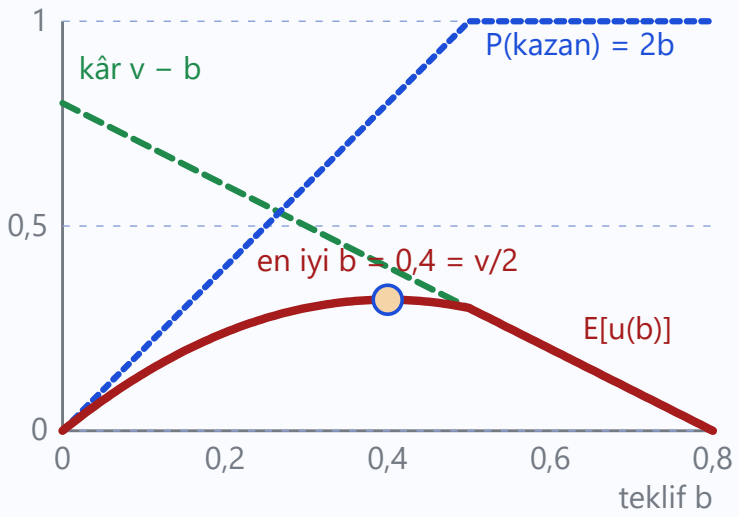
Formüldeki ilk çarpan kazanınca elde edilen kârdır, ikinci çarpan bu kârın gerçekleşme olasılığıdır. Kaybedince fayda 0 olduğundan formüle başka terim girmez. $b = 0$ iken kazanma olasılığı 0'dır; $b = v$ iken kâr 0'dır. İki uçta da beklenen fayda sıfırdır; en iyi teklif ikisinin arasında bir yerdedir.

Kazanma olasılığı rakiplerin nasıl teklif verdiğine bağlıdır. Bu yüzden birinci fiyatta en iyi teklif, ikinci fiyattaki gibi rakiplerden bağımsız değildir. Her teklifçinin stratejisi ötekilerin stratejisine bağlı olduğunda doğru çözüm kavramı bir dengedir. Ancak burada her teklifçi rakiplerinin değerlerini bilmez, yalnızca dağılımlarını bilir. Bu tür eksik bilgili oyunlarda kullanılan denge kavramı Bayes–Nash dengesidir.

Tanım 5.4

Her teklifçinin **stratejisi**, değerinden teklife giden bir fonksiyondur: $b_i = \beta_i(v_i)$. Ötekiler kendi stratejilerini oynarken ve değerleri dağılımlarına göre rastgeleyken, her teklifçi i ve her değer v_i için $\beta_i(v_i)$ beklenen faydayı maksimize ediyorsa strateji profili bir **Bayes–Nash dengesidir** (*Bayes–Nash equilibrium*). Herkes aynı fonksiyonu kullanıyorsa denge **simetrik**tir.

Tanım, Nash dengesinin “kimse tek başına sapmak istemez” fikrini eksik bilgiye taşır. Fark, sapmanın beklenen fayda üzerinden değerlendirilmesidir: teklifçi rakiplerin değerlerini bilmediği için her teklifin ortalama sonucunu düşünür.



Şekil 5.10: İki teklifçi, değer $v = 0{,}8$; rakip değerinin yarısını teklif ediyor. Yeşil kesikli doğru kazanınca kalan kâr $v - b$, mavi kesikli doğru kazanma olasılığı $P(\text{kazan}) = 2b$ ($b \geq 0{,}5$ için 1), kırmızı eğri bunların çarpımı olan beklenen faydadır. Eğrinin tepesi $b = 0{,}4 = v/2$'dedir.

Şekil 5.10 ödünleşimi somutlaştırır. Rakibin değeri $U[0, 1]$ ve teklifi değerinin yarısıdır; yani rakibin teklifi 0 ile 0,5 arasında düzgün dağılır. Teklif b ile kazanmak için rakibin teklifi b'den küçük olmalıdır: $P(v_r/2 < b) = P(v_r < 2b) = 2b$ ($b \leq 0{,}5$ iken). Böylece beklenen fayda $(0{,}8 - b) \cdot 2b$ olur. Bu bir parabol parçasıdır; türevi $2(0{,}8 - b) - 2b = 1{,}6 - 4b$, $b = 0{,}4$'te sıfırdır. Tepe noktasında beklenen fayda $(0{,}8 - 0{,}4) \cdot 0{,}8 = 0{,}32$'dir.

Hollanda açık artırması bu çözümlemeyle stratejik olarak aynıdır. “Fiyat kaça inince kabul ederim?” sorusunun cevabı kırpılmış bir tekliftir. Değeri 0,8 olan teklifçi, rakibi aynı biçimde davrandığında fiyat 0,4'e inince kabul eder.

5.3.5 Denge teklifinin türetilmesi

Genel durumda n teklifçi vardır ve değerler bağımsız $U[0, 1]$'dir. Simetrik bir denge aranır. Bunun için tahmin edilen bir biçim varsayılır, en iyi tepki hesaplanır ve sonucun varsayımla tutarlı olduğu gösterilir. Varsayım şudur: rakipler değerlerinin sabit bir oranını teklif eder.

Örnek 5.7

1. **Varsayım.** Diğer $n - 1$ teklifçi $b_j = \beta \cdot v_j$ teklif etsin; burada β 0 ile 1 arasında bir sabittir ve $v_j \sim U[0, 1]$'dir.
2. **Kazanma olasılığı.** Teklif b ile kazanmak için her rakibin teklifi b'den küçük olmalıdır. Bir rakip için $P(\beta\, v_j < b) = P(v_j < b/\beta) = b/\beta$'dır ($b \leq \beta$ iken; düzgün dağılımda olasılık aralığın uzunluğudur). Rakipler bağımsız olduğundan olasılıklar çarpılır: $P(\text{kazan}) = (b/\beta)^{n-1}$.
3. **Beklenen fayda.** Kazanınca kâr çarpı kazanma olasılığı: $E[u(b)] = (v - b) \cdot (b/\beta)^{n-1}$.

4. **Türev.** $(v - b)$'nin türevi -1, b^{n-1}'in türevi $(n-1)\,b^{n-2}$'dir. Çarpım kuralıyla toplanır ve ortak çarpan b^{n-2}/β^{n-1} dışarı alınır: $dE[u(b)]/db = [\,-b + (n-1)(v-b)\,] \cdot b^{n-2}/\beta^{n-1}$. En büyük noktada türev sıfırdır.
5. **Çözüm.** Dışarıdaki çarpan $b > 0$ iken pozitiftir; demek ki köşeli parantez sıfır olmalıdır: $-b + (n-1)(v-b) = 0$. Açılırsa $-b + (n-1)v - (n-1)b = 0$, yani $n\,b = (n-1)v$ ve $b = (n-1)v/n$. Sadeleşince çözüm β'dan bağımsız çıkar.
6. **Tutarlılık.** Bulunan en iyi tepki "değerin $(n-1)/n$ katını teklif et"tir. Bu, varsayılan biçimle aynıdır ve $\beta = (n-1)/n$ seçilirse herkes rakiplerine en iyi tepkiyi vermiş olur: simetrik Bayes–Nash dengesi. $n = 2$ için $b = v/2$: iki kişide değerin yarısı.

Beşinci adımda bulunan çözümün en büyük nokta olduğu, türevin işaretine bakarak da doğrulanabilir. Köşeli parantez $(n-1)v - n\,b$'ye eşittir. b denge teklifinden küçükken pozitif, büyükken negatiftir. Yani beklenen fayda önce artar, denge teklifinde tepe yapar, sonra azalır. Bulunan nokta gerçekten en büyük noktadır. Ayrıca $b > \beta$ teklifleri her zaman kazanır, ama $b = \beta$ de her zaman kazandığından bu teklifler yalnızca daha çok ödeme demektir; en iyi teklif bu bölgeye düşmez.

Dengede değeri v olan teklifçinin beklenen faydası da hesaplanabilir. Rakipler $\beta\,v_j$ teklif ederken teklifçi $\beta\,v$ teklif eder; bu teklif ancak her rakibin değeri v'den küçükse kazanır. Kazanma olasılığı v^{n-1}, kâr $v - (n-1)v/n = v/n$'dir. Çarpım v^n/n'dir. $n = 2,\ v = 0{,}8$ için $0{,}64/2 = 0{,}32$, Şekil 5.10'daki tepe değeriyle aynıdır.

Formül sayısal olarak da denetlenebilir. $n = 2, 3, 5, 10$ ve $v = 0{,}3;\ 0{,}6;\ 0{,}8$ için $E[u(b)]$ 10.001 noktalı bir ızgarada maksimize edildiğinde en iyi teklif her durumda $(n-1)v/n$ ile çakışır.

Sık yapılan hata

Denge teklifini rakiplerin davranışından bağımsız bir kural sanmak. $b = (n-1)v/n$ yalnızca rakipler de aynı kuralı kullandığında ve değerler $U[0, 1]$ olduğunda en iyidir. Rakip örneğin dürüst teklif verseydi, kazanma olasılığı $P(v_r < b) = b$ olurdu ve en iyi tepki $(v-b)\cdot b$'yi maksimize eden yine $b = v/2$ çıkardı; ama dağılım değişince formül de değişir. Vickrey'deki dürüst teklif ise her durumda en iyidir; iki sonucun gücü aynı değildir.

5.3.6 Kod izleme: en_iyi_teklif()

Türetmenin sonucu, bir bilgisayar programının kaba kuvvetle bulduğu en iyi teklifle karşılaştırılabilir. Aşağıdaki fonksiyon, rakipler $\beta\,v$ teklif ederken değeri v olan teklifçinin beklenen faydasını $0,\ adim,\ 2{\cdot}adim,\ \ldots$ ızgarasındaki her teklif için hesaplar ve en büyüğünü döndürür.

```
def en_iyi_teklif(v, n, adim=0.1):
  beta = (n - 1) / n        # rakipler beta*v
  en, b_en = -1, 0
  b = 0.0
  while b <= v:
    p = min(1, b / beta) ** (n - 1)  # P(kazan)
    eu = (v - b) * p
    if eu > en:
      en, b_en = eu, b
    b = round(b + adim, 10)
  return b_en, en
```

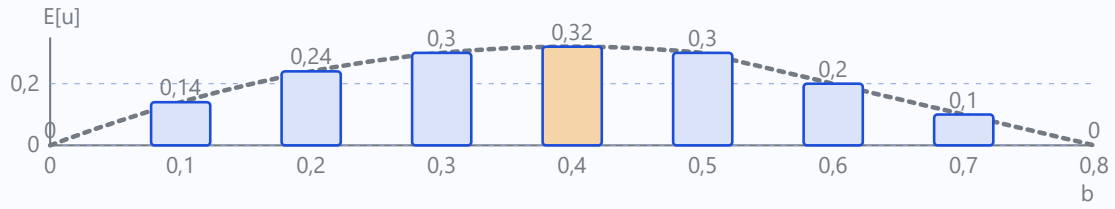
Kod satır satır şöyle çalışır:

- `beta = (n - 1) / n`: rakiplerin değerlerinin hangi oranını teklif ettiği. *n = 2* için 0,5'tir.
- `en, b_en = -1, 0`: şimdiye kadarki en büyük beklenen fayda ve onu veren teklif. Başlangıçta `en` −1'dir; beklenen fayda hiçbir zaman negatif olmadığı için ilk teklif kesin olarak "yeni en iyi" olur.
- `while b <= v`: teklifler 0'dan başlayıp değere kadar denenir. Değerin üstünde teklif vermek, kazanınca negatif fayda verdiği için anlamsızdır.
- `p = min(1, b / beta) ** (n - 1)`: kazanma olasılığı $(b/\beta)^{n-1}$. `min(1, …)` koruması, $b > \beta$ iken oranın 1'i geçmesini engeller; olasılık 1'den büyük olamaz.
- `eu = (v - b) * p`: beklenen fayda. Daha büyükse en iyi güncellenir.
- `b = round(b + adim, 10)`: bir sonraki teklif. `round`, 0,1'in ikilik tabanda tam gösterilememesinden doğan küçük hataları (ör. 0,30000000000000004) temizler; aksi hâlde son adımda `b` 0,8'i çok az aşıp döngü erken bitebilirdi.

`en_iyi_teklif(0.8, 2)` çağrısında $\beta = 0{,}5$'tir ve ızgara $b = 0;\ 0{,}1;\ \ldots;\ 0{,}8$'dir. Tablo 5.5 her turu gösterir.

b	P(kazan)	E[u] = (0,8 − b) · P	Karşılaştırma	(en, b_en)
0	0	0,8 · 0 = 0	0 > −1: yeni en iyi	(0; 0)
0,1	0,2	0,7 · 0,2 = 0,14	yeni en iyi	(0,14; 0,1)
0,2	0,4	0,6 · 0,4 = 0,24	yeni en iyi	(0,24; 0,2)
0,3	0,6	0,5 · 0,6 = 0,30	yeni en iyi	(0,30; 0,3)
0,4	0,8	0,4 · 0,8 = 0,32	yeni en iyi	(0,32; 0,4)
0,5	1	0,3 · 1 = 0,30	0,30 < 0,32: değişmez	(0,32; 0,4)
0,6	1	0,2 · 1 = 0,20	değişmez	(0,32; 0,4)
0,7	1	0,1 · 1 = 0,10	değişmez	(0,32; 0,4)
0,8	1	0 · 1 = 0	değişmez	(0,32; 0,4)

Tablo 5.5: `en_iyi_teklif(0.8, 2)` izlemesi. *P(kazan) = min(1, b/0,5).*



Şekil 5.11: `en_iyi_teklif(0.8, 2)` izlemesinin sonu. Çubuklar her *b* için beklenen faydadır; en yüksek çubuk (turuncu) *b = 0,4*'te, *E[u] = 0,32*'dir.

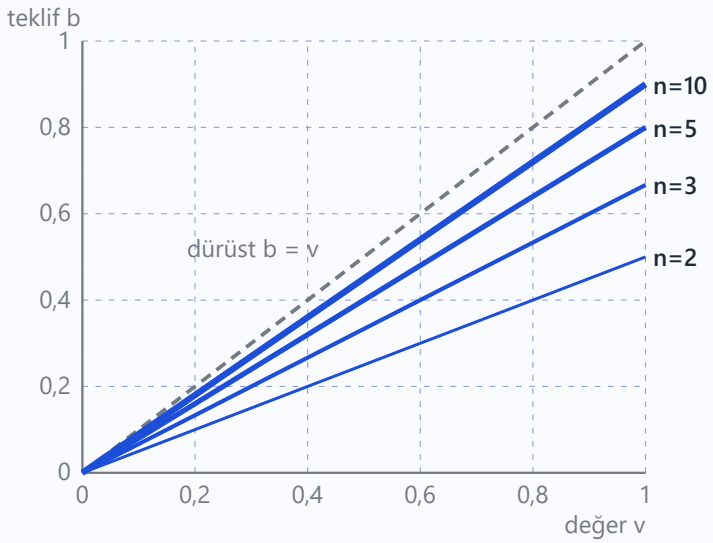
Fonksiyon `(0.4, 0.32)` döndürür: en iyi teklif *b = 0,4 = v/2*, beklenen fayda 0,32. Sonuç türetmeyle aynıdır. Şekil 5.11 ödünleşimin iki ucunu gösterir: düşük teklif kazanmayı, yüksek teklif kârı öldürür. *b = 0,5*'ten sonra kazanma olasılığı 1'e ulaştığı için artık yalnızca kâr azalır ve çubuklar doğrusal olarak kısalır.

5.3.7 Rekabet arttıkça kırpma azalır

Denge teklifi *b(v) = (n − 1)v/n*, teklifçi sayısı *n*'ye bağlıdır. Kırpma miktarı, yani değer ile teklif arasındaki fark *v − b(v) = v/n*'dir. Teklifçi sayısı arttıkça kırpma küçülür.

n	b(v)	v = 0,8 → b	kırpma v − b
2	1/2 · v	0,4	0,4
3	2/3 · v	0,53	0,27
5	4/5 · v	0,64	0,16
10	9/10 · v	0,72	0,08

Tablo 5.6: Teklifçi sayısına göre denge teklifi ve kırpma (değer *v = 0,8*).

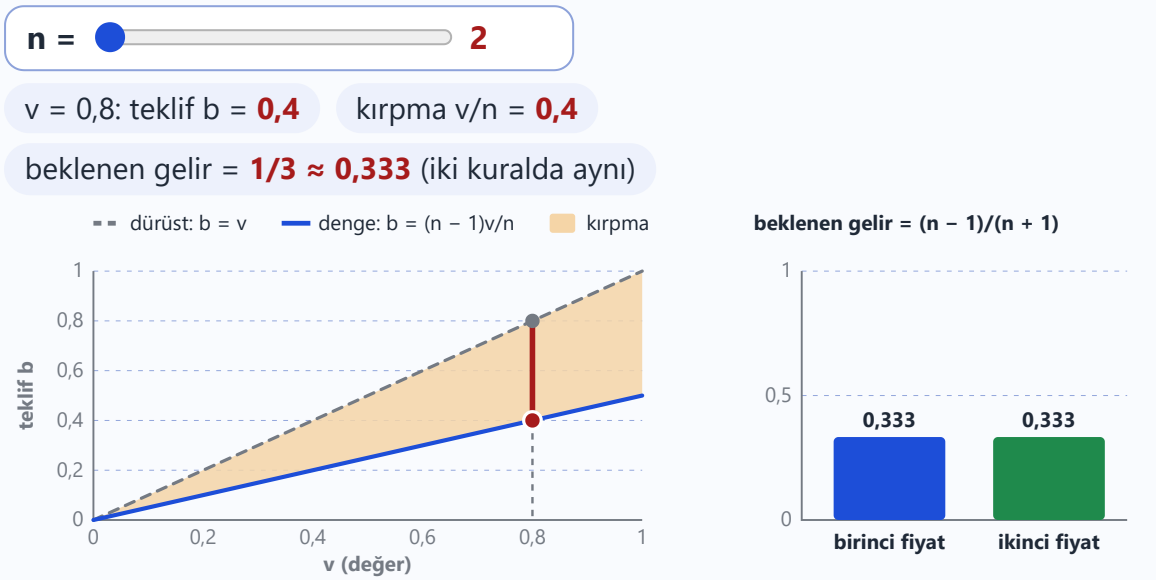


Şekil 5.12: Denge teklifi doğruları *b = (n − 1)v/n*, *n = 2, 3, 5, 10* için. Kesikli doğru dürüst teklif *b = v*'dir. *n* büyü-

dükçe doğrular dürüst teklife yaklaşır.

Şekil 5.12 sonucun mantığını gösterir. Çok teklifçi varken rakiplerden birinin değeri büyük olasılıkla teklifçinin değerine yakındır; kazanmak için değere yakın teklif vermek gerekir. Az teklifçi varken rakibin değeri büyük olasılıkla çok daha düşüktür ve teklif rahatça kırpılabilir. Limitte $n \to \infty$ iken $b \to v$ olur: kalabalık bir piyasada birinci fiyat kuralı da teklifçileri neredeyse dürüst davranmaya zorlar.

Aşağıdaki etkileşimli şekil aynı ilişkiyi teklifçi sayısı değiştikçe gösterir. Solda dürüst teklif ve denge teklifi doğruları ile aradaki kırpma bölgesi, sağda iki kuralın beklenen geliri vardır. Varsayılan değer $n = 2$'dir.



Şekil 5.13: Etkileşimli teklif kırpma. Kaydırıcı teklifçi sayısı n'yi 2'den 10'a değiştirir. Solda $v = 0{,}8$ için denge teklifi ve kırpma v/n, sağda birinci ve ikinci fiyat kurallarının beklenen geliri $(n-1)/(n+1)$ görünür. Varsayılan $n = 2$'de teklif 0,4, kırpma 0,4 ve beklenen gelir iki kuralda da 1/3'tür.

Şekil 5.13'te dikkat çeken ikinci olgu sağdaki çubuklardır: birinci ve ikinci fiyat kurallarının beklenen geliri her n için eşittir ve n arttıkça artar. $n = 2$'de 1/3, $n = 3$'te 1/2, $n = 10$'da 9/11'dir. Bu eşitliğin nedeni gelir eşdeğerliğidir (5.4.2).

Örnek 5.8

$n = 4$ teklifçi var ve değerler $U[0, 1]$. Değeri 0,6 olan teklifçinin birinci fiyattaki denge teklifi, kırpma miktarı ve beklenen faydası nedir?

1. Denge teklifi: $b = (4 - 1)/4 \cdot 0{,}6 = 0{,}75 \cdot 0{,}6 = 0{,}45$.
2. Kırpma: $v - b = 0{,}6 - 0{,}45 = 0{,}15 = v/n$.
3. Kazanma olasılığı: üç rakibin de değeri 0,6'dan küçük olmalı: $0{,}6^3 = 0{,}216$.

4. Beklenen fayda: $0{,}15 \cdot 0{,}216 = 0{,}0324 = 0{,}6^4/4$.

5.4 Gelir ve bilgi

Buraya kadar soru teklifçinin bakış açısından soruldu: ne teklif etmeliyim? Satıcının sorusu ise farklıdır: hangi kural bana daha çok kazandırır? Cevap şaşırtıcıdır: standart varsayımlar altında bütün temel kurallar ortalamada aynı geliri getirir. Bu eşitliğin hangi varsayımlar bozulunca ortadan kalktığı, satıcının gelirini nasıl artırabileceği (rezerv fiyatı) ve bilginin ortak olduğu durumda ortaya çıkan kazananın laneti de aynı çerçevede ele alınır.

5.4.1 İki teklifçide beklenen gelir

En basit durumla başlanır: iki teklifçi vardır ve değerleri v_1, v_2 bağımsız $U[0, 1]$'dir. Satıcı ikinci fiyat kuralını mı, birinci fiyat kuralını mı seçmelidir? Her kuralda önce satıcının bir açık artırmadaki geliri yazılır, sonra bu gelirin beklentisi hesaplanır.

Örnek 5.9

1. **İkinci fiyat: gelir.** Herkes dürüst teklif verir (Teorem 5.1). Kazanan büyük değerli teklifçidir ve küçük değeri öder. Satıcının geliri $\min(v_1, v_2)$'dir.
2. **İkinci fiyat: beklenti.** İki değerin küçüğü t'yi aşıyorsa ikisi de t'yi aşar. Bağımsızlıktan $P(\min > t) = (1 - t) \cdot (1 - t) = (1 - t)^2$. Kuyruk formülüyle $E[\min] = \int_0^1 (1 - t)^2\, dt$. Değişken $s = 1 - t$ alınırsa integral $\int_0^1 s^2\, ds = 1/3$ olur.
3. **Birinci fiyat: gelir.** Dengede herkes $b = v/2$ teklif eder (5.3.5). Kazanan yine büyük değerli teklifçidir ve kendi teklifini, yani büyük değerin yarısını öder. Gelir $\max(v_1, v_2)/2$'dir.
4. **Birinci fiyat: beklenti.** İki değerin büyüğü t'den küçükse ikisi de küçüktür: $P(\max < t) = t^2$, yani $P(\max > t) = 1 - t^2$. Kuyruk formülüyle $E[\max] = \int_0^1 (1 - t^2)\, dt = 1 - 1/3 = 2/3$. Gelir bunun yarısıdır: $2/3 \cdot 1/2 = 1/3$.
5. **İngiliz ve Hollanda.** İngiliz açık artırması özel değerlerde ikinci fiyatla aynı sonucu verir: 1/3. Hollanda açık artırması birinci fiyatla stratejik olarak özdeştir: 1/3.
6. **Sonuç.** Dört kural aynı beklenen geliri verir: 1/3.

Kural	Satıcının geliri	Beklenen gelir
İkinci fiyat	$\min(v_1, v_2)$	$E[\min] = \int_0^1 (1 - t)^2\, dt = 1/3$
Birinci fiyat	$\max(v_1, v_2)/2$	$E[\max] = \int_0^1 (1 - t^2)\, dt = 2/3 \to 1/3$
İngiliz	≈ ikinci fiyat	1/3
Hollanda	≡ birinci fiyat	1/3

Kural	Satıcının geliri	Beklenen gelir
Simülasyon	200 bin çekiliş	ikinci fiyat 0,333 · birinci fiyat 0,333

Tablo 5.7: İki teklifçi, değerler bağımsız *U[0, 1]*: dört kuralın beklenen geliri.

Son satır bir Monte Carlo simülasyonunun sonucudur: bilgisayar 200 bin kez iki rastgele değer çeker, her çekilişte iki kuralın gelirini hesaplar ve ortalamasını alır. İki ortalama da 0,333'e çok yakındır. Simülasyon, integral hesabını bilmeyen biri için de sonucu doğrulamanın yoludur.

Eşitlik yalnızca *beklenen* gelir için geçerlidir. Tek tek açık artırmalarda gelirler farklıdır. Değerler 0,9 ve 0,2 ise ikinci fiyat 0,2, birinci fiyat *0,9/2 = 0,45* getirir. Değerler 0,6 ve 0,5 ise ikinci fiyat 0,5, birinci fiyat 0,3 getirir. Değerler birbirine yakınken ikinci fiyat, uzakken birinci fiyat daha çok kazandırır. Ortalamada bu farklar tam olarak birbirini götürür.

5.4.2 Gelir eşdeğerliği teoremi

İki teklifçideki eşitlik rastlantı değildir; daha genel bir teoremin özel durumudur.

Teorem 5.2 (Gelir eşdeğerliği)

Bağımsız özel değerler, risk nötr ve simetrik teklifçiler ve aynı tahsis kuralı (en yüksek değerli kazanır) altında, en düşük değerli tipin ödemesi sıfır olan standart açık artırmalar aynı beklenen geliri sağlar (Vickrey, 1961; Myerson, 1981; Riley ve Samuelson, 1981).

Teoremin neden doğru olduğu Myerson'un ödeme formülüyle açıklanır. Bir açık artırmada değeri v olan teklifçinin kazanma olasılığı $x(v)$, beklenen ödemesi $p(v)$ olsun. Teklifçiler dengede davranıyorsa bu ikisi arasında şu ilişki vardır:

$$p(v) = v \cdot x(v) - \int_0^v x(z)\, dz$$

Formülün sağ tarafında yalnızca tahsis kuralı $x(\cdot)$ vardır. Formülün arkasındaki fikir mekanizma tasarımındaki Myerson lemmasıdır: değeri v olan bir teklifçi değeri daha düşükmüş gibi davranmayı da seçebileceğinden, dengede beklenen faydası v arttıkça tam olarak kazanma olasılığı hızında artmak zorundadır. Bu koşul, en düşük tipin faydası (burada 0) sabitlenince beklenen ödemeyi tek biçimde belirler. Sonuç şudur: aynı $x(\cdot) \to$ aynı $p(\cdot) \to$ aynı beklenen gelir. Standart açık artırmaların hepsinde en yüksek değerli teklifçi kazandığı için $x(\cdot)$ aynıdır; bu yüzden gelirleri de aynıdır.

Örnek 5.10

İki teklifçi, *U[0, 1]*. Değeri v olan teklifçinin beklenen ödemesi üç yoldan hesaplanır.

1. **Formülle.** Kazanmak için rakibin değeri v'den küçük olmalı: $x(v) = v$. O hâlde $p(v) = v \cdot v - \int_0^v z\, dz = v^2 - v^2/2 = v^2/2$.

2. **Birinci fiyatta.** Teklif $v/2$, kazanma olasılığı v; kazanınca teklif ödenir. Beklenen ödeme $v/2 \cdot v = v^2/2$.
3. **İkinci fiyatta.** Kazanma olasılığı v; kazanınca rakibin değeri ödenir. Rakibin değeri v'den küçük olduğu bilindiğinde $[0, v]$'de düzgün dağılır ve ortalaması $v/2$'dir. Beklenen ödeme $v \cdot v/2 = v^2/2$.
4. **Gelire geçiş.** Üç yol aynı sonucu verir. Satıcının beklenen geliri iki teklifçinin beklenen ödemelerinin toplamıdır: $2 \cdot E[v^2/2] = E[v^2] = \int_0^1 t^2\, dt = 1/3$.

Aynı hesap n teklifçi için de yapılabilir. Bu durumda $x(v) = v^{n-1}$, $p(v) = v^n - v^n/n = (n-1)v^n/n$ ve beklenen gelir $n \cdot (n-1)/n \cdot E[v^n] = (n-1)/(n+1)$ çıkar. Aynı sonuca ikinci fiyat yönünden de ulaşılır: n bağımsız $U[0, 1]$ değerinin ikinci büyüğünün beklentisi $(n-1)/(n+1)$'dir. Birinci fiyat yönünden ise en büyük değerin beklentisi $n/(n+1)$ ile teklif oranı $(n-1)/n$'nin çarpımı yine $(n-1)/(n+1)$'dir.

n	Kuram	İkinci fiyat (simülasyon)	Birinci fiyat (simülasyon)
2	1/3 = 0,33	0,332	0,333
3	1/2 = 0,5	0,500	0,500
4	3/5 = 0,6	0,600	0,600
5	2/3 = 0,67	0,668	0,667

Tablo 5.8: Beklenen gelir $(n-1)/(n+1)$ ve simülasyon sonuçları.

Tablo 5.8 iki sonucu birlikte gösterir: her n için iki kural aynı geliri getirir ve gelir n ile artar. Simülasyon değerleri kuramdan üçüncü basamakta ayrılır; bu, sonlu sayıda çekilişin doğal rastgele hatasıdır.

Teoremin varsayımları önemlidir. Riskten kaçınma, bütçe kısıtları, ortak değerler ya da asimetrik bilgi sonucu değiştirebilir. Teklifçiler simetrik değilse (örneğin birinin değeri genellikle daha yüksekse) birinci fiyatta en yüksek değerli kişi her zaman kazanmayabilir; tahsis kuralı değişir ve eşitlik bozulur. Teorem bu yüzden "hangi kural daha çok kazandırır?" sorusunu ortadan kaldırmaz; sorunun cevabını varsayımlara taşır. Pratikte açık artırma biçimi yalnızca gelirle değil; sadelik, şeffaflık ve katılım teşvikiyle birlikte seçilir.

Sık yapılan hata

Gelir eşdeğerliğini "her açık artırmada iki kural aynı fiyatı verir" diye okumak. Teorem *beklenen* gelir hakkındadır; tek bir açık artırmada gelirler çoğu zaman farklıdır (5.4.1'deki 0,9 ve 0,2 örneği). Bir başka hata da teoremi bütün kurallara genellemektir: nesneyi rastgele birine veren ya da en yüksek değerliye vermeyen bir kural farklı tahsis yapar ve farklı gelir getirir.

5.4.3 Varsayımı değiştirmek: riskten kaçınma

Gelir eşdeğerliğinin bir varsayımı teklifçilerin risk nötr olmasıdır. Bu varsayım kaldırılınca ne olur? İki teklifçi ve *U[0, 1]* değerlerle bir model kurulur. Kazananın kârı $x = v - fiyat$, faydası ise

$$u(x) = x^{\rho},\ 0 < \rho \le 1$$

olsun. $\rho = 1$ risk nötr teklifçidir: fayda kârın kendisidir. $\rho < 1$ riskten kaçınan teklifçidir; örneğin $\rho = ½$ iken $u(x) = \sqrt{x}$'tir. Karekök eğrisi küçük kârlarda dik, büyük kârlarda yataydır. Bu, riskten kaçınan bir kişinin "az ama kesin" bir kârı "çok ama belirsiz" bir kâra tercih etmesi demektir: ilk birkaç liralık kâr ona çok değerli gelir, fazlası o kadar değil.

İkinci fiyat değişmez. Fiyatı teklif değil, rakibin teklifi belirler. 5.3.2'deki ispat faydanın kârla artan bir fonksiyon olmasından başka bir şey kullanmadı; bu yüzden dürüst teklif hâlâ zayıf baskındır. Gelir yine $E[min] = 1/3$'tür.

Birinci fiyatta teklif yükselir. Rakip $\beta\, v_r$ teklif ediyorsa teklif b ile kazanma olasılığı b/β'dır ve beklenen fayda $(v - b)^{\rho} \cdot b/\beta$'dır. Türev alınırsa $[-\rho\,(v - b)^{\rho-1}\, b + (v - b)^{\rho}]/\beta$ bulunur. Sıfıra eşitleyip $(v - b)^{\rho-1}$ ortak çarpanı atılırsa $-\rho\, b + (v - b) = 0$, yani

$$b(v) = v / (1 + \rho)$$

elde edilir. $\rho = 1$ iken $v/2$, bildiğimiz risk nötr dengedir; $\rho = ½$ iken $2v/3$'tür. Riskten kaçınan teklifçi kaybetme riskini azaltmak için daha yüksek teklif verir; bir anlamda kazanma olasılığını artırmak için "sigorta primi" öder.

Beklenen gelir	ρ = 1	ρ = ½
İkinci fiyat	1/3 ≈ 0,333	1/3 ≈ 0,333
Birinci fiyat (E[max] = 2/3)	(2/3) · (1/2) = 1/3	(2/3) · (2/3) = 4/9 ≈ 0,444

Tablo 5.9: Riskten kaçınmanın beklenen gelire etkisi (iki teklifçi, *U[0, 1]*).

Tablo 5.9 sonucu özetler: riskten kaçınma gelir eşdeğerliğini bozar ve birinci fiyat kuralı satıcıya daha çok kazandırır. Satıcı teklifçilerin riskten kaçındığını düşünüyorsa birinci fiyat kuralını (ya da onunla özdeş Hollanda açık artırmasını) seçmek için bir nedeni vardır.

Örnek 5.11

İki teklifçi, $\rho = ½$. Değeri 0,6 olan teklifçi birinci fiyatta ne teklif eder ve kazanırsa faydası nedir?

1. Denge teklifi: $b = 0{,}6/(1 + 0{,}5) = 0{,}4$. Risk nötr teklifçi *0,3* teklif ederdi.
2. Kazanma olasılığı: rakip $2v_r/3$ teklif eder; $2v_r/3 < 0{,}4$ için $v_r < 0{,}6$ gerekir: olasılık 0,6.
3. Kazanınca kâr $0{,}6 - 0{,}4 = 0{,}2$, fayda $\sqrt{0{,}2} \approx 0{,}447$. Beklenen fayda $0{,}6 \cdot 0{,}447 \approx 0{,}268$.

5.4.4 Gelir maksimizasyonu: Myerson ve rezerv fiyatı

Gelir eşdeğerliği standart açık artırmaların aynı geliri getirdiğini söyler. Peki satıcı verimliliği bir yana bırakıp yalnızca beklenen geliri maksimize etmek isterse ne yapmalıdır? Bu soruyu Roger Myerson cevapladı (Myerson, 1981). Cevabın anahtarı sanal değer kavramıdır.

Tanım 5.5

Değer dağılımı F, yoğunluğu f olan bir teklifçinin **sanal değeri** (*virtual value*)

$$\psi(v) = v - (1 - F(v)) / f(v)$$

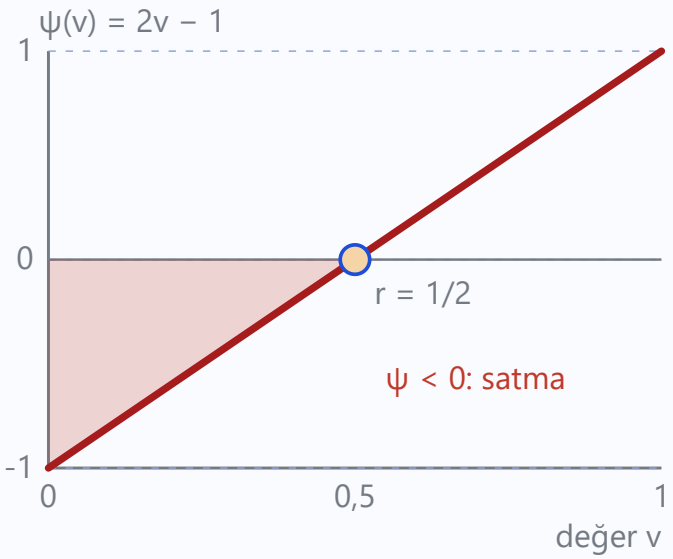
dir. $F(v)$ değerin v'den küçük olma olasılığı, $f(v)$ yoğunluktur. $U[0, 1]$'de $F(v) = v$, $f(v) = 1$ olduğundan $\psi(v) = v - (1 - v) = 2v - 1$'dir.

Sanal değer, satıcının bir teklifçiden gerçekte ne kadar gelir çıkarabileceğini ölçer. Değer v tam olarak alınamaz, çünkü teklifçi değerini saklayabilir ve satıcı ona dürüst olması için bir "bilgi kirası" bırakmak zorundadır. Çıkarılan terim $(1 - F)/f$ bu kiradır. Myerson'un temel sonucu şudur: *beklenen gelir, kazananın sanal değerinin beklentisine eşittir*. O hâlde geliri en büyük yapan mekanizma nesneyi sanal değeri en büyük olan teklifçiye verir; en büyük sanal değer bile negatifse nesneyi satmaz, çünkü negatif sanal değerli birine satmak beklenen geliri azaltır.

Sanal değer artan bir fonksiyonsa ve teklifçiler simetrikse "en büyük sanal değer" ile "en büyük değer" aynı kişidir. Tek fark, sanal değeri negatif olanlara satılmamasıdır. Bu, **rezerv fiyatlı ikinci fiyat açık artırmasıdır**: satıcı önceden bir **rezerv fiyat** (*reserve price*) r ilan eder; r'nin altındaki teklifler geçersizdir. Rezerv, sanal değerin sıfır olduğu noktadır:

$$\psi(r) = 0 \Rightarrow 2r - 1 = 0 \Rightarrow r = 1/2$$

Bu eşik yalnızca değer dağılımına bağlıdır; teklifçi sayısı n'den bağımsızdır. İki teklifçi için de, on teklifçi için de $U[0, 1]$'de en iyi rezerv $1/2$'dir.



Şekil 5.14: *U[0, 1]*'de sanal değer *ψ(v) = 2v − 1*. Doğru *v = 1/2*'de sıfırı keser. Pembe bölgede *ψ < 0*'dır: değeri 1/2'nin altındaki teklifçiye satış beklenen geliri azaltır; satıcı satmaz.

Rezervli ikinci fiyat açık artırmasının kuralları şöyledir. En yüksek teklif *r*'nin altındaysa nesne satılmaz. *r*'nin üstünde yalnızca en yüksek teklif varsa kazanan *r* öder. İki teklif de *r*'nin üstündeyse kazanan ikinci en yüksek teklifi öder. Kısaca kazanan *max(r, ikinci en yüksek teklif)* öder. Rezerv, satıcının kendisinin de bir teklif verdiği gibidir; dürüst teklif bu kuralda da zayıf baskındır.

İki teklifçide rezervli ikinci fiyat açık artırmasının beklenen geliri sanal değer sonucuyla hesaplanabilir. Gelir, kazananın sanal değerinin beklentisidir ve nesne yalnız en büyük değer *r*'yi aşınca satılır. İki değerin büyüğünün yoğunluğu *2t*'dir (*P(max < t) = t^2*'nin türevi). O hâlde

$$E[\text{gelir}] = \int_r^1 (2t - 1) \cdot 2t \, dt = [4t^3/3 - t^2]_r^1 = 1/3 + r^2 - 4r^3/3$$

bulunur. *r = 0* iken rezervsiz ikinci fiyatın geliri 1/3 çıkar. *r = 1/2* iken *1/3 + 1/4 − 1/6 = 5/12 ≈ 0,417*'dir. Rezerv, geliri yaklaşık %25 artırır. Bunun bir bedeli vardır: bazen hiç satış olmaz ve nesneye değer veren bir alıcı olduğu hâlde nesne satıcıda kalır. Bu verimlilik kaybı 5.4.6'da hesaplanır.

Örnek 5.12

İki teklifçi, *U[0, 1]*, rezerv *r = 1/2*. Üç değer çifti için sonuç bulunur.

1. **(0,8; 0,3):** yalnız 0,8 rezervi geçer. Kazanan *max(0,5; 0,3) = 0,5* öder. Rezervsiz olsaydı 0,3 öderdi.
2. **(0,9; 0,7):** ikisi de rezervi geçer. Kazanan *max(0,5; 0,7) = 0,7* öder; rezerv bu açık artırmayı etkilemez.
3. **(0,4; 0,2):** ikisi de rezervin altındadır; nesne satılmaz. Rezervsiz olsaydı 0,4'lük teklifçi 0,2'ye alırdı.

Rezerv, rakibin zayıf olduğu durumlarda fiyatı yukarı çeker (1. çift), ama iki teklifçi de zayıfsa satışı kaybettirir (3. çift). Ortalamada ilk etki baskındır.

5.4.5 Kod izleme: en_iyi_rezerv()

En iyi rezervi sayısal olarak bulmak için iki fonksiyon yazılabilir. `gelir(r)`, rezervi *r* olan iki teklifçili ikinci fiyat açık artırmasının beklenen gelirini bir ızgara üzerinde yaklaşık hesaplar; `en_iyi_rezerv()` ise *r = 0; 0,1; …; 1* değerlerini dener.

```
def gelir(r, m=100):      # 2 teklifçi, U[0,1]
  v = [(i + .5) / m for i in range(m)]
  top = sum(max(min(a, c), r) for a in v
            for c in v if max(a, c) >= r)
  return top / m**2       # ortalama ödeme
def en_iyi_rezerv(adim=0.1):
  en, r_en = -1, None
  for k in range(round(1/adim) + 1):
    R = gelir(k * adim)
    if R > en: en, r_en = R, k * adim
  return r_en, en
```

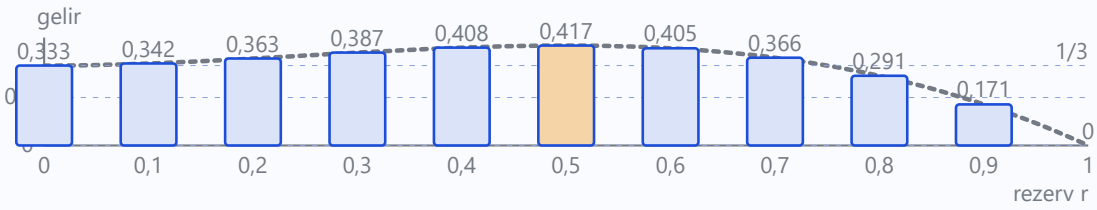
Kod satır satır şöyle çalışır:

- `v = [(i + .5) / m for i in range(m)]` : *[0, 1]* aralığı *m = 100* eşit parçaya bölünür ve her parçanın orta noktası alınır: 0,005; 0,015; …; 0,995. Bu noktalar düzgün dağılımı temsil eder.
- İç içe iki döngü bütün *(a, c)* değer çiftleri üzerinde gezinir; *100 × 100 = 10.000* çift vardır ve her biri eşit olasılıklıdır.
- `if max(a, c) >= r` : en büyük değer rezervi geçmiyorsa satış yoktur ve gelir 0'dır; bu çiftler toplama katılmaz.
- `max(min(a, c), r)` : satış varsa kazanan, küçük değer ile rezervin büyüğünü öder.
- `top / m**2` : toplam ödeme çift sayısına bölünür; satışsız çiftler 0 katkıyla sayılmış olur. Sonuç beklenen gelirin yaklaşığıdır.
- `en_iyi_rezerv` : `k = 0, 1, …, 10` için `r = k · adim` dener. `range(round(1/adim) + 1)`, *r = 1*'in de denenmesini sağlar. Daha büyük gelir bulununca en iyi güncellenir.

r	gelir(r)	Karşılaştırma	(en, r_en)	Formül
0	0,333	0,333 > −1: yeni en iyi	(0,333; 0)	0,333
0,1	0,342	yeni en iyi	(0,342; 0,1)	0,342
0,2	0,363	yeni en iyi	(0,363; 0,2)	0,363
0,3	0,387	yeni en iyi	(0,387; 0,3)	0,387
0,4	0,408	yeni en iyi	(0,408; 0,4)	0,408

r	gelir(r)	Karşılaştırma	(en, r_en)	Formül
0,5	0,417	yeni en iyi	(0,417; 0,5)	0,417
0,6	0,405	< 0,417: değişmez	(0,417; 0,5)	0,405
0,7	0,366	değişmez	(0,417; 0,5)	0,366
0,8	0,291	değişmez	(0,417; 0,5)	0,291
0,9	0,171	değişmez	(0,417; 0,5)	0,171
1	0	değişmez	(0,417; 0,5)	0

Tablo 5.10: `en_iyi_rezerv()` izlemesi. Son sütun formül $1/3 + r^2 - 4r^3/3$'tür.



Şekil 5.15: `en_iyi_rezerv()` izlemesinin sonu. Çubuklar her rezerv için beklenen gelirdir; en yüksek çubuk (turuncu) $r = 0{,}5$'te, gelir ≈ 0,417'dir. Kesikli eğri formüldür.

Fonksiyon $r = 0{,}5$ ve gelir ≈ 0,417 döndürür; bu, kuramsal sonuç 5/12 ile aynıdır. Şekil 5.15 gelirin rezerve göre önce artıp sonra azaldığını gösterir. Rezervsiz durumda ($r = 0$) gelir 1/3'tür. Yüksek rezervler sık sık satışsız kalır: $r = 1$'de nesne hiç satılmaz ve gelir 0'dır. Izgara hesabının formülle bu kadar iyi uyuşmasının nedeni orta nokta kuralının düzgün dağılımda çok küçük hata vermesidir.

5.4.6 Rezervin bedeli

Rezerv satıcının gelirini artırır, ama toplumun nesneden elde ettiği toplam faydayı, yani **sosyal refahı**, azaltabilir. İki teklifçi ve rezerv r ile üç büyüklük hesaplanır.

- **Beklenen gelir:** $1/3 + r^2 - 4r^3/3$ (5.4.4).
- **Hiç satılmama olasılığı:** iki değerin ikisi de r'nin altında olmalıdır: $r \cdot r = r^2$.
- **Beklenen sosyal refah kaybı:** verimli sonuçta nesneyi değeri en büyük teklifçi alır ve refah bu değere eşittir. Bu değer r'nin altındaysa nesne satılmaz ve değer kaybolur. En büyük değerin yoğunluğu $2t$ olduğundan kayıp $\int_0^r t \cdot 2t\, dt = 2r^3/3$'tür.

r	Beklenen gelir	Hiç satılmama olasılığı	Beklenen refah kaybı
0	1/3 ≈ 0,333	0	0
0,25	3/8 = 0,375	1/16 = %6,25	1/96 ≈ 0,010
0,5	5/12 ≈ 0,417	1/4 = %25	1/12 ≈ 0,083
0,75	1/3 ≈ 0,333	9/16 = %56,25	9/32 ≈ 0,281

Tablo 5.11: Rezervin gelir ve verimlilik etkisi (iki teklifçi, $U[0, 1]$).

Rezervsiz sosyal refah, en büyük değerin beklentisi olan $E[max] = 2/3$'tür. Tablo 5.11'e göre $r = 0{,}5$ geliri maksimize eder, ama nesne %25 olasılıkla satılmaz ve beklenen sosyal refah $2/3 - 1/12 = 7/12$'ye düşer. Kayıp $(1/12)/(2/3) = 1/8$, yani %12,5'tir. Satıcı ile toplumun çıkarı burada ayrışır: satıcı için en iyisi, toplum için en iyisi değildir.

Tablonun son satırı bir başka dersi de verir: rezervi aşırı yükseltmek geliri de düşürür. $r = 0{,}75$'te gelir rezervsiz duruma geri iner (1/3), ama nesne yarıdan fazla olasılıkla satılmaz ve refah kaybı çok büyüktür.

> **Sık yapılan hata**
>
> "Rezerv ne kadar yüksekse gelir o kadar yüksek" sanmak. Rezerv yükseldikçe kazananın ödemesi artar, ama satış olasılığı düşer. İki etki $r = 1/2$'de dengelenir; ötesinde ikinci etki baskın gelir. Bir diğer hata da en iyi rezervin teklifçi sayısıyla değiştiğini düşünmektir: $U[0, 1]$'de $\psi(r) = 0$ koşulu n'yi içermez.

5.4.7 Tümü öder ve çok birimli açık artırmalar

Tümü öder açık artırması (*all-pay auction*), en yüksek teklifin kazandığı ama *herkesin* teklifini ödediği bir kuraldır. İlk bakışta tuhaf görünür, ama birçok gerçek yarışmanın modelidir: lobi faaliyetleri, ar-ge yarışları, seçim kampanyaları. Bu yarışmaların hepsinde ödülü tek kişi kazanır, ama harcanan emek ve para kaybedenlere geri dönmez.

İki teklifçi ve $U[0, 1]$ değerlerle simetrik dengede teklif $b(v) = v^2/2$'dir. Bu sonuç gelir eşdeğerliğinden kolayca çıkar. Tümü öder kuralında da en yüksek değerli kazanır, yani tahsis kuralı aynıdır. O hâlde değeri v olan teklifçinin beklenen ödemesi birinci ve ikinci fiyattakiyle aynı, $v^2/2$'dir (5.4.2). Tümü öderde teklif her durumda ödendiği için beklenen ödeme teklifin kendisidir: $b(v) = v^2/2$. Değeri 0,8 olan teklifçi 0,32 teklif eder; birinci fiyatta teklifi 0,4 olurdu. Kazanma olasılığı düşük tipler neredeyse hiç teklif vermez: değeri 0,2 olan teklifçinin teklifi yalnızca 0,02'dir.

Satıcının beklenen geliri yine 1/3'tür. Bu, gelir eşdeğerliğinin güçlü bir gösterimidir: kurallar çok farklı görünse de en yüksek değerli kazandığı sürece ortalama gelir değişmez.

Değerler eşit ve herkesçe biliniyorsa tümü öder modelinin ilginç bir sonucu vardır. Örneğin iki firma değeri V olan bir ödül (ör. bir sözleşme) için lobi harcaması yapıyorsa, dengede beklenen toplam harcama ödülün değerine eşitlenir. Ödülün bütün değeri yarışmada "erir"; bu olguya rant kaybı denir ve kamu politikasında rekabetçi lobiciliğin maliyetini açıklar.

Çok birimli açık artırmalar (*multi-unit auction*) birden çok birimin aynı anda satıldığı açık artırmalardır. Birimler özdeşse (örneğin devlet tahvilleri) iki temel kural vardır. **Tek tip fiyatlı** açık artırmada (*uniform-price auction*) bütün kazananlar aynı fiyatı öder. **Ayrımcı fiyatlı** açık artırmada (*discriminatory (pay-as-bid) auction*) her kazanan kendi teklifini öder. Birimler birlikte daha değerliyse (örneğin komşu bölgelerin frekans lisansları) teklifçilere birim demetleri için *paket teklif* verme hakkı tanınır; bu tasarımların ödeme kuralları VCG mekanizmasına dayanır ve hesaplaması zordur.

Gelir artırmanın bir başka yolu daha vardır. Bulow ve Klemperer (1996), rezervsiz bir açık artırmaya bir ek teklifçi çekmenin, en iyi rezerv fiyatıyla yapılan açık artırmadan daha çok kazandırdığını gösterdi. *U[0, 1]* örneğinde bu açıkça görülür: iki teklifçi ve en iyi rezervle gelir *5/12 ≈ 0,417*'dir; üç teklifçi ve rezervsiz gelir ise *(3 − 1)/(3 + 1) = 1/2*'dir. Satıcı için rekabeti artırmak, kuralları ince ayarlamaktan daha değerli olabilir.

5.4.8 Kazananın laneti

Ortak değerli bir açık artırmada nesnenin gerçek değeri herkes için aynıdır, ama kimse bu değeri bilmez. Her katılımcı değeri tahmin eder ve tahminler hatalıdır: bazıları gerçek değerin üstünde, bazıları altındadır. Açık artırmada en yüksek teklif, büyük olasılıkla değeri en fazla tahmin eden katılımcıdan gelir. Bu katılımcı, kazandığı için kendi tahmin hatasının pozitif ve büyük olduğunu da öğrenmiş olur.

Tanım 5.6

Kazananın laneti (*winner's curse*): ortak değerli bir açık artırmada kazananın, tahmin hatasını hesaba katmadan teklif verdiği için nesnenin gerçek değerinden fazla ödemesi ve ortalamada zarar etmesidir.

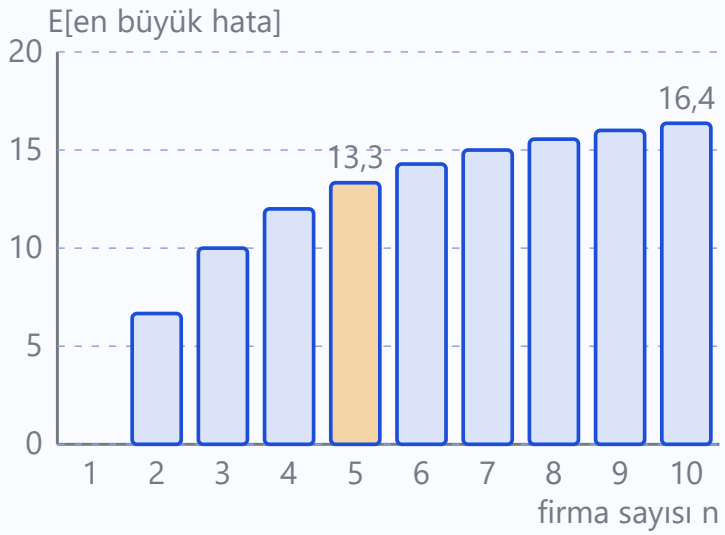
Lanetin özü bir istatistik olgusudur: *tahminler ortalamada doğru olsa bile, en büyük tahmin sistematik olarak yüksektir*. Beş tahminin ortalaması gerçek değere eşit olabilir; ama beşinin en büyüğü neredeyse her zaman gerçek değerin üstündedir. Kazanan tam olarak bu en büyük tahminin sahibidir. Katılımcılar bu riski azaltmak için tekliflerini gözlemlerinin altında tutar ve bilgilerinin kalitesini dikkate alır.

Olgu ilk kez Meksika Körfezi'ndeki petrol sahası açık artırmalarında belgelendi. Atlantic Richfield şirketinin mühendisleri, kazandıkları sahalardan beklenenden sistematik olarak az getiri elde ettiklerini fark etti ve bu olguya “kazananın laneti” adını verdi (Capen, Clapp ve Campbell, 1971).

Lanetin büyüklüğü bir modelle hesaplanabilir. Her firmanın tahmini $V + e_i$ olsun; tahmin hataları e_i bağımsız ve *U[−20, 20]* olsun. Bir hata $-20 + 40\,u$ biçiminde yazılabilir; burada $u \sim U[0, 1]$'dir. n tane *U[0, 1]* sayının en büyüğünün beklentisi $n/(n + 1)$ olduğundan, en büyük hatanın beklentisi

$$-20 + 40 \cdot n/(n + 1) = 20(n - 1)/(n + 1)$$

olur. İki firmada 6,7, beş firmada 13,3, on firmada 16,4'tür. Rakip arttıkça lanet büyür, çünkü kalabalık bir grubun en iyimser üyesi daha iyimserdir.



Şekil 5.16: Tahmin hataları *U[−20, 20]* iken kazananın (en büyük tahminin) beklenen fazla tahmini *20(n − 1)/(n + 1)*, firma sayısına göre. Beş firmada 13,3 (turuncu), on firmada 16,4'tür. Simülasyon da aynı değerleri verir.

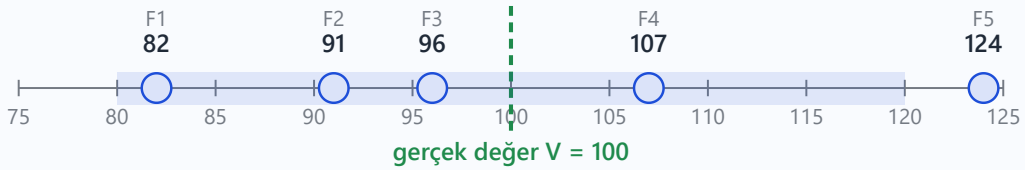
Şekil 5.16'dan sezgiye ters bir sonuç çıkar. Özel değerde rakip arttıkça teklif değere yaklaşır (5.3.7). Ortak değerde ise rakip arttıkça teklif *tahminden uzaklaşmalıdır*; çünkü kazanmak, kalabalığın en iyimseri olmak anlamına gelir. Rakip sayısı arttıkça daha çok kırpmak gerekir.

5.4.9 Petrol sahası açık artırması adım adım

Örnek 5.13

1. **Durum.** Sahanın gerçek değeri *V = 100*'dür ama kimse bilmez. Beş firmanın tahminleri 82, 91, 96, 107, 124'tür. Ortalama *500/5 = 100*'dür: tahminler ortalamada doğrudur.
2. **Birinci fiyat, kırpılmamış teklif.** Herkes tahmini kadar teklif verirse en iyimser firma F5 kazanır ve 124 öder. Kâr *100 − 124 = −24*.
3. **İkinci fiyat, kırpılmamış teklif.** F5 yine kazanır, ama ikinci en yüksek teklifi, 107'yi öder. Kâr *100 − 107 = −7*. İkinci fiyat kuralı bile laneti ortadan kaldırmaz: kazanmak, “en çok abartan ben miyim?” sorusuna evet demektir.
4. **Beklenti.** Beş tahmin tek bir gerçekleşmedir; ortalama davranışı görmek için bir hata modeli gerekir. Hatalar *U[−20, 20]* ise beş tahminin en büyüğünün hatası ortalamada *20 · 4/6 ≈ 13,3*'tür. Birinci fiyatta tahmini kadar teklif vermenin beklenen kârı ≈ −13,3'tür. İkinci fiyatta ödenen ikinci büyük tahminin hatası ortalamada *20 · (n − 3)/(n + 1) = 20 · 2/6 ≈ 6,7*'dir; beklenen kâr ≈ −6,7'dir.
5. **Simülasyon.** 100 bin açık artırmalık bir birinci fiyat simülasyonunda tahmini kadar teklif vermenin ortalama kârı −13,3 çıkar. Teklif tahminden 13,3 kırpılınca ortalama kâr ≈ 0 olur.

6. **Rakip sayısı.** On firmada gereken kırpma *20 · 9/11 ≈ 16,4*'tür: rakip arttıkça daha çok kırpılmalıdır.



Şekil 5.17: Petrol sahası açık artırması. Beş firmanın tahminleri (82, 91, 96, 107, 124) gerçek değer *V = 100*'ün (yeşil kesikli çizgi) iki yanına dağılır; gölgeli bant *[80, 120]* aralığıdır. Kazanan F5, gerçek değerden 24 fazla tahmin etmiştir.

Durum	Kazanan / ödeme	Kâr
Birinci fiyat, kırpılmamış teklif	F5 → 124	−24
İkinci fiyat, kırpılmamış teklif	F5 → 107	−7
En büyük hatanın beklentisi (n = 5)	20 · 4/6	≈ 13,3
Simülasyon, birinci fiyat, kırpılmamış teklif	ortalama	−13,3
Simülasyon, birinci fiyat, 13,3 kırpılmış	ortalama	0

Tablo 5.12: Petrol sahası örneğinin özeti.

Şekil 5.17 ve Tablo 5.12 lanetin mantığını birlikte gösterir. Firmaların hiçbiri sistematik olarak iyimser değildir; tahminleri gerçek değerin iki yanına simetrik dağılır. Sorun, kazananın rastgele seçilmemesidir: kazanan, tanımı gereği en büyük tahmini yapandır. Rasyonel bir firma teklifini verirken şu soruyu sormalıdır: “Bu teklifle kazanırsam, bu benim tahminim hakkında ne söyler?” Kazanma olayına koşullu olarak düşünmek laneti ortadan kaldırır. Örnekteki beş tahmin yalnızca bir gösterimdir: F5'in +24'lük hatası *U[−20, 20]* modelinin dışındadır. Hata aralığı daha geniş olsaydı (ör. *U[−30, 30]*) beklenen fazla tahmin de aynı oranda büyürdü: beş firmada *30 · 4/6 = 20*.

Sık yapılan hata

Kazananın lanetini özel değerli açık artırmalara uygulamak. Özel değerde teklifçi kendi değerini bilir; başkasının ne düşündüğü bu değeri değiştirmez ve kazanmak hiçbir kötü haber taşımaz. Lanet yalnızca değerin ortak (ya da kısmen ortak) olduğu ve tahminlerin gürültülü olduğu durumlarda ortaya çıkar. Bir başka hata da “rakip arttıkça lanet azalır” sanmaktır; tersine, rakip arttıkça en büyük tahmin daha yüksek olur.

5.5 Çift taraflı piyasalar

Şimdiye kadar tek bir satıcı ve birçok alıcı vardı. Borsalarda ise iki taraf da kalabalıktır: birçok alıcı ve birçok satıcı aynı anda emir verir. Böyle piyasaları çift taraflı açık artırma düzenler. Eşleştirme ve fiyat belirleme kuralı, bu kuralın teşvik sorunları ve sorunlara bir çözüm olan McAfee kuralı bu yapının temel parçalarıdır. Açık artırmalar, daha geniş bir alanın, piyasa tasarımının da parçasıdır.

5.5.1 Çift taraflı açık artırma

Tanım 5.7

Çift taraflı açık artırma (*double auction*), birden çok alıcı ve birden çok satıcının aynı anda emir verdiği bir mekanizmadır. Alıcılar **alış teklifi** (*bid*) verir: bir birim için ödemeye razı oldukları en yüksek tutar. Satıcılar **satış isteği** (*ask*) verir: bir birim için kabul edecekleri en düşük tutar. Bütün emirlerin listesine **emir defteri** (*order book*) denir.

Borsalar, elektrik piyasaları ve karbon izni piyasaları bu yapıdadır. Mekanizma iki iş yapar: kimin kiminle işlem yapacağını belirler (eşleştirme) ve işlemin hangi fiyattan yapılacağını belirler.

- **Eşleştirme:** alış teklifleri büyükten küçüğe, satış istekleri küçükten büyüğe sıralanır. En istekli alıcı en ucuz satıcıyla, ikinci en istekli alıcı ikinci en ucuz satıcıyla eşleştirilir ve böyle devam edilir. Alış teklifi satış isteğinden büyük ya da ona eşit olduğu sürece çiftler eşleşir.
- **Fiyat:** eşleşen son çift ile ilk eşleşmeyen çift arasındaki aralıktan tek bir piyasa fiyatı seçilir; örneğin aralığın orta noktası. Eşleşen herkes bu fiyattan işlem yapar.
- **Toplam artık:** bir işlemin yarattığı değer, alıcının değeri ile satıcının maliyeti arasındaki farktır. Eşleşen çiftlerde Σ *(alış − satış)* toplamına **toplam artık** (*surplus*) denir. Bu artık alıcı artığı (alıcının değeri − fiyat) ile satıcı artığının (fiyat − satıcının maliyeti) toplamıdır. Yukarıdaki eşleştirme, mümkün olan en büyük toplam artığı verir.

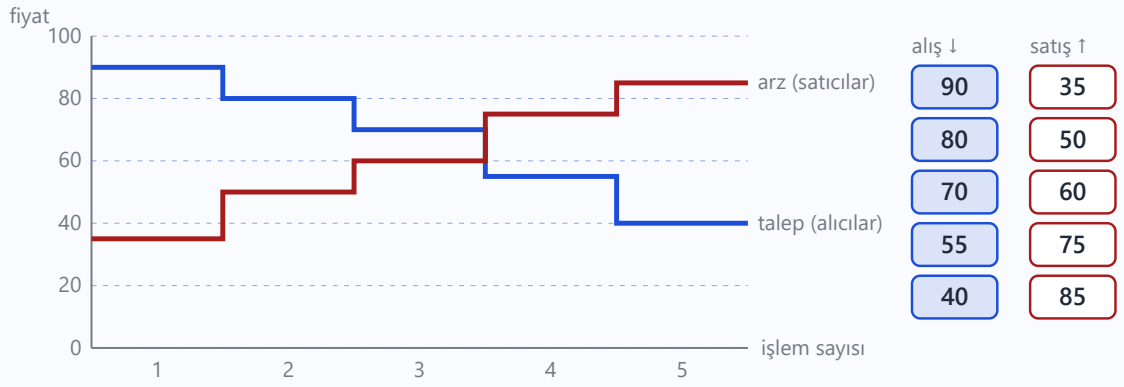
Sıralı eşleştirmenin neden en büyük artığı verdiği şöyle görülür. Bir alış teklifi bir satış isteğinden büyükse o işlem pozitif değer yaratır. En büyük alış teklifleri ile en küçük satış isteklerini eşleştirmek, pozitif değerli işlemleri önce yapmak demektir. Sıralı listelerde alış teklifleri azaldığı ve satış istekleri arttığı için fark da çift çift küçülür; ilk negatif farkta durmak, bütün pozitif değerli işlemleri yapıp hiçbir negatif değerli işlemi yapmamak anlamına gelir.

5.5.2 Eşleştirme adım adım

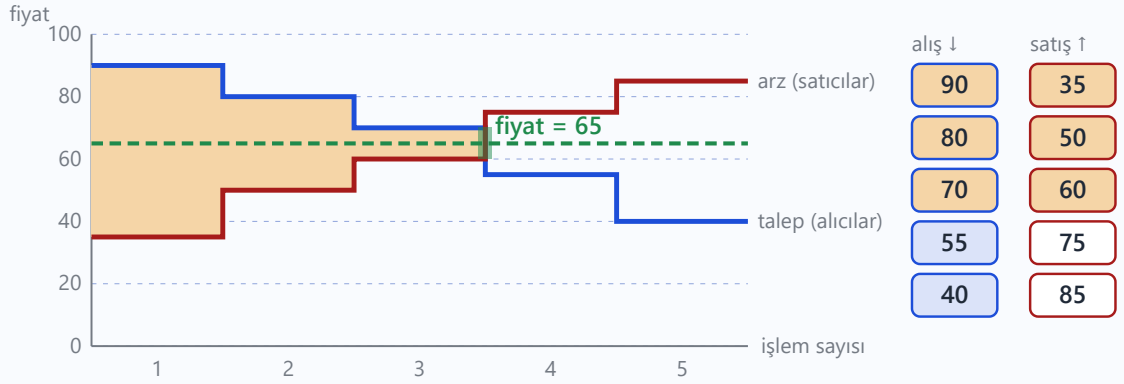
Emir defterinde beş alıcı ve beş satıcı vardır. Alıcılar 70, 40, 90, 55 ve 80 ödemeye; satıcılar 60, 85, 35, 75 ve 50 almaya razıdır.

Örnek 5.14

1. **Sıralama.** Alış teklifleri azalan: 90, 80, 70, 55, 40 (talep eğrisi). Satış istekleri artan: 35, 50, 60, 75, 85 (arz eğrisi).
2. **Çift 1:** 90 ≥ 35 → eşleşir; bu işlemin yarattığı artık *90 − 35 = 55*.
3. **Çift 2:** 80 ≥ 50 → eşleşir; artık *80 − 50 = 30*.
4. **Çift 3:** 70 ≥ 60 → eşleşir; artık *70 − 60 = 10*.
5. **Çift 4:** 55 < 75 → alıcı satıcının istediğini ödemez; dur. 3 işlem gerçekleşir.
6. **Fiyat aralığı.** Alt sınır, eşleşen son satıcının isteği ile ilk eşleşmeyen alıcının teklifinin büyüğüdür: *max(60, 55) = 60*. Üst sınır, eşleşen son alıcının teklifi ile ilk eşleşmeyen satıcının isteğinin küçüğüdür: *min(70, 75) = 70*. Aralık *[60, 70]*, orta nokta fiyat 65'tir.
7. **Artıklar.** Alıcı artığı *(90 − 65) + (80 − 65) + (70 − 65) = 45*. Satıcı artığı *(65 − 35) + (65 − 50) + (65 − 60) = 50*. Toplam *95 = 55 + 30 + 10*, olası en büyük toplam artıktır.



Şekil 5.18: Sıralanmış emir defteri. Mavi basamaklı çizgi azalan alış teklifleri (talep eğrisi), kırmızı basamaklı çizgi artan satış istekleri (arz eğrisi). Sağda sıralı listeler görünür.



Şekil 5.19: Eşleştirmenin sonu. Turuncu alan, talep eğrisinin arz eğrisinin üstünde kaldığı ilk üç işlemdir ve toplam artığı (95) gösterir. Yeşil kesikli çizgi piyasa fiyatı 65'tir; çizginin üstündeki turuncu kısım alıcı artığı, al-

tındaki kısım satıcı artığıdır.

Şekil 5.18 ve Şekil 5.19 eşleştirmeyi iktisatta alışılan arz ve talep eğrileriyle gösterir. Talep eğrisi (alış teklifleri) azalan, arz eğrisi (satış istekleri) artan bir basamaklı çizgidir. İki eğrinin kesiştiği yer işlem sayısını ve fiyat aralığını belirler. Kesişmeden önceki her basamakta talep arzın üstündedir: işlem değer yaratır. Kesişmeden sonra arz talebin üstüne çıkar: işlem değer yok eder.

Fiyat aralığının iki sınırının mantığı önemlidir. Alt sınır 60'tır, çünkü fiyat 60'ın altına inerse eşleşen son satıcı (maliyeti 60) zarar eder ve satmak istemez. Ayrıca fiyat 55'in altına inseydi dışarıda kalan alıcı (55) da almak isterdi ve piyasa dengede olmazdı. Üst sınır 70'tir, çünkü fiyat 70'in üstüne çıkarsa eşleşen son alıcı almak istemez; 75'in üstüne çıkarsa dışarıda kalan satıcı da satmak isterdi. Aralıktaki her fiyat bu dört kişiyi de yerinde tutar.

5.5.3 Kod izleme: cift_tarafli()

Eşleştirme ve orta nokta fiyatı kısa bir fonksiyonla yazılabilir.

```
def cift_tarafli(alis, satis):
  b = sorted(alis, reverse=True)  # alıcı
  s = sorted(satis)               # satıcı
  k, n = 0, min(len(b), len(s))
  while k < n and b[k] >= s[k]:
    k += 1                   # çift eşleşti
  if k == 0:
    return 0, None
  lo = max([s[k-1]] + b[k:k+1])
  hi = min([b[k-1]] + s[k:k+1])
  return k, (lo + hi) / 2
```

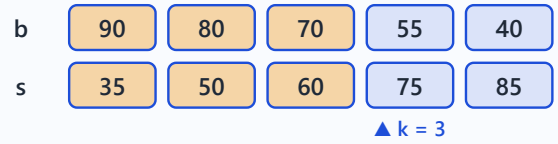
Kod satır satır şöyle çalışır:

- `b = sorted(alis, reverse=True)` : alış teklifleri büyükten küçüğe sıralanır; en istekli alıcı başa gelir.
- `s = sorted(satis)` : satış istekleri küçükten büyüğe sıralanır; en ucuz satıcı başa gelir.
- `k` eşleşen çift sayısıdır; `n` kısa listenin uzunluğudur. Alıcı ve satıcı sayısı farklı olabilir; uzun listenin fazlası zaten eşleşemez.
- `while k < n and b[k] >= s[k]` : sıradaki alıcı sıradaki satıcının isteğini karşılıyorsa çift eşleşir ve `k` bir artar.
- `if k == 0` : hiç işlem yoksa fiyat da yoktur; `(0, None)` döner.
- `lo = max([s[k-1]] + b[k:k+1])` : eşleşen son satıcının isteği ile (varsa) ilk eşleşmeyen alıcının teklifinin büyüğü. `b[k:k+1]` bir dilimdir: `k` listenin içindeyse tek elemanlı bir liste, dışındaysa boş liste verir. Böylece bütün alıcılar eşleşmişse kod hata vermez.

- `hi = min([b[k-1]] + s[k:k+1])` : eşleşen son alıcının teklifi ile (varsa) ilk eşleşmeyen satıcının isteğinin küçüğü.
- Fonksiyon işlem sayısını ve aralığın orta noktasını döndürür.

Adım	İşlem	Sonuç
1	`b = sorted(…, reverse=True)`	[90, 80, 70, 55, 40]
2	`s = sorted(…)`	[35, 50, 60, 75, 85]
3	`k, n = 0, 5`	k = 0
4	b[0] = 90 ≥ s[0] = 35?	evet → k = 1
5	b[1] = 80 ≥ s[1] = 50?	evet → k = 2
6	b[2] = 70 ≥ s[2] = 60?	evet → k = 3
7	b[3] = 55 ≥ s[3] = 75?	hayır → döngü biter
8	`k == 0` ?	hayır; fiyat hesaplanır
9	`lo = max(s[2], b[3])`	max(60, 55) = 60
10	`hi = min(b[2], s[3])`	min(70, 75) = 70
11	dönüş	(3, 65.0)

Tablo 5.13: `cift_tarafli([70, 40, 90, 55, 80], [60, 85, 35, 75, 50])` izlemesi.



Şekil 5.20: `cift_tarafli()` izlemesinin sonu. Üst satır sıralı alış teklifleri `b`, alt satır sıralı satış istekleri `s`; turuncu hücreler eşleşen üç çifttir ve döngü `k = 3`'te durur.

Şekil 5.20 sonucu iki dizi olarak gösterir. Fonksiyon 3 işlem ve 65 fiyatını döndürür; bu, elle yapılan hesapla aynıdır. Algoritmanın maliyetini sıralama belirler: m emir için $O(m \log m)$. Eşleştirme döngüsü listelerin üzerinden bir kez geçtiği için doğrusal zamanlıdır.

5.5.4 Çift taraflı piyasada teşvikler

Tek taraflı açık artırmada Vickrey kuralı hem verimli hem de stratejiye dayanıklıydı. Çift taraflı piyasada aynı şeyi elde etmek mümkün müdür? İstenen dört özellik şunlardır:

- **Verimlilik:** bütün değer yaratan işlemler yapılır.
- **Teşvik uyumluluğu** (IC): herkes için dürüst emir vermek en iyisidir.
- **Bireysel rasyonellik** (IR): kimse katıldığı için zarar etmez.
- **Bütçe dengesi**: piyasa yapıcı dışarıdan para koymaz; alıcıların ödediği satıcılara gider.

Teorem 5.3 (Myerson–Satterthwaite, 1983)

İki taraflı ticarette verimlilik, teşvik uyumluluğu, bireysel rasyonellik ve bütçe dengesi birlikte sağlanamaz.

Bu bir imkânsızlık sonucudur: dört özellikten en az birinden vazgeçmek gerekir. Pratik kurallar hangisinden vazgeçtikleriyle ayrılır.

Orta nokta kuralı (*midpoint rule*) verimli ve bütçe dengelidir, ama teşvik uyumlu değildir. Eşleşen son alıcı ya da satıcı, emrini değiştirerek fiyatı kendi lehine kaydırabilir. Örnekteki eşleşen son satıcının maliyeti 60'tır ve dürüst emirle 65'ten satıp 5 artık elde eder. 60 yerine 64 isterse ne olur? $70 \geq 64$ olduğu için işlem yine gerçekleşir, ama aralık $[\max(64, 55), \min(70, 75)] = [64, 70]$ ve fiyat 67 olur. Satıcı artığı $67 - 60 = 7 > 5$'tir: yalan kârlıdır. Aynı şeyi eşleşen son alıcı da teklifini düşürerek yapabilir. Abartmanın bir sınırı vardır: satıcı 70'in üstünde isterse işlem kaybolur. Bu, birinci fiyattaki kırpma ödünleşiminin aynısıdır.

McAfee kuralı (*McAfee double auction*) teşvik uyumluluğunu sağlamak için bazen bir işlemden vazgeçer (McAfee, 1992). Kural şöyledir. Eşleştirme yukarıdaki gibi yapılır ve k verimli işlem bulunur. Dışarıda kalan ilk alıcı ile ilk satıcının emirlerinin ortalaması p_0 hesaplanır.

- p_0, eşleşen son satıcının isteği ile eşleşen son alıcının teklifi arasına düşerse, k işlemin hepsi p_0'dan yapılır.
- Düşmezse son (en az artık getiren) çift çıkarılır. Kalan $k - 1$ çiftte alıcılar çıkarılan alıcının teklifini öder, satıcılar çıkarılan satıcının isteğini alır.

Kuralın teşvik uyumlu olmasının nedeni, işlem yapan hiç kimsenin kendi fiyatını belirleyememesidir. Birinci durumda fiyatı dışarıda kalan çift belirler. İkinci durumda fiyatları çıkarılan çift belirler ve o çift işlem yapmaz. Bu, Vickrey kuralındaki "ödemeni rakip belirler" fikrinin çift taraflı sürümüdür. Dürüstlük baskındır; bedeli, ikinci durumda bir işlemin kaybolmasıdır. İkinci durumda alıcıların ödediği satıcıların aldığından fazladır; aradaki fark piyasa yapıcıda kalır. Böylece kural bütçe dengesini zayıf anlamda sağlar: piyasa yapıcı dışarıdan para koymaz, ama bazen elinde para kalır.

Örneğimizde dışarıda kalan ilk çift 55 (alıcı) ve 75'tir (satıcı). $p_0 = (55 + 75)/2 = 65$'tir. Bu değer $[60, 70]$ aralığındadır; üç işlem de 65'ten gerçekleşir ve hiçbir işlem kaybolmaz. Bu örnekte McAfee kuralı orta nokta kuralıyla aynı sonucu verir.

5.5.5 Kod izleme: mcafee()

McAfee kuralı şöyle yazılabilir:

```
def mcafee(alis, satis):
  b = sorted(alis, reverse=True) + [0]
  s = sorted(satis) + [float('inf')]
  k = 0
  while b[k] >= s[k]:
    k += 1                # verimli işlem
  if k == 0: return 0, None, None
  p0 = (b[k] + s[k]) / 2   # dışarıdaki çift
  if s[k-1] <= p0 <= b[k-1]:
    return k, p0, p0      # tek fiyat
  return k - 1, b[k-1], s[k-1]  # iki fiyat
```

Kodun iki ayrıntısı vardır:

- **Bekçiler.** Alış listesinin sonuna 0, satış listesinin sonuna sonsuz (`float('inf')`) eklenir. 0 teklif eden bir alıcı hiçbir pozitif isteği karşılayamaz; sonsuz isteyen bir satıcıyı hiçbir alıcı karşılayamaz. Böylece döngü listenin sonunda kendiliğinden durur ve `b[k]`, `s[k]` her zaman tanımlıdır. Bu, `cift_tarafli()`'deki dilim hilesinin yerini tutar.
- **İki dönüş biçimi.** Fonksiyon `(işlem sayısı, alıcı fiyatı, satıcı fiyatı)` döndürür. Tek fiyat durumunda iki fiyat aynıdır; ikinci durumda farklıdır.

Yeni bir örnek ele alınsın: alıcılar 90, 80, 70, 40; satıcılar 30, 50, 65, 85.

Adım	İşlem	Sonuç
1	sıralama ve bekçiler	b = [90, 80, 70, 40, 0], s = [30, 50, 65, 85, ∞]
2	b[0] = 90 ≥ s[0] = 30?	evet: verimli çift, k = 1
3	b[1] = 80 ≥ s[1] = 50?	evet: k = 2
4	b[2] = 70 ≥ s[2] = 65?	evet: k = 3
5	b[3] = 40 ≥ s[3] = 85?	hayır: döngü biter
6	`p0 = (b[3] + s[3]) / 2`	(40 + 85)/2 = 62,5
7	s[2] = 65 ≤ 62,5 ≤ b[2] = 70?	hayır: 62,5 < 65
8	son çift (70, 65) çıkarılır	(2, 70, 65)

Tablo 5.14: `mcafee([90, 80, 70, 40], [30, 50, 65, 85])` izlemesi.



Şekil 5.21: `mcafee()` izlemesinin sonu. Üçüncü çift (70, 65) soluk gösterilmiştir: çıkarılmıştır. Kalan iki işlemde alıcılar 70 öder, satıcılar 65 alır.

Adım 7'deki kontrolün anlamı şudur: $p_0 = 62{,}5$ fiyatından işlem yapılsaydı üçüncü satıcı (maliyeti 65) zarar ederdi. Bu yüzden en zayıf çift (70, 65) çıkarılır. Kalan iki işlemde alıcılar 70 öder, satıcılar 65 alır; her işlemde artan $70 - 65 = 5$, toplamda $2 \cdot 5 = 10$ piyasa yapıcıda kalır (Şekil 5.21).

Aynı emirler orta nokta kuralına verilseydi `cift_tarafli()` üç işlem ve $(65 + 70)/2 = 67{,}5$ fiyatını döndürürdü. Toplam artık $60 + 30 + 5 = 95$ olurdu. McAfee kuralında toplam artık $60 + 30 = 90$'dır; 5 birimlik artık, kaybolan üçüncü işlemle birlikte gider. Dürüstlüğün bedeli budur ve en fazla bir işlemdir. Çok sayıda katılımcılı piyasalarda bir işlemin kaybı toplam artığın küçük bir kısmı olduğundan kural pratikte çekicidir.

5.5.6 Açık artırma ve piyasa tasarımı

Piyasa tasarımı (*market design*), katılımcıların kimlerle ve hangi koşullarla eşleşeceğini belirleyen kuralları kurma işidir. Temel hedefler verimlilik, teşvik uyumluluğu, adalet, kararlılık ve uygulanabilirliktir. Kurallar stratejiye dayanıklı olmalıdır: katılımcıların kuralları kendi lehine çevirmek için yalan söylemesi kârlı olmamalıdır.

Açık artırmalar, kıt kaynakları fiyat ve tekliflere göre tahsis eden piyasa tasarımı araçlarıdır. Elektrik piyasaları, emisyon izinleri, kamu ihaleleri ve frekans lisansları fiyatlı uygulamalardır. Para kullanılmayan piyasalarda ise (örneğin öğrencilerin okullara yerleştirilmesi ya da organ nakli için böbrek takası) fiyatın yerini öncelik ve eşleştirme kuralları alır. Her iki durumda da tasarımcı aynı ödünleşimlerle karşılaşır: gelir, verimlilik, basitlik ve stratejiye dayanıklılık her zaman birlikte sağlanamaz. Bölümdeki örnekler bunu açıkça gösterir: rezerv gelir için verimlilikten, McAfee kuralı dürüstlük için bir işlemden vazgeçer.

5.5.7 Kuramdan uygulamaya: kısa tarihçe

Eski bir kurum. Açık artırmalar çok eskidir. Herodot, MÖ 5. yüzyılda Babil'de yapılan açık artırmaları anlatır. MS 193'te Roma'da Pretoryen Muhafızları imparatorluk tahtını açık artırmaya çıkarıp Didius Julianus'a sattı; Julianus iki ay sonra öldürüldü. Bu, kazananın lanetinin tarihteki en çarpıcı örneklerinden biri sayılır.

Kuram. Açık artırma kuramı 1961'de Vickrey'nin makalesiyle başladı. Vickrey dört türü karşılaştırdı ve gelir eşdeğerliğinin ilk biçimini buldu (Vickrey, 1961). Myerson geliri maksimize eden açık artırmayı tanımladı (Myerson, 1981). İkisi de Nobel Ekonomi Ödülü aldı: Vickrey 1996'da, Myerson 2007'de.

Spektrum açık artırmaları. ABD Federal İletişim Komisyonu (FCC) 1994'te ilk spektrum açık artırmasını Paul Milgrom, Robert Wilson ve Preston McAfee'nin önerdiği *eşzamanlı çok turlu artırma* ile yaptı. Bu tasarımda birçok lisans aynı anda ve turlar hâlinde satılır; teklifçiler her turda öteki lisansların fiyatlarını görerek tekliflerini günceller. Milgrom ve Wilson 2020'de Nobel Ekonomi Ödülü aldı.

Eşleştirme piyasaları. Alvin Roth ve Lloyd Shapley 2012'de, fiyatın kullanılmadığı piyasalarda kararlı eşleştirme ve takas kuralları üzerine çalışmalarıyla Nobel Ekonomi Ödülü aldı.

Bir tasarımcının elindeki araçlar tahsis ve ödeme kuralıyla sınırlı değildir. Bilgi açıklama da bir araçtır: hangi teklifler, ne zaman gösterilir? Katılım koşulları (kimin teklif verebileceği, depozito gerekip gerekmediği) da öyledir. Gerçek tasarımlar uygulamaya konmadan önce laboratuvar deneyleri ve saha verisiyle sınanır.

5.6 Değerlendirme

Bölümün sonuçları uygulamada iki soruyla tartılır: kuramın en zarif sonucu olan Vickrey açık artırması neden az kullanılır ve hangi açık artırma ne kadar adildir? Dört türün özeti ve örnek problemler bu tartışmayı tamamlar.

5.6.1 Vickrey açık artırmaları neden az kullanılıyor?

Vickrey açık artırmasının kuramsal özellikleri çok çekicidir: dürüst teklif baskındır, nesne en yüksek değerliye gider, gelir standart kurallarla aynıdır. Yine de saf ikinci fiyat kapalı zarf açık artırması pratikte seyrek görülür. Rothkopf, Teisberg ve Kahn (1990) ile Ausubel ve Milgrom (2006) bunun nedenlerini tartışır.

- **Güven.** Katılımcılar ikinci en yüksek teklifin doğru açıklandığına güvenmek zorundadır. Kapalı zarfları yalnızca açık artırmacı görür; satıcı sahte bir "ikinci teklif" ekleyerek fiyatı kazananın teklifinin hemen altına çekebilir.
- **Sezgisellik.** "İkinci fiyatı öde" kuralı çoğu insana doğal gelmez. Laboratuvarda fazla teklif sık görülür (5.3.3).
- **Model varsayımları.** Ortak ya da birbirine bağlı değerlerde dürüst teklifin baskın olduğu sonucu geçerliliğini yitirebilir.
- **Çok nesne.** Bütçe kısıtları ve tamamlayıcı ürünler ödeme kuralını karmaşıklaştırır. Çok nesneli genelleme olan VCG'de gelir çok düşük olabilir ve teklifçiler sahte kimliklerle kural dışı kazanç sağlayabilir.
- **Satıcı kaygısı.** Kazananın teklifi ile ödediği fiyat arasındaki fark büyük olabilir. Kazananın 1 milyon teklif edip 100 bin ödemesi kamuoyunda "düşük fiyata satış" olarak algılanabilir. Ayrıca teklifçiler arası danışıklı teklif (gizli anlaşma) riski vardır.

Bu nedenlerle pratik tasarımlar çoğu zaman dürüstlük teşvikiyle uygulama kolaylığı arasında denge kurar. İngiliz açık artırması özel değerlerde aynı sonucu şeffaf biçimde verir: fiyat herkesin gözü önünde oluşur ve kimse açıklanmayan bir teklife güvenmek zorunda kalmaz. Bu yüzden İngiliz açık artırması uygulamada Vickrey açık artırmasının "görünür" sürümü gibi çalışır.

5.6.2 Tartışma: hangi açık artırma, ne kadar adil?

Bir katılımcı açısından türler farklı deneyimler sunar. İngiliz açık artırması şeffaftır; fiyatı adım adım görmek bir avantajdır. Vickrey açık artırmasında dürüst teklif en iyisidir ve strateji kurmaya gerek yoktur. Hollanda ve birinci fiyat açık artırmaları hızlıdır, ama kırpma miktarını tahmin etmek gerekir.

Hangi türün tercih edileceği kişisel stratejiye ve risk iştahına bağlıdır: riskten kaçınan biri, 5.4.3'te görüldüğü gibi birinci fiyatta daha az kırpar ve kazanma olasılığını yükseltir.

Adalet sorusu daha karmaşıktır. Bir yandan kuralların önceden belirlenmesi öngörülebilirlik sağlar ve manipülasyonu azaltabilir; herkes aynı kurallarla yarışır. Öte yandan bütçesi büyük oyuncular avantajlıdır: bir açık artırma ödeme gücünü ölçer, ihtiyacı ölçmez. Katılım erişimi ve bilgi eşitsizliği de önemlidir: daha iyi bilgisi olan bir teklifçi ortak değerli bir açık artırmada lanetten kaçınırken daha az bilgili rakipleri zarar eder. Satıcının geliri ile fırsat eşitliği farklı hedeflerdir. Rezerv fiyatı örneğinde görüldüğü gibi bir hedef için iyi olan diğeri için kötü olabilir. Hiçbir kural "herkes için tam adalet"i garanti etmez.

5.6.3 Dört tür yan yana ve örnek problemler

Tür	Öder	Denge teklifi	Eşdeğeri
İngiliz	son rakibin çekildiği fiyat	değerine kadar kal	ikinci fiyat
İkinci fiyat	ikinci en yüksek teklif	$b = v$ (zayıf baskın)	İngiliz
Hollanda	kabul edilen fiyat	$b = (n-1)v/n$	birinci fiyat
Birinci fiyat	kendi teklifi	$b = (n-1)v/n$	Hollanda

Tablo 5.15: Dört tür yan yana (IPV, $U[0, 1]$, n teklifçi).

Tablo 5.15'e üç ek sonuç eşlik eder. Dört türde de beklenen gelir $(n - 1)/(n + 1)$'dir; iki kişide 1/3. Ortak değerde kazanan fazla ödeyebilir; rakip sayısı arttıkça daha çok kırpmak gerekir. Çift taraflı piyasada alış teklifleri azalan, satış istekleri artan sıralanıp eşlenir ve fiyat $[lo, hi]$ aralığından seçilir.

Aşağıdaki örnek problemler bölümün temel hesaplarını tekrar eder.

Örnek 5.15

1. **Dört kural.** Sınırlar 40, 65, 80, 90 ve herkes sınırını teklif ediyor. 90'lık teklifçi kazanır. İngiliz ≈ 80, ikinci fiyat 80; Hollanda 90, birinci fiyat 90 (5.2.5).
2. **Denge teklifi.** $n = 4$, değer 0,6: $b = (4 - 1)/4 \cdot 0{,}6 = 0{,}45$.
3. **Beklenen gelir.** Üç teklifçi, ikinci fiyat: $(n - 1)/(n + 1) = 2/4 = 1/2$.
4. **Vickrey'de fazla teklif.** Değer 50, rakiplerin en yüksek teklifi 55, teklif 60: kazanılır ama 55 ödenir; fayda $50 - 55 = -5$. Dürüst teklifle kaybedilir ve fayda 0 olur.
5. **Çift taraflı.** Alıcılar 100, 70, 45; satıcılar 30, 60, 80. Çiftler 100 ≥ 30 ve 70 ≥ 60 eşleşir; 45 < 80'de durulur: 2 işlem. Aralık $[\max(60, 45), \min(70, 80)] = [60, 70]$, fiyat 65.
6. **Rezerv.** İki teklifçi, $U[0, 1]$, ikinci fiyat, $r = 0{,}4$: $1/3 + 0{,}16 - 4 \cdot 0{,}064/3 = 1/3 + 0{,}16 - 0{,}0853 \approx 0{,}408$. Bu, rezervsiz 1/3 ≈ 0,333'ten fazladır. En iyisi $r = 0{,}5$: 5/12 ≈ 0,417.

5.7 Özet

- **Açık artırma** bir tahsis kuralı (kim alır?) ile bir ödeme kuralının (ne öder?) birleşimidir. Teklifçinin **değeri** özel bilgidir, **teklifi** stratejik bir karardır; faydası kazanınca $v - p$, kaybedince 0'dır. Temel model **bağımsız özel değerler**dir (IPV); ortak değer modeli kazananın lanetine yol açar.
- **Dört temel tür:** İngiliz (fiyat yükselir, son kalan son rakibin çekildiği fiyatı öder), Hollanda (fiyat düşer, ilk kabul eden o fiyatı öder), birinci fiyat kapalı zarf (kendi teklifini öder), ikinci fiyat/Vickrey (ikinci en yüksek teklifi öder). Özel değerlerde İngiliz ≈ ikinci fiyat; her durumda Hollanda ≡ birinci fiyat.
- **Vickrey teoremi:** ikinci fiyat açık artırmasında dürüst teklif zayıf baskın stratejidir. Teklif yalnızca kazanıp kazanmamayı belirler, ödemeyi rakipler belirler; her B için dürüst teklif $\max(v - B, 0)$'ı verir.
- **Teklif kırpma:** birinci fiyatta $E[u(b)] = (v - b) \cdot P(\text{kazan})$ ödünleşimi simetrik Bayes–Nash dengesinde $b = (n-1)v/n$ teklifini verir; iki kişide $v/2$. Kırpma v/n'dir ve rekabet arttıkça azalır.
- **Gelir eşdeğerliği:** IPV, risk nötr simetrik teklifçiler ve aynı tahsis kuralı altında standart açık artırmaların beklenen geliri aynıdır: $(n-1)/(n+1)$, iki kişide 1/3. Nedeni Myerson ödeme formülüdür: $p(v) = v\,x(v) - \int_0^v x(z)\,dz$. Tümü öder açık artırması da ($b = v^2/2$) aynı geliri getirir.
- **Riskten kaçınma** eşitliği bozar: birinci fiyatta teklif $v/(1+\rho)$'ya yükselir, gelir $\rho = ½$'de 4/9'a çıkar; ikinci fiyat 1/3'te kalır.
- **Myerson rezerv fiyatı:** beklenen gelir kazananın sanal değerinin beklentisidir; $U[0,1]$'de $\psi(v) = 2v - 1$, en iyi rezerv $r = 1/2$ (n'den bağımsız). İki teklifçide gelir $1/3 + r^2 - 4r^3/3$, en iyi rezervde 5/12. Bedeli: %25 olasılıkla satış yok, sosyal refah 2/3'ten 7/12'ye iner.
- **Kazananın laneti:** ortak değerde kazanan en iyimser tahmini yapandır; hatalar $U[-a, a]$ ise beklenen fazla tahmin $a(n-1)/(n+1)$'dir. Rakip arttıkça daha çok kırpmak gerekir.
- **Çift taraflı açık artırma:** alış teklifleri azalan, satış istekleri artan sıralanıp eşlenir; fiyat $[lo, hi]$ aralığından seçilir ve toplam artık en büyüktür. Myerson–Satterthwaite'e göre verimlilik, IC, IR ve bütçe dengesi birlikte sağlanamaz; orta nokta kuralı IC değildir, McAfee kuralı bir işlem pahasına dürüstlüğü baskın yapar.

Kavram	Ne söyler?	Anahtar sayı / formül
Vickrey açık artırması	Dürüst teklif zayıf baskın	$b = v$
Teklif kırpma	Birinci fiyatta değerin altında teklif	$b = (n-1)v/n$
Gelir eşdeğerliği	Aynı tahsis → aynı beklenen gelir	$(n-1)/(n+1)$
Riskten kaçınma	Birinci fiyat daha çok kazandırır	$b = v/(1+\rho)$
Sanal değer ve rezerv	Negatif sanal değere satma	$\psi(v) = 2v - 1$, $r = 1/2$, 5/12
Kazananın laneti	Ortak değerde kazanan fazla öder	$20(n-1)/(n+1)$
Çift taraflı açık artırma	Sıralı eşleştirme, tek fiyat	$[lo, hi]$ orta noktası

Kavram	Ne söyler?	Anahtar sayı / formül
McAfee kuralı	Dürüstlük için bazen bir işlem kaybı	p_0 = dışarıdaki çiftin ortalaması

Tablo 5.16: Bölümün kavramları bir bakışta.

5.8 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

açık artırma *auction* · teklifçi *bidder* · değer *value* · teklif *bid* · değerleme *valuation* · yarı doğrusal fayda *quasi-linear utility* · tahsis kuralı *allocation rule* · ödeme kuralı *payment rule* · fiyat keşfi *price discovery* · bağımsız özel değerler *independent private values (IPV)* · ortak değer *common value* · birbirine bağlı değerler *interdependent values* · İngiliz açık artırması *English (ascending) auction* · Hollanda açık artırması *Dutch (descending) auction* · kapalı zarf *sealed bid* · birinci fiyat kapalı zarf açık artırması *first-price sealed-bid auction* · ikinci fiyat (Vickrey) açık artırması *second-price (Vickrey) auction* · zayıf baskın strateji *weakly dominant strategy* · dürüst teklif *truthful bid* · stratejiye dayanıklı *strategy-proof* · teklif kırpma *bid shading* · Bayes–Nash dengesi *Bayes–Nash equilibrium* · gelir *revenue* · gelir eşdeğerliği teoremi *revenue equivalence theorem* · gelir maksimizasyonu *revenue maximization* · en iyi (optimal) açık artırma *optimal auction (Myerson)* · sanal değer *virtual value* · rezerv fiyat *reserve price* · tümü öder açık artırması *all-pay auction* · çok birimli açık artırma *multi-unit auction* · tek tip fiyatlı açık artırma *uniform-price auction* · ayrımcı fiyatlı açık artırma *discriminatory (pay-as-bid) auction* · spektrum açık artırması *spectrum auction* · kazananın laneti *winner's curse* · çift taraflı açık artırma *double auction* · alış teklifi *bid (buy order)* · satış isteği *ask (sell order)* · emir defteri *order book* · artık *surplus* · alıcı / satıcı artığı *buyer / seller surplus* · orta nokta kuralı *midpoint rule* · McAfee kuralı *McAfee double auction* · piyasa yapıcı *market maker* · bütçe dengesi *budget balance* · piyasa tasarımı *market design* · VCG mekanizması *Vickrey–Clarke–Groves (VCG) mechanism*

5.9 Kendini Dene

1. Beş teklifçinin sınırları 35, 62, 78, 81 ve 50'dir; herkes sınırını teklif ediyor (Hollanda'da fiyat sü-

rekli düşüyor). Dört kuralda kazanan kimdir, ne öder ve faydası kaçtır?

Cevap

Kazanan her kuralda sınırı 81 olan teklifçidir. İngiliz: son rakip 78'de çekilir, ödeme ≈ 78, fayda ≈ 3. İkinci fiyat: ödeme ikinci en yüksek teklif 78, fayda 3. Hollanda: fiyat 81'e inince kabul eder, ödeme 81, fayda 0. Birinci fiyat: ödeme 81, fayda 0. İngiliz ile ikinci fiyat, Hollanda ile birinci fiyat aynı sonucu verir.

2. Vickrey açık artırmasında bir teklifçinin değeri $v = 40$'tır. Rakiplerin en yüksek teklifi B 20, 35, 50 ya da 60 olabilir. Dürüst teklif, $b = 30$ ve $b = 55$ için her durumda fayda kaçtır?

Cevap

Dürüst (40): B = 20 → kazanır, 20 öder, fayda 20; 35 → fayda 5; 50 ve 60 → kaybeder, 0. $b = 30$: 20 → 20 (aynı); 35 → kaybeder, 0 (dürüstten 5 kötü); 50, 60 → 0 (aynı). $b = 55$: 20 → 20; 35 → 5 (aynı); 50 → kazanır, 50 öder, fayda −10 (dürüstten kötü); 60 → 0 (aynı). Hiçbir sapma hiçbir B'de dürüstten iyi değildir; kötüleştiği durumlar B'nin b ile v arasına düştüğü durumlardır.

3. Birinci fiyat açık artırmasında üç teklifçi var; iki rakip değerlerinin 2/3'ünü teklif ediyor ve değerler $U[0, 1]$. Üçüncü teklifçinin değeri 0,9'dur. $b = 0{,}3;\ 0{,}45;\ 0{,}6;\ 0{,}75$ tekliflerinin beklenen faydası nedir? En iyisi hangisidir?

Cevap

Bir rakibi geçme olasılığı $\min(1, b/(2/3)) = \min(1, 1{,}5b)$; iki rakip için karesi alınır. $b = 0{,}3$: $P = 0{,}45^2 = 0{,}2025$, $E[u] = 0{,}6 \cdot 0{,}2025 \approx 0{,}122$. $b = 0{,}45$: $P = 0{,}675^2 \approx 0{,}456$, $E[u] \approx 0{,}45 \cdot 0{,}456 \approx 0{,}205$. $b = 0{,}6$: $P = 0{,}9^2 = 0{,}81$, $E[u] = 0{,}3 \cdot 0{,}81 = 0{,}243$. $b = 0{,}75$: $P = 1$, $E[u] = 0{,}15$. En iyisi $b = 0{,}6 = (3 - 1)/3 \cdot 0{,}9$'dur, denge teklifiyle aynı.

4. Üç teklifçi, değerler $U[0, 1]$. Myerson ödeme formülüne göre değeri v olan teklifçinin beklenen ödemesi nedir? Tümü öder açık artırmasında değeri 0,9 olan teklifçinin denge teklifi nedir?

Cevap

Kazanmak için iki rakibin de değeri v'den küçük olmalı: $x(v) = v^2$. $p(v) = v \cdot v^2 - \int_0^v z^2\,dz = v^3 - v^3/3 = 2v^3/3$. Denetim: birinci fiyatta teklif $2v/3$, kazanma olasılığı v^2, çarpım $2v^3/3$. Tümü öderde teklif her durumda ödendiğinden teklif beklenen ödemeye eşittir: $b(0{,}9) = 2 \cdot 0{,}729/3 = 0{,}486$.

5. İki teklifçi, değerler $U[0, 1]$, fayda $u(x) = x^\rho$ ve $\rho = 1/3$. Birinci fiyatta denge teklifi ve satıcının bekle-

nen geliri nedir? İkinci fiyata göre fark ne kadardır?

Cevap

$b(v) = v/(1 + \rho) = v/(4/3) = 3v/4$. Kazanan büyük değerli teklifçidir; gelir $3/4 \cdot E[max] = 3/4 \cdot 2/3 = 1/2$. İkinci fiyatta dürüst teklif baskın kalır ve gelir 1/3'tür. Birinci fiyat $1/2 - 1/3 = 1/6$ daha çok kazandırır; riskten kaçınma arttıkça (ρ küçüldükçe) fark büyür.

6. İki teklifçi, $U[0, 1]$, rezervi $r = 0{,}3$ olan ikinci fiyat açık artırması. Beklenen gelir, hiç satılmama olasılığı ve beklenen sosyal refah kaybı nedir? Beş teklifçi olsaydı en iyi rezerv değişir miydi?

Cevap

Gelir $1/3 + 0{,}09 - 4 \cdot 0{,}027/3 = 1/3 + 0{,}09 - 0{,}036 \approx 0{,}387$. Satılmama olasılığı $0{,}3^2 = 0{,}09$. Refah kaybı $2 \cdot 0{,}027/3 = 0{,}018$. En iyi rezerv $\psi(r) = 2r - 1 = 0$'dan gelir ve teklifçi sayısını içermez; beş teklifçide de $r = 1/2$'dir.

7. Ortak değerli bir açık artırmada tahmin hataları $U[-15, 15]$'tir. Üç firma tahmini kadar teklif verirse birinci fiyatta ve ikinci fiyatta kazananın beklenen kârı nedir? Altı firmada birinci fiyat için cevap ne olur?

Cevap

En büyük hatanın beklentisi $15(n - 1)/(n + 1)$; $n = 3$ için $15 \cdot 2/4 = 7{,}5$. Birinci fiyatta kazanan kendi tahminini öder: beklenen kâr $\approx -7{,}5$. İkinci fiyatta ikinci büyük tahmin ödenir; ikinci büyük hatanın beklentisi $15(n - 3)/(n + 1) = 0$: beklenen kâr 0. Altı firmada birinci fiyat beklenen kârı $-15 \cdot 5/7 \approx -10{,}7$'dir; rakip arttıkça lanet büyür.

8. Alıcılar 75, 62, 58, 30; satıcılar 20, 41, 55, 70. Orta nokta kuralı ve McAfee kuralı kaç işlem ve hangi fiyatları verir? McAfee kuralında kaybolan artık ne kadardır?

Cevap

Sıralı listeler zaten azalan/artan: $75 \geq 20$, $62 \geq 41$, $58 \geq 55$ eşleşir; $30 < 70$'te durulur, $k = 3$. Orta nokta: aralık $[\max(55, 30), \min(58, 70)] = [55, 58]$, fiyat 56,5; üç işlem. McAfee: $p_0 = (30 + 70)/2 = 50$; 50 $[55, 58]$'de değil. Son çift (58, 55) çıkarılır: 2 işlem, alıcılar 58 öder, satıcılar 55 alır; piyasa yapıcıda $2 \cdot 3 = 6$ kalır. Kaybolan artık üçüncü işlemin artığı $58 - 55 = 3$'tür.

5.10 İleri okuma

- V. Krishna, *Auction Theory* (2010): bu bölümdeki bütün sonuçların (denge teklifleri, gelir eşdeğerliği, Myerson açık artırması, ortak değer) standart ve ayrıntılı ispatlarını veren temel ders kitabıdır.

- T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory* (2016), Ders 2–5: Vickrey açık artırması, Myerson lemması ve gelir maksimizasyonunu bilgisayar bilimi bakışıyla, kısa ve açık biçimde anlatır.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 9: açık artırma türlerini ve kazananın lanetini az matematikle, sezgisel örneklerle sunar.
- P. Milgrom, *Putting Auction Theory to Work* (2004): spektrum açık artırmalarının tasarımını kuramın uygulamaya nasıl taşındığı üzerinden anlatır.

5.11 Kaynaklar

1. L. M. Ausubel, P. Milgrom, "The lovely but lonely Vickrey auction", P. Cramton, Y. Shoham, R. Steinberg (ed.), *Combinatorial Auctions*, MIT Press, 2006.
2. J. Bulow, P. Klemperer, "Auctions versus negotiations", *American Economic Review* 86(1), 180–194, 1996.
3. E. C. Capen, R. V. Clapp, W. M. Campbell, "Competitive bidding in high-risk situations", *Journal of Petroleum Technology* 23(6), 641–653, 1971.
4. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010 (Bölüm 9: Auctions).
5. J. H. Kagel, R. M. Harstad, D. Levin, "Information impact and allocation rules in auctions with affiliated private values: a laboratory study", *Econometrica* 55(6), 1275–1304, 1987.
6. A. R. Karlin, Y. Peres, *Game Theory, Alive*, American Mathematical Society, 2017 (açık artırma bölümleri).
7. V. Krishna, *Auction Theory*, 2. baskı, Academic Press, 2010.
8. R. P. McAfee, "A dominant strategy double auction", *Journal of Economic Theory* 56(2), 434–450, 1992.
9. P. Milgrom, *Putting Auction Theory to Work*, Cambridge University Press, 2004.
10. R. B. Myerson, "Optimal auction design", *Mathematics of Operations Research* 6(1), 58–73, 1981.
11. R. B. Myerson, M. A. Satterthwaite, "Efficient mechanisms for bilateral trading", *Journal of Economic Theory* 29(2), 265–281, 1983.
12. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 9 ve 13).
13. J. G. Riley, W. F. Samuelson, "Optimal auctions", *American Economic Review* 71(3), 381–392, 1981.
14. M. H. Rothkopf, T. J. Teisberg, E. P. Kahn, "Why are Vickrey auctions rare?", *Journal of Political Economy* 98(1), 94–109, 1990.
15. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press, 2016.
16. W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders", *Journal of Finance* 16(1), 8–37, 1961.

KISIM II · BÖLÜM 6

Çevrim İçi Platformlar

Bir pazar yeri, bir arama motoru ya da bir yolculuk uygulaması kendi başına pek bir şey üretmez. Değeri, bir araya getirdiği insanların birbirine erişmesinden doğar. Bu yüzden platformun kuralları (kim katılır, kim kimi görür, kim ne öder) birer mekanizmadır ve kullanıcılar bu kurallar altında stratejik oyunculardır. Bu not, iki taraflı pazarlardan başlayarak ağ etkilerini ve kritik kitleyi, hangi tarafın neden sübvanse edildiğini, arama reklamlarındaki GSP ve VCG açık artırmalarını, bütçeli çevrim içi tahsisi, itibar sistemlerini ve platform rekabetini adım adım anlatır. Her konuda sayısal bir örnek ara adımlarıyla çözülür.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum06`

ÖĞRENME ÇIKTILARI

Platformu modellemek
İki taraflı pazarı, çapraz ağ etkilerini ve platformu bir mekanizma tasarımcısı olarak tanımlamak.

Kritik kitleyi hesaplamak
Basit bir benimseme modelinde dengeleri, eşiği ve "devrilme" (*tipping*) dinamiğini bulmak.

Fiyat yapısını açıklamak
İki taraflı sayısal örnekte hangi tarafın neden sübvanse edildiğini göstermek.

Reklam açık artırmaları
GSP ve VCG ödemelerini aynı tekliflerle hesaplamak; GSP'de dürüst teklifin neden denge olmadığını göstermek.

İtibarı ölçmek
Bayesçi ortalamayla sıralamak; sahte değerlendirmenin ne zaman kârlı olduğunu hesaplamak.

Rekabeti tartışmak
Devrilme, geçiş maliyeti, çoklu kullanım ve düzenlemenin (AB Dijital Piyasalar Yasası, DMA) platform rekabetine etkisini yorumlamak.

6.1 Platformlar ve pazarlar

Bu bölümün bütün konuları tek bir gözleme dayanır: bir platformun değeri, onu kullanan insanların sayısına ve kimliğine bağlıdır. Bu gözlem, alışılmış piyasa sezgilerini değiştirir. Bir tarafa para ödemek kârlı olabilir. Herkesin beklentisi pazarın kaderini belirleyebilir. "İkinci fiyat" adlı bir açık artırmada dürüst teklif en iyisi olmayabilir.

6.1.1 Bir oyunla başlangıç: sponsorlu arama açık artırması

Bir arama motoruna "ucuz uçak bileti" yazıldığında sonuç sayfasının üstünde birkaç reklam görünür. Bu reklamların yeri rastgele belirlenmez. Her aramada, o anahtar kelimeye teklif vermiş reklamverenler arasında küçük bir açık artırma yapılır. Reklamverenler **tık başına** teklif verir: "reklamıma bir kullanıcı tıklarsa en çok şu kadar öderim." Reklamlar, teklifle kalitenin çarpımı olan bir puana göre üç **yuvaya** (*slot*) yerleştirilir. Üstteki yuva daha çok göze çarpar ve daha çok tıklanır.



Şekil 6.1: Sponsorlu Arama Açık Artırması oyununun ekranı. Üç reklam yuvası teklif × kalite puanına göre sıralanır. Yeşil çubuklar yuvaların tıklanma oranıdır: oyunun "Search Ads" senaryosunda %12, %7 ve %3,5.

Şekil 6.1'de oyunun ekranı görülür. Oyun iki basit soru sorar. Birincisi: bir tık reklamverene 8 TL kazandırıyorsa 8 TL teklif etmek en iyisi midir? İkincisi: en yüksek teklifi veren her zaman en üst yuvayı mı alır? İki sorunun cevabı da "hayır"dır. Bu notun büyük bölümü, bu iki cevabın nedenini açıklar. Birinci sorunun cevabı GSP açık artırmasının yapısından gelir ve 6.4.11'de sayılarla gösterilir. İkinci sorunun cevabı kalite puanından gelir ve 6.4.10'da hesaplanır.

Bu soruları yanıtlamak için önce tek nesneli ikinci fiyat açık artırmasını hatırlayalım. Bu açık artırmada (Vickrey açık artırması) en yüksek teklifi veren nesneyi alır, ama kendi teklifini değil ikinci en yüksek teklifi öder. Bu kuralda dürüst teklif, yani nesnenin teklif veren için değeri kadar teklif vermek, **baskın stratejidir**: ötekiler ne teklif ederse etsin, dürüst teklif hiçbir zaman daha kötü sonuç vermez.

Nedeni şudur: teklif ödemeyi belirlemez, yalnız kazanıp kazanmamayı belirler. Değerinden yüksek teklif veren, ödemenin değerini aştığı durumlarda da kazanır ve zarar eder. Değerinden düşük teklif veren ise kâr edebileceği durumlarda kaybeder. Sponsorlu aramadaki GSP kuralı bu fikri birden çok yuvaya genişletmeye çalışır, ama bu genişletme dürüstlüğü korumaz.

6.1.2 Çevrim içi platform nedir?

Tanım 6.1

Çevrim içi platform (*online platform*), kullanıcıları, satıcıları ve hizmet sağlayıcıları buluşturan dijital bir aracıdır. Değer, platformun kendisinden çok tarafların birbirine erişiminden doğar. Bu yapıya **iki taraflı pazar** (*two-sided market*) denir.

Tanımın ikinci cümlesi önemlidir. Bir fırın ekmek üretir ve ekmeğin değeri fırının kaç müşterisi olduğuna bağlı değildir. Bir pazar yeri ise kendisi bir şey üretmez. Alıcı için değeri, orada kaç satıcı ve ne kadar çeşit ürün olduğuna bağlıdır. Satıcı için değeri de orada kaç alıcı olduğuna bağlıdır. Platformun sattığı asıl hizmet, iki grubun birbirini bulma ve eşleşme maliyetini düşürmektir.

Platform yalnız bir buluşma yeri de değildir. Kuralları o koyar: kim katılabilir, kim kimi görür, kim ne öder. Bir pazar yeri satıcıların hangi sırayla listeleneceğine karar verir. Bir arama motoru reklamların hangi yuvaya gideceğine karar verir. Bir yolculuk uygulaması hangi sürücünün hangi yolcuya gideceğine ve fiyata karar verir. Bu kuralların tümü birlikte bir **mekanizmadır**: katılımcıların bildirdiği bilgilerden ve seçimlerinden bir sonuç ve ödeme üreten kural.

Tür	1. taraf	2. taraf	Örnek
E-ticaret pazar yeri	alıcılar	satıcılar	Trendyol, Hepsiburada, Amazon
Paylaşım ekonomisi	misafir / yolcu	ev sahibi / sürücü	Airbnb, BlaBlaCar, Uber
Sosyal medya	kullanıcılar	reklamveren, içerik üreticisi	Instagram, TikTok, Facebook
Arama motoru	arayanlar	reklamverenler	Google, Bing

Tablo 6.1: Platform türleri ve iki tarafı.

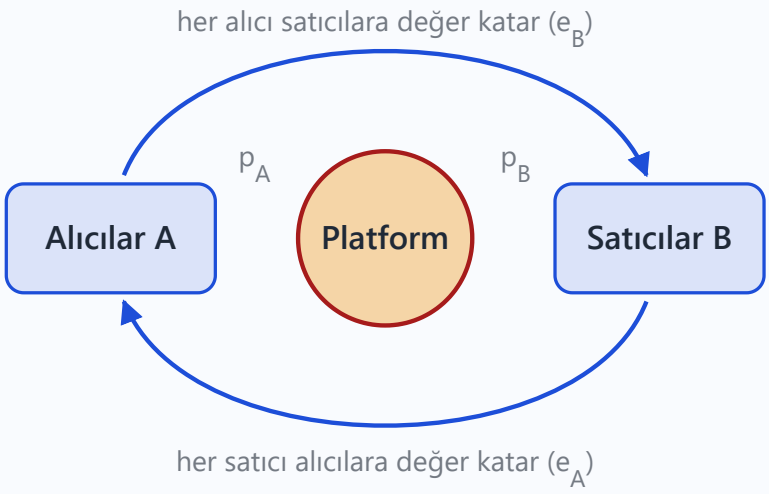
Tablo 6.1 dört yaygın platform türünü gösterir. Her satırda iki farklı kullanıcı grubu vardır ve iki grup da öteki grubun varlığından yarar görür. Sosyal medyada ikinci taraf iki gruptan oluşur: reklamverenler kullanıcıların dikkatini satın alır, içerik üreticileri ise kullanıcıları platforma çeker. Arama motorunda arayan kişi hiç ödeme yapmaz; gelirin tamamı reklamverenden gelir. Bu asimetri rastlantı değildir; nedeni 6.3'te hesaplanır.

6.1.3 İki taraflı pazar ve çapraz ağ etkisi

İki taraflı pazarı tek taraflı bir piyasadan ayıran özellik, tarafların birbirine değer katmasıdır. Bu bağ şöyle tanımlanır.

Tanım 6.2

Çapraz ağ etkisi (*cross-side network effect*): bir taraftaki katılımcı sayısı, diğer tarafın elde ettiği değeri belirler. Pazar yerinde çok satıcı, alıcı için çok seçenek demektir; çok alıcı da satıcı için çok müşteri demektir. Bu etkiye **dolaylı ağ etkisi** de denir.



Şekil 6.2: İki taraflı pazar. Platform alıcılardan p_A, satıcılardan p_B ücret alır. Üstteki ok her alıcının satıcılara kattığı değeri (e_B), alttaki ok her satıcının alıcılara kattığı değeri (e_A) gösterir.

Şekil 6.2 notun geri kalanında kullanılan gösterimi tanıtır. A tarafı alıcılar, B tarafı satıcılardır. p_A ve p_B iki tarafın platforma ödediği ücretlerdir. e_B, bir alıcının *satıcılara* kattığı değeri ölçer; yani B tarafının A tarafını ne kadar önemsediğidir. e_A da bir satıcının alıcılara kattığı değerdir. Gösterimde alt indis, değeri *alan* tarafı belirtir. Bu ayrıntı fiyat kuralında önem kazanır.

Şekildeki yapı üç sonuç doğurur.

- **Çapraz ağ etkisi.** Bir taraftaki katılımcı sayısı diğer tarafın elde ettiği değeri belirler. Platformun bir tarafı büyütmesi, öteki tarafı da kendiliğinden çeker.
- **Fiyat yapısı.** Platform yalnız toplam fiyat düzeyini değil, ücretin taraflar arasında nasıl bölüşüleceğini de seçer: (p_A, p_B). İki pazar yeri aynı toplam ücreti alabilir ama birinde ücreti alıcı, ötekinde satıcı öder. Sonuçları çok farklı olur.
- **Tavuk-yumurta sorunu.** Satıcı yoksa alıcı gelmez, alıcı yoksa satıcı gelmez. Yeni bir platform başlangıçta boştur ve iki taraf da ötekinin gelmesini bekler. Kısır döngüyü kırmak için taraflardan biri önce ücretsiz kullanım, indirim ya da doğrudan ödemeyle platforma "çekilmelidir".

Tavuk-yumurta sorunu, bu bölümün iki ana sorusunu doğurur. Birincisi: platform hangi koşulda büyür, hangi koşulda çöker? Bu, kritik kitle sorusudur (6.2). İkincisi: hangi taraf çekilmeli, yani sübvanse edilmelidir? Bu, fiyat yapısı sorusudur (6.3).

6.1.4 Platform bir mekanizma, kullanıcılar oyuncu

Mekanizma tasarımı, istenen bir sonucu doğuracak oyun kurallarını seçme sanatıdır. Platformlar bu sanatın en büyük uygulama alanlarından biridir. Platform tasarımı (ücret, sıralama ve puanlama kuralları) kullanıcıların stratejik davranışlarını yönlendirir. Kullanıcılar da teşviklere göre strateji seçer. Ortaya çıkan davranış, kurallar altındaki **Nash dengesidir**: her kullanıcının, ötekilerin davranışı veriyken kendi davranışını değiştirmek istemediği durum.

Bunun pratik bir sonucu vardır: bir kural, kullanıcıların "iyi niyetle" davranacağı varsayılarak değil, kendi çıkarlarını gözetecekleri varsayılarak değerlendirilmelidir. Platform bir teşvik koyduğunda, kullanıcılar o teşviki en iyi nasıl kullanacaklarını hesaplar. Bazen bu hesap platformun istediği davranışı üretir, bazen beklenmedik bir istismar üretir.

Platformun teşvikleri	Kullanıcıların stratejik tepkileri
Bonus: yoğun bölgede sürücüye ek ödeme (Uber)	Sahte yorumlar ve puan şişirme
İndirim, kupon, ücretsiz deneme	Çoklu hesap açma, bonus istismarı
Sıralama ve öneri algoritması → görünürlük	Algoritmayı manipüle etme (tıklama çiftlikleri)
Puan ve rozetler (güvenilir satıcı)	Misilleme amaçlı kötü değerlendirme

Tablo 6.2: Platformun teşvikleri ve kullanıcıların stratejik tepkileri.

Tablo 6.2 iki sütunu yan yana koyar. Soldaki her araç belirli bir davranışı ödüllendirmek için tasarlanmıştır. Sağdaki her davranış ise bir aracın kötüye kullanılmasıdır. Örneğin ücretsiz deneme yeni kullanıcı çekmek için konur; ama bir kişi çok sayıda hesap açarak denemeyi tekrar tekrar kullanabilir. Puan ve rozet sistemi güvenilir satıcıyı öne çıkarmak için konur; ama puan satın alınabiliyorsa rozet güvenilirliği değil, hile yapma isteğini ölçer.

Tablonun özeti şudur: doğru teşvik işbirliği ve verimlilik getirir, yanlış teşvik manipülasyon ve güven kaybı getirir. Platform tasarımcısının işi, kullanıcıların kendi çıkarını izleyerek platformun ve toplumun istediği sonuca ulaşmasını sağlayan kuralları bulmaktır. Notun geri kalanı bu işi dört alanda inceler: katılım (ağ etkileri), fiyat yapısı, reklam açık artırmaları ve itibar sistemleri.

Sık yapılan hata

"Platform yalnız bir aracıdır, kullanıcıların davranışı platformdan bağımsızdır." Bu yanlıştır. Kullanıcıların davranışı, platformun koyduğu kurallar altındaki bir dengedir. Aynı kullanıcılar farklı bir sıralama ya da ücret kuralı altında farklı davranır. Örneğin puan ortalaması sıralamayı belirliyorsa sahte yorum yazmanın getirisi artar; sıralama az yorumlu ürünlerin puanını ortalamaya çekiyorsa bu getiri azalır (6.5.3).

6.2 Ağ etkileri ve kritik kitle

Bir ürünün değeri onu kullanan kişi sayısıyla artıyorsa, “kaç kişi kullanacak?” sorusu ürünün kendisi kadar önemlidir. Ağ etkisinin türlerinden sonra kurulan tek parametreli basit model üç olguyu açıklar. Aynı platform için birden çok olası sonuç vardır. Bu sonuçlardan hangisine gidileceğini beklentiler belirler. Küçük bir başlangıç farkı büyük bir sonuç farkına dönüşebilir.

6.2.1 Ağ etkisi türleri

Tanım 6.3

Bir ürünü ya da platformu kullanmanın bir kişiye sağladığı değer, onu kullanan başka kişilerin sayısına bağlıysa bu bağa **ağ etkisi** (*network effect*) denir. Ek kullanıcı değeri artırıyorsa ağ etkisi *pozitif*, azaltıyorsa *negatiftir*.

Ağ etkileri dört başlıkta toplanır.

Doğrudan (aynı taraf)
: Değer, aynı taraftaki kullanıcı sayısından gelir. Bir kişinin arkadaşlarından ne kadar çoğu WhatsApp kullanıyorsa uygulama o kişi için o kadar değerlidir. Telefon, e-posta ve mesajlaşma uygulamaları bu türdendir. *(direct network effect)*

Dolaylı / çapraz
: Değer, öteki taraftaki kullanıcı sayısından gelir. Uber'de çok sürücü kısa bekleme süresi demektir; kısa bekleme çok yolcu çeker; çok yolcu da sürücüye daha çok iş demektir. Yolcu için önemli olan öteki yolcular değil sürücülerdir.

Negatif
: Kalabalık bazen değeri düşürür: tıkanıklık, bilgi kirliliği, uzun kuyruklar. İki taraflı pazarda aynı taraftaki satıcılar birbirine rakiptir; bir satıcı için yeni bir satıcı, alıcıların paylaşılacağı anlamına gelir.

Yerel
: Değer, tüm kullanıcılardan değil yakın çevreden gelir. İstanbul'daki bir yolcu için Ankara'daki sürücüler önemsizdir. Bir arkadaş grubunun tamamı başka bir mesajlaşma uygulamasına geçerse, uygulamanın toplam kullanıcı sayısı o grup için bir şey ifade etmez. Bu yüzden rekabet yerelde kazanılır: şehir şehir, grup grup.

Pozitif ağ etkisi bir **geri besleme döngüsü** (*feedback loop*) kurar. Büyüyen platform daha değerli olur, daha değerli olduğu için daha da büyür. Küçülen platform değer kaybeder, değer kaybettiği için daha da küçülür. Böyle bir döngüde pazarın bir yöne “devrilmesi” beklenir: ya platform kitleyi yakalar ve büyür ya da kitleyi yakalayamaz ve çöker. Ara durumlar kararsızdır. Bu fikri kesinleştirmek için bir model gerekir.

6.2.2 Basit bir benimseme modeli

Aşağıdaki model Easley ve Kleinberg'in (2010, Bölüm 17) ağ etkisi modelinin basitleştirilmiş bir biçimidir. Modelin amacı gerçekçi olmak değil, dinamiğin özünü en az sayıda varsayımla göstermektir.

Tanım 6.4

Benimseme modeli (*adoption model*). $N = 1000$ potansiyel kullanıcı vardır. Kullanıcı i'nin **tipi** θ_i, platforma ne kadar önem verdiğini gösteren 0 ile 1 arasında bir sayıdır ve bu aralıkta düzgün dağılmıştır: $\theta_i \sim U[0, 1]$. Nüfusun x oranı katılırsa kullanıcı i'nin katılmaktan elde ettiği fayda

$$\theta_i \cdot x - p$$

olur. Burada $p = 0{,}16$ katılmanın maliyetidir (ücret ya da katılma çabası). Katılmayan kullanıcının faydası 0'dır.

Tanımdaki her parçanın bir anlamı vardır. $\theta_i \cdot x$ terimi ağ etkisidir: platformun değeri, katılanların oranı x ile orantılı artar. Hiç kimse yoksa ($x = 0$) platform kimseye değer vermez. θ_i ise kişisel ilgidir: aynı kalabalık, platformu çok önemseyen birine (θ 1'e yakın) çok, az önemseyene (θ 0'a yakın) az değer verir. p herkes için aynı olan sabit bir maliyettir.

$\theta \sim U[0, 1]$ gösterimi, "tipler 0 ile 1 arasında eşit olasılıkla dağılmıştır" demektir. Bunun pratik anlamı basittir: herhangi bir a eşiği için tipi a'dan büyük olanların oranı $1 - a$'dır. Örneğin tipi 0,64'ten büyük olanlar nüfusun %36'sıdır; 1000 kişide 360 kişi.

Kim katılır? Kullanıcı, faydası negatif değilse katılır:

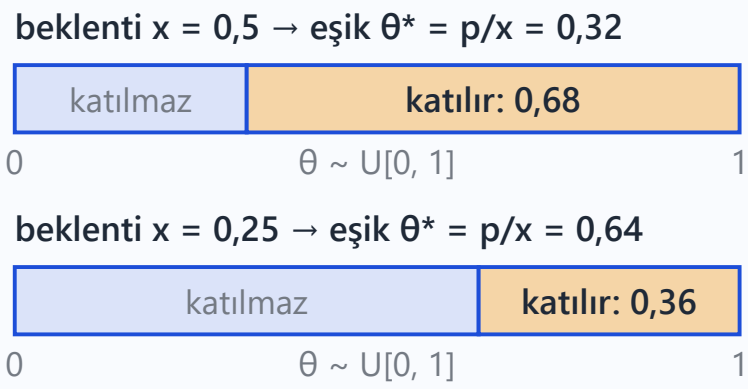
$$\theta_i \cdot x \geq p \iff \theta_i \geq p / x$$

İkinci adımda iki taraf x'e bölünmüştür; x pozitif olduğu için eşitsizliğin yönü değişmez. Bu eşitsizlik, katılmak için gereken en düşük tipi verir: $\theta^* = p / x$. Tipi bu eşiğin üstünde olanlar katılır. Düzgün dağılım nedeniyle bunların oranı $1 - p/x$'tir.

Burada ince ama önemli bir nokta vardır. Kullanıcı katılıp katılmayacağına karar verirken x'i henüz bilmez; çünkü x herkesin kararına bağlıdır. Kullanıcı bir **beklentiye** göre karar verir: "nüfusun x oranının katılacağını düşünüyorum." Bu beklentiyle gerçekte katılanların oranı şu fonksiyonla verilir:

$$F(x) = 1 - p/x \ (\text{x} \leq \text{p ise F(x)} = 0)$$

Parantez içindeki durum şöyle açıklanır. $x \leq p$ ise eşik $p/x \geq 1$ olur. Tipi 1'den büyük kimse yoktur; bu yüzden kimse katılmaz. Beklenen kalabalık çok küçükse, platformu en çok önemseyen kullanıcı bile maliyeti karşılayacak değeri göremez.



Şekil 6.3: Beklenti katılım eşiğini belirler. Beklenti $x = 0{,}5$ iken eşik $\theta^* = 0{,}32$'dir ve nüfusun 0,68'i katılır. Beklenti $x = 0{,}25$'e düşünce eşik 0,64'e çıkar ve yalnız 0,36'sı katılır.

Örnek 6.1

1. **Beklenti x = 0,5.** Herkes nüfusun yarısının (500 kişinin) katılacağını düşünsün. Eşik $\theta^* = p/x = 0{,}16/0{,}5 = 0{,}32$'dir. Tipi 0,32'den büyük olanlar katılır: $F(0{,}5) = 1 - 0{,}32 = 0{,}68$, yani 680 kişi.
2. **Beklenti x = 0,25.** Herkes 250 kişinin katılacağını düşünsün. Eşik $0{,}16/0{,}25 = 0{,}64$'e çıkar. Katılan oran $F(0{,}25) = 1 - 0{,}64 = 0{,}36$, yani 360 kişidir.
3. **Karşılaştırma.** Beklenti yarıya inince eşik iki katına çıkmıştır. Kalabalık küçüldükçe platformun değeri düştüğü için yalnız platformu daha çok önemseyenler katılır.
4. **Beklenti ile gerçekleşme.** İki durumda da gerçekleşen katılım beklentiden farklıdır: 0,5 beklenirken 0,68 katılmış, 0,25 beklenirken 0,36 katılmıştır. Beklenti ile gerçekleşmenin farkı bir sonraki dönemin beklentisini değiştirir.

Şekil 6.3 bu iki durumu çubuklarla gösterir. Her çubuk 0 ile 1 arasındaki tip ekseni üzerindedir. Turuncu kısım katılanlar, mavi kısım katılmayanlardır. Beklenti düştükçe sınır sağa kayar ve turuncu kısım daralır.

6.2.3 Kendini doğrulayan beklentiler: dengeler

Örnek 6.1'deki iki durumda da beklenti yanlış çıkmıştır. Beklentinin doğru çıktığı, yani herkesin beklediği kadar kişinin gerçekten katıldığı durum özel bir önem taşır.

Tanım 6.5

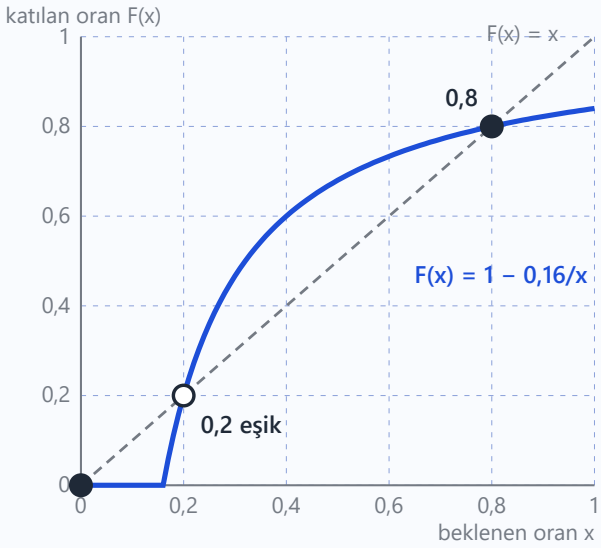
Benimseme modelinde bir **denge**, **kendini doğrulayan bir beklentidir** (*self-fulfilling expectation*): herkes x oranının katılacağını bekler ve bu beklentiyle gerçekten x oranı katılır. Biçimsel olarak denge, $F(x) = x$ denkleminin çözümüdür.

Bu tanım Nash dengesinin bu modeldeki karşılığıdır. Dengede hiçbir kullanıcı kararını değiştirmek istemez: katılanların faydası negatif değildir, katılmayanların faydası da katılsalar negatif olurdu. Beklenti doğru olduğu için kimse yanılmamıştır.

Örnek 6.2

1. **Denklemi kur.** $x > p$ için $F(x) = x$ koşulu $1 - 0{,}16/x = x$ olur.
2. **Kesirden kurtul.** İki taraf x ile çarpılır (x pozitiftir): $x - 0{,}16 = x^2$. Terimler bir tarafa toplanınca $x^2 - x + 0{,}16 = 0$ çıkar.
3. **İkinci derece denklemi çöz.** $ax^2 + bx + c = 0$ denkleminin kökleri $x = (-b \pm \sqrt{(b^2 - 4ac)}) / 2a$'dır. Burada $a = 1$, $b = -1$, $c = 0{,}16$'dır. Kökün içi (diskriminant) $1 - 4 \cdot 0{,}16 = 1 - 0{,}64 = 0{,}36$, karekökü 0,6'dır. Kökler $(1 - 0{,}6)/2 = 0{,}2$ ve $(1 + 0{,}6)/2 = 0{,}8$'dir.
4. **Denetle.** $F(0{,}2) = 1 - 0{,}16/0{,}2 = 1 - 0{,}8 = 0{,}2$ ✓ ve $F(0{,}8) = 1 - 0{,}16/0{,}8 = 1 - 0{,}2 = 0{,}8$ ✓.
5. **Sıfırı unutma.** $x = 0$ da bir dengedir: kimsenin katılmayacağı beklenirse $x \le p$ olduğu için $F(0) = 0$'dır ve gerçekten kimse katılmaz.

Modelin üç dengesi vardır: $x = 0$ (kimse yok), $x = 0{,}2$ (200 kullanıcı) ve $x = 0{,}8$ (800 kullanıcı).



Şekil 6.4: Benimseme eğrisi $F(x) = 1 - 0{,}16/x$ (mavi) ve 45° doğrusu $F(x) = x$ (kesikli). Kesişimler dengelerdir: 0 ve 0,8 (dolu noktalar) ile 0,2 (içi boş nokta). $x \le 0{,}16$ iken eğri sıfırdadır.

Şekil 6.4 dengeleri grafikle gösterir. Yatay eksen beklenen oran *x*'tir, dikey eksen gerçekte katılan oran *F(x)*'tir. Kesikli 45° doğrusu üzerindeki her noktada beklenti ile gerçekleşme eşittir. Dengeler de mavi eğrinin kesikli doğruyu kestiği noktalardır. Eğri 0,16'ya kadar yataydır, sonra hızla yükselir ve 1'e yaklaşır ama ulaşmaz.

Grafikte iki tür nokta vardır. Dolu noktalar (0 ve 0,8) **kararlı**, içi boş nokta (0,2) **kararsız** dengedir. Kararsız denge şu demektir: beklenti bu noktadan biraz saparsa, dinamik onu dengeden uzaklaştırır.

Bir platformun aynı koşullarda hem başarılı (800 kullanıcı) hem başarısız (0 kullanıcı) olabilmesi, bu modelin en önemli dersidir. Ürün, fiyat ve kullanıcılar aynıdır; sonucu belirleyen yalnız beklentidir. Bu, oyun kuramındaki **çoklu denge** sorununun platformlardaki biçimidir: koordinasyon oyunlarında olduğu gibi, herkes ötekilerin ne yapacağına göre hareket eder ve "herkes gelir" de "kimse gelmez" de kendini doğrular.

6.2.4 Devrilme adım adım

Dengeyi tanımlamak, oraya nasıl gidildiğini söylemez. Bunun için zamana bağlı basit bir dinamik varsayılır: her dönem kullanıcılar, bir önceki dönemde gözledikleri katılım oranını bu dönemin beklentisi olarak alır. Bu dönem katılanlar da gelecek dönemin beklentisini belirler:

$$x_{t+1} = F(x_t)$$

Burada x_t, t döneminde beklenen oran; x_{t+1}, o beklentiyle gerçekleşen ve bir sonraki dönemin beklentisi olan orandır. Dinamik, bir başlangıç beklentisi x_0 ile başlar.

Örnek 6.3

Eşiğin üstünden başlangıç: $x_0 = 0{,}25$. Platform başlangıçta 250 kullanıcı bekleniyormuş gibi bir izlenim yaratmış olsun.

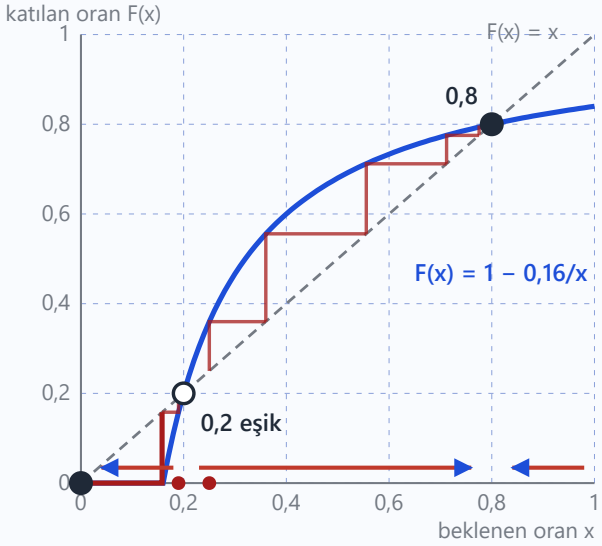
1. $t = 0 \to 1$: eşik $\theta \geq 0{,}16/0{,}25 = 0{,}64$; katılan $x_1 = 1 - 0{,}64 = 0{,}36$.
2. $t = 1 \to 2$: katılım beklentiyi aştı (0,36 > 0,25). Yeni beklenti 0,36 ile eşik $0{,}16/0{,}36 \approx 0{,}444$; $x_2 = 1 - 0{,}444 = 0{,}556$.
3. $t = 2 \to 3$: geri besleme sürer: $x_3 = 1 - 0{,}16/0{,}556 = 0{,}712$.
4. $t = 3 \to 4$: $x_4 = 1 - 0{,}16/0{,}712 = 0{,}775$. Artışlar küçülmeye başladı: 0,11, 0,20, 0,16, 0,06.
5. $t = 4 \to 6$: $x_5 = 0{,}794$, $x_6 \approx 0{,}798$. Yol kararlı denge 0,8'e yaklaşıyor. Pazar platform lehine devrildi: yaklaşık 800 kullanıcı.

Eşiğin altından başlangıç: $x_0 = 0{,}19$. Başlangıç beklentisi kritik eşiğin yalnız biraz altında olsun.

1. $x_1 = 1 - 0{,}16/0{,}19 = 1 - 0{,}842 = 0{,}158$. Katılım beklentinin *altında* kaldı.
2. $x_1 = 0{,}158 \leq p = 0{,}16$ olduğu için $x_2 = 0$: kimse katılmaz, platform çöker.

t	$x_0 = 0{,}25$	$x_0 = 0{,}19$
0	0,25	0,19
1	0,36	0,158
2	0,556	0
3	0,712	0
4	0,775	0
5	0,794	0
6	0,798	0

Tablo 6.3: $x_{t+1} = F(x_t)$ dinamiği, $p = 0{,}16$, iki başlangıç beklentisiyle.



Şekil 6.5: Devrilme dinamiğinin "örümcek ağı" çizimi. Kırmızı basamaklar $x_0 = 0{,}25$'ten başlayıp 0,8'e tırmanır; $x_0 = 0{,}19$'dan başlayan kısa yol 0'a düşer. Alttaki oklar dinamiğin yönünü gösterir: 0,2'nin sağında sağa (0,8'e doğru), solunda sola (0'a doğru).

Şekil 6.5 dinamiği grafik üzerinde izlemenin bir yolunu gösterir. Bu çizime **örümcek ağı çizimi** denir ve şöyle okunur. Yatay eksende x_t'den başlanır. Dikey olarak eğriye çıkılır; eğrinin yüksekliği $x_{t+1} = F(x_t)$'dir. Sonra yatay olarak 45° doğrusuna gidilir; bu, yeni değeri yatay eksene "taşır". Oradan yine dikey olarak eğriye çıkılır ve böyle devam eder. Eğri doğrunun üstündeyse basamaklar sağa, altındaysa sola ilerler.

Grafikte bu kuralın sonucu açıktır. 0,2 ile 0,8 arasında eğri doğrunun üstündedir: her dönem katılım beklentiyi aşar ve oran büyür. 0,8'in sağında eğri doğrunun altındadır: oran küçülür ve 0,8'e geri döner. 0,2'nin solunda da eğri doğrunun altındadır: oran küçülür ve 0'a düşer. Bu yüzden 0,8 ve 0 iki taraftan gelen yolları kendine çeker, 0,2 ise iki yöne iter.

Tablo 6.3 çarpıcı bir karşılaştırma verir. İki başlangıç arasındaki fark yalnız 0,06'dır (60 kullanıcı). Ama biri 800 kullanıcıya, öteki sıfıra gider. Platformlardaki "ya hep ya hiç" davranışının matematiksel nedeni budur.

Tanım 6.6

Benimseme modelindeki kararsız dengeye **kritik kitle** (*critical mass*) denir. Başlangıç beklentisi kritik kitlenin üstündeyse dinamik büyük kararlı dengeye, altındaysa sıfıra gider. Pazarın bu iki uçtan birine kendiliğinden gitmesine **devrilme** (*tipping*) denir.

Bu örnekte kritik kitle 0,2'dir, yani 200 kullanıcı. Beklenti bunun üstündeyse 0,8'e, altındaysa 0'a gidilir.

6.2.5 Hangi denge kararlıdır?

Kararlılık grafikte kolayca görülür; aynı sonuca hesapla da varılır. Bir dengenin yakınında küçük bir sapma olsun: $x_t = x^* + d$. Bir dönem sonra sapma yaklaşık olarak $F'(x^*) \cdot d$ olur. Burada $F'(x^*)$, eğrinin denge noktasındaki eğimidir. Eğim mutlak değerce 1'den küçükse sapma her dönem küçülür ve denge kararlıdır. 1'den büyükse sapma büyür ve denge kararsızdır.

$F(x) = 1 - p/x$ için eğim $F'(x) = p/x^2$'dir. Bu, p/x teriminin $p \cdot x^{-1}$ olarak yazılıp türevinin $-p \cdot x^{-2}$ olmasından ve önündeki eksi işaretinden gelir.

- $x^* = 0{,}8$: $F'(0{,}8) = 0{,}16/0{,}64 = 0{,}25 < 1$. Sapma her dönem dörtte birine iner: kararlı. Örneğin $x = 0{,}81$'den başlanırsa $F(0{,}81) \approx 0{,}802$ olur; 0,01'lik sapma 0,002'ye inmiştir.
- $x^* = 0{,}2$: $F'(0{,}2) = 0{,}16/0{,}04 = 4 > 1$. Sapma her dönem yaklaşık dört katına çıkar: kararsız. Örneğin $x = 0{,}21$'den başlanırsa $F(0{,}21) \approx 0{,}238$ olur; 0,01'lik sapma yaklaşık 0,038'e çıkmıştır.
- $x^* = 0$: 0'ın hemen yakınında F sıfırdır (eğri yataydır); eğim 0'dır. Küçük bir beklenti hemen sıfıra döner: kararlı.

Bu hesap, Şekil 6.4'teki dolu ve boş noktaların neden öyle çizildiğini açıklar. Eğri 45° doğrusunu *diktен* (eğimi 1'den büyük) kesiyorsa denge kararsız, *yatıktan* (eğimi 1'den küçük) kesiyorsa kararlıdır.

6.2.6 Ücret ve eşik: tasarım dersleri

Model tek bir parametreye, p'ye bağlıdır. Platform bu parametreyi etkileyebilir: ücreti düşürebilir, katılmayı kolaylaştırabilir, ilk kullanıcılara bonus verebilir. p değiştiğinde dengeler de değişir. Her p için aynı denklem çözülür: $x^2 - x + p = 0$.

p	dengeler	eşik (kritik kitle)
0,09	0, 0,1, 0,9	0,1
0,16	0, 0,2, 0,8	0,2

p	dengeler	eşik (kritik kitle)
0,21	0, 0,3, 0,7	0,3
0,25	0 ve 0,5 (teğet)	0,5
0,3	yalnız 0	—

Tablo 6.4: Katılma maliyeti p'ye göre dengeler ve kritik kitle.

Örnek 6.4

p = 0,21. Denklem $x^2 - x + 0{,}21 = 0$. Diskriminant $1 - 0{,}84 = 0{,}16$, karekökü 0,4. Kökler $(1 - 0{,}4)/2 = 0{,}3$ ve $(1 + 0{,}4)/2 = 0{,}7$. Denetim: $F(0{,}3) = 1 - 0{,}21/0{,}3 = 1 - 0{,}7 = 0{,}3$ ✓, $F(0{,}7) = 1 - 0{,}21/0{,}7 = 1 - 0{,}3 = 0{,}7$ ✓.

p = 0,25. Diskriminant $1 - 1 = 0$; tek kök $x = 1/2 = 0{,}5$. Eğri bu noktada 45° doğrusuna yalnız değer, kesmez (teğet). 0,5'in iki yanında da eğri doğrunun altındadır. Bu yüzden 0,5'in altından başlanırsa yol 0'a gider. Üstünden başlanırsa katılım azalarak 0,5'e yaklaşır. Bu denge yalnız bir yönden kararlıdır: küçük bir düşüş pazarı çökertir.

p = 0,3. Diskriminant $1 - 1{,}2 = -0{,}2 < 0$: gerçek kök yoktur. Eğri 45° doğrusunun hep altındadır; hangi beklentiyle başlanırsa başlansın katılım küçülür ve 0'a gider. Pazar hiç oluşmaz.

Tablo 6.4 ve Örnek 6.4 iki genel sonucu gösterir. Birincisi: ücret düştükçe eşik düşer ve büyük dengedeki kullanıcı sayısı artar. p 0,21'den 0,09'a inince kritik kitle 300'den 100 kullanıcıya iner. İkincisi: $x^2 - x + p = 0$ denkleminin gerçek kökü olması için diskriminant $1 - 4p \geq 0$, yani $p \leq 0{,}25$ olmalıdır. $p > 0{,}25$ ise pazar hiç oluşmaz. Kritik kitle ile büyük dengenin toplamı her zaman 1'dir (0,2 + 0,8, 0,3 + 0,7); çünkü ikinci derece denklemde köklerin toplamı $-b/a = 1$'dir. Eşik yükseldikçe büyük denge aynı miktarda düşer.

Bu sonuçlar, platformların başlangıçta neden olağandışı davrandığını açıklar. Dört yaygın strateji vardır:

Başlangıç sübvansiyonu
: Ücretsiz dönem ya da bonus p'yi küçültür ve kritik kitleyi düşürür. Platform kitle oluştuktan sonra ücreti yükseltebilir; çünkü büyük denge kararlıdır ve küçük bir ücret artışı onu yıkmaz.

Tohumlama
: İlk içeriği, satıcıyı ya da sürücüyü platform kendisi sağlar (*seeding*). Böylece gerçek katılım x, kullanıcıların katılımı başlamadan önce sıfırdan büyük olur.

Beklenti yönetimi
: "Herkes burada" algısı başlangıç beklentisi x_0'ı eşiğin üstüne taşır. Model, beklentinin kendisinin bir girdi olduğunu gösterir; reklam ve tanıtım bu girdiyi değiştirir.

Tek başına değer
: Ağ olmadan da işe yarayan bir araçla başlanır, ağ sonra kurulur. Örneğin fotoğraf düzenleme aracı olarak başlayan bir uygulama, kullanıcıları ağ etkisi olmadan da çeker; paylaşım özelliği

eklendiğinde kitle hazırdır. Modelde bu, faydaya x'ten bağımsız bir terim eklemek demektir; böylece $x = 0$ artık denge olmaz.

Tarih iki yönü de gösterir. WhatsApp'ın değeri kullanıcı arttıkça katlanarak büyüdü. Friendster ise kullanıcı kaybı başlayınca aynı geri besleme döngüsüyle ters yönde çöktü. İki durumda da işleyen mekanizma aynıdır; fark, beklentinin eşiğin hangi tarafında olduğudur.

6.2.7 Ağ etkilerinde sık hatalar

Sık yapılan hata

Beklentiyi unutmak. $F(x)$ hesaplanırken $1 - p$ yazmak, yani katılımın x'e bağlı olduğunu unutmak. $p = 0{,}16$ ve $x = 0{,}4$ için doğru cevap $F(0{,}4) = 1 - 0{,}16/0{,}4 = 1 - 0{,}4 = 0{,}6$'dır, 0,84 değildir.

Eşik tipiyle katılan oranı karıştırmak. Aynı örnekte 0,4 katılma eşiği olan tiptir ($\theta^* = p/x$), katılan oran değil. Katılan oran eşiğin üstündeki kısımdır: $1 - \theta^*$.

Sıfır dengesini atlamak. $x^2 - x + p = 0$ denklemi yalnız pozitif dengeleri verir. Denklem kurulurken x ile çarpıldığı için $x = 0$ ayrıca denetlenmelidir; kimsenin katılmadığı durum her zaman bir dengedir.

Kararsız dengeyi "orta yol" sanmak. 0,2 bir denge olsa da pazar orada kalmaz. Gerçek hayatta beklentiler hiçbir zaman tam 0,2 olmayacağı için 0,2 bir hedef değil, bir sınırdır.

6.3 Platform fiyatlandırması

Tek taraflı bir piyasada fiyat sorusu basittir: ürün kaça satılmalı? İki taraflı bir platformda soru ikiye ayrılır: toplam ne kadar alınmalı ve bu toplam taraflar arasında nasıl bölüşülmeli? İkinci sorunun cevabı sayısal bir modelle bulunur ve ilk bakışta şaşırtıcıdır: kâr maksimize eden platform taraflardan birine para öder.

6.3.1 İki taraflı fiyatlandırma ilkeleri

Rochet ve Tirole'nin (2003) gözlemine göre bir platform toplam fiyat düzeyinin yanında **fiyat yapısına** (*price structure*) da karar verir: ücreti hangi taraf, ne kadar öder? Tek taraflı bir firma için "fiyat yapısı" diye bir soru yoktur. İki taraflı pazarda ise yapı, toplam düzey kadar önemlidir. Çünkü bir tarafın ücreti yalnız o tarafın katılımını değil, çapraz ağ etkisi yoluyla öteki tarafın katılımını da değiştirir.

Temel ilke iki maddede özetlenir:

- Diğer tarafa daha çok değer katan ya da fiyata daha duyarlı olan taraf ucuzlatılır, hatta **sübvanse** edilir (*subsidy*). Sübvansiyon negatif fiyat demektir: platform o taraftaki her katılımcıya ödeme yapar.
- Bu kayıp, sübvanse edilen tarafın platforma çektiği öteki taraftan alınan ücretle karşılanır.

Günlük hayatta bu ilkenin birçok örneği vardır:

Arama ve sosyal medya
: Kullanıcı ücretsiz kullanır, reklamveren öder. Kullanıcıların dikkati reklamveren için değerlidir; kullanıcıdan ücret almak bu dikkati azaltır.

Kredi kartı
: Kart sahibine puan ve mil verilir, işyerinden komisyon alınır. Kart sahibinin kartı kullanması işyeri için değerlidir; puan bu kullanımı artırır.

Pazar yeri
: Alıcıya kupon verilir, satıcıdan komisyon alınır.

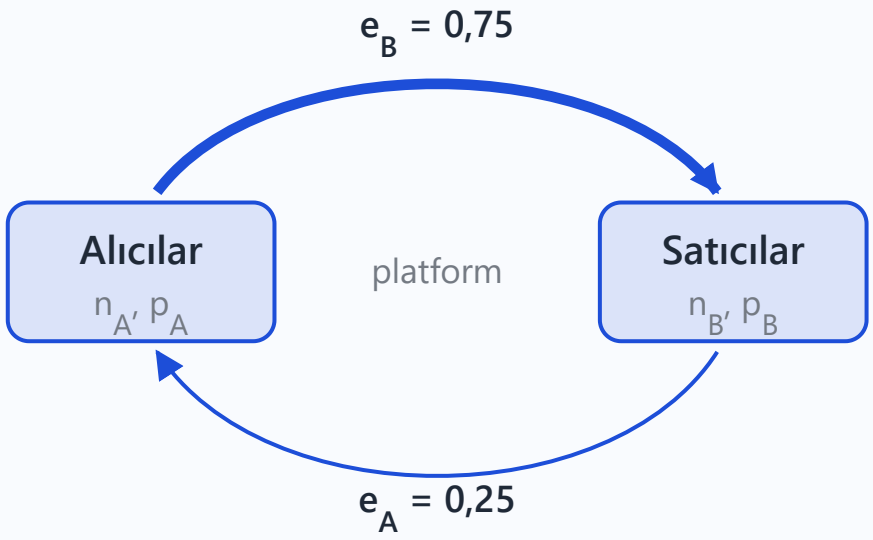
Gelir modelleri
: Komisyon, üyelik, reklam, işlem başına ücret. Bu modeller ücretin hangi tarafa ve hangi biçimde yükleneceğini belirler.

Armstrong (2006) basit bir doğrusal modelde bu ilkeyi tek bir kurala indirger:

> $$p_A = \text{kâr payı} - \text{bir A katılımcısının B tarafına kattığı değer}$$

Kurala göre ikinci terim büyükse fiyat negatife düşer. **Kâr payı** (*markup*), ağ etkisi olmasaydı platformun bir katılımcıdan alacağı fiyat ile maliyet arasındaki farktır. Kuralın mantığı şudur: platform bir A katılımcısını çektiğinde iki şey kazanır. Birincisi, o katılımcıdan aldığı ücret. İkincisi, o katılımcının B tarafına kattığı değer; çünkü B tarafı bu değer için daha çok ödemeye razı olur. İkinci kazanç büyükse platform, A katılımcısını çekmek için ona para vermeyi göze alır. Kural 6.3.5'te genel olarak türetilir.

6.3.2 Sayısal örnek: pazar yeri modeli



Şekil 6.6: Pazar yeri modeli. Kalın üst ok: her alıcı satıcılara $e_B = 0{,}75$ değer katar. İnce alt ok: her satıcı alıcılara $e_A = 0{,}25$ değer katar.

Bir pazar yeri düşünelim. n katılımcı sayısını (bin kişi), p aylık ücreti (TL) gösterir. Bir katılımcıya hizmet vermenin platforma ek maliyeti yoktur (marjinal maliyet 0). Satıcı için alıcı çok değerlidir ($e_B = 0{,}75$), alıcı için satıcı çeşitliliği daha az önemlidir ($e_A = 0{,}25$). Katılım şu iki denklemle belirlenir:

$$n_A = 20 + 0{,}25 \cdot n_B - p_A$$
$$n_B = 80 + 0{,}75 \cdot n_A - p_B$$

Platformun kârı iki taraftan alınan ücretlerin toplamıdır:

$$\Pi = p_A \cdot n_A + p_B \cdot n_B$$

Denklemleri tek tek okumak modelin varsayımlarını açığa çıkarır. Alıcı denklemindeki 20, ücret sıfır ve hiç satıcı yokken bile gelecek alıcı sayısıdır (bin kişi). $0{,}25 \cdot n_B$, satıcıların çektiği ek alıcılardır: bin satıcı başına 250 alıcı. $-p_A$, ücretin kaçırdığı alıcılardır: ücretteki her 1 TL artış bin alıcı kaybettirir. Satıcı denklemi aynı biçimdedir ama sabit terim (80) ve ağ etkisi katsayısı (0,75) daha büyüktür. Satıcıların alıcılara verdiği önem, alıcıların satıcılara verdiği önemin üç katıdır.

Platformun sorusu şudur: kârı maksimize eden (p_A, p_B) nedir ve hangi taraf sübvanse edilir?

6.3.3 En kârlı fiyatlar adım adım

Doğrudan fiyatlar üzerinden hesap yapmak zordur, çünkü her fiyat iki katılımı birden değiştirir ve katılımlar birbirine bağlıdır. Hesabı kolaylaştıran bir yol vardır: fiyat yerine katılımları karar değişkeni yapmak. Her (n_A, n_B) çiftine tek bir fiyat çifti karşılık gelir. Platform "şu kadar alıcı ve şu kadar satıcı

istiyorum” diye düşünebilir ve bu sayıları getiren fiyatları denklemlerden okuyabilir.

Çözümde **kısmi türev** kullanılır. İki değişkenli bir fonksiyonun bir değişkene göre kısmi türevi, öteki değişken sabit tutularak alınan türevdir ve ∂ ile gösterilir. Örneğin n_A n_B'nin n_A'ya göre kısmi türevi n_B'dir; çünkü n_B sabit bir sayı gibi davranır. İki değişkenli bir fonksiyonun tepe noktasında iki kısmi türev de sıfırdır: tepede hangi yöne adım atılırsa atılsın eğim sıfırdır.

Örnek 6.5

1. **Katılımları karar değişkeni yap.** Katılım denklemleri fiyatlar için çözülür: $p_A = 20 + 0{,}25\, n_B - n_A$ ve $p_B = 80 + 0{,}75\, n_A - n_B$.
2. **Kâr ifadesine yerleştir.** $p_A\, n_A = 20\, n_A + 0{,}25\, n_A n_B - n_A^2$ ve $p_B\, n_B = 80\, n_B + 0{,}75\, n_A n_B - n_B^2$. Toplanınca

   $$\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$$

   $n_A n_B$ teriminin katsayısı iki çapraz etkinin toplamıdır (burada 1). Bu terim, iki tarafın birbirine kattığı değeri platformun toplayabildiğini gösterir.
3. **Kısmi türevleri sıfıra eşitle.** n_B sabitken $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$. n_A sabitken $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$. İki doğrusal denklem elde edilir.
4. **Denklemleri çöz.** İlkinden $n_A = (20 + n_B)/2$. İkincisine yerleştirilince $80 + (20 + n_B)/2 - 2n_B = 0$. İki taraf 2 ile çarpılır: $160 + 20 + n_B - 4n_B = 0$, yani $3n_B = 180$ ve $n_B = 60$. Geri yerleştirince $n_A = 80/2 = 40$.
5. **Fiyatları oku.** $p_A = 20 + 0{,}25 \cdot 60 - 40 = 20 + 15 - 40 = -5$ ve $p_B = 80 + 0{,}75 \cdot 40 - 60 = 80 + 30 - 60 = 50$.
6. **Kârı hesapla.** $\Pi = (-5) \cdot 40 + 50 \cdot 60 = -200 + 3000 = 2800$.
7. **Tepe mi?** Kârda $-n_A^2$ ve $-n_B^2$ terimleri fonksiyonu bir “kubbe” biçimine sokar; çapraz terimin katsayısı (1) bu kubbeyi bozacak kadar büyük değildir (koşul: katsayı 2'den küçük olmalı). Bulunan nokta bu nedenle bir tepedir.

Çözüm: alıcılara ayda 5 TL ödenir ($p_A = -5$), satıcılar 50 TL öder. Platformda 40 bin alıcı ve 60 bin satıcı bulunur; kâr 2800 bin TL'dir.

Sonucun yorumu bölümün ana fikridir. Platform alıcılara para öder ama zarar etmez. Alıcılara ödenen 200, satıcılardan alınan 3000'in yanında küçüktür. Satıcılar 50 TL ödemeye razıdır, çünkü platformda 40 bin alıcı vardır ve her alıcı satıcılara değer katar.

6.3.4 Senaryoların karşılaştırması

Sübvansiyonun gerçekten kârlı olduğunu görmek için başka fiyat yapıları denenir. Her senaryoda alıcı ücreti sabitlenir ve satıcı ücreti o koşul altında en iyi biçimde seçilir.

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π
Optimum	0,25 / 0,75	−5	50	40	60	2800
Alıcıya ücretsiz	0,25 / 0,75	0	47,5	34,6	58,5	2776,9
Alıcıdan 10 TL	0,25 / 0,75	10	42,5	23,8	55,4	2592,3
Etkiler ters	0,75 / 0,25	25	30	40	60	2800

Tablo 6.5: Pazar yeri modelinde dört senaryo (katılım bin kişi, ücret TL, kâr bin TL).

Örnek 6.6

Alıcıya ücretsiz ($p_A = 0$). Platform alıcıya para ödemek yerine yalnız ücret almamayı seçsin. Bu kez katılımlar fiyatlardan hesaplanır.

1. Satıcı denklemi alıcı denklemine yerleştirilir: $n_A = 20 + 0{,}25(80 + 0{,}75\, n_A - p_B)$. Açılınca $n_A = 40 + 0{,}1875\, n_A - 0{,}25\, p_B$.
2. n_A'lı terimler bir yana toplanır: $0{,}8125\, n_A = 40 - 0{,}25\, p_B$. 0,8125 = 13/16 olduğu için $n_A = (640 - 4\, p_B)/13$.
3. Satıcı sayısı: $n_B = 80 - p_B + 0{,}75\, n_A = (1520 - 16\, p_B)/13$.
4. Kâr yalnız satıcılardan gelir: $\Pi = p_B\,(1520 - 16\, p_B)/13$. Bu bir ters parabol; tepesi türevin sıfır olduğu yerdedir: $1520 - 32\, p_B = 0$, yani $p_B = 47{,}5$.
5. Katılımlar: $n_A = (640 - 190)/13 = 450/13 \approx 34{,}6$, $n_B = (1520 - 760)/13 = 760/13 \approx 58{,}5$. Kâr: $47{,}5 \cdot 760/13 = 36100/13 \approx 2776{,}9$.

Alıcıya ücretsiz yapı optimumdan yalnız 23,1 bin TL az kazandırır, ama yine de geride kalır: alıcı sayısı 40'tan 34,6'ya düşer ve satıcılar daha az öder.

Alıcıdan 10 TL. Aynı yöntemle $n_A = (480 - 4\, p_B)/13$ ve $n_B = (1400 - 16\, p_B)/13$ bulunur. Kâr $\Pi = [10(480 - 4p_B) + p_B(1400 - 16p_B)]/13$; türev $1360 - 32\, p_B = 0$ verir, yani $p_B = 42{,}5$. Katılımlar $310/13 \approx 23{,}8$ ve $720/13 \approx 55{,}4$, kâr $33700/13 \approx 2592{,}3$'tür.

Tablo 6.5 üç dersi bir arada verir.

- **Sübvansiyon kârlıdır.** Alıcıdan ücret alındıkça kâr düşer: 2800 → 2776,9 → 2592,3. Alıcıdan 10 TL alınırsa alıcı sayısı 23,8 bine düşer, satıcılar da azalır. Alıcıdan alınan 238 bin TL'lik gelir, satıcı tarafındaki kaybı karşılamaz.
- **Neden?** Fazladan bir alıcı satıcılara $0{,}75 \cdot 60 = 45$ TL değer katar ve platform bunu p_B ile geri alır. Alıcının kâr payı ise 40'tır. $45 > 40$ olduğu için alıcı sübvanse edilir: $40 - 45 = -5$.
- **Karşı olgu.** Çapraz etkiler ters olsaydı, yani alıcılar satıcılara az değer katsaydı ($e_A = 0{,}75$, $e_B = 0{,}25$), sübvansiyon kalkardı. Kâr ifadesi değişmez (çapraz terim yine $0{,}75 + 0{,}25 = 1$), bu yüzden katılımlar yine 40 ve 60'tır. Ama fiyatlar değişir: $p_A = 20 + 0{,}75 \cdot 60 - 40 = 25$, $p_B = 80 + 0{,}25 \cdot 40 - 60 = 30$. Bu kez alıcılar ödemeye razıdır, çünkü 60 bin satıcı onlara çok değer katar.

Son satır bir inceliği gösterir. Kâr ve katılımlar yalnız çapraz etkilerin *toplamına* bağlıdır. Fiyat yapısı ise etkilerin *dağılımına* bağlıdır. Platformun toplam olarak ne kadar değer topladığı değişmez; değişen, bu değerin hangi taraftan toplandığıdır. Sunumdaki betik (p_A, p_B) ızgarasını 0,5 TL adımla tarar ve en yüksek kârı $(-5, 50)$ noktasında bulur. Bu, türevle bulunan çözümün sayısal denetimidir.

6.3.5 Genel kural: kâr payı eksi katılan değer

Sayısal örnekteki fiyatlar $p_A = n_A - e_B \cdot n_B = 40 - 0{,}75 \cdot 60 = -5$ kuralına uyar. Bu bir rastlantı değildir. Aynı biçimdeki her doğrusal modelde kural geçerlidir.

Teorem 6.1 (Doğrusal modelde fiyat kuralı)

Katılımlar $n_A = a + e_A\, n_B - p_A$ ve $n_B = b + e_B\, n_A - p_B$ ile verilsin, marjinal maliyet 0 ve $e_A + e_B < 2$ olsun. Kârı maksimize eden fiyatlar şunlardır:

$$p_A = n_A - e_B \cdot n_B,\; p_B = n_B - e_A \cdot n_A$$

Burada n_A, A tarafının kâr payıdır; $e_B \cdot n_B$, bir A katılımcısının B tarafına kattığı toplam değerdir.

İspat. Örnekteki adımlar genel sayılarla tekrarlanır. Fiyatlar katılımlar cinsinden $p_A = a + e_A\, n_B - n_A$ ve $p_B = b + e_B\, n_A - n_B$ olarak yazılır. Kâr ifadesine yerleştirilince $\Pi = a\, n_A + b\, n_B + (e_A + e_B)\, n_A n_B - n_A^2 - n_B^2$ çıkar.

n_A'ya göre kısmi türev sıfıra eşitlenir: $a + (e_A + e_B)\, n_B - 2n_A = 0$. Buradan $a = 2n_A - (e_A + e_B)\, n_B$ elde edilir.

Bu ifade p_A formülüne yerleştirilir: $p_A = 2n_A - (e_A + e_B)\, n_B + e_A\, n_B - n_A$. $e_A\, n_B$ terimleri birbirini götürür ve $p_A = n_A - e_B\, n_B$ kalır. B tarafı için hesap simetriktir.

Son olarak bulunan noktanın bir tepe olduğu gösterilir. Kârın ikinci türevleri -2, -2 ve çapraz türev $e = e_A + e_B$'dir. İki değişkenli bir fonksiyonda $(-2)(-2) - e^2 > 0$ ve köşegen terimler negatifse durağan nokta bir tepedir. Bu koşul $e < 2$'ye denktir ve teoremin varsayımıdır. □

Kuralın neden "kâr payı" terimi içerdiği şöyle açıklanır. Ağ etkisi olmasaydı ($e_A = e_B = 0$), A tarafı için tek taraflı bir tekel problemi kalırdı: $n_A = a - p_A$ talebinde kâr $p_A(a - p_A)$'dır ve en iyi fiyat $a/2$'dir; bu fiyatta $n_A = a/2$ olur. Yani doğrusal talepte tekel fiyatı katılımcı sayısına eşittir. Kuraldaki n_A terimi bu tekel fiyatıdır. İkinci terim ise ağ etkisinin düzeltmesidir.

Örnek 6.7

Teorem 6.1 tablodaki senaryolara uygulanır.

1. **Optimum ($e_A = 0{,}25$, $e_B = 0{,}75$).** $p_A = 40 - 0{,}75 \cdot 60 = -5$ ✓; $p_B = 60 - 0{,}25 \cdot 40 = 50$ ✓.

2. **Etkiler ters.** $p_A = 40 - 0{,}25 \cdot 60 = 25$ ✓; $p_B = 60 - 0{,}75 \cdot 40 = 30$ ✓.
3. **Etkiler eşit ($e_A = e_B = 0{,}5$).** Toplam yine 1 olduğu için katılımlar 40 ve 60'tır. $p_A = 40 - 0{,}5 \cdot 60 = 10$, $p_B = 60 - 0{,}5 \cdot 40 = 40$. Kâr $400 + 2400 = 2800$'dür. Bu durumda sübvansiyon yoktur: bir alıcının satıcılara kattığı değer (30) kâr payından (40) küçüktür.

Kural, Tablo 6.5'teki "Neden?" sorusunu tek satırda cevaplar. Sübvansiyonun ölçütü tarafın kalabalığı, gelir getirmesi ya da satıcı olması değildir. Ölçüt, o taraftaki bir katılımcının öteki tarafa kattığı değerin kâr payını aşıp aşmadığıdır.

6.3.6 Kod izleme: katilim()

Örnek 6.5'te katılımlar iki denklemin ortak çözümü olarak bulundu. Gerçekte kullanıcılar bir denklem sistemi çözmez; her taraf ötekinin o anki büyüklüğüne tepki verir. Aşağıdaki kısa fonksiyon bu süreci taklit eder: fiyatlar sabittir, taraflar sırayla birbirine tepki verir.

```
def katilim(pA, pB, tur=8):
  nA, nB = 0, 0
  for t in range(tur):
    nA = 20 + 0.25 * nB - pA   # alıcılar
    nB = 80 + 0.75 * nA - pB   # satıcılar
  return nA, nB
```

Kodun mantığı şöyledir:

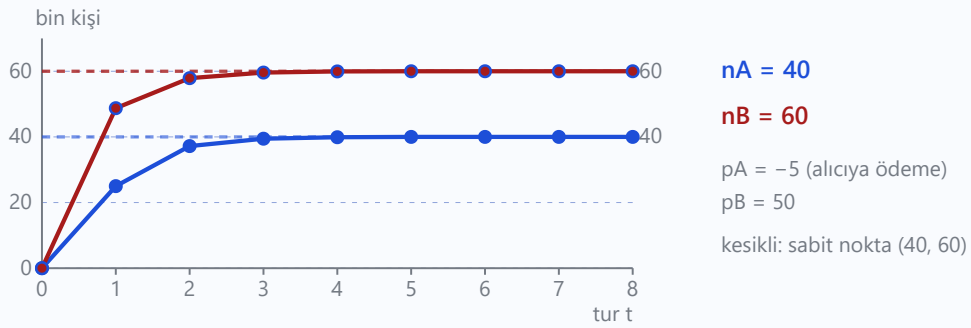
- Başlangıçta platform boştur: `nA, nB = 0, 0`.
- Her turda önce alıcılar, *mevcut* satıcı sayısına tepki verir: `nA = 20 + 0.25 * nB - pA`.
- Ardından satıcılar, *yeni* alıcı sayısına tepki verir: `nB = 80 + 0.75 * nA - pB`. İkinci satır, aynı turda birinci satırın güncellediği `nA`'yı kullanır.
- Döngü `tur` kez tekrarlandıktan sonra son katılımlar döndürülür. Varsayılan 8 tur, bu örnek için yeterlidir.

Fonksiyon optimum fiyatlarla, `katilim(-5, 50)` olarak çağrılır. `pA = -5` olduğu için alıcı denkleminde `- pA` terimi +5 olur.

Tur	nA hesabı	nA	nB hesabı	nB
başlangıç	—	0	—	0
1	20 + 0,25 · 0 + 5	25	80 + 0,75 · 25 − 50	48,8
2	20 + 0,25 · 48,8 + 5	37,2	80 + 0,75 · 37,2 − 50	57,9
3	20 + 0,25 · 57,9 + 5	39,5	80 + 0,75 · 39,5 − 50	59,6
4	20 + 0,25 · 59,6 + 5	39,9	80 + 0,75 · 39,9 − 50	59,9

Tur	nA hesabı	nA	nB hesabı	nB
5–8	…	40	…	60

Tablo 6.6: `katilim(-5, 50)` izlemesi (bin kişi, bir ondalığa yuvarlanmış).



Şekil 6.7: `katilim(-5, 50)` sırasında katılımların turlara göre değişimi. Mavi alıcılar (n_A), kırmızı satıcılar (n_B). İki eğri de birkaç turda kesikli sabit nokta çizgilerine, 40 ve 60'a yerleşir.

Tablo 6.6 ve Şekil 6.7 sürecin iki özelliğini gösterir. Birincisi: ilk turda alıcı sübvansiyonu satıcıyı çeker. 25 bin alıcı, ücreti 50 TL olmasına rağmen 48,75 bin satıcıyı platforma getirir. İkincisi: çapraz etki iki tarafı birlikte büyütür. Her turda alıcı artışı satıcıyı, satıcı artışı alıcıyı artırır. Süreç (40, 60) sabit noktasına varır. Bu nokta, iki tarafın beklentilerinin birbirini doğruladığı **katılım dengesidir**. Kâr *−5 · 40 + 50 · 60 = 2800*'dür.

Neden yakınsar? Sabit noktadan sapma her turda belli bir oranda küçülür. Satıcı sayısının 60'tan uzaklığı ilk turda 11,25'tir (60 − 48,75). Bu uzaklık alıcı denkleminden geçerken $e_A = 0{,}25$ ile, satıcı denkleminden geçerken $e_B = 0{,}75$ ile çarpılır. Bir turda sapma $e_A \cdot e_B = 0{,}25 \cdot 0{,}75 = 0{,}1875$ katına iner: 11,25 → 2,11 → 0,40 → 0,07. Beş turdan sonra fark yuvarlamada görünmez olur.

Yakınsama koşulu çapraz etkilerin çarpımıdır: $e_A \cdot e_B < 1$. Çarpım 1'i aşsaydı sapma her turda büyür, katılım patlar ya da çökerdi. Bu, ağ etkisinin fazla güçlü olduğu ve 6.2.4'teki devrilme davranışının ortaya çıktığı durumdur. Sabit noktanın kendisi yine vardır ama artık kararsızdır; benimseme modelindeki 0,2 eşiği gibi.

Sık yapılan hata

İzlemede ikinci satırda eski `nA` değerini kullanmak. Kod sırayla çalışır: tur 1'de `nB` hesaplanırken `nA` zaten 25'tir. Doğru değer *80 + 0,75 · 25 − 50 = 48,75*'tir. 30 (= 80 − 50) cevabı alıcıların katkısını, 18,75 cevabı ise sabit terimi ve ücreti unutmuştur. Fiyatların döngü boyunca değişmediği de unutulmamalıdır: yakınsamayı fiyat güncellemesi değil, $e_A \cdot e_B < 1$ koşulu sağlar.

6.3.7 Oyundaki model

İki Taraflı Platform Fiyatlandırması oyunu aynı modeli iki parametre ekleyerek kullanır:

$$n_A = taban_A - s_A \cdot p_A + e_A \cdot n_B$$
$$n_B = taban_B - s_B \cdot p_B + e_B \cdot n_A$$

- **taban**: ücret 0 ve karşı taraf boşken katılan kullanıcı sayısıdır.
- **s**: fiyat duyarlılığıdır, yani ücretteki 1 birimlik artışın kaç kullanıcı kaybettirdiği. Pazar yeri senaryosunda $s_A = 11$, $s_B = 5$'tir; alıcılar fiyata çok daha duyarlıdır. Sayısal modelde $s = 1$ alınmıştı.
- Katılım 0–100 kullanıcıyla sınırlıdır; ücretler −5 ile 10 arasında tam sayıdır.
- Kâr $\Pi = p_A \cdot n_A + p_B \cdot n_B$'dir ve katılım yine karşılıklı tepkinin sabit noktasıdır (oyunun kodu `katilim()` gibi bir döngüyle hesaplar).

Seviye	e_A / e_B	En kârlı (p_A, p_B)	n_A / n_B	Π
1	0,10 / 0,30	(2, 4)	25,3 / 22,6	140,8
2	0,15 / 0,60	(0, 6)	50,3 / 35,2	211,0
3	0,20 / 0,85	(−2, 9)	78,3 / 56,6	352,5

Tablo 6.7: Oyunun pazar yeri senaryosunda (taban 45 / 35) her seviyenin en kârlı tam sayı ücretleri.

Tablo 6.7 kuralın oyundaki karşılığını gösterir. B tarafı A'ya ne kadar çok değer verirse (e_B büyüdükçe) A'nın en kârlı ücreti o kadar düşer: 2, sonra 0, sonra −2. Seviye 3'te A sübvanse edilir. Bu seviyede iki tarafa eşit ücret, örneğin (3, 3), yalnız 167,0 kâr getirir. A'yı sübvanse edip B'den yüksek ücret almak kârı iki katından fazlasına çıkarır. Seviye 3'teki dengede A'ya ödenen $2 \cdot 78{,}3 \approx 157$, B'den alınan $9 \cdot 56{,}6 \approx 509$'un yanında küçüktür.

Fiyat duyarlılığı kuralı biraz değiştirir. Aynı türetme s katsayılarıyla yapılınca $p_A = n_A/s_A - e_B \cdot n_B/s_B$ çıkar. Kâr payı n_A/s_A'ya döner: fiyata duyarlı tarafın kâr payı küçüktür. Bu, ilkedeki "fiyata daha duyarlı olan taraf ucuzlatılır" ifadesinin kaynağıdır. Kuralın ikinci terimi de B'nin duyarlılığına bölünür: B tarafı fiyata az duyarlıysa, ondan alınabilecek ek ücret büyüktür. Seviye 3'te ücretler tam sayı olmak zorunda olmasaydı en kârlı ücretler yaklaşık $(-2{,}1,\ 9{,}5)$ olurdu; bu değerler de kurala uyar.

6.3.8 Fiyatlandırmada sık hatalar

Sık yapılan hata

"Kalabalık taraf ucuzlatılır." Sübvansiyonun ölçütü kalabalık değildir. Sayısal örnekte satıcılar (60 bin) alıcılardan (40 bin) kalabalıktır ama sübvanse edilen alıcılardır. Belirleyici olan, bir katılımcının öteki tarafa kattığı değerdir.

"Sübvansiyon zarardır." Sübvansiyon bir yatırımdır. Tek tarafa bakıldığında platform alıcılardan 200 kaybeder; ama bu alıcılar satıcıların 50 TL ödemesini mümkün kılar. Kâr bütün olarak hesaplanmalıdır.

Kâr payını ağ etkisiyle karıştırmak. Kuralda n_A A'nın kâr payı, e_B n_B ise A'nın B'ye kattığı değerdir. e_A (B'nin A'ya kattığı değer) p_A formülünde yer almaz; o, p_B formülündedir. Alt indis, değeri alan tarafı gösterir.

6.4 Sponsorlu arama

Arama reklamları, platformların en büyük gelir kaynaklarından biridir ve her gün milyarlarca kez yapılan bir açık artırmayla satılır. Birden çok yuva aynı anda satıldığında ödeme kuralının seçimi önem kazanır: GSP ve VCG aynı tekliflerle farklı gelir getirir ve GSP'de dürüst teklif denge değildir. Gerçek sistemlerde kalite puanı ve bütçe kısıtları da işin içine girer.

6.4.1 Sponsorlu arama modeli

Her arama sorgusunda, o anahtar kelimeye teklif veren reklamverenler arasında bir açık artırma yapılır. Satılan nesneler reklam yuvalarıdır. Yuvalar aynı değildir: üstteki yuva daha çok tıklanır. Açık artırma, tek bir nesneyi değil sıralı birkaç nesneyi aynı anda satar.



Şekil 6.8: "Kiralık araba" aramasında üç reklam yuvası. Bin gösterimde yuvalar sırasıyla 300, 200 ve 100 tık alır.

Tanım 6.7

Sponsorlu arama açık artırması (*sponsored search auction*): m yuva vardır ve yukarıdan aşağı tıklanma sayıları $\alpha_1 > \alpha_2 > \ldots > \alpha_m$'dir. Reklamveren i'nin tık başına değeri v_i'dir. Reklamveren tık başına b_i teklif verir; yuva j'ye yerleşirse tık başına p_j öder. Yuva j'deki reklamverenin faydası

$$\alpha_j \cdot (v_i - p_j)$$

olur. Yuva alamayan reklamverenin faydası 0'dır.

Fayda formülü iki çarpandan oluşur. α_j, reklamverenin aldığı tık sayısıdır. $v_i - p_j$, her tıktan kalan kârdır: tıkın değeri eksi tık başına ödeme. Üst yuva daha çok tık getirir ama genellikle daha pahalıdır. Reklamveren için soru, "daha çok tık mı, daha ucuz tık mı?" sorusudur. Bu gerilim, GSP'deki stratejik davranışın kaynağıdır.

Bu kesimdeki bütün sayısal örnekler aynı veriyi kullanır: üç yuva, bin gösterimde 300, 200 ve 100 tık; dört reklamveren, tık başına değerleri A 10, B 8, C 5 ve D 2 TL. Başlangıçta herkesin değerini teklif ettiği varsayılır: $b = v$.

6.4.2 İki ödeme kuralı: GSP ve VCG

İki kural da reklamları teklife göre sıralar; en yüksek teklif en üst yuvaya gider. Fark yalnız ödemededir. Gösterimde $b_{(k)}$, k. en yüksek teklifi belirtir: $b_{(1)} \geq b_{(2)} \geq \ldots$.

Tanım 6.8

Genelleştirilmiş ikinci fiyat (GSP) açık artırması (*Generalized Second Price auction*): reklamlar teklife göre sıralanır. Yuva j'deki reklamveren tık başına bir alttakinin teklifini öder:

$$p_j = b_{(j+1)}$$

GSP, tek nesneli ikinci fiyat açık artırmasının doğal genellemesi gibi görünür. Tek yuva varsa ($m = 1$) gerçekten Vickrey açık artırmasının aynısıdır: kazanan ikinci en yüksek teklifi öder. Ancak birden çok yuvada dürüstlük kaybolur. Bunun nedeni 6.4.5'te gösterilir.

Tanım 6.9

VCG açık artırması: sıralama aynıdır. Reklamveren, varlığıyla diğerlerine verdiği kaybı öder. Yuva j'deki reklamverenin toplam ödemesi

$$\alpha_j \cdot p_j = \Sigma_{k \geq j} (\alpha_k - \alpha_{k+1}) \cdot b_{(k+1)}$$

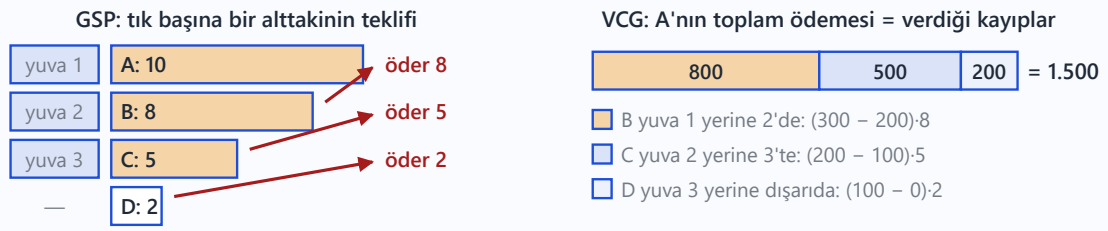
olur; burada $\alpha_{m+1} = 0$ alınır. $\alpha_j \cdot p_j$ toplam ödeme, p_j tık başına ödemedir. VCG'de dürüst teklif baskın stratejidir.

VCG kuralı genel bir ilkenin uygulamasıdır. Bir katılımcının ödemesi, onun varlığının öteki katılımcılara yüklediği maliyettir:

> *ödeme = (o olmasaydı diğerlerinin refahı) − (o varken diğerlerinin refahı)*

Bu maliyete **dışsallık** (*externality*) denir. Dışsallığı ödeyen bir katılımcı, kendi faydası ile toplam refahı aynı anda maksimize eder. Değerini yanlış bildirmekle de hiçbir şey kazanamaz. VCG'nin dürüstlüğü bu hizalamadan gelir.

Formüldeki toplam, bu ilkenin sponsorlu aramadaki biçimidir. Yuva j'deki reklamveren olmasaydı ne olurdu? Üstündekiler etkilenmez, yerlerinde kalırdı; çünkü onlar zaten daha yüksek teklif vermiştir. Altındakilerin her biri bir yuva yukarı çıkardı. Yuva $k+1$'deki reklamveren yuva k'ye çıkar ve $\alpha_k - \alpha_{k+1}$ fazladan tık kazanırdı. Her tıkın ona değeri $b_{(k+1)}$'dir (dürüst teklifte değerine eşit). Toplam, bu kayıpların $k = j$'den sona kadar toplamıdır. Yani VCG'de yuva j'deki reklamveren, altındakilerin birer yuva aşağı kaymasıyla kaybettiği tık değerlerinin toplamını öder.



Şekil 6.9: Solda GSP ödeme merdiveni: her reklamveren tık başına bir alttakinin teklifini öder (8, 5, 2). Sağda A'nın VCG ödemesinin bileşenleri: B'nin yuva 1 yerine 2'de kalmasıyla kaybı $(300 - 200) \cdot 8 = 800$, C'nin $(200 - 100) \cdot 5 = 500$, D'nin $(100 - 0) \cdot 2 = 200$; toplam 1.500.

Şekil 6.9 iki kuralı yan yana gösterir. Soldaki merdivende her ok bir alttaki tekliften üstteki yuvanın ödemesine gider. Sağdaki çubuk, A'nın toplam VCG ödemesinin üç parçadan oluştuğunu gösterir. Her parça, A'nın altındaki bir reklamverenin A yüzünden kaybettiği tık değeridir.

6.4.3 Aynı tekliflerle GSP ve VCG ödemeleri

Teklifler değerlere eşitken (teklif = değer) iki kuralın ödemeleri adım adım hesaplanır. Sıralama teklife göredir: A yuva 1'e, B yuva 2'ye, C yuva 3'e gider; D açıkta kalır.

Örnek 6.8

GSP. Her yuva bir alttakinin teklifini tık başına öder.

1. Yuva 1: A, altındaki B'nin teklifini öder: *8* TL/tık × 300 tık = 2400.
2. Yuva 2: B, C'nin teklifini öder: *5 × 200 = 1000*.
3. Yuva 3: C, D'nin teklifini öder: *2 × 100 = 200*. GSP geliri *2400 + 1000 + 200 = 3600*.

VCG. Ödeme = "o olmasaydı" diğerlerinin refahı − "o varken" diğerlerinin refahı. En alttan başlamak hesabı kolaylaştırır.

1. **C (yuva 3).** C olmasaydı D yuva 3'e çıkardı: diğerlerinin (yalnız alttakilerin) refahı *100 · 2 = 200*. C varken D dışarıdadır: 0. C öder *200 − 0 = 200*, tık başına *200/100 = 2*.
2. **B (yuva 2).** B olmasaydı C yuva 2'ye (*200 · 5 = 1000*), D yuva 3'e (*100 · 2 = 200*) çıkardı: toplam 1200. B varken C yuva 3'tedir: *100 · 5 = 500*, D dışarıdadır. B öder *1200 − 500 = 700*, tık başına 3,5.
3. **A (yuva 1).** A olmasaydı B, C, D birer yuva yukarı çıkardı: *300 · 8 + 200 · 5 + 100 · 2 = 2400 + 1000 + 200 = 3600*. A varken *200 · 8 + 100 · 5 = 1600 + 500 = 2100*. A öder *3600 − 2100 = 1500*, tık başına 5.
4. **Formülle denetim.** *(300 − 200) · 8 + (200 − 100) · 5 + (100 − 0) · 2 = 800 + 500 + 200 = 1500* ✓. VCG geliri *1500 + 700 + 200 = 2400*.

Üstteki reklamverenlerin refahı hiçbir hesapta değişmez. Yalnız alttakiler sayılır, çünkü bir reklamverenin çıkması yalnız onun altındakileri etkiler.

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400	1500	5
2	200	B (b = 8)	5	1000	700	3,5
3	100	C (b = 5)	2	200	200	2
—	0	D (b = 2)	—	0	0	—
		Gelir		3600	2400	

Tablo 6.8: Aynı tekliflerle GSP ve VCG ödemeleri (bin gösterimde, TL).

Tablo 6.8 iki gözlem verir. Birincisi: aynı tekliflerde GSP her yuvada VCG'den fazla ya da eşit alır: *8 ≥ 5*, *5 ≥ 3,5*, *2 ≥ 2* TL/tık. Bu her zaman böyledir. VCG'nin tık başına ödemesi, alttaki tekliflerin ağırlıklı bir ortalamasıdır ve bu ortalama bir alttaki tekliften büyük olamaz. En alt yuvada iki ödeme her zaman

eşittir. İkincisi: gelir GSP'de 3600, VCG'de 2400'dür. Ama bu karşılaştırma adil değildir. GSP'de dürüst teklif reklamverenin çıkarına değildir (6.4.5). Gelirler, iki kuralın kendi dengelerinde karşılaştırılmalıdır (6.4.7).

6.4.4 Kod izleme: gsp_vcg()

İki ödeme kuralı tek bir kısa fonksiyonla hesaplanır. Fonksiyon teklif listesini (`bids`) ve yuvaların tıklanma sayılarını (`ctr`) alır; her yuvanın toplam GSP ve VCG ödemelerini döndürür.

```
def gsp_vcg(bids, ctr):
  m = len(ctr)
  b = sorted(bids, reverse=True) + [0]*m
  a = ctr + [0]     # sanal yuva: 0 tık
  gsp, vcg = [0]*m, [0]*(m + 1)
  for j in range(m):
    gsp[j] = a[j] * b[j+1]   # alttaki
  for j in range(m-1, -1, -1):  # alttan
    ek = (a[j] - a[j+1]) * b[j+1]
    vcg[j] = vcg[j+1] + ek
  return gsp, vcg[:m]
```

Kodun mantığı satır satır şöyledir:

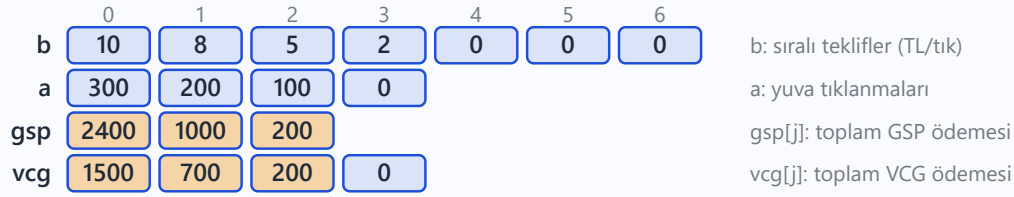
- `m = len(ctr)` : yuva sayısı. Örnekte 3.
- `b = sorted(bids, reverse=True) + [0]*m` : teklifler büyükten küçüğe sıralanır ve sona `m` tane sıfır eklenir. Böylece "alttaki teklif" her yuva için var olur; reklamveren sayısı yuva sayısından az olsa bile `b[j+1]` dizinin dışına taşmaz.
- `a = ctr + [0]` : 0 tıklı sanal bir *(m+1)*. yuva eklenir. Bu, formüldeki $\alpha_{m+1} = 0$ varsayımıdır.
- `gsp, vcg = [0]*m, [0]*(m + 1)` : sonuç dizileri sıfırlanır. `vcg` 'nin fazladan son elemanı (0) yinelemenin başlangıç değeridir.
- İlk döngü: `gsp[j] = a[j] * b[j+1]` . Yuva *j*'nin tık sayısı çarpı bir alttaki teklif.
- İkinci döngü alttan yukarı gider: `ek = (a[j] - a[j+1]) * b[j+1]` , bir alttaki reklamverenin bu yuva yerine bir aşağıda kaldığı için kaybettiği tık değeridir. `vcg[j] = vcg[j+1] + ek` : yuva *j*'nin ödemesi, bir alttaki yuvanın ödemesi artı bu ek kayıptır. Alttakilere verilen kayıplar böylece yukarı taşınır.
- `return gsp, vcg[:m]` : sanal yuvanın sıfırı atılarak sonuçlar döndürülür.

İkinci döngü, VCG formülündeki toplamı her yuva için baştan hesaplamaz. Yuva *j*'nin toplamı, yuva *j + 1*'in toplamına tek bir terim eklenerek bulunur. Bu yüzden iki döngü de *O(m)* adımda biter; sıralama ile birlikte toplam iş *O(n log n)*'dir (*n* reklamveren sayısı).

Fonksiyon `gsp_vcg([10, 8, 5, 2], [300, 200, 100])` olarak çağrılınca izleme şöyledir:

Döngü	j	Hesap	Sonuç
GSP	0	a[0] · b[1] = 300 · 8	gsp[0] = 2400
GSP	1	a[1] · b[2] = 200 · 5	gsp[1] = 1000
GSP	2	a[2] · b[3] = 100 · 2	gsp[2] = 200
VCG	2	ek = (100 − 0) · 2 = 200	vcg[2] = 0 + 200 = 200
VCG	1	ek = (200 − 100) · 5 = 500	vcg[1] = 200 + 500 = 700
VCG	0	ek = (300 − 200) · 8 = 800	vcg[0] = 700 + 800 = 1500

Tablo 6.9: `gsp_vcg([10, 8, 5, 2], [300, 200, 100])` izlemesi. b = [10, 8, 5, 2, 0, 0, 0], a = [300, 200, 100, 0].



Şekil 6.10: `gsp_vcg` çalıştıktan sonra dizilerin son hâli. `b` : sıralı teklifler ve eklenen sıfırlar; `a` : tıklanmalar ve sanal 0; `gsp` ve `vcg` : her yuvanın toplam ödemesi.

Sonuç Şekil 6.10'da görülür: GSP [2400, 1000, 200] (toplam 3600), VCG [1500, 700, 200] (toplam 2400). Bu, Tablo 6.8'in sütunlarıyla aynıdır. Tek yuvalı bir çağrıda (`gsp_vcg([10, 8, 5], [100])`) iki dizi de [800] olur: tek yuvada GSP ile VCG çakışır.

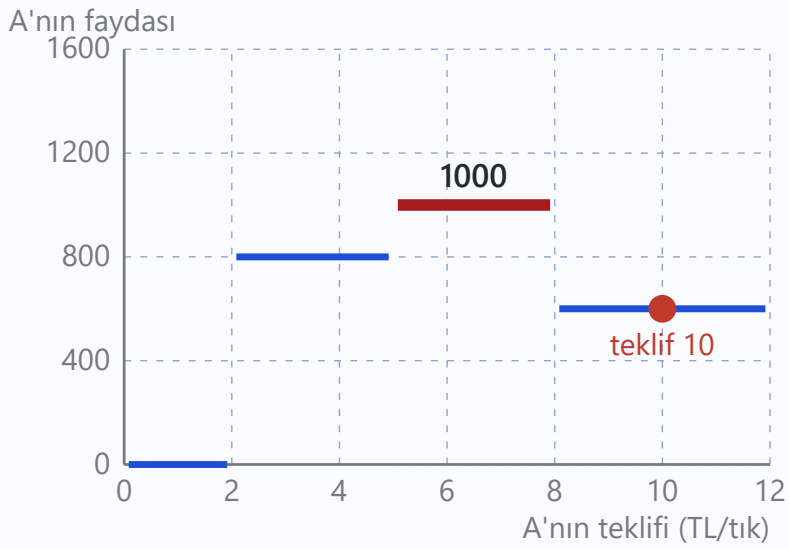
6.4.5 GSP'de dürüst teklif denge değildir

Bir teklif profilinin Nash dengesi olması için hiçbir reklamverenin, ötekiler sabitken teklifini değiştirerek faydasını artıramaması gerekir. Dürüst profil bu sınavı geçemez. Bunu görmek için A'nın seçeneklerine bakmak yeter.

Örnek 6.9

1. **Dürüst profilde faydalar.** Herkes değerini teklif ediyor. A yuva 1'de, B'nin teklifini (8) öder: *300 · (10 − 8) = 600*. B yuva 2'de 5 öder: *200 · (8 − 5) = 600*. C yuva 3'te 2 öder: *100 · (5 − 2) = 300*. D dışarıda: 0. Platform geliri 3600.
2. **A'nın faydası teklifine göre.** Ötekiler (8, 5, 2) sabitken A'nın teklifi yalnız hangi yuvaya düşeceğini belirler. 2'nin altında A dışarıda kalır: 0. 2 ile 5 arasında yuva 3'e düşer ve D'nin 2'sini öder: *100 · (10 − 2) = 800*. 5 ile 8 arasında yuva 2'ye düşer ve C'nin 5'ini öder: *200 · (10 − 5) = 1000*. 8'in üstünde yuva 1'de kalır ve 8 öder: 600.
3. **En iyi tepki.** En büyük fayda 1000'dir ve 5 ile 8 arasındaki bir teklifle elde edilir. Örneğin A 6 teklif ederse yuva 2'ye iner, C'nin 5'ini öder: *200 · (10 − 5) = 1000 > 600*.

4. **Sonuç.** Dürüst teklif A için en iyi tepki değildir. Dürüst profil Nash dengesi değildir.



Şekil 6.11: Ötekiler dürüstken (8, 5, 2) A'nın faydası teklifine göre basamaklıdır: 2–5 arası yuva 3 (800), 5–8 arası yuva 2 (1000), 8 üstü yuva 1 (600). Kırmızı nokta dürüst teklif 10'dur; en yüksek basamak (kırmızı) bunun altındadır.

Şekil 6.11 A'nın fayda fonksiyonunu çizer. Fonksiyon basamaklıdır, çünkü A'nın ödemesi kendi teklifine değil, altındaki teklife bağlıdır. Teklif bir basamak içinde değiştiğinde yuva da ödeme de değişmez. Basamaklar yalnız öteki tekliflerin (2, 5, 8) üstünden geçerken değişir. Dürüst teklif en sağdaki basamağa düşer; ama bu basamak en yüksek değildir.

Neden böyle olur? Yuva 1, yuva 2'den yalnız 100 fazla tık getirir (300'e karşı 200). Bu ek tıklar için A'nın tık başına ödemesi 5'ten 8'e çıkar. Üstelik bu yüksek fiyat yalnız ek 100 tık için değil, *bütün* 300 tık için ödenir. Yukarı çıkmanın marjinal maliyeti marjinal getirisini aşar. A daha az tık alır ama tık başına daha az öder: *1000 > 600*.

Bu gözlem, tek nesneli Vickrey açık artırmasının neden dürüst olduğunu da açıklar. Tek nesnede kaybedenin faydası 0'dır; “daha az ama ucuz” bir seçenek yoktur. Birden çok yuvada ise her alt yuva böyle bir seçenektir. GSP'de bir reklamveren teklifini düşürerek ödeme kuralının kendisini (hangi teklifin altında kalacağını) değiştirebilir. VCG'de ise ödeme, reklamveren hangi yuvada olursa olsun ötekilere verdiği kayba eşittir. Teklifi düşürmek yalnız yuvayı değiştirir ve bu değişiklik toplam refahı, dolayısıyla A'nın faydasını, hiçbir zaman artırmaz.

Bir denge örneği. Dürüst profil denge değilse GSP'nin dengesi nasıl görünür? Bir aday şudur: A 10, B 5, C 3,5, D 2 teklif eder. Sıralama değişmez (A, B, C) ama ödemeler düşer.

Rek.	v	teklif	yuva	öder/tık	fayda
A	10	10	1	5	1500
B	8	5	2	3,5	900

Rek.	v	teklif	yuva	öder/tık	fayda
C	5	3,5	3	2	300
D	2	2	—	—	0
Platform geliri					2400

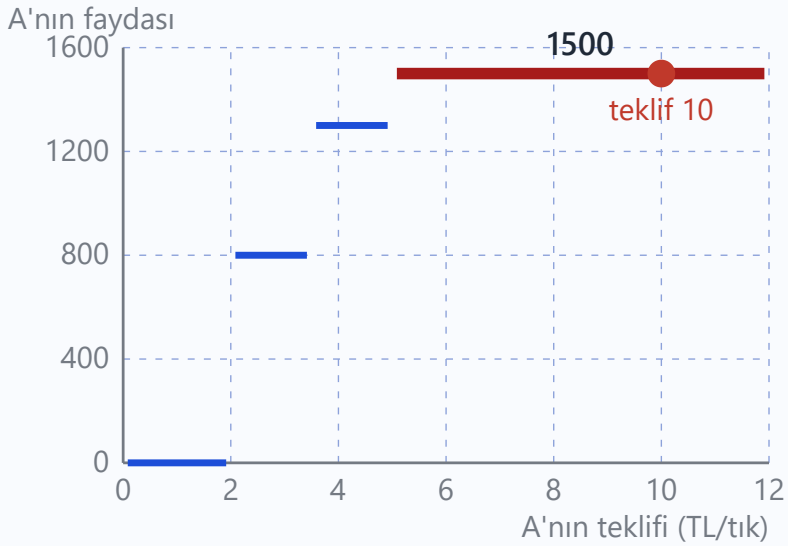
Tablo 6.10: GSP'de bir Nash dengesi. Teklifler A 10, B 5, C 3,5, D 2.

Örnek 6.10

Tablo 6.10'daki profilin bir Nash dengesi olduğu, her reklamverenin bütün sapmaları denenerek gösterilir.

1. **A (fayda 1500).** 5'in üstündeki her teklif yuva 1'i ve 5 ödemeyi verir: 1500. 3,5 ile 5 arası yuva 2: $200 \cdot (10 - 3{,}5) = 1300$. 2 ile 3,5 arası yuva 3: $100 \cdot (10 - 2) = 800$. 2'nin altı: 0. Hiçbiri 1500'ü geçmez.
2. **B (fayda 900).** Yuva 1 için 10'dan fazla teklif edip 10 ödemesi gerekir: $300 \cdot (8 - 10) = -600$. Yuva 3 için 2 ile 3,5 arası teklif eder ve 2 öder: $100 \cdot (8 - 2) = 600$. Dışarıda 0. Sapmaz.
3. **C (fayda 300).** Yuva 2 için 5 ile 10 arası teklif edip B'nin 5'ini öder: $200 \cdot (5 - 5) = 0$. Yuva 1 için 10'dan çok ödemesi gerekir: negatif. Dışarıda 0. Sapmaz.
4. **D (fayda 0).** Yuva 3 için 3,5'ten fazla teklif edip 3,5 ödemesi gerekir: $100 \cdot (2 - 3{,}5) < 0$. Sapmaz.

Hiçbir sapma kazandırmaz; profil bir Nash dengesidir. Sunumdaki betik aynı denetimi dört reklamverenin hepsi için 0,25 adımlı teklif ızgarasında yapar ve kârlı bir sapma bulmaz.



Şekil 6.12: Dengede (B 5, C 3,5, D 2) A'nın faydası teklifine göre: 2–3,5 arası 800, 3,5–5 arası 1300, 5 üstü 1500. A'nın teklifi 10, en yüksek basamaktadır.

Şekil 6.12 dengedeki A'nın durumunu gösterir. Basamaklar değişmiştir, çünkü ötekilerin teklifleri değişmiştir. Artık en yüksek basamak en sağdadır ve A'nın teklifi oradadır. Bu dengede tık başına ödemeler 5 / 3,5 / 2'dir. Bunlar Tablo 6.8'deki VCG ödemelerinin aynısıdır ve gelir 2400'dür. Bu rastlantı değildir; nedeni 6.4.7'de açıklanır.

6.4.6 Kod izleme: karli_sapma()

Bir profilin denge olup olmadığını denetlemek, her reklamveren için kârlı bir sapma aramaktır. Aşağıdaki fonksiyon bir reklamveren için bu aramayı yapar.

```
def karli_sapma(i, b, v, ctr, adaylar):
  eski = gsp_fayda(i, b, v, ctr)
  for x in adaylar:
    b2 = b[:]; b2[i] = x   # yalnız i değişir
    u = gsp_fayda(i, b2, v, ctr)
    if u > eski:
      return x, u   # kârlı sapma
  return None     # sapma yok
```

Fonksiyon yardımcı bir `gsp_fayda(i, b, v, ctr)` fonksiyonu kullanır. Bu yardımcı teklifleri sıralar; *i*, *j*. yuvadaysa `ctr[j] · (v[i] − bir alttaki teklif)` döndürür, yuva dışındaysa 0 döndürür. Tekliflerin birbirinden farklı olduğu varsayılır; eşit tekliflerde sıralamanın nasıl yapılacağı ayrı bir kural gerektirir.

`karli_sapma` şöyle çalışır:

- `eski` : *i*'nin mevcut tekliflerle faydası.
- Her aday teklif `x` için teklif listesi kopyalanır (`b2 = b[:]`) ve yalnız *i*'nin teklifi değiştirilir. Kopyalama önemlidir: asıl liste değişirse sonraki adaylar yanlış profille denenir.
- `u` , yeni profilde *i*'nin faydasıdır. `u > eski` ise kârlı sapma bulunmuştur; `(x, u)` döner.
- Hiçbir aday kârlı değilse `None` döner.

Fonksiyon ilk kârlı sapmada durur; en iyi sapmayı değil, *bir* kârlı sapmayı bulur. Denge denetimi için bu yeterlidir; çünkü tek bir kârlı sapma profilin denge olmadığını göstermeye yeter. Buna karşılık `None` sonucu yalnız denenen adaylar için geçerlidir. Bir profilin denge olduğunu söylemek için adayların bütün anlamlı teklifleri kapsaması gerekir. Basamaklı fayda fonksiyonunda her basamaktan bir aday yeterlidir: örnekteki adaylar (9, 6, 4, 3, 1) dürüst profilin basamaklarının ve dengedeki basamakların her birine en az bir teklif düşürür.

Profil b	eski	x	A'nın yuvası	u	Sonuç
[10, 8, 5, 2]	300 · (10 − 8) = 600	9	1	600	600 ≤ 600, devam
		6	2	1000	1000 > 600 → `return (6, 1000)`

Profil b	eski	x	A'nın yuvası	u	Sonuç
[10, 5, 3,5, 2]	300 · (10 − 5) = 1500	9	1	1500	devam
		6	1	1500	devam
		4	2	1300	devam
		3	3	800	devam
		1	dışarıda	0	`return None`

Tablo 6.11: `karli_sapma(0, b, v, ctr, [9, 6, 4, 3, 1])` izlemesi, A için iki profilde (v = [10, 8, 5, 2], ctr = [300, 200, 100]).



Şekil 6.13: `karli_sapma` izlemesinin iki anı. Üstte: dürüst profilde A'nın 6 teklifi onu yuva 2'ye indirir ve *u = 1.000 > 600*; fonksiyon `(6, 1.000)` döndürür. Altta: denge profilinde hiçbir aday 1.500'ü geçmez; fonksiyon `None` döndürür.

Tablo 6.11 ve Şekil 6.13 iki çağrının farkını gösterir. Dürüst profilde ilk aday (9) faydayı değiştirmez, çünkü A yine yuva 1'dedir ve yine 8 öder. İkinci aday (6) A'yı yuva 2'ye indirir ve fonksiyon hemen döner. Denge profilinde beş aday da denenir ve hiçbiri 1500'ü aşmaz. Aynı test dört reklamverenin hepsi için yapılırsa (B, C ve D için de `None`) profil bir Nash dengesidir.

Bu fonksiyonun maliyeti küçüktür: her aday için bir sıralama, yani $O(n \log n)$. K aday ve n reklamveren için tam denetim $O(K \cdot n^2 \log n)$ işlem gerektirir. Gerçek bir arama motorunda reklamverenin elinde de benzer bir hesap vardır: teklif yönetim araçları, rakiplerin teklifleri veriyken en kârlı teklifi bu tür bir aramayla bulur.

6.4.7 Verili teklifler ve denge teklifleri

GSP sonuçları her zaman verili tekliflerle hesaplanır. 3600, herkesin dürüst teklif ettiği profildeki gelirdir. Ama dürüst teklif GSP'de baskın strateji değildir; bu yüzden 3600 gerçek bir tahmin değildir. Üç profili yan yana koymak durumu netleştirir.

Profil	Teklifler (A, B, C, D)	A'nın yuvası ve faydası	Platform geliri
Dürüst	10, 8, 5, 2	yuva 1: 300 · (10 − 8) = 600	3600
A daha düşük teklif verir	6, 8, 5, 2	yuva 2: 200 · (10 − 5) = 1000	3000
Yerel kıskançsız denge	10, 5, 3,5, 2	yuva 1: 300 · (10 − 5) = 1500	2400 = VCG

Tablo 6.12: Aynı değerlerle (A 10, B 8, C 5, D 2) üç teklif profili.

İkinci satırda B yuva 1'e çıkar ve A'nın teklifini (6) öder: *300 · 6 = 1800*. A yuva 2'de 5 öder: 1000. C yuva 3'te 2 öder: 200. Gelir 3000'dir. A daha alt yuvada daha az tık alır ama tık başına daha az öder: *1000 > 600*.

Üçüncü satır, Tablo 6.10'daki dengedir. GSP'nin genellikle birçok Nash dengesi vardır. Edelman, Ostrovsky ve Schwarz (2007) bu dengeler arasından makul bir alt sınıfı seçmiştir.

Tanım 6.10

Bir GSP teklif profili Nash dengesiyse ve hiçbir reklamveren bir üstündeki yuvayı o yuvanın şu anki tık başına fiyatıyla almayı kendi yerine tercih etmiyorsa, bu profile **yerel kıskançsız denge** (*locally envy-free equilibrium*) denir. Yuva *j*'deki reklamveren için koşul:

$$\alpha_j \cdot (v - p_j) \geq \alpha_{j-1} \cdot (v - p_{j-1})$$

Koşul her yuva için (ve yuvası olmayan ilk reklamveren için en alt yuvaya karşı) sağlanmalıdır. Hiçbir reklamverenin hiçbir yuvayı kıskanmadığı profile **kıskançsız** (*envy-free*) denir.

Koşulun anlamı şudur. Nash dengesi yalnız tek başına yapılabilen sapmalara bakar. B'nin yuva 1'e çıkmak için A'nın teklifini (10) geçmesi ve 10 ödemesi gerekir; bu kârsızdır. Ama B'nin gözünde yuva 1'in "şu anki fiyatı" 5'tir (A'nın ödediği). Kıskançsızlık, "A'nın ödediği fiyatla A'nın yerinde olmak isterdim" düşüncesinin bile kimsede olmamasını ister. Bu daha güçlü bir koşuldur ve tekrarlanan teklif ortamlarında ayakta kalan dengeleri tarif eder: biri bir üstündekini kıskanıyorsa kendi teklifini yükselterek üstündekinin ödemesini artırabilir; bu baskı dengeyi bozar.

Örnek 6.11

1. **B (yuva 2).** Yerinde *200 · (8 − 3,5) = 900* alır. Yuva 1'i şu anki fiyatı 5'ten alsa *300 · (8 − 5) = 900* alırdı. Kazancı yoktur; kıskanmaz.
2. **C (yuva 3).** Yerinde *100 · (5 − 2) = 300*. Yuva 2'yi 3,5'ten alsa *200 · (5 − 3,5) = 300*. Kıskanmaz.
3. **D (dışarıda).** Yerinde 0. Yuva 3'ü 2'den alsa *100 · (2 − 2) = 0*. Kıskanmaz.

Her reklamveren bir üstündeki yuvaya karşı tam olarak kayıtsızdır. Kıskançsızlık koşulu eşitlikle sağlanır. Bu, en düşük gelirli kıskançsız dengenin ayırt edici özelliğidir: fiyatlar bir kuruş daha düşük olsaydı biri üstündekini kıskanırdı.

Teorem 6.2 (Edelman–Ostrovsky–Schwarz, Varian)

GSP açık artırmasının her zaman yerel kıskançsız bir dengesi vardır. Her yerel kıskançsız dengede GSP geliri en az VCG geliri kadardır. Bu dengelerden en düşük gelirli olanında yuvalar VCG'deki gibi dağılır ve her reklamveren VCG'deki tık başına fiyatı öder.

Teoremin ispatı bu notun kapsamı dışındadır (Edelman, Ostrovsky ve Schwarz, 2007; Varian, 2007). Ama en düşük dengenin nasıl kurulduğu örnekten okunabilir: yuva $j + 1$'deki reklamverenin teklifi, yuva j'nin VCG tık başına fiyatına eşitlenir. B'nin teklifi yuva 1'in VCG fiyatı 5'tir; C'ninki yuva 2'nin VCG fiyatı 3,5'tir; D'ninki yuva 3'ün VCG fiyatı 2'dir. GSP'de herkes bir alttakinin teklifini ödediği için ödemeler VCG ödemelerine eşit olur.

Sonuç olarak bu dengelerde GSP geliri en az 2400'dür ve en düşüğü VCG ile aynıdır. Dürüst tekliflerle bulunan 3600 ise dengedeki gelir değildir. GSP'nin "aynı tekliflerle daha çok gelir" üstünlüğü, teklifler değiştiğinde büyük ölçüde kaybolur.

6.4.8 GSP ile VCG'nin karşılaştırması

	GSP	VCG
Dürüst teklif baskın mı?	Hayır (A 10 yerine 6 verir)	Evet
Aynı tekliflerle gelir	3600 (her yuvada ≥ VCG)	2400
Dengede gelir	Yerel kıskançsız dengelerde ≥ 2400; en düşüğü VCG ile aynı	2400 (dürüst teklifle)
Açıklaması	Kolay: "bir alttakinin teklifi"	Dışsallık toplamı, daha zor
Uygulama	Arama reklamlarında uzun süre yaygın (kalite puanıyla)	Facebook reklam açık artırmasında kullanıldı (Varian ve Harris, 2014)

Tablo 6.13: GSP ve VCG'nin karşılaştırması (örnekteki sayılarla).

Tablo 6.13 iki kuralın artı ve eksilerini özetler. VCG kuramsal olarak daha temizdir: reklamverenin stratejik düşünmesine gerek yoktur, değerini bildirmesi yeter. GSP ise anlatması kolaydır ve tarihsel olarak önce yerleşmiştir. Bir platform kuralını değiştirdiğinde binlerce reklamverenin alışkanlıkları ve teklif yazılımları da değişmek zorunda kalır; bu yüzden kural değişikliği pahalıdır. Pratikte sıralamada teklifle birlikte reklamın tıklanma kalitesi (kalite puanı) de kullanılır (6.4.10).

Bu karşılaştırmanın genel dersi şudur: "ikinci fiyat" adı dürüstlüğü garanti etmez. Tek nesnede geçerli olan bir özellik, çok yuvada bozulabilir. Bir mekanizmanın dürüst olup olmadığı adından değil, en iyi tepki hesabından anlaşılır.

6.4.9 Bütçeler ve çevrim içi tahsis

Gerçek reklamverenlerin günlük bütçesi vardır. Ayrıca aramalar tek tek gelir ve platform her birini, gelecekte hangi aramaların geleceğini bilmeden, hemen bir reklamverene vermek zorundadır. Bu tür karara **çevrim içi** (*online*) karar denir: girdi parça parça gelir ve her parça geldiği anda geri alınamaz biçimde işlenir.

Örnek 6.12

A ve B'nin bütçesi 100'er TL'dir. "Koşu ayakkabısı" aramaları için ikisi de 1 TL, "spor çanta" aramaları için yalnız A 1 TL teklif verir. Önce 100 ayakkabı, sonra 100 çanta araması gelir. Platform her aramayı bütçesi yeten bir reklamverene verir ve teklif kadar gelir elde eder.

1. **Açgözlü kural** (en yüksek teklif, eşitlikte A). Ayakkabı aramalarında iki teklif eşittir; 100'ü de A'ya gider. A'nın bütçesi biter. Çanta aramalarında yalnız A teklif vermişti ama parası kalmamıştır: 0. Gelir 100.
2. **Dengeleme kuralı** (*Balance*): arama, kalan bütçesi büyük olana verilir. Ayakkabı aramaları sırayla A ve B arasında bölünür: A 50, B 50. Çanta aramalarında A'nın kalan 50 TL'si kullanılır: 50. Gelir $100 + 50 = 150$.
3. **Geleceği bilen optimum.** Çantaların geleceği bilinseydi ayakkabıların 100'ü B'ye, çantaların 100'ü A'ya verilirdi. Gelir 200.

Kural	Ayakkabı aramaları	Çanta aramaları	Gelir
Açgözlü (en yüksek teklif, eşitlikte A)	100'ü A'ya; A'nın bütçesi biter	A'nın parası yok: 0	100
Dengeleme: kalan bütçesi büyük olana	A 50, B 50	A'nın kalan 50'si	150
Geleceği bilen optimum	100'ü B'ye	100'ü A'ya	200

Tablo 6.14: Bütçeli çevrim içi tahsiste üç kural.

Tablo 6.14 açgözlü kuralın zaafını gösterir. Her aramada en yüksek teklifi seçmek tek tek kararlarda doğrudur. Ama bütçe varken açgözlü kural esnek reklamvereni (birden çok aramaya teklif veren A'yı) erkenden tüketir. Sonra yalnız A'nın değer verdiği aramalar gelince kimse onları alamaz.

Çevrim içi algoritmaların başarısı **rekabet oranı** (*competitive ratio*) ile ölçülür: en kötü arama sırasında algoritmanın gelirinin geleceği bilen optimuma oranı. Açgözlü kural en kötü durumda optimumun yalnız 1/2'sini garanti eder; bu örnek tam o en kötü durumdur (100/200). Kalan bütçeyi gözeten kurallar, arama sırası ne olursa olsun optimumun en az $1 - 1/e \approx \%63$'ünü elde eder (Mehta, Saberi, Vazirani ve Vazirani, 2007). Örnekte dengeleme kuralı %75 almıştır; garanti %63'tür ama bazı sıralarda daha iyisi elde edilir. Burada $e \approx 2{,}718$ doğal logaritmanın tabanıdır.

Sık yapılan hata

"Her aramada en yüksek teklifi seçmek en iyisidir." Bütçe yoksa doğrudur. Bütçe varsa bir aramayı birine vermek, o reklamverenin gelecekteki aramalar için kullanabileceği parayı tüketir. Çevrim içi tahsiste her karar geleceği de etkiler; bu yüzden kalan bütçeyi hesaba katan kurallar daha iyi garanti verir.

6.4.10 Kalite puanıyla GSP

Buraya kadar her reklamın bir yuvada aynı sayıda tık aldığı varsayıldı. Gerçekte bazı reklamlar daha ilgi çekicidir. Bir platformun geliri tıklardan geldiği için, yüksek teklif veren ama kimsenin tıklamadığı bir reklamı en üste koymak platform için kötüdür. Bu sorunu kalite puanı çözer.

Tanım 6.11

Kalite puanlı GSP: reklam i'nin **kalitesi** q_i, tıklanma eğilimidir. Yuva j'de i'nin tık sayısı $\alpha_j \cdot q_i$'dir. Platform reklamları $b_i \cdot q_i$ **puanına** göre sıralar. Yuva j'deki reklamveren, yerini korumak için gereken en düşük teklifi tık başına öder:

$$p_j = puan_{j+1} / q_i = b_{(j+1)}\, q_{(j+1)} / q_i$$

Puanın anlamı basittir: $b \cdot q$, bir gösterimin platforma getireceği beklenen gelirin ölçüsüdür. Teklif tık başına ödemedir, kalite tıklanma olasılığıyla orantılıdır; çarpımları gösterim başına beklenen ödemedir. Ödeme formülü de aynı mantıktan gelir. Reklamveren i yerini korumak için puanının bir alttakinin puanından az olmamasını ister: $b_i\, q_i \geq puan_{j+1}$. Bu, $b_i \geq puan_{j+1}/q_i$'ye denktir. GSP'nin "yerini korumak için gereken en düşük teklifi öde" ilkesi bu eşitsizliğin sınırını ödeme yapar. Tüm kaliteler 1 ise formül sıradan GSP'ye döner.

Örnek 6.13

Aynı dört reklamveren ve aynı teklifler (A 10, B 8, C 5, D 2). Kaliteler A 0,4, B 0,8, C 0,6, D 1'dir. Yuva tıklanmaları $\alpha = 300, 200, 100$.

1. **Puanlar.** A: $10 \cdot 0{,}4 = 4$; B: $8 \cdot 0{,}8 = 6{,}4$; C: $5 \cdot 0{,}6 = 3$; D: $2 \cdot 1 = 2$.
2. **Sıralama.** B (6,4) > A (4) > C (3) > D (2). En yüksek teklifi veren A artık birinci değildir.
3. **Ödemeler.** B yuva 1'de, alttakinin puanı 4: $4/0{,}8 = 5$ TL/tık. A yuva 2'de, alttakinin puanı 3: $3/0{,}4 = 7{,}5$ TL/tık. C yuva 3'te, alttakinin puanı 2: $2/0{,}6 \approx 3{,}33$ TL/tık.
4. **Teklifi aşmama.** Her ödeme teklifin altındadır: $5 \leq 8$, $7{,}5 \leq 10$, $3{,}33 \leq 5$. Bu her zaman böyledir, çünkü reklamverenin puanı alttakinin puanından büyüktür.
5. **Tıklar.** B: $300 \cdot 0{,}8 = 240$; A: $200 \cdot 0{,}4 = 80$; C: $100 \cdot 0{,}6 = 60$. Toplam 380.

6. **Gelir.** *240 · 5 + 80 · 7,5 + 60 · 3,33 = 1200 + 600 + 200 = 2000*.
7. **Karşılaştırma: yalnız teklife göre sıralama.** Sıra A, B, C olur. Tıklar *300 · 0,4 = 120*, *200 · 0,8 = 160*, *100 · 0,6 = 60*; toplam 340. Ödemeler sıradan GSP ile 8, 5, 2: gelir *120 · 8 + 160 · 5 + 60 · 2 = 960 + 800 + 120 = 1880*.



Şekil 6.14: Kalite puanıyla GSP. Çubuklar $b \cdot q$ puanlarıdır; sıra B (6,4), A (4), C (3), D (2). Sağda tık başına ödemeler: B *4/0,8 = 5*, A *3/0,4 = 7,5*, C *2/0,6 = 3,33*; hepsi teklifin altında.

Sıralama	yuva 1–3	toplam tık	gelir (TL)
Yalnız teklife göre	A, B, C	340	1.880
Kalite puanıyla (b·q)	B, A, C	380	2.000

Tablo 6.15: Kalite puanının tıklara ve gelire etkisi (bin gösterimde).

Tablo 6.15 kalite puanının iki etkisini gösterir. Bu örnekte kalite puanı hem tıkları (340 → 380) hem geliri (1.880 → 2.000) artırmıştır. Kullanıcı ilgili reklamı görür, platform daha çok tık satar. Ayrıca kaliteli reklam aynı yuvaya daha ucuza çıkar: B yuva 1'de tık başına 5 öder, oysa yalnız teklife göre sıralamada yuva 1'deki A 8 ödüyordu. Kalite ödüllendirilir; böylece reklamveren reklamını iyileştirmeye teşvik edilir.

Kalite puanı, oyunun açılış sorusunun ikincisini de cevaplar: en yüksek teklifi veren her zaman en üst yuvayı almaz. A 10 teklif etmesine rağmen ikinci yuvadadır.

Sık yapılan hata

Kalite puanlı GSP'de ödemeyi $b_{(j+1)}$ (alttakinin teklifi) olarak almak. Doğru ödeme alttakinin *puanının* reklamverenin kendi kalitesine bölümüdür. Örnekte B, alttaki A'nın teklifini (10) değil, *4/0,8 = 5* öder. Alttakinin teklifi ödenseydi B teklifinden (8) fazla öderdi.

6.4.11 Oyuna dönüş

Açılıştaki ilk soru şuydu: bir tık reklamverene 8 TL kazandırıyorsa 8 TL teklif etmek en iyisi midir? Kuramın cevabı hayırdır. Oyunun "Search Ads" senaryosunun tıklanma oranlarıyla (%12, %7, %3,5) bin gösterimde yuvalar 120, 70 ve 35 tık alır. Reklamverenin tık değeri 8, rakiplerin teklifleri 6, 4 ve 2, bütün kaliteler 1 olsun. Oyunun Seviye 2'sinde kural GSP'dir.

Teklif	Yuva	Ödeme/tık	Fayda
8 (dürüst)	1	6	120 × 2 = 240
5	2	4	70 × 4 = 280
3	3	2	35 × 6 = 210

Tablo 6.16: Oyunda değer 8 iken üç teklifin sonucu (rakipler 6, 4, 2; GSP).

Tablo 6.16 A'nın örneğindeki mantığı tekrarlar. Dürüst teklif yuva 1'i getirir ama tık başına 6 ödenir; her tıktan 2 kalır. 5 teklifi yuva 2'yi getirir; tık başına 4 ödenir ve her tıktan 4 kalır. Tık sayısı 120'den 70'e düşse de toplam fayda 240'tan 280'e çıkar. Alt yuvaya inmek daha kârlıdır. Oyunun "Find best response" düğmesi tam olarak bu sapmayı arar; yani `karli_sapma` fonksiyonunun yaptığını yapar.

Gerçek dünyada da aynı davranış gözlenmiştir. Sponsorlu aramanın ilk dönemlerinde kullanılan birinci fiyatlı açık artırmada (oyunun Seviye 1 kuralı: herkes kendi teklifini öder) teklifler sürekli yükselip çöktü; reklamverenler birbirinin teklifinin biraz üstüne çıkıyor, sonra kârsız hâle gelince düşüyordu (Edelman ve Ostrovsky, 2007). Google'ın kullandığı GSP bu dalgalanmayı azalttı ama GSP'de de dürüst teklif baskın değildir (Edelman, Ostrovsky ve Schwarz, 2007). Platformlar ayrıca rezerv fiyat (en düşük kabul edilen teklif) koyarak geliri artırabilir; Ostrovsky ve Schwarz (2011) bunu bir saha deneyiyle göstermiştir.

Bölümün reklam kısmının özeti şudur: "ikinci fiyat" adı tek yuvada dürüstlüğü getirir; çok yuvada getirmez. Dürüstlüğü garanti eden kural VCG'dir.

6.4.12 Reklam açık artırmalarında sık hatalar

Sık yapılan hata

GSP'yi Vickrey sanmak. "GSP ikinci fiyat açık artırması, o hâlde dürüst teklif baskındır." Yalnız tek yuvada doğrudur. Çok yuvada dürüst teklif bazen en iyi tepki olabilir, ama her zaman değil.

VCG ödemesinde üsttekileri saymak. Bir reklamverenin çıkması yalnız altındakileri etkiler. A'nın VCG ödemesinde B, C ve D'nin kayıpları sayılır; C'nin ödemesinde yalnız D sayılır.

Toplam ve tık başına ödemeyi karıştırmak. VCG formülü $\alpha_j \cdot p_j$'yi, yani toplam ödemeyi verir. Tık başına ödeme için toplam, yuvanın tık sayısına bölünür: B için *700/200 = 3,5*.

Gelirleri yanlış profille karşılaştırmak. “GSP, VCG'den %50 fazla gelir getirir (3600'e karşı 2400).” Bu karşılaştırma iki kurala aynı teklifleri verir. GSP'de teklifler değişir; en düşük kıskançsız dengede iki kuralın geliri aynıdır.

6.5 İtibar ve rekabet

Platformlar yabancıları buluşturur. Bir alıcı, hiç tanımadığı bir satıcıdan ürün alır; bir yolcu, hiç görmediği bir sürücünün arabasına biner. Bu güven nereden gelir? Büyük ölçüde itibar sistemlerinden. İtibar sistemleri bilgi asimetrisini azaltır, ama az yorumlu ürünleri adil sıralamak ve sahte yorumu kârsız kılmak için dikkatli tasarlanmalıdır. Kullanıcıların hangi platformda kalacağını ise rekabet, düzenleme ve öneri sistemleri belirler.

6.5.1 İtibar sistemleri ve bilgi asimetrisi

Tanım 6.12

Bir alışverişte taraflardan birinin elinde ötekinin bilmediği önemli bir bilgi varsa **bilgi asimetrisi** (*information asymmetry*) vardır. Çevrim içi pazarda tipik durum şudur: satıcı ürününün kalitesini bilir, alıcı önceden bilmez.

Akerlof (1970), bilgi asimetrisinin bir piyasayı nasıl çökertebileceğini ikinci el araba örneğiyle gösterdi. Kalite bilinmeyince alıcı ortalama bir fiyat önerir. Bu fiyat iyi satıcıyı tatmin etmez ve iyi satıcı piyasadan çekilir. Piyasada kötü ürünler kalır. Amerikan argosunda kusurlu arabaya “limon” dendiği için bu duruma **limonlar piyasası** (*market for lemons*) denir.

Örnek 6.14

Piyasadaki arabaların yarısı iyi, yarısı kötü olsun. İyi bir araba satıcısına 8, alıcıya 10 değerindedir (bin TL). Kötü bir araba satıcısına 3, alıcıya 4 değerindedir. Her arabada alıcının değeri satıcınınkinden yüksektir; yani bütün arabaların el değiştirmesi iki tarafa da kazanç sağlar.

1. **Kalite bilinseydi.** İyi araba 8 ile 10 arasında, kötü araba 3 ile 4 arasında bir fiyata satılırdı. Bütün arabalar satılırdı.
2. **Kalite bilinmiyorsa.** Alıcı arabanın iyi ya da kötü olduğunu ayırt edemez. Yarı yarıya olasılıkla beklenen değer $0{,}5 \cdot 10 + 0{,}5 \cdot 4 = 7$'dir. Alıcı en çok 7 önerir.
3. **İyi satıcı çekilir.** 7, iyi satıcının değeri olan 8'in altındadır. İyi satıcı satmaz; yalnız kötü satıcılar 7'yi kabul eder.

4. **Alıcı bunu öngörür.** Piyasada yalnız kötü araba kaldığını bilen alıcı en çok 4 önerir. Sonuçta yalnız kötü arabalar el değiştirir; iyi arabaların her birindeki 2'lik kazanç kaybolur.

Örnek 6.14 çevrim içi pazarlar için de geçerlidir. Alıcı, satıcının iyi olduğunu bilemiyorsa iyi satıcının emeğine değecek bir fiyat ödemez; iyi satıcı da platformdan ayrılır. İtibar sistemi bu bilgi boşluğunu doldurmak için vardır ve iki iş görür:

- **Seçim:** İyi satıcıyı bulmak. Önceki alıcıların deneyimleri, yeni alıcıya satıcının kalitesi hakkında bilgi verir.
- **Disiplin:** Kötü davranışı gelecekte cezalandırmak. Bugün kalitesiz ürün gönderen satıcı, yarın düşük puan yüzünden müşteri kaybeder. Bu, tekrarlanan oyundaki işbirliği mantığıdır: bugünkü hile, gelecekteki kazançların kaybıyla ödenir.

İtibar sistemleri de kendi stratejik sorunlarını doğurur:

Sahte değerlendirme
: Satın alınmış olumlu yorum ya da rakibe yazılan olumsuz yorum. Luca ve Zervas (2016), restoran değerlendirme sitesi Yelp'te sahte yorumların rekabet koşullarıyla ilişkili olduğunu göstermiştir: itibarı zayıflayan ve rekabetin yoğun olduğu işletmeler sahte yoruma daha çok başvurur.

Misilleme korkusu
: Karşılıklı puanlamada (ev sahibi misafiri, misafir ev sahibini puanlar) kötü deneyim bildirilmez; çünkü bildiren, karşılığında kötü puan almaktan korkar.

Not enflasyonu
: Herkes 4,8 ve üstü alıyorsa puan ayırt edici olmaz. 4,6 puan “ortalamanın altında” anlamına gelir.

Seçici katılım
: Orta düzeyde memnun kalanlar çoğunlukla sessiz kalır; yorumlar uçlarda toplanır. Yeni satıcı ise hiç yorum alamaz; yorum almadığı için öne çıkamaz, öne çıkamadığı için satış ve yorum alamaz. Buna **soğuk başlangıç** (*cold start*) sorunu denir.

Bu sorunlara karşı tasarım önlemleri vardır: doğrulanmış satın alma şartı (yalnız ürünü gerçekten alanın yorum yazabilmesi), eski yorumlara daha az ağırlık, az yorumlu puanın belirsizliğini hesaba katmak (Bayesçi ortalama) ve karşılıklı puanlamada iki tarafın puanını aynı anda açıklamak. Son önlem misilleme korkusunu ortadan kaldırır: kimse ötekinin puanını görmeden kendi puanını verir; bu yüzden puan bir misilleme aracı olamaz.

6.5.2 Bayesçi ortalama

Ürün sıralamasında şu soruyla sık karşılaşılır: 2 yorumla 5,0 puanlı ürün mü, 200 yorumla 4,7 puanlı ürün mü daha iyidir? Ham ortalamaya göre ilki birincidir. Ama iki yorum çok az bilgidir; iki mutlu müşteri ya da iki sahte yorum ortalamayı 5,0 yapabilir.

Tanım 6.13

Bayesçi ortalama (*Bayesian average*): n yorumu olan bir ürünün puanı

$$(C \cdot m + \Sigma\,puan) / (C + n)$$

ile hesaplanır. Burada m platformun genel ortalaması (örnekte 3,5), C bir güven sabitidir (örnekte 10), $\Sigma\,puan$ ürünün aldığı puanların toplamıdır.

Formülün yorumu kolaydır: her ürüne C tane "ortalama" (puanı m olan) sanal yorum eklenmiş gibi davranılır. Sanal yorumlar, az yorumlu ürünün ortalamasını platform ortalamasına çeker. Çok yorumlu ürün ise neredeyse kendi ortalamasında kalır; çünkü 10 sanal yorum yüzlerce gerçek yorumun yanında küçüktür.

Aynı formül bir ağırlıklı ortalama olarak da yazılabilir:

$$Bayesçi\ puan = w \cdot (ham\ ortalama) + (1 - w) \cdot m,\ w = n / (n + C)$$

Ağırlık w, ürünün kendi verisine ne kadar güvenildiğini gösterir. 2 yorumda $w = 2/12 \approx 0{,}17$'dir: puanın yalnız %17'si ürünün kendi ortalamasından gelir. 200 yorumda $w = 200/210 \approx 0{,}95$'tir. "Bayesçi" adı şuradan gelir: platform ortalaması bir **ön olasılık** (önceki inanç) gibi kullanılır, ürünün yorumları da bu inancı güncelleyen kanıttır. Kanıt arttıkça son olasılık kanıta yaklaşır.

Örnek 6.15

Üç ürün: X (2 yorum, ortalama 5,0), Y (200 yorum, ortalama 4,7), Z (20 yorum, ortalama 4,2). $m = 3{,}5$, $C = 10$, yani $C \cdot m = 35$.

1. **Ham sıra.** X (5,0) > Y (4,7) > Z (4,2). X'in yalnız 2 yorumu vardır.
2. **Puan toplamları.** X: $2 \cdot 5 = 10$; Y: $200 \cdot 4{,}7 = 940$; Z: $20 \cdot 4{,}2 = 84$.
3. **Bayesçi puanlar.** X: $(35 + 10)/(10 + 2) = 45/12 = 3{,}75$. Y: $(35 + 940)/(10 + 200) = 975/210 \approx 4{,}64$. Z: $(35 + 84)/(10 + 20) = 119/30 \approx 3{,}97$.
4. **Yeni sıra.** Y (4,64) > Z (3,97) > X (3,75). Az yorumlu X ortalamaya çekilip sona düşmüştür.
5. **Sahte yorum denemesi.** X'e 3 sahte 5★ eklensin: 5 yorum, toplam 25. Ham ortalama yine 5,0'dır; X ham sıralamada zirvede kalır. Bayesçi puan $(35 + 25)/(10 + 5) = 60/15 = 4{,}0$'dır; X hâlâ Y'nin altındadır.

Ürün	yorum	ortalama	sıra	Bayesçi puan	sıra
X	2	5	1	3,75	3
Y	200	4,7	2	4,64	1

Ürün	yorum	ortalama	sıra	Bayesçi puan	sıra
Z	20	4,2	3	3,97	2
X + 3 sahte	5	5	1	4	3

Tablo 6.17: Ham ortalama ve Bayesçi ortalama sıralamaları (m = 3,5, C = 10).

Tablo 6.17 Bayesçi ortalamanın asıl yararını gösterir: sahte yorumların etkisini azaltır. Ham ortalamada X'in zirveye çıkması için hiç sahte yoruma gerek yoktur; iki gerçek ya da sahte yorum yeter. Bayesçi puanda X'in Y'ye yetişmesi için $(45 + 5k)/(12 + k) \geq 975/210$ olmalıdır; bu $k \geq 30$ verir. Aynı etkiyi yaratmak için çok daha fazla sahte yorum gerekir.

Sabitlerin seçimi bir tasarım kararıdır. C büyüdükçe az yorumlu ürünler daha çok cezalandırılır; sahteciliğe karşı direnç artar ama gerçekten iyi yeni ürünler de geç fark edilir. m genellikle platformun ya da kategorinin ortalamasıdır. m'yi düşük seçmek yeni ürünlere karşı temkinli, yüksek seçmek iyimser bir platform demektir.

Sık yapılan hata

Bayesçi formülde ortalamayı toplam yerine yazmak. Pay $C \cdot m + \Sigma\, puan$'dır, yani ürünün puanlarının *toplamı* eklenir. Y için pay $35 + 4{,}7$ değil, $35 + 940$'tır. Ortalama verildiyse önce yorum sayısıyla çarpılıp toplam bulunur.

6.5.3 Sahte değerlendirme ne zaman kârlı?

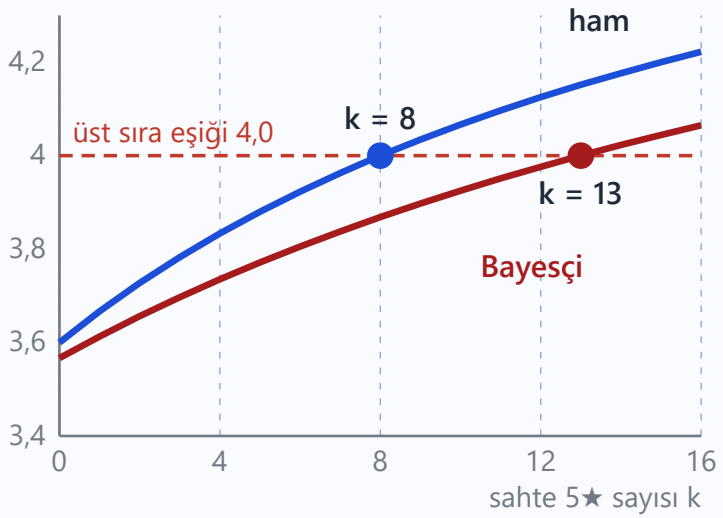
Sahte yorum yasak olsa da bazı satıcılar ona başvurur. Mekanizma tasarımı açısından soru şudur: bir satıcı için sahte yorum hangi koşullarda kârlıdır? Sahte yorum bir yatırım kararı gibi modellenebilir: getirisi, maliyeti ve beklenen cezası vardır.

Örnek 6.16

Satıcının 20 yorumu ve 3,6 ortalaması vardır. Puanı 4,0 ve üstü olanlar arama sonuçlarında üst sırada gösterilir; üst sıra satıcıya aylık $G = 600$ TL ek kâr getirir. Sahte 5★ yorumun tanesi 15 TL'dir. Platform sahteciliği ρ olasılıkla tespit eder; yakalanan satıcı $L = 3000$ TL kaybeder (hesap askıya alınır, satış kaybedilir).

1. **Kaç sahte yorum gerekir (ham ortalama)?** Puan toplamı $20 \cdot 3{,}6 = 72$'dir. k sahte 5★ eklenince ortalama $(72 + 5k)/(20 + k)$ olur. Koşul $(72 + 5k)/(20 + k) \geq 4$. Paydayla çarpılınca $72 + 5k \geq 80 + 4k$, yani $k \geq 8$. Denetim: $k = 8$ için $112/28 = 4{,}0$ ✓; $k = 7$ için $107/27 \approx 3{,}96$, yetmez.
2. **Maliyet.** $8 \times 15 = 120$ TL.
3. **Beklenen kazanç.** Satıcı 600 TL ek kâr bekler, 120 TL öder ve ρ olasılıkla 3000 TL kaybeder: $600 - 120 - \rho \cdot 3000$.

4. **Kârlılık koşulu.** Beklenen kazanç pozitifse hile kârlıdır: $480 > \rho \cdot 3000 \iff \rho < 0{,}16$. Örneğin $\rho = 0{,}1$'de beklenen kazanç $480 - 300 = 180$ TL'dir; $\rho = 0{,}2$'de $480 - 600 = -120$ TL'dir.
5. **Bayesçi puanla.** $m = 3{,}5$, $C = 10$ ile pay $35 + 72 = 107$, payda 30'dur; satıcının şu anki Bayesçi puanı $107/30 \approx 3{,}57$'dir. Koşul $(107 + 5k)/(30 + k) \geq 4$: $107 + 5k \geq 120 + 4k$, yani $k \geq 13$. Maliyet $13 \times 15 = 195$ TL. Kârlılık: $600 - 195 > \rho \cdot 3000 \iff \rho < 0{,}135$.



Şekil 6.15: Sahte 5★ yorum sayısı k'ye göre ham ortalama (mavi) ve Bayesçi puan (kırmızı). Kesikli çizgi üst sıra eşiği 4,0'dır. Ham ortalama eşiğe $k = 8$'de, Bayesçi puan $k = 13$'te ulaşır.

Şekil 6.15 iki eğriyi karşılaştırır. Bayesçi eğri ham eğrinin altında başlar (3,57'ye karşı 3,6) ve daha yavaş yükselir; çünkü her sahte yorum, gerçek yorumlarla birlikte 10 sanal yorumu da aşmak zorundadır. Aynı eşik bu nedenle 13 sahte yorum ister; hile pahalanır ve kârlılık sınırı $\rho < 0{,}135$'e iner.

Model platformun elindeki üç kolu açıkça gösterir. Beklenen kazanç $G - (\text{yorum maliyeti}) - \rho \cdot L$ biçimindedir ve platform her terimi etkileyebilir:

- **ρ'yu artırmak:** doğrulanmış alım şartı, anormallik tespiti (aynı günde gelen çok sayıda 5★, yeni hesaplardan yorumlar).
- **L'yi artırmak:** hesap kapatma, satıcının birikmiş itibarını silme, para cezası.
- **G'yi azaltmak ya da yorum maliyetini artırmak:** sağlam puanlama (Bayesçi ortalama), sahte yorumların etkisini azaltır; aynı sıralama kazancı için daha çok sahte yorum gerekir.

Mekanizma tasarımı bakışı burada açıkça görülür: platform hileyi yalnız yasaklamakla yetinmez, hilenin beklenen getirisini negatife çeker. Rasyonel bir satıcı, kuralın kendisine değil, kuralın kendi kazancına etkisine bakar.

Sık yapılan hata

Kesirli k'yi yukarı yuvarlamamak. Eşitsizlik $k \geq 7{,}5$ verirse 7 sahte yorum yetmez; sayı tam olmak zorunda olduğu için 8 gerekir.

Ceza olasılığını unutmak. "Sahte yorum 120 TL'ye 600 TL getiriyor, o hâlde her zaman kârlı." Beklenen ceza $\rho \cdot L$ hesaba katılmalıdır; $\rho \geq 0{,}16$ olduğunda hile zarar getirir.

6.5.4 Platform rekabeti ve düzenleme

Ağ etkileri yalnız tek bir platformun büyümesini değil, platformlar arasındaki rekabeti de belirler. 6.2'deki devrilme dinamiği iki platformlu bir pazara uygulandığında şu sonuç çıkar: kullanıcılar daha kalabalık olana gitmeye eğilimlidir; kalabalık olan daha da kalabalıklaşır. Bu, rekabetin üç özelliğini belirler.

Kazanan her şeyi alır
: Devrilme nedeniyle tek platform pazarın çoğunu alabilir (*winner-takes-all*). İkinci platformun ürünü iyi olsa da kullanıcıları başka yerdeyse değeri düşüktür.

Geçiş maliyeti
: Arkadaşlar, geçmiş puanlar, birikmiş veriler ve alışkanlıklar kullanıcıyı elde tutar (*switching cost*). Yüzlerce olumlu yorumu olan bir satıcı, yeni bir platformda sıfırdan başlamak istemez. Geçiş maliyeti, platformun ücretleri artırsa bile kullanıcılarını kaybetmemesini sağlar.

Çoklu kullanım
: Kullanıcılar aynı anda birden çok platform kullanabilir (*multi-homing*): bir sürücü hem Uber'de hem rakibinde çalışabilir. Çoklu kullanım tekelleşmeyi zayıflatır. Küçük platform da aynı sürücülere erişebildiği için yolcu açısından değeri düşmez; devrilme zayıflar.

Türkiye'den ve dünyadan örnekler bu dinamiği gösterir: Hepsiburada ile Trendyol, Uber ile Lyft, Instagram ile TikTok. Bu çiftlerde iki platform da ayakta kalabilmiştir. Bunun bir nedeni çoklu kullanımdır: satıcılar iki pazar yerinde birden satış yapar, kullanıcılar iki uygulamayı birden kullanır. Bir başka nedeni yerel ağ etkileridir: bir platform bazı şehirlerde ya da bazı kullanıcı gruplarında güçlü olabilir.

Ağ etkilerinin yarattığı yoğunlaşma etik ve düzenleme sorularını da doğurur:

- **Veri mahremiyeti ve güvenlik:** büyük platformlar çok sayıda kişinin verisini toplar.
- **Tekelleşme riski ve kendi ürününü kayırma** (*self-preferencing*): pazar yeri aynı zamanda satıcıysa kendi ürünlerini sıralamada öne çıkarabilir. Platform hem hakem hem oyuncu olur.
- **Sıralama ve öneri algoritmalarının şeffaflığı:** satıcı ya da içerik üreticisi neden görünmez olduğunu bilemez.

Avrupa Birliği'nin **Dijital Piyasalar Yasası** (*Digital Markets Act, DMA*) bu sorunlara yönelik bir düzenlemedir. Yasa, kullanıcı tabanı çok büyük olan ve iş kullanıcıları ile tüketiciler arasında kilit konumda duran platformları **geçit bekçisi** (*gatekeeper*) olarak tanımlar ve onlara yükümlülükler getirir.

Örneğin geçit bekçisi kendi ürünlerini rakiplerinkinden daha iyi sıralayamaz ve kullanıcıların verilerini başka platformlara taşıyabilmesine izin vermelidir. Oyun kuramı açısından bakıldığında düzenleme, platformun kendisi için belirlediği kurallara dışarıdan konan bir kısıttır: mekanizma tasarımcısının da uyması gereken bir üst mekanizma.

Veri taşınabilirliği ve birlikte çalışabilirlik, geçiş maliyetini düşürerek devrilmeyi zayıflatır. Ağ etkisi modeli açısından bu, kullanıcının bir platformda biriktirdiği değeri yeni platforma götürebilmesi demektir; böylece yeni platformun başlangıç beklentisi x_0 sıfırdan başlamaz.

6.5.5 Öneri, bağlılık ve mahremiyet

Bir platformun öneri ve sıralama algoritması görünürlüğü tahsis eder: kimin öne çıkacağına karar verir. Bu, bir **tahsis kuralıdır**; tıpkı reklam açık artırmasının yuvaları tahsis etmesi gibi. Katılımcılar stratejilerini bu kurala göre seçer. Dört konu öne çıkar.

Platformun hedefi
: Algoritma kısa vadeli etkileşimi ya da geliri maksimize ediyorsa kullanıcının uzun vadeli yararıyla çatışabilir. Tıklama tuzağı başlıklar ve sonsuz akış, etkileşim ölçüsünü artırır ama kullanıcının zamanını ve dikkatini tüketir. Platformun amaç fonksiyonu ile kullanıcının faydası aynı şey değildir.

Üreticiler de oyuncu
: Üreticiler ve satıcılar görünürlük için yarışır ve algoritmanın ödüllendirdiği biçime (başlık, etiket, video süresi) yönelir. Herkes aynı ödülü kovaladığında içerik tekdüzeleşebilir. Bu, katılımcıların kurala en iyi tepki vermesinin toplu sonucudur.

Bağlılık programları
: Sadakat puanı, ücretli üyelik (örneğin hızlı kargo), oyunlaştırma (rozet, seri) geçiş maliyetini bilerek artırır ve çoklu kullanımı azaltır. Ücretli üyeliğe para ödemiş bir kullanıcı, alışverişini o platformda toplamaya yönelir; çünkü üyeliğin değeri ancak orada alışveriş yaptıkça artar.

Veri ve mahremiyet
: "Ücretsiz" hizmetin bedeli veridir. Daha çok veri daha iyi öneri sağlar, ama sızıntı ve kötüye kullanım riskini de artırır. Kullanıcı ne kadar veri paylaşacağını da stratejik olarak seçer: daha iyi öneri ile daha az mahremiyet arasında bir değiş tokuş yapar.

Bu dört başlığın ortak sorusu, bütün bölümün sorusudur: platformun amacı ile kullanıcıların ve üreticilerin çıkarı uyumlu mu? Uyumlu değilse, kullanıcılar kendi çıkarlarını izlerken platformun ya da toplumun istemediği sonuçlar doğabilir. Mekanizma tasarımının görevi bu uyumu kurmaktır.

6.6 Tartışma

Bölümün kavramları üç genel soruya cevap verir. Her soru, notta hesaplanan bir modele dayanır.

6.6.1 Neden bazı platformlar hızla büyür, bazıları çöker?

Kritik kitle ve devrilme bu soruyu cevaplar. Beklenti eşiğin üstündeyse geri besleme büyütür, altındaysa çökertir. Benimseme modelinde iki başlangıç arasındaki 60 kullanıcılık fark, 800 kullanıcı ile sıfır arasındaki farka dönüşmüştü. Kolay kullanım, güven ve erken teşvikler eşiği aşmayı kolaylaştırır: kolay kullanım ve teşvik p'yi düşürür, güven ve tanıtım x_0'ı yükseltir. Aynı mekanizma ters yönde de çalışır: kullanıcı kaybeden bir platform eşiğin altına düştüğünde çöküş hızlanır.

6.6.2 Platformlar davranışı nasıl şekillendirir?

Bonus ve indirimler belirli davranışları artırır: yoğun bölgede sürücüye ek ödeme, sürücüleri o bölgeye çeker. Öneri algoritmaları tercihleri yönlendirir: görünür olan seçilir. Spam ve sahte yorum denetimi manipülasyonu sınırlar: tespit olasılığı ve ceza, hilenin beklenen getirisini düşürür. Her durumda kullanıcı, platformun kuralına en iyi tepkiyi verir; davranış kuralın bir fonksiyonudur.

6.6.3 Tasarım herkese adil olabilir mi?

Tam adalet zordur. Birinci neden, kullanıcıların tercihlerinin farklı olmasıdır: alıcıyı sübvanse eden bir fiyat yapısı satıcıya daha pahalıya gelir. İkinci neden, ağ etkisinin büyük grupları kayırmasıdır: büyük bir şehirdeki kullanıcı küçük bir şehirdekinden daha iyi hizmet alır, çünkü ağ etkileri yereldir. Üçüncü neden, stratejik manipülasyondur: adil bir kural bile kötüye kullanılabilir. Yine de teşvik, kural ve düzenleme ile durum iyileştirilebilir. Bayesçi ortalama az yorumlu ürünleri hem korur hem cezalandırır; kalite puanı tıklanan reklamı ödüllendirir; DMA platformun kendi ürününü kayırmasını sınırlar.

> **Sık yapılan hata**
>
> Platform sorunlarını yalnız teknik hata ya da kötü niyet olarak görmek. Sahte yorum, bonus istismarı ve içerik tekdüzeliği çoğu zaman rasyonel davranışın, yani kurala en iyi tepkinin sonucudur. Çözüm de genellikle kullanıcıları suçlamak değil, kuralı değiştirmektir.

6.7 Özet

- **Çevrim içi platform**, tarafları buluşturan bir aracıdır; değeri tarafların birbirine erişiminden doğar (iki taraflı pazar). Platformun kuralları (kim katılır, kim kimi görür, kim ne öder) bir mekanizmadır; kullanıcı davranışı bu kurallar altındaki Nash dengesidir.
- **Ağ etkisi** doğrudan, çapraz, negatif ya da yerel olabilir. Pozitif ağ etkisi bir geri besleme döngüsü kurar ve pazar bir yöne devrilebilir.
- **Benimseme modelinde** ($\theta \sim U[0, 1]$, fayda $\theta x - p$) katılan oran $F(x) = 1 - p/x$'tir. Dengeler $F(x) = x$'in çözümleridir: $p = 0{,}16$ için 0 ve 0,8 kararlı, 0,2 kararsızdır. Kararsız denge **kritik kitledir**: beklenti 0,2'nin üstündeyse 0,8'e, altındaysa 0'a gidilir. $p > 0{,}25$ ise pazar hiç oluşmaz.

- **Fiyat yapısı**, ücretin taraflar arasında nasıl bölüşüldüğüdür. Doğrusal modelde en kârlı fiyat $p_A = n_A - e_B\, n_B$'dir: kâr payı eksi bir A katılımcısının B tarafına kattığı değer. Pazar yeri örneğinde $p_A = -5$, $p_B = 50$, $\Pi = 2800$'dür; alıcı sübvanse edilir çünkü satıcılara kattığı değer (45) kâr payını (40) aşar.
- **Katılım dengesi**, tarafların birbirine sırayla tepki vermesiyle bulunabilir (`katilim()`); çapraz etkilerin çarpımı 1'den küçükse ($0{,}25 \cdot 0{,}75 = 0{,}1875$) sapma her turda bu kata küçülür ve süreç yakınsar.
- **GSP**'de yuva j tık başına bir alttakinin teklifini öder; dürüst teklif baskın değildir (A, 10 yerine 6 teklif edip faydasını 600'den 1000'e çıkarır). **VCG**'de reklamveren diğerlerine verdiği kaybı öder; dürüst teklif baskındır. Aynı tekliflerle GSP 3600, VCG 2400 gelir getirir.
- **Yerel kıskançsız dengelerde** GSP geliri en az VCG geliri kadardır; en düşüğünde ödemeler VCG ile aynıdır (teklifler 10, 5, 3,5, 2 → 5 / 3,5 / 2 TL/tık).
- **Kalite puanı** ($b \cdot q$) sıralamayı değiştirir ve ödemeyi $puan_{j+1}/q_i$ yapar; örnekte tıkları 340'tan 380'e, geliri 1.880'den 2.000'e çıkarır.
- **Bütçeli çevrim içi tahsiste** açgözlü kural en kötü durumda optimumun yarısını alır; kalan bütçeyi gözeten kurallar $1 - 1/e \approx \%63$'ünü garanti eder.
- **İtibar sistemleri** bilgi asimetrisini (limonlar piyasası) azaltır. **Bayesçi ortalama** $(C \cdot m + \Sigma\, puan)/(C + n)$ az yorumlu ürünü ortalamaya çeker ve sahte yorumu pahalılaştırır: üst sıra için gereken sahte yorum 8'den 13'e çıkar, hile $\rho < 0{,}16$ yerine ancak $\rho < 0{,}135$'te kârlı olur.
- **Rekabet**: devrilme "kazanan her şeyi alır" sonucunu doğurur; geçiş maliyeti tekeli güçlendirir, çoklu kullanım zayıflatır. DMA gibi düzenlemeler geçit bekçisi platformlara yükümlülük getirir.

Kavram	Açıklama	Örnek
Ağ etkisi, devrilme	$F(x) = 1 - p/x$; iki kararlı denge ve aradaki eşik	$p = 0{,}16$: 0 ve 0,8 kararlı, eşik 0,2
İki taraflı fiyat	$p_A = n_A - e_B n_B$: diğer tarafa çok değer katan ucuzlar	$p_A = -5$, $p_B = 50$, $\Pi = 2800$
GSP	Yuva j, bir alttakinin teklifini öder; dürüst teklif baskın değil	aynı tekliflerle gelir 3600; A 10 yerine 6 verir
VCG	Diğerlerine verilen kaybı öder; dürüst teklif baskın	1500 / 700 / 200, gelir 2400
GSP dengesi	Kıskançsız dengelerin en düşüğü VCG sonucunu verir	teklifler 10, 5, 3,5, 2 → 5 / 3,5 / 2 TL/tık
Kalite puanı	$b \cdot q$'ya göre sırala, $puan_{j+1}/q_i$ öde	sıra B, A, C; tık 380, gelir 2.000
Çevrim içi tahsis	Kalan bütçeyi gözetmek açgözlüden iyidir	açgözlü 100, dengeleme 150, optimum 200
İtibar	Bayesçi ortalama, sahte yorumu pahalılaştırır	gereken sahte yorum 8 → 13

Tablo 6.18: Bölümün kavramları bir bakışta.

6.8 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

çevrim içi platform *online platform* · iki taraflı pazar *two-sided market* · pazar yeri *marketplace* · ağ etkisi *network effect* · doğrudan ağ etkisi *direct network effect* · çapraz ağ etkisi *cross-side (indirect) network effect* · geri besleme döngüsü *feedback loop* · tavuk-yumurta sorunu *chicken-and-egg problem* · benimseme modeli *adoption model* · kendini doğrulayan beklenti *self-fulfilling expectation* · kararsız denge *unstable equilibrium* · kritik kitle *critical mass* · devrilme *tipping* · tohumlama *seeding* · fiyatlandırma *pricing* · fiyat yapısı *price structure* · sübvansiyon *subsidy* · kâr payı *markup* · sponsorlu arama açık artırması *sponsored search auction* · reklamveren *advertiser* · yuva *slot (ad position)* · tıklanma oranı *click-through rate (CTR)* · genelleştirilmiş ikinci fiyat (GSP) açık artırması *Generalized Second Price (GSP) auction* · ikinci fiyat (Vickrey) açık artırması *second-price (Vickrey) auction* · VCG mekanizması *Vickrey–Clarke–Groves (VCG) mechanism* · dışsallık *externality* · baskın strateji *dominant strategy* · kârlı sapma *profitable deviation* · kıskançsız denge *envy-free equilibrium (GSP)* · yerel kıskançsız denge *locally envy-free equilibrium (GSP)* · kalite puanı *quality score* · çevrim içi tahsis *online allocation* · dengeleme (Balance) kuralı *Balance algorithm* · rekabet oranı *competitive ratio* · itibar sistemi *reputation system* · bilgi asimetrisi *information asymmetry* · limonlar piyasası *market for lemons* · soğuk başlangıç *cold start* · Bayesçi ortalama *Bayesian average* · kazanan her şeyi alır *winner-takes-all* · geçiş maliyeti *switching cost* · çoklu kullanım *multi-homing* · kendi ürününü kayırma *self-preferencing* · Dijital Piyasalar Yasası *Digital Markets Act (DMA)* · geçit bekçisi *gatekeeper (DMA)*

6.9 Kendini Dene

1. Benimseme modelinde ($\theta \sim U[0, 1]$, fayda $\theta x - p$) katılma maliyeti $p = 0{,}1875$'tir. Dengeler nelerdir, kritik kitle kaçtır? Başlangıç beklentisi $x_0 = 0{,}3$ ve $x_0 = 0{,}2$ için dinamiğin ilk adımları nereye gider?

Cevap

$x^2 - x + 0{,}1875 = 0$: diskriminant $1 - 0{,}75 = 0{,}25$, karekökü 0,5; kökler $(1 \pm 0{,}5)/2$, yani 0,25 ve 0,75. Ayrıca $x = 0$. Kararlı dengeler 0 ve 0,75'tir; kritik kitle 0,25'tir (250 kullanıcı). $x_0 = 0{,}3$ eşiğin üstündedir: $x_1 = 1 - 0{,}1875/0{,}3 = 0{,}375$, katılım artar ve 0,75'e gider. $x_0 = 0{,}2$ eşiğin altındadır: $x_1 = 1 - 0{,}1875/0{,}2 = 0{,}0625$; bu p'den küçük olduğu için $x_2 = 0$ olur ve platform çöker.

2. Bir pazar yerinde $n_A = 30 + 0{,}25\, n_B - p_A$ ve $n_B = 60 + 0{,}75\, n_A - p_B$'dir (marjinal maliyet 0). Kârı maksimize eden katılımlar, fiyatlar ve kâr nedir? Alıcı sübvanse edilir mi? Sonuç $p_A = n_A - e_B\, n_B$ kuralına uyar mı?

Cevap

$\Pi = 30\, n_A + 60\, n_B + n_A n_B - n_A^2 - n_B^2$. Kısmi türevler: $30 + n_B - 2n_A = 0$, $60 + n_A - 2n_B = 0$. İlkinden $n_A = (30 + n_B)/2$; ikinciye yerleştirince $120 + 30 + n_B - 4n_B = 0$, yani $n_B = 50$, $n_A = 40$. Fiyatlar $p_A = 30 + 12{,}5 - 40 = 2{,}5$, $p_B = 60 + 30 - 50 = 40$; kâr $100 + 2000 = 2100$. Kural: $40 - 0{,}75 \cdot 50 = 2{,}5$ ✓, $50 - 0{,}25 \cdot 40 = 40$ ✓. Alıcı sübvanse edilmez: satıcılara kattığı değer (37,5) kâr payından (40) küçüktür. Alıcının ücreti yine de satıcınınkinden çok düşüktür.

3. Üç yuva bin gösterimde 400, 250 ve 50 tık alır. Dört reklamveren tık başına değerini teklif eder: A 9, B 7, C 4, D 3. GSP ve VCG altında her yuvanın toplam ödemesi ve platform geliri nedir? VCG'de tık başına ödemeler nedir?

Cevap

GSP: A $400 \cdot 7 = 2800$, B $250 \cdot 4 = 1000$, C $50 \cdot 3 = 150$; gelir 3950. VCG alttan yukarı: C $(50 - 0) \cdot 3 = 150$; B $(250 - 50) \cdot 4 + 150 = 950$; A $(400 - 250) \cdot 7 + 950 = 1050 + 950 = 2000$; gelir 3100. Tık başına: A $2000/400 = 5$, B $950/250 = 3{,}8$, C $150/50 = 3$. Her yuvada GSP ≥ VCG (7 ≥ 5, 4 ≥ 3,8, 3 ≥ 3).

4. 3. sorudaki GSP'de herkes dürüst teklif veriyor. A'nın (değer 9) faydası nedir? A teklifini değiştirerek faydasını artırabilir mi? Dürüst profil bir Nash dengesi midir?

Cevap

Dürüst teklifle A yuva 1'de 7 öder: $400 \cdot (9 - 7) = 800$. 4 ile 7 arasında bir teklifle yuva 2'ye iner ve C'nin 4'ünü öder: $250 \cdot (9 - 4) = 1250 > 800$. 3 ile 4 arasında yuva 3'e iner ve 3 öder: $50 \cdot 6 = 300$. En iyi tepki 4 ile 7 arası bir tekliftir (fayda 1250). A kârlı sapabildiği için dürüst profil Nash dengesi değildir. Yuva 1 ile 2 arasındaki tık farkı (150) küçük, fiyat farkı (7'ye karşı 4) büyük olduğu için alt yuva cazip gelir.

5. İki yuva bin gösterimde 100 ve 50 tık alır; yuva j'de i'nin tık sayısı $\alpha_j\, q_i$'dir. Teklifler ve kaliteler: A 5 (q = 0,6), B 4 (q = 0,9), C 6 (q = 0,4), D 2 (q = 1). Kalite puanlı GSP'de sıra, tık başına ödemeler,

toplam tık ve gelir nedir? Yalnız teklife göre sıralayan GSP'ye göre ne değişir?

Cevap

Puanlar: A 3, B 3,6, C 2,4, D 2 → sıra B, A, C, D. B yuva 1'de $3/0{,}9 \approx 3{,}33$ TL/tık, A yuva 2'de $2{,}4/0{,}6 = 4$ TL/tık öder. Tıklar: B $100 \cdot 0{,}9 = 90$, A $50 \cdot 0{,}6 = 30$; toplam 120. Gelir $90 \cdot 10/3 + 30 \cdot 4 = 300 + 120 = 420$. Yalnız teklife göre sıra C (6), A (5): C $100 \cdot 0{,}4 = 40$ tık alır ve 5 öder (200), A $50 \cdot 0{,}6 = 30$ tık alır ve B'nin 4'ünü öder (120); toplam 70 tık, gelir 320. Kalite puanı tıkları 70'ten 120'ye, geliri 320'den 420'ye çıkarır; en yüksek teklifi veren C yuva alamaz.

6. Bayesçi ortalama $m = 3{,}5$, $C = 10$ ile hesaplanıyor. Ürün P'nin 5 yorumu (ortalama 4,8), ürün R'nin 50 yorumu (ortalama 4,4) vardır. Ham ve Bayesçi sıralama nedir? P'ye en az kaç sahte 5★ eklenirse Bayesçi puanı R'ninkine ulaşır?

Cevap

Ham sıra: P (4,8) > R (4,4). Bayesçi: P $(35 + 24)/(10 + 5) = 59/15 \approx 3{,}93$; R $(35 + 220)/(10 + 50) = 255/60 = 4{,}25$. Bayesçi sıra R > P. Sahte yorumla: $(59 + 5k)/(15 + k) \geq 4{,}25 \iff 59 + 5k \geq 63{,}75 + 4{,}25k \iff 0{,}75k \geq 4{,}75 \iff k \geq 6{,}33$; tam sayı olduğu için $k = 7$ ($94/22 \approx 4{,}27$). 6 sahte yorum yetmez ($89/21 \approx 4{,}24$).

7. Bir satıcının 30 yorumu var, ortalama 3,7. Puanı 4,0 ve üstü olanlar üst sıraya çıkar ve aylık $G = 500$ TL ek kâr eder. Sahte 5★ yorumun tanesi 10 TL; tespit olasılığı ρ, yakalanınca kayıp $L = 2000$ TL. Ham ortalamayla ve Bayesçi puanla (m = 3,5, C = 10) kaç sahte yorum gerekir ve sahtecilik hangi ρ değerlerinde kârlıdır?

Cevap

Puan toplamı $30 \cdot 3{,}7 = 111$. Ham: $(111 + 5k)/(30 + k) \geq 4 \iff 111 + 5k \geq 120 + 4k \iff k \geq 9$. Maliyet 90 TL; kârlı $\iff 500 - 90 > 2000\rho \iff \rho < 0{,}205$. Bayesçi: $(35 + 111 + 5k)/(40 + k) \geq 4 \iff 146 + 5k \geq 160 + 4k \iff k \geq 14$. Maliyet 140 TL; kârlı $\iff 360 > 2000\rho \iff \rho < 0{,}18$. Bayesçi puan gereken sahte yorumu 9'dan 14'e çıkarır ve kârlılık aralığını daraltır.

8. A'nın bütçesi 80 TL, B'ninki 40 TL'dir. “Kitap” aramaları için ikisi de 1 TL, “defter” aramaları için yalnız A 1 TL teklif verir. Önce 80 kitap, sonra 80 defter araması gelir. Açgözlü kural (eşitlikte A), dengeleme kuralı (kalan bütçesi büyük olana; eşitlikte A) ve geleceği bilen optimum ne kadar gelir

getirir?

Cevap

Açgözlü: 80 kitabın hepsi A'ya gider ve A'nın bütçesi biter; defterlerde kimse yok: gelir 80. Dengeleme: A'nın kalanı 40'a inene kadar (40 arama) kitaplar A'ya gider; sonra kalanlar eşit olduğu için kalan 40 kitap 20–20 bölünür. A 60, B 20 harcamıştır. Defterlerde A'nın kalan 20 TL'si kullanılır: gelir *80 + 20 = 100*. Optimum: kitapların 40'ı B'ye, 40'ı A'ya; defterlerin 40'ı A'ya: gelir 120 (iki bütçenin toplamı). Açgözlü optimumun 2/3'ünü, dengeleme 5/6'sını almıştır.

6.10 İleri okuma

- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 15 ve 17: sponsorlu arama açık artırmalarını ve ağ etkisi modelini (kritik kitle, devrilme) az matematikle, çok sayıda şekille anlatır; kitabın taslağı yazarların sitesinde ücretsiz okunabilir.
- T. Roughgarden, Stanford CS364A ders notları (2013), 2. ve 3. dersler: mekanizma tasarımının temel kavramlarını ve Myerson lemmasını sponsorlu arama örneği üzerinden, bilgisayar bilimi bakışıyla kısa ve düzenli biçimde anlatır.
- B. Edelman, M. Ostrovsky ve M. Schwarz, "Internet Advertising and the Generalized Second-Price Auction" (2007): GSP'nin tarihini, dürüst olmamasını ve yerel kıskançsız dengeleri tanıtan temel makaledir; giriş ve örnek bölümleri teknik ayrıntıya girmeden okunabilir.
- J.-C. Rochet ve J. Tirole, "Platform Competition in Two-Sided Markets" (2003): iki taraflı pazar kavramını ve fiyat yapısının önemini ortaya koyan makaledir; kredi kartı ve yazılım gibi örneklerle sübvansiyon mantığını tartışır.

6.11 Kaynaklar

1. G. A. Akerlof, "The Market for 'Lemons': Quality Uncertainty and the Market Mechanism", *Quarterly Journal of Economics* 84(3), 488–500, 1970.
2. M. Armstrong, "Competition in Two-Sided Markets", *RAND Journal of Economics* 37(3), 668–691, 2006.
3. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*, Cambridge University Press, 2010 (Bölüm 15 ve 17).
4. B. Edelman, M. Ostrovsky, "Strategic Bidder Behavior in Sponsored Search Auctions", *Decision Support Systems* 43(1), 192–198, 2007.
5. B. Edelman, M. Ostrovsky, M. Schwarz, "Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords", *American Economic Review* 97(1), 242–259, 2007.
6. M. Luca, G. Zervas, "Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud", *Management Science* 62(12), 3412–3427, 2016.

7. A. Mehta, A. Saberi, U. Vazirani, V. Vazirani, “AdWords and Generalized Online Matching”, *Journal of the ACM* 54(5), 2007.
8. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 27–28).
9. M. Ostrovsky, M. Schwarz, “Reserve Prices in Internet Advertising Auctions: A Field Experiment”, *ACM Conference on Electronic Commerce (EC)*, 2011.
10. J.-C. Rochet, J. Tirole, “Platform Competition in Two-Sided Markets”, *Journal of the European Economic Association* 1(4), 990–1029, 2003.
11. T. Roughgarden, Stanford CS364A: Algorithmic Game Theory ders notları, 2013 (2.–3. dersler: sponsorlu arama açık artırmaları).
12. H. R. Varian, “Position Auctions”, *International Journal of Industrial Organization* 25(6), 1163–1178, 2007.
13. H. R. Varian, C. Harris, “The VCG Auction in Theory and Practice”, *American Economic Review: Papers & Proceedings* 104(5), 442–445, 2014.

KISIM II · BÖLÜM 7

Sosyal Seçim ve Oylama

Bir grup insan birlikte karar vermek zorunda kaldığında herkesin isteği aynı değildir. Bir kulüp başkan seçer, bir komite projelerden birini onaylar, bir arama motoru farklı kaynaklardan gelen sıralamaları tek listede birleştirir. Bu kararların hepsinde aynı soru vardır: bireylerin tercihlerinden tek bir toplu karar nasıl çıkarılır? Bu soruyu inceleyen alan sosyal seçim kuramıdır. En çok kullanılan oylama kuralları tek bir tercih profiline adım adım uygulandığında aynı oylar dört farklı kazanan verir. Condorcet döngüsü, gündemin gücü ve monotonluk ihlali gibi paradokslar, Kemeny kuralı ve seçmenlerin yalan söyleyerek sonucu değiştirebildiği stratejik oy sayısal örneklerle işlenir. Arrow, Gibbard–Satterthwaite ve May teoremleri, tek tepeli tercihler, hesaplamalı sosyal seçim, yargı birleştirme ve nispi temsil notun öteki konularıdır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum07`

ÖĞRENME ÇIKTILARI

Sayım
Aynı tercih profilinde mutlak çoğunluk, çoğulluk, Borda, onay, anında ikinci tur ve Condorcet kuralları-nı adım adım uygulamak.

Paradokslar
Condorcet döngüsünü, gündem sırasının etkisini ve IRV'nin monotonluk ihlalini göstermek; Kemeny sıralamasını bulmak.

Stratejik oy
Çoğullukta taktik oyu ve Borda'da "gömme" manipülasyonunu sayısal olarak göstermek.

Teoremler
Arrow, Gibbard–Satterthwaite ve May teoremlerinin ne söylediğini açıklamak; tek tepeli tercihlerle kaçış yolunu tanımak.

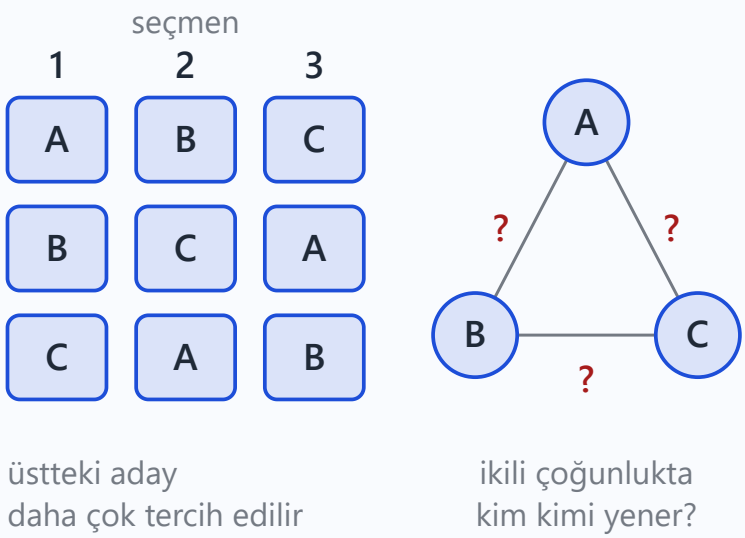
7.1 Sosyal seçim

Oyun kuramındaki oyunlarda her oyuncu kendi kazancını büyütmeye çalışır ve sonuç, oyuncuların seçimlerinin bir araya gelmesiyle oluşur. Oylamada da durum benzerdir: her seçmen kendi tercihini bildirir, bir kural bu bildirimleri toplar ve herkes için geçerli tek bir sonuç üretir. Fark, sonucu açıkça yazılmış bir kuralın belirlemesidir. Bu bölümün sorusu şudur: bu kural nasıl seçilmeli ve seçilen kuralın hangi iyi ve kötü özellikleri vardır?

7.1.1 Bir oyunla başlangıç: Condorcet paradoksu

Bölüm, sınıfta oynanan küçük bir oyunla açılır. Üç seçmen A, B ve C adlı üç adayı kendi zevklerine göre sıralar. Ardından her aday çifti ikili çoğunlukla karşılaştırılır: A ile B karşılaştırılırken her seçmen bu ikisinden kendi sıralamasında hangisi daha yukarıdaysa ona oy verir. Bütün rakiplerini bu yolla yenen adaya **Condorcet kazananı** (*Condorcet winner*) denir. Ad, bu tür karşılaştırmaları 18. yüzyılda inceleyen Fransız matematikçi Marquis de Condorcet'den gelir.

Oyunun ilk seviyesinde seçmenlerin sıralamaları şöyledir: birinci seçmen A ≻ B ≻ C, ikinci seçmen B ≻ C ≻ A, üçüncü seçmen C ≻ A ≻ B. Burada "≻" işareti "daha çok tercih edilir" anlamına gelir; A ≻ B ≻ C, "A'yı B'ye, B'yi de C'ye tercih ederim" diye okunur. Şekil 7.1'de üç sıralama sütunlar hâlinde gösterilmiştir; üstteki aday daha çok tercih edilir. Sağdaki üçgende her kenar bir ikili karşılaştırmadır ve soru işareti, o karşılaştırmayı kimin kazandığının henüz bilinmediğini gösterir.



Şekil 7.1: Condorcet Paradoksu oyununun ilk seviyesi. Solda üç seçmenin sıralamaları (üstteki aday daha çok tercih edilir), sağda ikili çoğunlukla doldurulacak karşılaştırma üçgeni.

Oyun iki soru sorar. Birincisi: her seçmenin sıralaması kendi içinde tutarlıysa, topluluğun tercihi de tutarlı olmak zorunda mıdır? İkincisi: oylamaların sırasını belirleme yetkisi bir kişide olsaydı, o kişi istediği adayı kazandırabilir miydi? İki sorunun cevabı 7.3'te hesaplanır. Kısaca: topluluğun tercihi tu-

tarsız olabilir ve oylama sırasını belirleyen kişi sonucu seçebilir. Bu iki gözlem, "halkın iradesi" denen şeyin her zaman kendiliğinden belli olmadığını gösterir.

7.1.2 Sosyal seçim nedir?

Sosyal seçim (*social choice*), bireysel tercihleri toplu bir karara ya da toplumsal bir sıralamaya dönüştüren kuralları inceleyen alandır. Alanın amacı tek bir evrensel adalet tanımı bulmak değildir. Amaç, kuralların hangi ölçütleri sağladığını ve hangi ödünleşimleri yarattığını anlamaktır. Bir kural bazı iyi özellikleri sağlarken başka iyi özellikleri bozar; hangi özelliğin daha önemli olduğu ise kararın bağlamına göre değişir.

Uygulama alanları geniştir: genel seçimler, komite ve kulüp kararları, bir sınıfın hangi etkinliğe gideceğinin seçilmesi, ortak bir kaynağın nasıl kullanılacağı ve çevrim içi anketler. Bilgisayar biliminde de aynı sorular karşımıza çıkar. Birden çok arama motorunun sonuç listelerini tek listede birleştirmek, birden çok sınıflandırıcının kararlarını oylatmak ya da kullanıcı puanlarından bir sıralama üretmek, hep bireysel sıralamalardan toplu bir sıralama çıkarma sorunlarıdır.

Sosyal seçimin temel nesnesi tercihtir. Bir seçmenin tercihi, adayları en çok sevdiğinden en az sevdiğine doğru dizen bir sıralamadır. Bu sıralamanın iki özelliği olduğu varsayılır.

Tanım 7.1

Aday kümesi A olsun. Bir seçmenin **tercihi**, A üzerinde bir sıralamadır ve $x \succ y$ ile gösterilir ("x'i y'ye tercih eder"). Bu sıralama

- **tamdır**: farklı her x, y için ya $x \succ y$ ya da $y \succ x$'tir;
- **geçişlidir** (*transitivity*): $x \succ y$ ve $y \succ z$ ise $x \succ z$'dir.

Tamlık, seçmenin her iki adayı karşılaştırabildiğini söyler; "bilmiyorum" cevabı yoktur. Geçişlilik ise sıralamanın kendi içinde çelişkisiz olmasıdır. A'yı B'ye, B'yi de C'ye tercih eden biri C'yi A'ya tercih ederse, bu kişi, A, B ve C arasında hangisini seçeceği sorulduğunda tutarlı bir cevap veremez: hangisini seçerse seçsin daha çok sevdiği başka bir aday vardır. Bu bölümde her seçmenin tercihinin tam ve geçişli olduğu varsayılır. Bölümün şaşırtıcı sonuçlarından biri şudur: bu varsayım bireyler için sağlansa bile topluluk için sağlanmayabilir.

Tanım 7.2

n seçmenin tercihlerinin listesine **tercih profili** (*preference profile*) denir: $(\succ_1, \succ_2, \ldots, \succ_n)$. Bir seçmenin bildirdiği sıralamaya **oy pusulası** (*ballot*) denir.

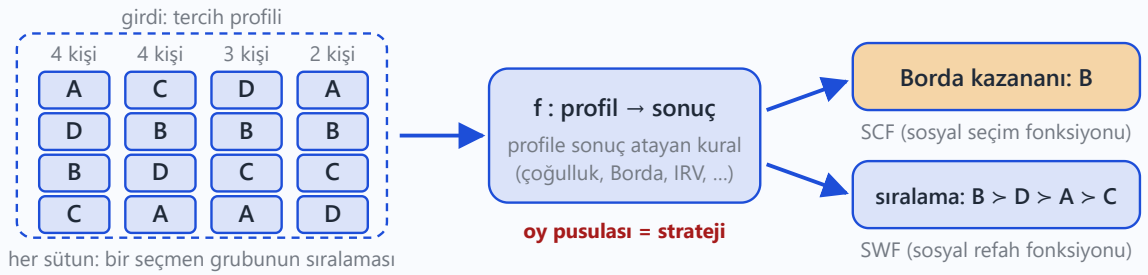
Pusula ile gerçek tercih aynı şey değildir. Seçmen gerçekte A ≻ B ≻ C düşünürken pusulaya B ≻ A ≻ C yazabilir. Oyun kuramı açısından pusula, seçmenin *stratejisidir*: seçmen pusulasını, kuralın sonucunu kendi gerçek tercihine göre en iyi yapacak şekilde seçebilir. Bu bakış açısı 7.4'teki stratejik oy konusunun temelidir.

Bir oylama kuralı tercih profilini girdi olarak alır ve bir sonuç üretir. Sonucun türüne göre iki tür kural ayrılır.

Tanım 7.3

- **Sosyal seçim fonksiyonu** (*social choice function, SCF*) her profile tek bir kazanan aday atar: $f(\succ_1, \ldots, \succ_n) \in A$.
- **Sosyal refah fonksiyonu** (*social welfare function, SWF*) her profile adayların toplumsal bir sıralamasını atar: $F(\succ_1, \ldots, \succ_n)$, A üzerinde bir sıralamadır.

Şekil 7.2 bu yapıyı özetler. Solda girdi olan tercih profili vardır; her sütun bir seçmen grubunun sıralamasıdır. Ortada profile sonuç atayan kural f durur. Sağda iki tür çıktı görülür: üstte tek bir kazanan (SCF), altta tam bir toplumsal sıralama (SWF). Şekildeki örnekte kural Borda sayımıdır; kazanan B, sıralama ise B ≻ D ≻ A ≻ C'dir. Bu sayıların nereden geldiği 7.2.3'te adım adım hesaplanır.



Şekil 7.2: Sosyal seçim kuralının girdisi ve çıktısı. Tercih profili kurala girer; çıktı ya tek bir kazanan (SCF) ya da toplumsal bir sıralamadır (SWF). Oy pusulası seçmenin stratejisidir.

Şeklin altındaki "oy pusulası = strateji" notu bu bölümün mekanizma tasarımıyla bağını gösterir. Mekanizma tasarımında bir tasarımcı, oyuncuların özel bilgilerini (tercihlerini) bildirmesini ister ve bu bildirimlere göre bir sonuç seçer; çoğu zaman oyunculardan ödeme de alır ya da onlara ödeme yapar. Oylama, mekanizma tasarımının *parasız* özel hâlidir: ödeme yoktur, yalnızca sıralamalar vardır. Para olmadığı için bir seçmenin yalan söylemesini "cezalandırmak" ya da doğruyu söylemesini "ödüllendirmek" mümkün değildir. Bu kısıtın sonuçları 7.5.4'teki Gibbard–Satterthwaite teoreminde açıkça görülür.

Sık yapılan hata

SCF ile SWF'yi karıştırmak. Bir SWF her zaman bir SCF verir: toplumsal sıralamanın ilk adayı kazanandır. Tersi doğru değildir: bir kazananı bilmek, kalan adayların toplumsal sırasını vermez. Bu ayrım önemlidir, çünkü Arrow teoremi SWF'ler (sıralamalar), Gibbard–Satterthwaite teoremi ise SCF'ler (tek kazanan) hakkındadır.

7.1.3 Örnek tercih profili

Bu bölümdeki kuralların çoğu tek bir tercih profili üzerinde karşılaştırılır. Böylece farklı kuralların aynı oylardan farklı sonuçlar çıkardığı açıkça görülür. Profil Tablo 7.1'de verilmiştir.

sıra	4 kişi	4 kişi	3 kişi	2 kişi
1.	A	C	D	A
2.	D	B	B	B
3.	B	D	C	C
4.	C	A	A	D

Tablo 7.1: Örnek tercih profili: 13 seçmen, 4 aday. Her sütun aynı sıralamayı veren seçmen grubudur.

Tablo şöyle okunur. Aynı sıralamayı veren seçmenler bir sütunda toplanmıştır; sütun başlığı o sıralamayı kaç seçmenin verdiğini söyler. Birinci sütun “4 seçmen A ≻ D ≻ B ≻ C sıralamasını verdi” demektir. Bu dört seçmen için A en iyi, C en kötü adaydır. İkinci sütundaki dört seçmen C ≻ B ≻ D ≻ A, üçüncü sütundaki üç seçmen D ≻ B ≻ C ≻ A, dördüncü sütundaki iki seçmen A ≻ B ≻ C ≻ D sıralamasını verir. Not boyunca bu dört sütundan sırasıyla *1. grup*, *2. grup*, *3. grup* ve *4. grup* (kısaca G1–G4) diye söz edilir.

Toplam seçmen sayısı *4 + 4 + 3 + 2 = 13*'tür. Bazı kurallarda bir adayın kazanması için oyların yarısından fazlasını alması gerekir. 13'ün yarısı 6,5'tir; yarısından fazla demek en az 7 oy demektir. Bu sayıya **mutlak çoğunluk eşiği** denir ve bu profilde 7'dir.

Profilde dikkat çeken birkaç özellik önceden not edilebilir. A, altı seçmenin (G1 ve G4) birinci tercihidir ama yedi seçmenin (G2 ve G3) son tercihidir: A seçmenleri kutuplaştıran bir adaydır. B'yi hiçbir seçmen birinci sıraya koymamıştır, ama dokuz seçmen onu ikinci sıraya koymuştur ve kimse onu sona koymamıştır: B herkesin “idare eder” dediği uzlaşmacı adaydır. D ise üç seçmenin birinci, dört seçmenin ikinci tercihidir. Bu özellikler, her kuralın neden farklı bir aday seçtiğini açıklar: aynı profile uygulanan kurallar dört farklı kazanan verir.

Örnek 7.1

Soru: profilde kaç seçmen D'yi B'ye tercih eder? Her grup için D ile B'nin yeri karşılaştırılır.

1. G1 (A ≻ D ≻ B ≻ C): D ikinci, B üçüncü sırada; D ≻ B. 4 seçmen.

2. G2 ($C \succ B \succ D \succ A$): B ikinci, D üçüncü; $B \succ D$. 0 seçmen.
3. G3 ($D \succ B \succ C \succ A$): D birinci, B ikinci; $D \succ B$. 3 seçmen.
4. G4 ($A \succ B \succ C \succ D$): B ikinci, D dördüncü; $B \succ D$. 0 seçmen.

Toplam: $4 + 3 = 7$ seçmen D'yi B'ye, $4 + 2 = 6$ seçmen B'yi D'ye tercih eder. İki sayının toplamı seçmen sayısına (13) eşittir; çünkü tam sıralamada her seçmen iki adaydan tam birini öne koyar. Bu kontrol, ikili karşılaştırma hesaplarında yapılan sayım hatalarını yakalamanın en kolay yoludur.

7.2 Oylama kuralları

En çok kullanılan oylama kurallarının hepsi Tablo 7.1'deki aynı profile uygulanır. Her kural için üç soru sorulur: seçmen ne bildirir, sayım nasıl yapılır ve kural bu profilde kimi seçer?

7.2.1 Temel oylama kurallarına genel bakış

Kurallar iki soruya göre ayrılır. Birincisi: seçmen pusulada ne bildirir? Bazı kurallarda yalnızca tek bir aday yazılır; bazılarında adayların tam sıralaması, bazılarında ise kabul edilen adayların kümesi istenir. İkincisi: bildirimler nasıl sayılır ve kazanan nasıl belirlenir? Tablo 7.2 altı temel kuralı bu iki soruya göre özetler.

Kural	Seçmen ne bildirir?	Kazanan
Mutlak çoğunluk (*majority*)	Tek aday	Oyların yarısından fazlasını alan (yoksa kazanan yok)
Çoğulluk (*plurality*)	Tek aday	En çok birinci tercih alan; yarıyı geçmesi gerekmez
Borda sayımı	Tam sıralama	m aday için sıradaki yer $m-1, m-2, \ldots, 0$ puan; en yüksek toplam
Onay (*approval*)	Onaylanan aday kümesi	En çok onay alan
Anında ikinci tur (IRV)	Tam sıralama	En az birinci tercih alan elenir, oyları aktarılır; çoğunluğa dek tekrarlanır
Condorcet	Tam sıralama	İkili karşılaştırmada her rakibini yenen aday (her zaman olmayabilir)

Tablo 7.2: Temel oylama kuralları.

Tablodaki kuralların bir kısmı sıralamanın yalnızca bir parçasını kullanır. Mutlak çoğunluk ve çoğulluk yalnızca birinci tercihe bakar; seçmenin ikinci, üçüncü tercihleri hiç sayılmaz. Borda, IRV ve Condorcet ise bütün sıralamayı kullanır. Daha çok bilgi kullanmak her zaman daha "adil" bir sonuç vermez, ama sonucun hangi bilgiye duyarlı olduğunu değiştirir.

Sık yapılan hata

"En çok oyu alan kazanır" cümlesi mutlak çoğunluğun değil çoğulluğun tanımıdır. Mutlak çoğunluk oyların %50'sinden fazlasını ister; bu eşik aşılmazsa kazanan yoktur. Günlük dilde ikisi de "çoğunluk" diye anılır, ama sonuçları farklıdır: 13 seçmenli profilde 6 oy alan aday çoğulluğu kazanır ama mutlak çoğunluğu sağlamaz.

7.2.2 Mutlak çoğunluk, çoğulluk ve iki turlu sistem

Tanım 7.4

Her seçmen yalnızca birinci tercihini bildirir. Bir adayın **birinci tercih sayısı**, onu birinci sıraya koyan seçmen sayısıdır.

- **Mutlak çoğunluk** (*majority rule*): birinci tercih sayısı $n/2$'den büyük olan aday kazanır; böyle bir aday yoksa kazanan yoktur.
- **Çoğulluk** (*plurality rule*): birinci tercih sayısı en büyük olan aday kazanır.

İki kural da aynı sayımı kullanır; fark yalnızca kazananın nasıl ilan edildiğindedir. Mutlak çoğunluk $n/2$'den fazla oy şartı koyar. Bu şart sağlandığında kazananın arkasında seçmenlerin çoğunluğu olduğu kesindir. Ama şart her zaman sağlanmaz; o zaman kural kazanan üretemez ve ek bir yöntem (ikinci tur gibi) gerekir. Çoğulluk ise her zaman bir kazanan üretir (eşitlik bozma kuralıyla birlikte), ama kazananın arkasında seçmenlerin küçük bir azınlığı olabilir.

Örnek 7.2

Tablo 7.1'deki profil için birinci tercihler sayılır. Yalnızca tablonun ilk satırına bakılır: A, C, D, A.

1. A: G1'in 4 seçmeni ve G4'ün 2 seçmeni A'yı birinci koyar; $4 + 2 = 6$.
2. B: hiçbir grup B'yi birinci koymaz; 0.
3. C: G2'nin 4 seçmeni; 4.
4. D: G3'ün 3 seçmeni; 3.
5. Kontrol: $6 + 0 + 4 + 3 = 13$, seçmen sayısına eşit.

Mutlak çoğunluk: eşik 7'dir ve en yüksek sayı $6 < 7$; kazanan yoktur. **Çoğulluk:** en çok birinci tercihi A alır ve A kazanır. Ama A'nın arkasında oyların yalnızca 6/13'ü, yani yaklaşık %46'sı vardır.

Mutlak çoğunluğun sağlanamadığı durumlar için uygulamada en yaygın çözüm **iki turlu sistemdir** (*two-round system, runoff*). Türkiye ve Fransa'daki cumhurbaşkanlığı seçimleri bu sistemi kullanır. İlk turda mutlak çoğunluğu sağlayan aday yoksa ilk turda en çok oy alan iki aday ikinci tura kalır. İkinci turda yalnızca bu iki aday yarışır; seçmenler ikisinden hangisini tercih ediyorsa ona oy verir.

Örnek 7.3

İlk turda A 6, C 4 oy aldığı için ikinci tura A ve C kalır. Her seçmenin bu iki aday arasındaki tercihi sıralamasından okunur.

1. G1 (A ≻ D ≻ B ≻ C): A, C'den yukarıda; A'ya 4 oy.
2. G2 (C ≻ B ≻ D ≻ A): C'ye 4 oy.
3. G3 (D ≻ B ≻ C ≻ A): favorileri D yarışta yok; sıralamalarında C üçüncü, A dördüncü olduğundan C'ye 3 oy.
4. G4 (A ≻ B ≻ C ≻ D): A'ya 2 oy.

İkinci tur: $A = 4 + 2 = 6$, $C = 4 + 3 = 7$. C kazanır. Birinci turun lideri A, ikinci turda kaybetmiştir; çünkü D'yi destekleyen 3 seçmen C'yi A'ya tercih eder.

Bu örnek, iki turlu sistemin çoğulluktan neden farklı sonuç verebildiğini gösterir: ikinci tur, elenen adayların seçmenlerine kalan adaylar arasında tercih yapma imkânı verir. Ama bu imkân sınırlıdır. İlk turda üçüncü veya dördüncü olan bir aday, bire bir yarışta herkesi yenebilecek olsa bile ikinci tura kalamaz. Bu profilde tam olarak bu olur: 7.2.7'de görüleceği gibi D bütün rakiplerini ikili karşılaştırmada yener, ama ilk turda yalnızca 3 oy aldığı için ikinci tura kalamaz.

7.2.3 Borda sayımı

Çoğulluk, seçmenin ikinci tercihini hiç dikkate almaz. Bir aday herkesin ikinci tercihi olsa bile sıfır oy alır. 18. yüzyılda Fransız denizci ve matematikçi Jean-Charles de Borda bu sorunu fark edip her sıraya puan veren bir sayım önermiştir.

Tanım 7.5

m aday varken **Borda sayımında** (*Borda count*) her seçmen tam sıralama bildirir. Bir seçmenin sıralamasında k. sıradaki aday o seçmenden $m - k$ puan alır: birinci $m - 1$, ikinci $m - 2$, …, sonuncu 0 puan. Adayın Borda puanı bütün seçmenlerden aldığı puanların toplamıdır; puanı en yüksek aday kazanır.

Borda puanının sade bir yorumu vardır: bir seçmenin sıralamasında x'in aldığı puan, o seçmenin x'in altına koyduğu aday sayısıdır. Birinci sıradaki aday $m - 1$ adayı geride bırakır, sonuncu hiçbirini. Dolayısıyla bir adayın toplam Borda puanı, bütün seçmenler üzerinden "x'in yendiği aday" sayılarının

toplamıdır. Bu yorum 7.2.7'deki ikili karşılaştırmalarla bağ kurmaya yarar: Borda puanı, bir adayın ikili karşılaştırma tablosundaki satır toplamına eşittir.

Dört aday için puanlar 1. sıra 3, 2. sıra 2, 3. sıra 1, 4. sıra 0'dır. Bir grubun bir adaya katkısı "seçmen sayısı × puan"dır. Sayım grup grup yapılır.

Örnek 7.4

1. **G1** (4 seçmen, A ≻ D ≻ B ≻ C): A $4 \times 3 = 12$, D $4 \times 2 = 8$, B $4 \times 1 = 4$, C $4 \times 0 = 0$.
2. **G2** (4 seçmen, C ≻ B ≻ D ≻ A): C $4 \times 3 = 12$, B $4 \times 2 = 8$, D $4 \times 1 = 4$, A 0. Ara toplamlar A 12, B 12, C 12, D 12: iki gruptan sonra dört aday eşittir.
3. **G3** (3 seçmen, D ≻ B ≻ C ≻ A): D $3 \times 3 = 9$, B $3 \times 2 = 6$, C $3 \times 1 = 3$, A 0. Ara toplamlar A 12, B 18, C 15, D 21.
4. **G4** (2 seçmen, A ≻ B ≻ C ≻ D): A $2 \times 3 = 6$, B $2 \times 2 = 4$, C $2 \times 1 = 2$, D 0.

Sonuç Tablo 7.3'te toplanmıştır: B = 22, D = 21, A = 18, C = 17. Borda kazananı B'dir.

aday	G1	G2	G3	G4	toplam
A	12	0	0	6	18
B	4	8	6	4	**22**
C	0	12	3	2	17
D	8	4	9	0	21

Tablo 7.3: Borda sayımında grupların katkısı ve toplamlar (puanlar 3, 2, 1, 0).

Sonucun en ilginç yanı, B'yi hiçbir seçmenin birinci sıraya koymamış olmasıdır. B'nin gücü, hiçbir seçmenin onu sona koymaması ve dokuz seçmenin onu ikinci sıraya koymasıdır. Borda bu tür uzlaşmacı adayları öne çıkarır. Bu, bir erdem (herkesin kabul edebileceği aday) ya da bir kusur (kimsenin gerçekten istemediği aday) olarak görülebilir.

Hesabı kontrol etmenin kolay bir yolu vardır. Her seçmen toplamda $3 + 2 + 1 + 0 = 6$ puan dağıtır. 13 seçmen toplam $13 \times 6 = 78$ puan dağıtmalıdır. Gerçekten $18 + 22 + 17 + 21 = 78$'dir. Toplam tutmazsa bir yerde katkı yanlış hesaplanmıştır.

Sık yapılan hata

- **Puanları 1'den başlatmak:** 4, 3, 2, 1 puan kullanmak her adaya her seçmenden 1 fazla puan ekler. Bu kazananı değiştirmez, ama sonuçlar bu nottaki sayılarla uyuşmaz ve aday sayısı değiştiğinde (bir aday çekildiğinde) karışıklığa yol açar. Tanımdaki ölçek $m - 1, \ldots, 0$'dır.

- **Grup büyüklüğünü unutmak:** bir sütun bir seçmen değil, bir seçmen grubudur. G3'ün D'ye katkısı 3 değil, $3 \times 3 = 9$'dur.

7.2.4 Onay oylaması

Tanım 7.6

Onay oylamasında (*approval voting*) her seçmen kabul ettiği (onayladığı) adayların kümesini işaretler; istediği kadar aday onaylayabilir. Her onay adaya 1 puan kazandırır; en çok onay alan aday kazanır.

Onay oylamasında seçmenin bildirdiği şey sıralama değil, bir kümedir. Seçmen "bu adaylar benim için kabul edilebilir, ötekiler değil" der. Onay oylamasının sonucu bu nedenle yalnızca seçmenlerin sıralamasına değil, her seçmenin *nerede çizgi çektiğine* de bağlıdır. Tercih profili tek başına onay kazananını belirlemez; her seçmenin kaç adayı onayladığı da bilinmelidir. Bu yüzden onay sayımı için bir varsayım gerekir.

Örnek 7.5

Varsayım: her seçmen sıralamasındaki ilk iki adayı onaylar. Tablonun ilk iki satırı okunur.

1. A: G1 (1. sırada) 4 ve G4 (1. sırada) 2; $4 + 2 = 6$.
2. B: G2 (2. sırada) 4, G3 (2. sırada) 3, G4 (2. sırada) 2; $4 + 3 + 2 = 9$.
3. C: G2 (1. sırada) 4; 4.
4. D: G1 (2. sırada) 4, G3 (1. sırada) 3; $4 + 3 = 7$.
5. Kontrol: her seçmen 2 onay verir; $13 \times 2 = 26 = 6 + 9 + 4 + 7$.

Onay kazananı B'dir (9). Herkes yalnızca birinci tercihini onaylasaydı onay sayıları birinci tercih sayılarıyla aynı olurdu (A 6, B 0, C 4, D 3) ve A kazanırdı.

Örnek iki önemli noktayı gösterir. Birincisi, herkes ilk iki adayını onayladığında onay oylaması *2-onay* kuralına (7.2.5) dönüşür; herkes yalnızca ilk adayını onayladığında ise çoğulluğa dönüşür. Onay oylaması bu iki uç arasında esnek bir yerde durur. İkincisi, sonuç onay eşiğine bağlıdır: aynı tercihlerle eşik değişince kazanan A'dan B'ye geçer. Eşiği seçmenin kendisi belirlediği için onay eşiği de stratejik bir karardır. Örneğin A'yı destekleyen bir seçmen, B'nin güçlü bir rakip olduğunu biliyorsa ikinci tercihi B'yi onaylamaktan vazgeçebilir.

Onay oylaması bazı bilimsel dernek seçimlerinde ve "hangi saatler uygun?" türü toplantı anketlerinde kullanılır. Toplantı anketinde her katılımcı uygun olduğu saatlerin hepsini işaretler ve en çok kişiye uyan saat seçilir; bu, onay oylamasının tam kendisidir.

7.2.5 Pozisyonel puanlama kuralları

Çoğulluk, Borda ve (herkesin aynı sayıda aday onayladığı) onay oylaması aslında aynı fikrin farklı hâlleridir: her sıraya sabit bir puan verilir ve puanlar toplanır. Bu ortak yapı tek bir tanımla yazılabilir.

Tanım 7.7

m aday için bir **pozisyonel puanlama kuralı** (*positional scoring rule*) bir puan vektörüyle verilir: $s = (s_1, s_2, \ldots, s_m)$, $s_1 \geq s_2 \geq \ldots \geq s_m$ ve $s_1 > s_m$. Bir seçmenin sıralamasında k. sıradaki aday s_k puan alır; toplam puanı en yüksek aday kazanır.

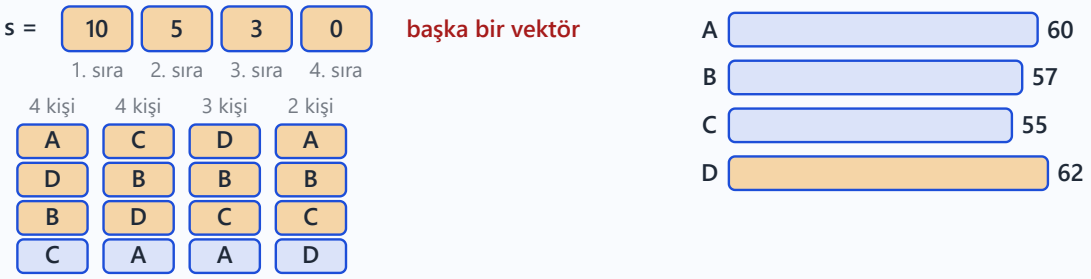
- **Çoğulluk**: $s = (1, 0, \ldots, 0)$.
- **k-onay** (*k-approval*): ilk k sıra 1, gerisi 0 puan; $k = 2$ için **2-onay**.
- **Borda**: $s = (m - 1, m - 2, \ldots, 0)$.
- **Veto kuralı** (*veto rule*): $s = (1, \ldots, 1, 0)$; yalnızca sonuncu sıra 0 puan alır.

Puanların azalan sırada olması, seçmenin daha çok sevdiği adaya daha az puan vermemesi içindir. Veto kuralının adı yorumundan gelir: her seçmen yalnızca bir adayı “veto eder”, yani reddeder; bir adayın veto puanı, onu sona koymayan seçmen sayısıdır. Veto kuralı herkesin en az itiraz ettiği adayı arar.

Şekil 7.3 ve Şekil 7.4 aynı profilin iki farklı puan vektörüyle sayımını gösterir. Solda profil vardır; turuncu hücreler puan alan sıralardır. Sağda her adayın toplam puanı bir çubukla çizilmiştir; turuncu çubuk kazananı gösterir.



Şekil 7.3: Çoğulluk, $s = (1, 0, 0, 0)$. Yalnızca birinci sıra puan alır: A = 6, C = 4, D = 3, B = 0; kazanan A.



Şekil 7.4: Başka bir vektör, *s = (10, 5, 3, 0)*. D = 62, A = 60, B = 57, C = 55; kazanan D. Bu vektör Condorcet kazananı D'yi seçer.

Beş farklı vektörün sonuçları Tablo 7.4'te toplanmıştır. Hesapların hepsi aynı yöntemle yapılır: her grubun her adaya katkısı "seçmen sayısı × o sıranın puanı"dır.

Kural	s	Puanlar	Kazanan
Çoğulluk	(1, 0, 0, 0)	A 6, C 4, D 3, B 0	A
2-onay (ilk iki)	(1, 1, 0, 0)	B 9, D 7, A 6, C 4	B
Borda	(3, 2, 1, 0)	B 22, D 21, A 18, C 17	B
Veto (sonuncuya 0)	(1, 1, 1, 0)	B 13, D 11, C 9, A 6	B
Başka bir vektör	(10, 5, 3, 0)	D 62, A 60, B 57, C 55	D

Tablo 7.4: Aynı profilde beş puan vektörü.

Örnek 7.6

Veto, *s = (1, 1, 1, 0)*: bir adayın puanı, onu sona koymayan seçmen sayısıdır. Sona konanlar tablonun son satırındadır: G1 C'yi, G2 ve G3 A'yı, G4 D'yi sona koyar.

- A'yı *4 + 3 = 7* seçmen sona koyar; A = *13 − 7 = 6*.
- B'yi kimse sona koymaz; B = 13.
- C'yi 4 seçmen sona koyar; C = 9.
- D'yi 2 seçmen sona koyar; D = 11.

(10, 5, 3, 0) vektörü:

- A: G1'de 1. (*4 × 10 = 40*), G2 ve G3'te 4. (0), G4'te 1. (*2 × 10 = 20*); A = 60.
- B: G1'de 3. (*4 × 3 = 12*), G2'de 2. (*4 × 5 = 20*), G3'te 2. (*3 × 5 = 15*), G4'te 2. (*2 × 5 = 10*); B = 57.
- C: G1'de 4. (0), G2'de 1. (40), G3'te 3. (*3 × 3 = 9*), G4'te 3. (*2 × 3 = 6*); C = 55.
- D: G1'de 2. (*4 × 5 = 20*), G2'de 3. (*4 × 3 = 12*), G3'te 1. (30), G4'te 4. (0); D = 62.

Bu tablo bölümün ana mesajlarından birini sayılarla verir: aynı oylarla yalnızca puan vektörü değişince kazanan A, B ve D arasında değişir. Hangi puan vektörünün "doğru" olduğuna dair matematiksel bir cevap yoktur. Birinci sıraya çok ağırlık veren vektörler çoğulluğa benzer ve kutuplaştıran adayları (A) öne çıkarır; sıralar arasında eşit aralık bırakan vektörler uzlaşmacı adayları (B) öne çıkarır.

Puanlama kurallarının ortak bir iyi özelliği vardır: *monotondurlar*. Bir seçmen kazanan adayı kendi pusulasında bir sıra yukarı taşırsa o aday ya aynı puanı ya da daha fazla puan alır, çünkü puan vektöründe üst sıra en az alt sıra kadar puan verir. Yukarı taşınan adayın yer değiştirdiği öteki aday ise en fazla aynı puanı alır. Dolayısıyla kazananın öteki adaylara karşı farkı azalmaz ve kazanan kazanmaya devam eder. Bu gözlem önemlidir, çünkü anında ikinci tur bu özelliği sağlamaz (7.3.7).

Sık yapılan hata

Anında ikinci turu bir puanlama kuralı sanmak. IRV de tam sıralama kullanır, ama bir adayın bir turda sayılan oyu hangi adayların daha önce elendiğine bağlıdır. Aynı pusula bir turda A'ya, bir sonraki turda C'ye sayılabilir. Bu yüzden IRV tek bir sabit puan vektörüyle yazılamaz.

7.2.6 Anında ikinci tur

İki turlu sistemin bir sorunu, seçmenleri iki kez sandığa götürmesidir. Bir başka sorunu, ikinci tura yalnızca iki aday kalması ve birinci turda üçüncü olan adayın şansının hemen bitmesidir. **Anında ikinci tur** (*instant-runoff voting, IRV*; tercihli oy diye de bilinir) iki soruna birlikte çözüm arar: seçmenden bir kez tam sıralama alınır ve turlar bu sıralamalar üzerinden bilgisayarla ya da elle "anında" yapılır.

Tanım 7.8

Her seçmen tam sıralama bildirir. Başta bütün adaylar yarıştadır. Her turda:

1. Her pusula, o pusulada *hâlâ yarışan* en üstteki adaya sayılır.
2. Bir adayın oy sayısı toplam oyun yarısından fazlaysa o aday kazanır.
3. Yoksa en az oy alan aday elenir; onun pusulaları bir sonraki turda sıralamadaki bir sonraki yarışan adaya geçer.

Bu kuralda hiçbir pusula boşa gitmez: seçmenin favorisi elenirse oyu sıralamasındaki bir sonraki adaya aktarılır. İki turlu sistemden farkı, adayların tek tek elenmesi ve her elemeden sonra oyların yeniden dağılmasıdır.

Örnek 7.7

1. **1. tur:** yarışan adaylar A, B, C, D. Birinci tercihler A = 6, B = 0, C = 4, D = 3. Çoğunluk için 7 gerekir; kimse ulaşamaz.

2. **Eleme:** en az oylu B'dir (0 oy). B elenir. B'yi birinci sıraya koyan kimse olmadığı için aktarılacak oy da yoktur.
3. **2. tur:** yarışan A, C, D. Her pusula yarışan en üst adaya sayılır; B'nin elenmesi hiçbir pusulanın sayıldığı adayı değiştirmez. A = 6, C = 4, D = 3. Yine çoğunluk yok.
4. **Eleme:** en az oylu D'dir (3). D elenir. G3'ün pusulası D ≻ B ≻ C ≻ A'dır; D ve B elendiği için sıradaki yarışan aday C'dir. 3 oy D'den C'ye geçer.
5. **3. tur:** yarışan A ve C. A = 6, C = *4 + 3 = 7*. *7 ≥ 7*: C mutlak çoğunlukla kazanır.

Bu profilde IRV'nin sonucu iki turlu sistemle aynıdır: C. Nedeni, iki aday elendikten sonra kalan ikilinin, iki turlu sistemin ikinci turuna kalan ikiliyle aynı olmasıdır. Ama genel olarak iki kural farklı sonuç verebilir, çünkü IRV'de eleme sırası oyların aktarılmasıyla değişebilir. Dikkat çeken bir başka nokta: IRV, Condorcet kazananı olan D'yi ikinci turda eler. D'nin birinci tercih sayısı az olduğu için D, bütün rakiplerini bire bir yarışta yenebilecek olsa da yarışın erken bir aşamasında çıkar.

Sık yapılan hata

Elenen adayın oylarını pusuladaki ikinci sıradaki adaya aktarmak. Oy, sıralamada bir sonraki *hâlâ yarışan* adaya geçer. G3'ün pusulasında D'den sonra B gelir, ama B daha önce elendiği için oy B'ye değil C'ye geçer. İkinci sıradaki aday elenmişse üçüncüye, o da elenmişse dördüncüye bakılır.

7.2.7 Condorcet: ikili karşılaştırmalar

Buraya kadarki kurallar adayları hep birlikte değerlendirir. Condorcet'nin önerisi farklıdır: her aday çifti ayrı ayrı, sanki yalnızca bu iki aday yarışıyormuş gibi karşılaştırılır. İki aday arasındaki yarışı, iki adaydan hangisini daha üstte gören seçmenlerin sayısı belirler.

Tanım 7.9

N(x, y), *x*'i *y*'ye tercih eden seçmen sayısı olsun. *N(x, y) > N(y, x)* ise *x*, *y*'yi **ikili çoğunlukla yener**.

- Bütün rakiplerini yenen aday **Condorcet kazananıdır** (*Condorcet winner*).
- Bütün rakiplerine yenilen aday **Condorcet kaybedenidir** (*Condorcet loser*).

Condorcet kazananı fikri çok çekicidir. Bir adayı her rakibine karşı seçmenlerin çoğunluğu tercih ediyorsa, yerine kim seçilirse seçilsin, o kişi yerine bu adayı isteyen bir çoğunluk memnuniyetsiz kalır. Ama Condorcet kazananı her profilde bulunmayabilir; bu durum 7.3'ün konusudur.

Dört aday için *4 · 3 / 2 = 6* çift vardır. Genel olarak *m* aday için çift sayısı *m(m − 1)/2*'dir. Sonuçlar bir tabloya yazılır: (x, y) hücresi, x'i y'ye tercih eden seçmen sayısı *N(x, y)*'dir.

Örnek 7.8

Her çift için gruplar tek tek incelenir. Sayılar seçmen sayısıdır; her satırda iki sayının toplamı 13 olmalıdır.

1. **A – B:** A ≻ B diyenler G1 (4) ve G4 (2); B ≻ A diyenler G2 (4) ve G3 (3). 6 – 7: B kazanır.
2. **A – C:** A ≻ C: G1 (4), G4 (2); C ≻ A: G2 (4), G3 (3). 6 – 7: C kazanır.
3. **A – D:** A ≻ D: G1 (4), G4 (2); D ≻ A: G2 (4), G3 (3). 6 – 7: D kazanır.
4. **B – C:** B ≻ C: G1 (4), G3 (3), G4 (2); C ≻ B: G2 (4). 9 – 4: B kazanır.
5. **B – D:** B ≻ D: G2 (4), G4 (2); D ≻ B: G1 (4), G3 (3). 6 – 7: D kazanır.
6. **C – D:** C ≻ D: G2 (4), G4 (2); D ≻ C: G1 (4), G3 (3). 6 – 7: D kazanır.

x ╲ y	A	B	C	D	galibiyet
A	—	6	6	6	0
B	7	—	9	6	2
C	7	4	—	6	1
D	**7**	**7**	**7**	—	**3**

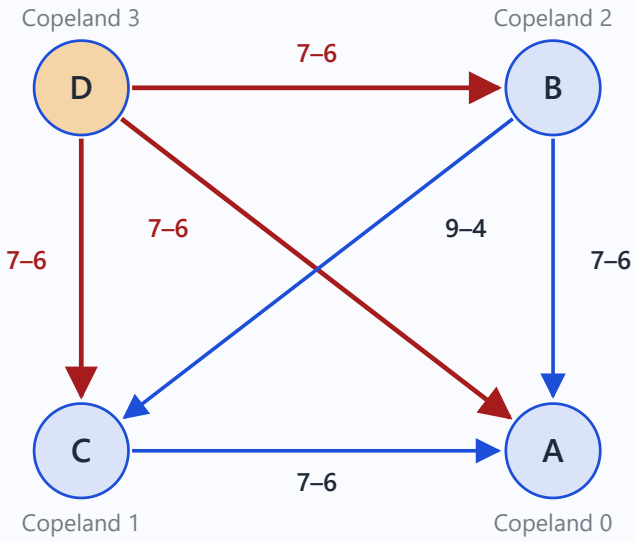
Tablo 7.5: İkili karşılaştırma tablosu. Hücre (x, y) = x'i y'ye tercih eden seçmen sayısı; galibiyet = satırda 6,5'ten büyük sayıların adedi.

Tablo 7.5 tamamlanınca sonuç açıktır: D üç rakibini de 7–6 yener ve Condorcet kazananıdır. A ise herkese 6–7 yenilir ve Condorcet kaybedenidir. Şaşırtıcı olan, çoğulluğun tam olarak bu Condorcet kaybedeni A'yı seçmiş olmasıdır. A'nın birinci tercih sayısı en yüksektir, ama A'yı birinci koymayan yedi seçmenin hepsi onu son sıraya koyar; bu yedi seçmen, A'ya karşı yarışan her adayı destekler.

Tablonun satır toplamları da bir şey söyler: A satırı $6 + 6 + 6 = 18$, B satırı $7 + 9 + 6 = 22$, C satırı $7 + 4 + 6 = 17$, D satırı $7 + 7 + 7 = 21$'dir. Bunlar tam olarak Borda puanlarıdır. Bu, 7.2.3'teki yorumun bir sonucudur: Borda puanı, bir adayın bütün ikili karşılaştırmalarda topladığı oyların toplamıdır. Borda ile Condorcet arasındaki fark da buradan görülür. Condorcet yalnızca kimin kazandığına bakar (7, 6'dan büyük mü?), Borda ise farkın büyüklüğünü de sayar. B'nin C'ye karşı 9–4'lük büyük farkı, D'ye karşı 6–7'lik küçük yenilgisini Borda toplamında telafi eder.

7.2.8 İkili çoğunluk çizgesi ve Copeland

İkili karşılaştırmaların sonuçları bir çizgeyle de gösterilebilir. Bu çizgede her aday bir düğümdür. x, y'yi ikili çoğunlukla yeniyorsa x'ten y'ye bir ok çizilir. Bu çizgeye **ikili çoğunluk çizgesi** (*pairwise majority graph*) denir.



Şekil 7.5: Örnek profilin ikili çoğunluk çizgesi. x → y oku, çoğunluğun x'i y'ye tercih ettiğini gösterir; etiketler oy sayılarıdır. D'den çıkan üç ok vurguludur. Düğümlerin yanında Copeland skorları yazılıdır.

Seçmen sayısı tek olduğunda iki aday arasında eşitlik olamaz; bu yüzden her çift arasında tam olarak bir ok vardır. Böyle bir çizgeye **turnuva** (*tournament*) denir; adı, her takımın her takımla bir kez karşılaştığı lig maçlarından gelir. Şekil 7.5'te D'den üç ok çıkar ve D'ye hiç ok girmez. Genel kural şudur:

- **Condorcet kazananı** = hiç ok almayan düğüm (D).
- **Condorcet kaybedeni** = bütün okları alan, hiç ok vermeyen düğüm (A).

Her aday en az bir ok alıyorsa, yani hiç ok almayan düğüm yoksa, Condorcet kazananı yoktur. Bu durumda çizgede bir döngü vardır: bir adaydan başlayıp okları izleyerek aynı adaya dönülebilir. Bütün adayları kapsayan bir döngü, Condorcet kazananının bulunmamasına yol açar.

Tanım 7.10

Copeland kuralında (*Copeland rule*) her adayın skoru ikili karşılaştırmalardaki galibiyet sayısıdır, yani çizgede o adaydan çıkan ok sayısıdır. Skoru en yüksek aday kazanır. (Eşitlik olabilen durumlarda beraberliklere yarım puan verilir.)

Örnek profilde Copeland skorları D 3, B 2, C 1, A 0'dır; bu skorlar Tablo 7.5'in son sütunuyla aynıdır. Copeland, D ≻ B ≻ C ≻ A sıralamasını verir ve kazanan D'dir. Copeland kuralının iyi bir özelliği, Condorcet kazananı varsa onu her zaman seçmesidir: Condorcet kazananı $m - 1$ galibiyetle en yüksek mümkün skoru alır ve başka hiçbir aday bu skora ulaşamaz, çünkü her öteki aday en az bir kez (Condorcet kazananına) yenilir. Condorcet kazananı yoksa Copeland yine de bir sıralama verir, ama bu sıralamada eşitlikler sık görülür.

7.2.9 Aynı profil, dört farklı kazanan

Bu noktaya kadar aynı 13 seçmenin oyları yedi farklı yolla sayıldı. Sonuçlar Tablo 7.6'da yan yana konulmuştur. Tablodaki Kemeny satırı 7.3.4'te hesaplanır.

Kural	Sayım	Kazanan
Mutlak çoğunluk	A 6 < 7	yok
Çoğulluk	A 6, C 4, D 3, B 0	A
İki turlu sistem	C 7 – A 6	C
Anında ikinci tur	C 7 – A 6 (3. tur)	C
Borda	B 22, D 21, A 18, C 17	B
2-onay (ilk iki)	B 9, D 7, A 6, C 4	B
Condorcet	D 3 galibiyet	D
Kemeny	D ≻ B ≻ C ≻ A (44 uyum)	D

Tablo 7.6: Aynı profil, farklı kurallar.

Tablodan iki ders çıkar. Birincisi: “halkın iradesi” kurala bağlıdır. Seçmenlerin oyları değişmediği hâlde kural değişince kazanan A, B, C ya da D olabilir; yani sonucu seçmenler kadar kural da belirler. Bu yüzden hangi kuralın kullanılacağı oylamadan önce açıkça kararlaştırılmalıdır. İkincisi: bazı kurallar akla aykırı sonuçlar verebilir. Çoğulluk, herkese yenilen Condorcet kaybedeni A'yı seçebilir; IRV ise herkesi yenen Condorcet kazananı D'yi erken eleyebilir.

Her kuralın neden farklı aday seçtiği, profilin 7.1.3'te not edilen özellikleriyle açıklanır. Çoğulluk yalnızca birinci sıraya baktığı için en çok “sadık” seçmeni olan A'yı seçer. İki tur ve IRV, birinci tercihleri güçlü olan adaylar arasında son yarışı yaptırır ve C'yi seçer. Borda ve 2-onay, ikinci sıraları da saydığı için uzlaşmacı B'yi seçer. Condorcet ve Kemeny ise ikili yarışlara baktığı için her ikili yarışı kazanan D'yi seçer.

7.2.10 Kod izleme: borda()

Oylama kurallarının sayımı birkaç satırlık programlarla yapılabilir. Aşağıdaki kısa Python fonksiyonu Borda sayımını yapar. Fonksiyonun girdisi `oylar` adlı bir listedir. Listenin her elemanı `(adet, sira)` biçiminde bir çifttir: `adet` o sıralamayı veren seçmen sayısı, `sira` ise adayları en çok tercih edilenden başlayarak dizen bir dizgidir. Örnek profil şöyle yazılır:

```
oylar = [(4, "ADBC"), (4, "CBDA"), (3, "DBCA"), (2, "ABCD")]
```

```
def borda(oylar):
  puan = {}
  for adet, sira in oylar:
    m = len(sira)
    for k, aday in enumerate(sira):
      p = adet * (m - 1 - k)
      puan[aday] = puan.get(aday, 0) + p
  kazanan = max(puan, key=puan.get)
  return kazanan, puan
```

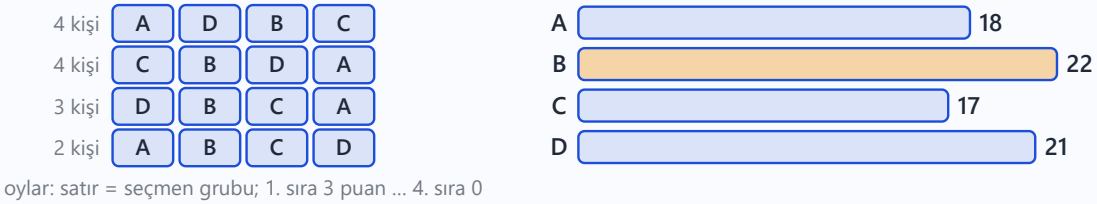
Kod şöyle çalışır:

- `puan` boş bir sözlüktür; her adayın birikmiş puanı burada tutulur.
- Dış döngü seçmen gruplarında birer kez gezinir: `adet` grubun büyüklüğü, `sira` grubun sıralamasıdır.
- `m = len(sira)` aday sayısıdır; burada 4.
- İç döngü sıralamadaki adaylarda yerleriyle birlikte gezinir. `enumerate`, sıradaki yeri 0'dan başlayarak `k`'ya verir: birinci sıra `k = 0`, sonuncu `k = m − 1`'dir.
- `p = adet * (m - 1 - k)` grubun bu adaya katkısıdır. `k = 0` için puan $m - 1 = 3$, `k = 3` için 0'dır. Bu, tanımdaki puan ölçeğinin 0'dan başlayan indisle yazılışıdır.
- `puan.get(aday, 0)`, aday sözlükte henüz yoksa 0 döndürür; böylece ilk kez görülen adayın puanı 0'dan başlar.
- `max(puan, key=puan.get)`, puanı en büyük olan anahtarı (adayı) döndürür.

Tablo 7.7 ilk grubun iç döngüsünü adım adım, öteki grupları ise toplu olarak izler.

grup	k	aday	p	puan (A, B, C, D)
4 × ADBC	0	A	4 × (3 − 0) = 12	12, –, –, –
4 × ADBC	1	D	4 × (3 − 1) = 8	12, –, –, 8
4 × ADBC	2	B	4 × (3 − 2) = 4	12, 4, –, 8
4 × ADBC	3	C	4 × (3 − 3) = 0	12, 4, 0, 8
4 × CBDA	0–3	C, B, D, A	12, 8, 4, 0	12, 12, 12, 12
3 × DBCA	0–3	D, B, C, A	9, 6, 3, 0	12, 18, 15, 21
2 × ABCD	0–3	A, B, C, D	6, 4, 2, 0	18, 22, 17, 21

Tablo 7.7: borda() izlemesi. Son sütun, adımdan sonraki puan sözlüğüdür.



Şekil 7.6: borda() izlemesinin son durumu. Solda girdi olan dört grup, sağda birikmiş puanlar; en uzun çubuk B'nin (22) çubuğudur.

Fonksiyon `("B", {"A": 18, "D": 21, "B": 22, "C": 17})` döndürür. Şekil 7.6 bu son durumu gösterir. Puanı en yüksek aday B'dir (22) ve hiçbir seçmen B'yi birinci sıraya koymamıştır.

Kodun bir inceliği eşitlik durumudur. İki aday aynı en yüksek puanı alırsa `max`, sözlükte önce gördüğünü döndürür. Bu bir oylama kuralı açısından keyfi bir seçimdir: sonucu adayların sözlüğe hangi sırayla girdiği, yani girdi listesinin düzeni belirler. Gerçek kurallarda bunun yerine önceden ilan edilmiş bir **eşitlik bozma kuralı** (*tie-breaking rule*) gerekir: kura çekmek, belirli bir aday sırası ya da ikinci bir ölçüt (örneğin birinci tercih sayısı).

Karmaşıklık açısından fonksiyon her grup için her adayı bir kez işler. n seçmen grubu ve m aday varsa iç döngü toplam $n \cdot m$ kez çalışır; çalışma süresi $O(nm)$'dir. Sayım kolaydır; zor olan, kuralın hangi özellikleri sağladığını anlamaktır.

7.2.11 Kod izleme: ikinci_tur()

Anında ikinci turun kodu Borda'dan biraz daha uzundur, çünkü sayım turlar hâlinde tekrarlanır. Fonksiyon `oylar` listesini ve aday listesini alır.

```
def ikinci_tur(oylar, adaylar):
  kalan = set(adaylar)
  while True:
    say = {a: 0 for a in kalan}
    for adet, sira in oylar:
      ilk = next(a for a in sira if a in kalan)
      say[ilk] += adet
    lider = max(say, key=say.get)
    if 2 * say[lider] > sum(say.values()):
      return lider
    kaybeden = min(say, key=say.get)
    kalan.remove(kaybeden)
```

Kod şöyle çalışır:

- `kalan`, yarışta kalan adayların kümesidir; başta bütün adaylar yarıştadır.
- `while True` döngüsünün her dönüşü bir turdur. Döngüden yalnızca `return` ile, yani bir kazanan bulununca çıkılır.

- `say`, her yarışan aday için 0'dan başlayan bir sayaçtır. Sayaçlar her turda sıfırlanır; oylar her turda baştan sayılır.
- `next(a for a in sira if a in kalan)`, sıralamayı baştan sona tarar ve *hâlâ yarışan* ilk adayı döndürür. Oyların aktarılması bu satırda kendiliğinden olur: elenen adaylar `kalan`'da olmadığı için atlanır.
- `lider`, en çok oyu alan adaydır. Koşul `2 * say[lider] > sum(say.values())`, liderin oyunun toplam oyun yarısını aşıp aşmadığını sınar. İki tarafı 2 ile çarpmak, 6,5 gibi kesirli eşiklerle uğraşmayı önler.
- Çoğunluk yoksa `min(say, key=say.get)` en az oylu adayı bulur ve `kalan.remove` onu yarıştan çıkarır.

tur	kalan	say	lider	koşul	sonuç
1	A, B, C, D	A 6, B 0, C 4, D 3	A	12 ≤ 13	B elenir
2	A, C, D	A 6, C 4, D 3	A	12 ≤ 13	D elenir
3	A, C	A 6, C 7	C	14 > 13	C döndürülür

Tablo 7.8: ikinci_tur() izlemesi (toplam oy 13; çoğunluk koşulu 2 · say > 13).

Birinci turda dört pusula grubu sırayla işlenir: ADBC grubunun yarışan ilk adayı A'dır ve `say[A] += 4`; CBDA için `say[C] += 4`; DBCA için `say[D] += 3`; ABCD için `say[A] += 2`. Sonuç A 6, B 0, C 4, D 3'tür. Lider A'dır ama *2 × 6 = 12 ≤ 13* olduğu için çoğunluk yoktur. En az oylu B (0) elenir. İkinci turda sayaçlar sıfırlanır ve sayım tekrarlanır; B'yi birinci sıraya koyan olmadığı için sayılar değişmez. D (3) elenir. Üçüncü turda DBCA grubunun taraması D'yi ve B'yi atlar ve C'de durur: `say[C] += 3`. Bu, D'nin oylarının C'ye geçmesidir. *2 × 7 = 14 > 13* olduğu için C döndürülür.



Şekil 7.7: ikinci_tur() izlemesinin son turu. Soluk hücreler elenmiş adaylardır; her grubun oyu yarışan ilk adaya sayılır. A 6, C 7; C çoğunluğu sağlar.

Sık yapılan hata

- **Eşikte "≥ yarı" kullanmak:** çift sayıda seçmende tam yarı oy alan aday çoğunluğu sağlamaz. Koşul kesin büyüklüktür: `2 * say > toplam`.
- **Eşitlikleri düşünmemek:** en az oylu iki aday eşitse `min` bunlardan birini seçer. `kalan` bir küme olduğu için hangisinin önce görüleceği program çalıştırmaları arasında değişebilir. Gerçek bir sayımda bu, önceden belirlenmiş bir eşitlik bozma kuralıyla çözülmelidir.

Her turda n grup için sıralama en fazla m adımda taranır, yani bir tur $O(nm)$ sürer. En fazla $m - 1$ tur olduğundan toplam süre $O(nm^2)$'dir. Bu da hızlı bir hesaptır; IRV'nin zorlukları sayımda değil, özelliklerindedir.

7.3 Condorcet paradoksu

Örnek profilde Condorcet kazananı vardır: D. Ama her profilde böyle bir aday bulunmaz. Böyle bir aday yoksa oylamaların sırası sonucu belirleyebilir. Kemeny kuralı bu durumda da bir toplumsal sıralama üretir.

7.3.1 Condorcet döngüsü

Dokuz seçmen ve üç adaylı şu profil ele alınsın: 4 seçmen A ≻ B ≻ C, 3 seçmen B ≻ C ≻ A, 2 seçmen C ≻ A ≻ B. Her seçmenin sıralaması tek başına tutarlıdır, yani geçişlidir.

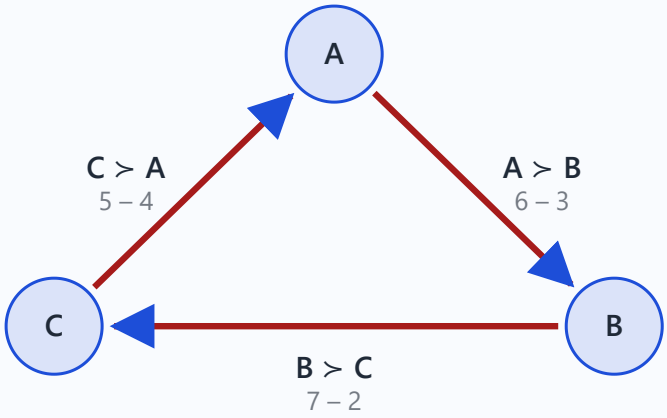
sıra	4 kişi	3 kişi	2 kişi
1.	A	B	C
2.	B	C	A
3.	C	A	B

Tablo 7.9: Döngü profili: 9 seçmen, 3 aday.

Örnek 7.9

1. **A – B:** A'yı B'ye tercih edenler 1. sütun (4) ve 3. sütun (2): $4 + 2 = 6$. B'yi tercih edenler 2. sütun: 3. Çoğunluk A ≻ B (6 – 3).
2. **B – C:** B'yi tercih edenler $4 + 3 = 7$, C'yi tercih edenler 2. Çoğunluk B ≻ C (7 – 2). Geçişlilik beklentisi: A ≻ B ve B ≻ C olduğuna göre A ≻ C olmalıdır.
3. **C – A:** C'yi A'ya tercih edenler 2. sütun (3) ve 3. sütun (2): $3 + 2 = 5$. A'yı tercih edenler 4. Çoğunluk C ≻ A (5 – 4)!

Sonuç: A ≻ B ≻ C ≻ A. Toplumsal tercih döngüseldir. Her adayı yenen bir çoğunluk vardır; Condorcet kazananı yoktur.



Şekil 7.8: Condorcet döngüsü. A, B'yi (6 – 3); B, C'yi (7 – 2); C de A'yı (5 – 4) yener. Oklar bir daire oluşturur; hiçbir düğüm ok almadan kalmaz.

Şekil 7.8 durumu ikili çoğunluk çizgesiyle gösterir. Üç ok bir daire oluşturur: her düğüme bir ok girer, bu yüzden Condorcet kazananı yoktur. Bu olguya **Condorcet paradoksu** ya da **Condorcet döngüsü** (*Condorcet paradox / cycle*) denir.

Tanım 7.11

İkili çoğunluk ilişkisi x_1'in x_2'yi, x_2'nin x_3'ü, …, x_k'nın da x_1'i yendiği bir aday dizisi içeriyorsa profilde bir **Condorcet döngüsü** vardır. Bu durumda ikili çoğunluk ilişkisi geçişli değildir.

Paradoksun önemi şudur: bireylerin tutarlı olması topluluğun tutarlı olmasını garanti etmez. “Çoğunluk A'yı B'ye tercih ediyor” ve “çoğunluk B'yi C'ye tercih ediyor” cümlelerindeki “çoğunluk” aynı insanlar değildir. İlk çoğunluk 1. ve 3. sütunlardan, ikincisi 1. ve 2. sütunlardan oluşur. Farklı çoğunluklar farklı kararlar verir ve bu kararlar bir araya getirildiğinde döngü oluşabilir.

Sık yapılan hata

Döngünün seçmenlerden birinin tutarsız olmasından kaynaklandığını sanmak. Profildeki üç sıralama da geçişlidir. Döngü, tutarlı sıralamaların çoğunluk kuralıyla *birleştirilmesinden* doğar. Bir başka hata da döngü varken “en çok ikili galibiyeti olan” adayı Condorcet kazananı saymaktır: döngüde her adayın birer galibiyeti vardır ve Condorcet kazananı tanım gereği yoktur.

7.3.2 Gündem sırası kazananı belirler

Döngünün pratik bir sonucu vardır. Pek çok komite ve meclis kararlarını **ikili elemeli oylama** (*sequential pairwise voting*) ile verir: önce iki seçenek oylanır, kazanan bir sonraki seçenekle karşılaşır ve bu, bütün seçenekler bitene dek sürer. Seçeneklerin hangi sırayla oylanacağına **gündem**

(*agenda*) denir. Döngü profilinde üç farklı gündem mümkündür (hangi çiftin önce oylanacağı seçilir; kalan aday sona kalır).

Örnek 7.10

1. **Gündem 1: önce A – B, sonra C.** A – B: 6 – 3, A kazanır. A – C: 4 – 5, C kazanır. Kazanan: C.
2. **Gündem 2: önce B – C, sonra A.** B – C: 7 – 2, B kazanır. B – A: 3 – 6, A kazanır. Kazanan: A.
3. **Gündem 3: önce A – C, sonra B.** A – C: 4 – 5, C kazanır. C – B: 2 – 7, B kazanır. Kazanan: B.

Üç gündemin üçü de farklı kazanan üretir.

Gündem	İlk oylama	Kazanan
(A – B), sonra C	A	C
(B – C), sonra A	B	A
(A – C), sonra B	C	B

Tablo 7.10: Döngü profilinde üç gündem.

Örnekteki örüntü genel bir kuraldır: üç adaylı bir döngüde *sona bırakılan aday kazanır*. Nedeni şudur: ilk oylamada iki aday karşılaşır ve kazanan, sona bırakılan adayla karşılaşır. Üç adaylı bir döngüde her aday tam bir adayı yener ve tam bir adaya yenilir. İlk oylamanın kazananı yenebildiği tek adayı ilk oylamada yenmiştir; demek ki yenildiği tek aday sona bırakılan adaydır ve son oylamayı kaybeder. Örneğin gündem 1'de A, B'yi yener; döngüde A'yı yenen aday C'dir ve C sondadır.

Bu, gündemi belirleyen kişiye (örneğin bir komite başkanına) büyük bir güç verir. Başkan, kazanmasını istediği adayı sona koyarak sonucu seçebilir; üstelik her oylama kendi içinde dürüst bir çoğunluk oylamasıdır. Bu duruma **gündem manipülasyonu** (*agenda manipulation*) denir. Condorcet kazananı olan bir profilde ise bu güç yoktur: Condorcet kazananı hangi sırayla oylanırsa oylansın karşılaştığı her adayı yener ve sonunda kazanır.

7.3.3 Oyuna dönüş: döngü ne kadar sık?

Bölümün başındaki oyunun iki sorusu artık cevaplanabilir. Oyunun birinci seviyesindeki profil (A ≻ B ≻ C, B ≻ C ≻ A, C ≻ A ≻ B) bir döngüdür: A – B 2 – 1, B – C 2 – 1, C – A 2 – 1. Toplumsal tercih döngüseldir, Condorcet kazananı yoktur. Birinci sorunun cevabı “hayır”dır: bireyler tutarlı olsa da topluluk tutarlı olmak zorunda değildir. İkinci sorunun cevabı “evet”tir: gündemi kuran kişi, kazanmasını istediği adayı sona bırakarak sonucu seçer. Gündemlerin sonuçları Tablo 7.10'daki ile aynı örüntüyü izler: (A – B) sonra C gündeminde C, (B – C) sonra A gündeminde A, (A – C) sonra B gündeminde B kazanır.

Doğal bir soru, döngülerin ne kadar sık görüldüğüdür. Bunun için bir olasılık modeli gerekir. En basit model, her seçmenin altı olası sıralamadan birini eşit olasılıkla ve ötekilerden bağımsız olarak seçmesidir. Üç seçmen ve üç aday için olası profil sayısı $6 \times 6 \times 6 = 6^3 = 216$'dır. Bu profillerin kaçında döngü olduğu sayılabilir: döngü yalnızca seçmenlerin sıralamaları A ≻ B ≻ C, B ≻ C ≻ A, C ≻ A ≻ B üçlüsü-

nün ya da bunun tersi olan A ≻ C ≻ B, C ≻ B ≻ A, B ≻ A ≻ C üçlüsünün bir dizilişi olduğunda oluşur. Her üçlü üç seçmene $3! = 6$ farklı şekilde dağıtılabilir; toplam $2 \times 6 = 12$ döngülü profil vardır. Olasılık $12/216 \approx 0{,}056$, yani yaklaşık %5,6'dır.

Seçmen sayısı çok büyüdükçe, aynı rastgele modelde üç adaylı döngü olasılığı yaklaşık %8,8'e yükselir (Guilbaud, 1952). Aday sayısı arttıkça bu oran da hızla artar. Bu sayılar endişe verici görünebilir, ama gerçek seçim ve anket verilerinde döngüler bu "rastgele tercih" hesabının öngördüğünden daha seyrek görülür (Regenwetter vd., 2006). Nedeni, gerçek tercihlerin rastgele olmamasıdır: insanların tercihleri birbirine benzer ve çoğu zaman bir ortak eksene (örneğin sol–sağ) göre düzenlenir. Bu tür düzenli tercihlerde döngünün neden oluşmadığı 7.5.6'da açıklanır.

Sık yapılan hata

%5,6 ya da %8,8 gibi oranları gerçek seçimlerde döngü görülme sıklığı olarak yorumlamak. Bu sayılar "her sıralama eşit olasılıklı" varsayımının sonucudur. Varsayım değişince sayı da değişir; gerçek veriler daha düşük oranlar gösterir.

7.3.4 Kemeny kuralı

Döngü olduğunda Condorcet fikri bir kazanan veremez. Yine de bir karar verilmesi gerekir. John Kemeny (1959) bu sorunu şöyle çözmeyi önerdi: seçmenlerin ikili tercihleriyle *en çok uyuşan* toplumsal sıralama seçilsin.

Tanım 7.12

Adayların bir σ sıralamasının **uyum skoru**, σ'da x'in y'den önce geldiği her (x, y) çifti için x'i y'ye tercih eden seçmen sayılarının toplamıdır:

$$\text{uyum}(\sigma) = \sum_{x \text{ önce } y \ (\sigma\text{'da})} N(x, y).$$

Kemeny kuralı (*Kemeny rule*) uyum skoru en büyük olan sıralamayı seçer.

Uyum skorunun bir yorumu daha vardır. Bir seçmenin sıralaması ile σ arasında bazı çiftlerde anlaşma, bazılarında anlaşmazlık vardır. Uyum skoru, bütün seçmenler üzerinden anlaşılan çift sayısının toplamıdır. Uyumu en büyük yapmak, anlaşılmayan çift sayısını (sıralamalar arasındaki toplam "uzaklığı") en küçük yapmakla aynıdır. Kemeny sıralaması, bu anlamda seçmenlerin sıralamalarına "ortalama olarak en yakın" sıralamadır.

Örnek 7.11

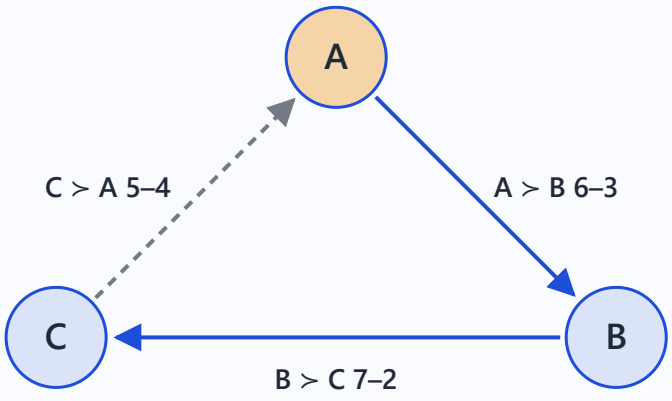
Ana profilde D ≻ B ≻ C ≻ A sıralamasının uyum skoru Tablo 7.5'ten okunur. Bu sıralamadaki altı çift ve ilgili sayılar:

- D önce B: *N(D, B) = 7*; D önce C: 7; D önce A: 7.
- B önce C: *N(B, C) = 9*; B önce A: 7.
- C önce A: *N(C, A) = 7*.

Uyum: *7 + 7 + 7 + 9 + 7 + 7 = 44*. Bu sıralamada her çift, o çiftin ikili çoğunluğuyla aynı yöndedir; her çiftin katkısı mümkün olan en büyük değerdir. Başka hiçbir sıralama 44'ü geçemez; 24 sıralamanın hepsi denendiğinde en yakın rakipler (D ≻ B ≻ A ≻ C ve B ≻ D ≻ C ≻ A) 43'te kalır.

Örnek genel bir gözlemi gösterir: ikili çoğunluk ilişkisi geçişliyse (döngü yoksa), bu ilişkiyi izleyen sıralama her çiftte en büyük katkıyı alır ve Kemeny sıralamasıdır. Özellikle Condorcet kazananı varsa Kemeny sıralamasında başa yerleşir. Bunun nedeni şudur: Condorcet kazananı bir sıralamada başta değilse, onun hemen önündeki adayla yer değiştirmesi, o çiftin katkısını küçük sayıdan büyük sayıya çıkarır ve başka hiçbir çifti değiştirmez; bu da uyumu artırır.

Döngüde de Kemeny bir karar verir. Döngü profilinde üç ikili çoğunluk birlikte sağlanamaz; en az birinin bozulması gerekir. Kemeny, bozulduğunda en az uyum kaybettiren çoğunluğu, yani *en zayıf* çoğunluğu bozar.



Şekil 7.9: Döngü profilinde Kemeny sıralaması. Kesikli ok, Kemeny'nin bozduğu en zayıf çoğunluktur (C ≻ A, 5 – 4). Kalan iki çoğunluk A ≻ B ≻ C sıralamasını verir.

Şekil 7.9'da döngünün üç kenarı ve oy sayıları görülür: A ≻ B 6 – 3, B ≻ C 7 – 2, C ≻ A 5 – 4. En dar farklı kenar C ≻ A'dır. Bu kenar bozulunca kalan iki kenar A ≻ B ≻ C sıralamasını verir. Bu sıralamanın uyumu *N(A, B) + N(A, C) + N(B, C) = 6 + 4 + 7 = 17*'dir. Altı sıralamanın tam karşılaştırması 7.3.5'teki kod izlemesindedir.

Kemeny kuralının güçlü özellikleri vardır, ama bir bedeli de vardır: en iyi sıralamayı bulmak NP-zordur (Bartholdi, Tovey ve Trick, 1989). NP-zor bir problem için bilinen bütün algoritmalar en kötü durumda girdinin büyüklüğüyle üstel olarak artan zaman alır. Aday sayısı küçükken bütün sıralamalar

denenebilir, ama 15–20 adayda bu artık pratik değildir. Kemeny kuralının önemli bir uygulaması **sıralama birleştirmedir** (*rank aggregation*; Dwork vd., 2001): birden çok arama motorunun sonuç listelerini tek listede birleştirmek, yarışma jürilerinin ve komitelerin sıralamalarını birleştirmek.

7.3.5 Kod izleme: kemeny()

Az sayıda aday için Kemeny sıralaması bütün sıralamalar denenerek bulunabilir. Aşağıdaki fonksiyon tam olarak bunu yapar. Fonksiyon, Python'un standart kütüphanesindeki `permutations` fonksiyonunu kullanır; bu yüzden programın başında `from itertools import permutations` satırı bulunmalıdır. `permutations(adaylar)`, adayların bütün sıralamalarını tek tek üretir.

```
def kemeny(oylar, adaylar):
  en_iyi, en_skor = None, -1
  for sira in permutations(adaylar):
    skor = 0
    for i, x in enumerate(sira):
      for y in sira[i + 1:]:
        skor += sum(n for n, oy in oylar
                    if oy.index(x) < oy.index(y))
    if skor > en_skor:
      en_iyi, en_skor = sira, skor
  return en_iyi, en_skor
```

Kod şöyle çalışır:

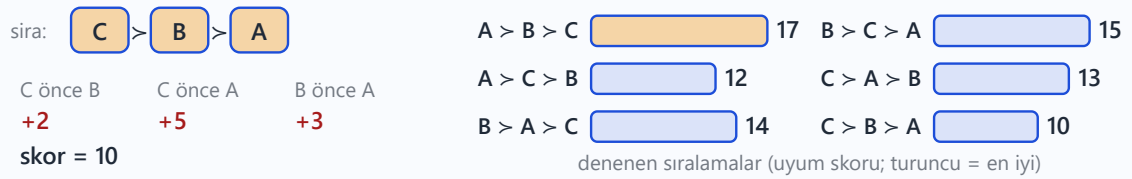
- `en_iyi` o ana kadar bulunan en iyi sıralamayı, `en_skor` onun uyum skorunu tutar. Başlangıç değeri *−1*'dir; böylece ilk denenen sıralama (skoru en az 0) her zaman en iyi kabul edilir.
- Dış döngü her `sira` sıralamasını dener.
- İki iç döngü sıralamadaki bütün *(x, y)* çiftleri üzerinde gezinir; `sira[i + 1:]` yalnızca `x`'ten sonra gelen adayları verir. Böylece her çift bir kez ve "sıralamada `x` önce" yönüyle ele alınır.
- `oy.index(x) < oy.index(y)`, o seçmen grubunun `x`'i `y`'den önce koyup koymadığını sınar. Koşulu sağlayan grupların büyüklükleri `n` toplanır; bu *N(x, y)*'dir.
- Bir sıralamanın skoru önceki en iyiden kesin büyükse en iyi güncellenir. Eşit skorlu sıralamalardan ilk bulunan korunur.

Döngü profili `oylar = [(4, "ABC"), (3, "BCA"), (2, "CAB")]` için izleme Tablo 7.11'de verilmiştir. Üç aday için *3! = 6* sıralama vardır ve her sıralamada *3 · 2 / 2 = 3* çift bulunur.

sira	1. çift	2. çift	3. çift	skor	en_iyi
A ≻ B ≻ C	A–B: 6	A–C: 4	B–C: 7	**17**	A ≻ B ≻ C (17 > −1)
A ≻ C ≻ B	A–C: 4	A–B: 6	C–B: 2	12	değişmez
B ≻ A ≻ C	B–A: 3	B–C: 7	A–C: 4	14	değişmez
B ≻ C ≻ A	B–C: 7	B–A: 3	C–A: 5	15	değişmez
C ≻ A ≻ B	C–A: 5	C–B: 2	A–B: 6	13	değişmez

sira	1. çift	2. çift	3. çift	skor	en_iyi
C ≻ B ≻ A	C–B: 2	C–A: 5	B–A: 3	10	değişmez

Tablo 7.11: kemeny() izlemesi (döngü profili). Çift sütunlarında sıralamadaki sırayla N(x, y).



Şekil 7.10: kemeny() izlemesinin son adımı. Üstte son denenen sıralama C ≻ B ≻ A ve çiftlerinin katkıları (2 + 5 + 3 = 10); altta altı sıralamanın uyum skorları, turuncu en iyisi A ≻ B ≻ C (17).

Fonksiyon `(("A", "B", "C"), 17)` döndürür. Kemeny sıralaması A ≻ B ≻ C'dir ve yalnızca en zayıf çoğunluk olan C ≻ A (5 – 4) bozulmuştur. İkinci en iyi sıralama B ≻ C ≻ A'dır (15); bu sıralama A ≻ B çoğunluğunu (6 – 3) bozar ve 2 puan daha fazla uyum kaybeder. Tablonun bir simetrisi de dikkat çeker: bir sıralamanın skoru ile tersinin skoru toplanınca her zaman $3 \times 9 = 27$ eder (17 + 10, 12 + 15, 14 + 13). Nedeni, her çiftte bir seçmenin iki sıralamadan tam biriyle anlaşmasıdır.

Bu kaba kuvvet yönteminin maliyeti hızla büyür. m aday için $m!$ sıralama denenir; her sıralamada $m(m-1)/2$ çift vardır ve her çift için n grup taranır. Toplam iş $m! \cdot m(m-1)/2 \cdot n$ düzeyindedir. 4 aday için 24, 10 aday için 3.628.800, 20 aday için yaklaşık $2{,}4 \times 10^{18}$ sıralama vardır. Kemeny'nin NP-zor olması, bu üstel büyümeden genel olarak kaçmanın bilinen bir yolunun olmadığını söyler.

7.3.6 Kurallar hangi ölçütleri sağlar?

Farklı kurallar farklı kazananlar seçer. Hangisinin “daha iyi” olduğunu tartışmak için ölçütler gerekir. Sosyal seçim kuramında kuralları karşılaştırmak için kullanılan ölçütlere *aksiyom* da denir. Dört temel ölçüt aşağıda tanımlanmış, kuralların bunları sağlayıp sağlamadığı da Tablo 7.12'de toplanmıştır.

Tanım 7.13

- **Condorcet tutarlılığı:** Condorcet kazananı varsa kural onu seçer.
- **Condorcet kaybedeni ölçütü:** kural Condorcet kaybedenini hiçbir zaman seçmez.
- **Monotonluk** (*monotonicity*): kazanan bir aday, bazı seçmenlerin pusulasında yukarı taşınırsa (öteki adayların birbirine göre sırası değişmeden) yine kazanır. Kısaca: kazananı desteklemek onu kaybettirmemelidir.
- **Stratejiye dayanıklılık** (*strategy-proofness*): hiçbir seçmen, gerçek tercihinden farklı bir sıralama bildirerek kendi gerçek tercihine göre daha iyi bir sonuç elde edemez.

Kural	Condorcet kazananını seçer	Condorcet kaybedeni seçilmez	Monoton	Stratejiye dayanıklı
Çoğulluk	hayır	hayır (A)	evet	hayır
Borda	hayır (B)	evet	evet	hayır
Anında ikinci tur	hayır (C)	evet	hayır	hayır
Onay	hayır	hayır	evet	hayır
Condorcet (Copeland)	evet	evet	evet	hayır
Kemeny	evet	evet	evet	hayır

Tablo 7.12: Kurallar ve ölçütler. Parantezdeki aday, örnek profildeki ihlali gösterir.

Tablo satır satır okunabilir. Çoğulluk örnek profilde Condorcet kazananı D'yi değil A'yı seçer; A aynı zamanda Condorcet kaybedenidir, yani çoğulluk iki Condorcet ölçütünü de bozar. Borda, Condorcet kazananı D yerine B'yi seçer. Ama Borda hiçbir zaman Condorcet kaybedenini seçmez. Bunun nedeni 7.2.7'deki gözlemdir: Borda puanı, ikili karşılaştırmalarda alınan oyların toplamıdır. Condorcet kaybedeni her karşılaştırmada oyların yarısından azını alır; bu yüzden Borda puanı, adayların ortalama Borda puanının altında kalır. Örnek profilde ortalama $78 / 4 = 19{,}5$'tir ve A'nın puanı 18'dir. Ortalamanın altında kalan bir aday en yüksek puanı alamaz.

IRV, Condorcet kazananı D'yi ikinci turda eler ve C'yi seçer. IRV Condorcet kaybedenini seçmez: son turda iki aday karşı karşıya kalır ve Condorcet kaybedeni bu son yarışı kaybeder. Ama IRV monoton değildir (7.3.7). Onay oylamasında sonuç onay eşiklerine bağlı olduğundan hiçbir Condorcet ölçütü garanti edilemez. Copeland ve Kemeny ikili karşılaştırmalara dayandıkları için iki Condorcet ölçütünü de sağlar.

Son sütundaki “hayır”ların hepsinin ortak bir nedeni vardır: üç ya da daha fazla adayda, diktatörlük dışında hiçbir makul kural stratejiye dayanıklı değildir. Bu genel sonuç Gibbard–Satterthwaite teoremidir ve 7.5.4'te ele alınır.

Condorcet yöntemlerinin bir eksiği, döngüde kazanan bulamamalarıdır. Döngüde Copeland skorları eşit çıkar (örneğin döngü profilinde üç adayın da birer galibiyeti vardır) ve bir eşitlik bozma kuralı gerekir. Bu eksiği gideren kurallardan biri maksimindir.

Tanım 7.14

Maksimin (*maximin*; Simpson) kuralında her aday için ikili karşılaştırmalarda aldığı *en düşük* oy sayısı bulunur: $skor(x) = \min_{y \neq x} N(x, y)$. Skoru en yüksek olan aday kazanır.

Örnek 7.12

Ana profilde Tablo 7.5'in her satırındaki en küçük sayı alınır:

- A: $\min(6, 6, 6) = 6$.

- B: *min(7, 9, 6) = 6*.
- C: *min(7, 4, 6) = 4*.
- D: *min(7, 7, 7) = 7*.

Kazanan D'dir. Döngü profilinde ise A'nın en kötü karşılaştırması C'ye karşı 4, B'ninki A'ya karşı 3, C'ninki B'ye karşı 2'dir; maksimin A'yı seçer. Bu, Kemeny sıralamasının başındaki adayla aynıdır.

Maksimin, Condorcet kazananı varsa onu seçer: Condorcet kazananı her karşılaştırmada oyların yarısından fazlasını alır, öteki her aday ise en az bir karşılaştırmada (Condorcet kazananına karşı) yarısından azını alır. Döngüde de karar verir: kazanan, en kötü yenilgisi en hafif olan adaydır.

7.3.7 Anında ikinci turda monotonluk ihlali

Monotonluk, bir oylama kuralından beklenecek en temel özelliklerden biri gibi görünür: bir adaya daha çok destek vermek ona zarar vermemelidir. Puanlama kuralları bu özelliği her zaman sağlar (7.2.5). IRV ise sağlamaz. 17 seçmenli bir örnek bunu gösterir.

sıra	6 kişi	6 kişi	5 kişi
1.	A	B	C
2.	B	C	A
3.	C	A	B

Tablo 7.13: Monotonluk örneği: 17 seçmen, 3 aday (çoğunluk için 9 oy gerekir).

Örnek 7.13

Kampanyadan önce (Tablo 7.13):

1. 1. tur: A = 6, B = 6, C = 5. Çoğunluk (9) yok; en az oylu C elenir.
2. C'nin 5 oyu sıradaki A'ya geçer. 2. tur: A = *6 + 5 = 11*, B = 6. A kazanır.

Kampanyadan sonra: A kampanya yapar ve B $\succ$ C $\succ$ A diyen 6 seçmenden 2'si A'yı başa alır; artık A $\succ$ B $\succ$ C derler. Profil: 8 seçmen A $\succ$ B $\succ$ C, 4 seçmen B $\succ$ C $\succ$ A, 5 seçmen C $\succ$ A $\succ$ B. A'nın desteği yalnızca artmıştır; başka hiçbir şey değişmemiştir.

1. 1. tur: A = 8, B = 4, C = 5. Çoğunluk yok; bu kez en az oylu B elenir (C değil).
2. B'nin 4 oyu sıradaki C'ye geçer. 2. tur: C = *5 + 4 = 9*, A = 8. C kazanır.

A, daha çok destek aldığı için kaybetti: IRV monoton değildir.

İhlalin mekanizması şudur. IRV'de bir adayın kazanıp kazanmaması yalnızca kendi oylarına değil, son turda *kiminle* karşılaşacağına da bağlıdır. İlk profilde A son turda B ile karşılaşır ve C'nin seçmenleri A'yı B'ye tercih ettiği için A kazanır. Kampanyada A'ya geçen 2 seçmen B'den alınmıştır; B'nin birinci tercih sayısı 6'dan 4'e düşer ve B, C'nin altına iner. Eleme sırası değişir: bu kez B elenir ve A son turda C ile karşılaşır. B'nin seçmenleri C'yi A'ya tercih ettiği için C kazanır. Yukarı taşıma, eleme sırasını değiştirmiştir.

Bu profilde ikili karşılaştırmalar da ilginçtir: ilk profilde A, B'yi 11 – 6, B, C'yi 12 – 5, C de A'yı 11 – 6 yener. Profil bir Condorcet döngüsüdür. Monotonluk ihlalleri çoğunlukla böyle, Condorcet kazananının olmadığı ya da yarışın çok yakın olduğu profillerde görülür.

Sık yapılan hata

Monotonluk ihlalini stratejik oy ile karıştırmak. Örnekteki 2 seçmen yalan söylemez; gerçekten fikir değiştirir ve A'yı daha çok desteklemeye başlar. Sorun, kuralın bu daha güçlü desteğe ters tepki vermesidir. Stratejik oy ise seçmenin gerçek tercihinden farklı bir pusula bildirmesidir.

7.4 Stratejik oy

Buraya kadar her seçmenin pusulasına gerçek tercihini yazdığı varsayıldı. Ama 7.1.2'de belirtildiği gibi pusula bir stratejidir. Bir seçmen, kuralın nasıl işlediğini ve öteki seçmenlerin nasıl oy vereceğini bildiğinde, gerçek tercihinden farklı bir pusula yazarak kendi gerçek tercihine göre daha iyi bir sonuç elde edebilir. Buna **stratejik oy** (*strategic voting*) ya da **manipülasyon** denir. Örnek profil iki tür stratejik oyu sayılarla gösterir.

7.4.1 Çoğullukta taktik oy

Dürüst oylamada çoğulluk kazananı A'dır: A 6, C 4, D 3, B 0. Bu sonuç 3. grubu ($D \succ B \succ C \succ A$) en çok rahatsız eden sonuçtur, çünkü A onların en kötü adayıdır.

Örnek 7.14

1. **Durumu okumak:** 3. grubun favorisi D yalnızca 3 oy alır; kazanma şansı yoktur. A'yı durdurabilecek tek aday, 4 oyla ikinci olan C'dir. 3. grup C'yi A'ya tercih eder (sıralamalarında C üçüncü, A dördüncü).
2. **Stratejik pusula:** 3. grup D yerine C'ye oy verir; yani pusulaya $C \succ D \succ B \succ A$ yazar.
3. **Yeni sayım:** A 6, C $4 + 3 = 7$, B 0, D 0. C kazanır.
4. **Grup için sonuç:** gerçek tercihleri $D \succ B \succ C \succ$ A'ya göre C, A'dan daha iyidir. Yalan söylemek bu grubun durumunu iyileştirmiştir.

Bu davranışa **taktik oy** (*tactical voting*) ya da "boşa oy vermeme" davranışı denir: seçmen favorisi yerine kazanma şansı olan ve daha az kötü bulduğu adaya oy verir. Taktik oy vermenin nedeni **oy bölünmesidir** (*vote splitting*): A'ya karşı olan yedi seçmenin oyları C ve D arasında bölündüğü için A, seçmenlerin azınlığının desteğiyle kazanabilmiştir.

Taktik oy vermenin uzun vadeli bir sonucu da vardır. Seçmenler küçük adaylara oy vermenin boşa gideceğini düşündükçe oylar en güçlü iki adayda toplanır. Siyaset biliminde bu gözleme **Duverger yasası** (*Duverger's law*) denir: tek turlu çoğulluk sistemi iki partili bir siyasi yapıya doğru iter. Birleşik Krallık ve ABD'deki tek üyeli seçim bölgeleri bu yasanın sık verilen örnekleridir.

7.4.2 Borda'da gömme

Borda sayımı daha çok bilgi kullanır ve oy bölünmesinden daha az etkilenir. Ama o da stratejik davranışa açıktır, üstelik farklı bir yolla. Dürüst Borda sayımında B = 22, D = 21'dir. 3. grubun favorisi D yalnızca 1 puan farkla kaybeder.

Örnek 7.15

1. **Manipülasyon:** 3. grubun 3 seçmeni asıl rakip B'yi 2. sıradan 3. sıraya iter. Gerçek sıralama $D \succ B \succ C \succ A$ iken pusula $D \succ C \succ B \succ A$ olur.
2. **Puan farkı:** her manipülatör için B 2 yerine 1 puan, C 1 yerine 2 puan alır. Üç seçmen için B -3, C $+3$. A ve D'nin puanı değişmez.
3. **Yeni toplamlar:** A 18, B $22 - 3 = 19$, C $17 + 3 = 20$, D 21.
4. **Sonuç:** D kazanır. 3. grup yalan söyleyerek favorisini seçtirdi.

aday	dürüst	fark	stratejik
A	18	0	18
B	**22**	−3	19
C	17	+3	20
D	21	0	**21**

Tablo 7.14: Borda'da gömme: dürüst ve stratejik sayım.

Bu manipülasyona **gömme** (*burying*) denir: seçmen, favorisinin asıl rakibini pusulasında gerçek yerinden daha aşağıya "gömer". Borda'da gömme çok etkilidir, çünkü her sıra farkı doğrudan puan farkına dönüşür. Bir rakibi en alta itmek, ona verilen puanı olabildiğince azaltır.

Gömmenin bir riski vardır: geri tepebilir. Rakibi aşağı itmek, onun yerine yukarı çıkan başka bir adayın puanını artırır. Bu başka aday gömen seçmenler için daha kötü bir adaysa ve yeterince yükselirse kazanan o olur. Örneğin 9 seçmenli şu profile bakalım: $4 \times A \succ B \succ C$, $3 \times C \succ B \succ A$, $2 \times B \succ C \succ A$. Dürüst Borda (2, 1, 0) puanları A 8, B 11, C 8'dir ve B kazanır. A seçmenleri B'yi gömüp $A \succ C \succ B$ bil-

dirirse C 4 puan kazanır, B 4 puan kaybeder: A 8, B 7, C 12 ve C kazanır. C, A seçmenlerinin en kötü adayıdır; gömme onların durumunu kötüleştirmiştir. Başarılı bir gömme, öteki seçmenlerin oyları hakkında doğru bilgi gerektirir.

> **Sık yapılan hata**
>
> Manipülasyonun başarısını manipülatörün favorisinin kazanıp kazanmadığıyla ölçmek. Ölçüt, sonucun manipülatörün *gerçek* tercihine göre iyileşip iyileşmediğidir. Çoğulluk örneğinde 3. grubun favorisi D kazanmaz, ama sonuç A'dan C'ye geçtiği için grup için iyileşir; bu da başarılı bir manipülasyondur.

7.4.3 Stratejik oy sistemi ne kadar bozar?

Stratejik davranışın etkisi kurala göre farklıdır:

- **Çoğulluk:** oy bölünmesi temel sorundur. Benzer iki aday oyları paylaşır, üçüncüsü kazanır. Taktik oy yaygın ve basittir; seçmenin yalnızca kimin önde olduğunu bilmesi yeter.
- **Borda:** rakibi en alta gömmek çok etkilidir. Herkes gömerse bildirilen sıralamalar seçmenlerin gerçek tercihlerinden uzaklaşır ve sonuç, kimsenin gerçekten istemediği bir aday olabilir.
- **Anında ikinci tur:** manipülasyonu hesaplamak daha zordur ama mümkündür; ayrıca kural monoton değildir. “IRV stratejik oya kapalıdır” yaygın ama yanlış bir kanıdır.
- **Hesaplama bariyeri:** bazı kurallarda (örneğin IRV ve Copeland'in bir türevinde) kârlı bir manipülasyonu bulmak NP-zordur (Bartholdi vd., 1989–91). Bu zorluk, manipülasyonu imkânsız değil ama en kötü durumda hesaplama açısından pahalı kılar.

Stratejik oy vermenin etkisi tercih dağılımına bağlıdır. Yakın yarışlarda birkaç seçmenin stratejik davranışı kazananı değiştirebilir; büyük farklı yarışlarda ise küçük grupların manipülasyonu sonuç üzerinde etkisizdir. Şeffaf kurallar ve anlaşılır bilgi bazı manipülasyonları azaltır, ama hiçbir genel kural stratejik davranışı her durumda ortadan kaldıramaz. Bu son cümle bir gözlem değil, bir teoremdir (7.5.4).

7.5 İmkânsızlık teoremleri

Önceki kısımlarda her kuralın bir kusuru bulundu: çoğulluk Condorcet kaybedenini seçebilir, Borda manipülasyona açıktır, IRV monoton değildir, Condorcet yöntemleri döngüde karar veremez. Doğal soru şudur: bütün bu kusurlardan arınmış, daha akıllıca tasarlanmış bir kural var mıdır? Bu kısımdaki teoremlerin cevabı, üç ya da daha fazla aday olduğunda “hayır”dır. Bu sonuçlar, kural aramayı bırakmayı değil, hangi özellikten ödün verileceğini bilinçli olarak seçmeyi öğretir.

7.5.1 Arrow'un koşulları

Kenneth Arrow (1951) soruyu çok genel bir biçimde sordu. Toplumsal sıralama üreten bir kural, yani bir sosyal refah fonksiyonu (SWF) düşünülsün. Böyle bir kuraldan beklenebilecek, ilk bakışta çok makul görünen dört koşul vardır.

Tanım 7.15

Bir SWF F için:

- **Evrensel alan** (*unrestricted domain*): her tercih profili kabul edilir ve sonuç her zaman tam ve geçişli bir sıralamadır.
- **Pareto (oybirliği)** (*Pareto, unanimity*): herkes x'i y'ye tercih ediyorsa toplum da x'i y'ye tercih eder.
- **İlgisiz alternatiflerden bağımsızlık** (*independence of irrelevant alternatives, IIA*): x ile y'nin toplumsal sırası yalnızca seçmenlerin x–y sırasına bağlıdır. İki profilde her seçmen x ile y'yi aynı sırayla diziyorsa, toplumsal sıralama da iki profilde x ile y'yi aynı sırayla dizer.
- **Diktatör yok** (*non-dictatorship*): tercihi her profilde toplumun tercihi olan tek bir seçmen yoktur.

Koşulların her birinin neden makul olduğu kısaca şöyle açıklanabilir. Evrensel alan, kuralın seçmenlerin ne düşündüğünü önceden kısıtlamamasını ve her durumda tutarlı bir sonuç vermesini ister; Condorcet döngüsü bu koşulun “geçişli sonuç” kısmını çoğunluk sıralaması için bozar. Pareto, oybirliğine saygıdır: bütün seçmenler aynı şeyi istiyorsa toplumun başka türlü karar vermesi için hiçbir neden yoktur. IIA, iki aday arasındaki kararın üçüncü bir adayın varlığından ya da yerinden etkilenmemesini ister; örneğin bir restoranda “balık ile tavuk arasında balığı seçerim” diyen birinin, menüye bir de makarna eklenince tavuğa geçmesi tuhaf olurdu. Diktatör yok koşulu ise en temel adalet beklentisidir.

Teorem 7.1 (Arrow, 1951)

En az 3 aday varsa evrensel alan, Pareto, IIA ve diktatör yok koşullarını birlikte sağlayan hiçbir sosyal refah fonksiyonu yoktur. Başka bir deyişle, evrensel alan, Pareto ve IIA'yı sağlayan her SWF bir diktatörlüktür.

Teoremin ispatı bu notun kapsamı dışındadır; okunabilir ispatlar İleri okuma kısmındaki kaynaklarda bulunur. Teorem iki aday için geçerli değildir: iki adayda basit çoğunluk dört koşulu da sağlar. Bu durum 7.5.5'teki May teoreminin konusudur.

Sık yapılan hata

IIA'yı "kazanan olmayan bir aday çekilirse kazanan değişmemeli" biçiminde anlamak. Bu yaygın ama eksik bir yorumdur. Arrow'un IIA'sı aday kümesi sabitken de bozulabilir: seçmenler yalnızca *başka* adayların yerlerini değiştirdiğinde x–y toplumsal sırasının değişmesi de IIA ihlalidir. 7.5.2'deki örnek iki türü de gösterir.

7.5.2 Borda ve ilgisiz alternatifler

Borda sayımı evrensel alanı sağlar (her profilde puanlar toplanır ve bir sıralama çıkar), Pareto'yu sağlar (herkes x'i y'nin üstüne koyuyorsa her seçmen x'e daha çok puan verir) ve diktatörlük değildir. Arrow teoremine göre IIA'yı bozmak zorundadır. Örnek profil bu bozulmayı açıkça gösterir.

Örnek 7.16

1. **Dört adayla:** Borda B = 22 > D = 21. Toplum B'yi D'nin önüne koyar.
2. **C yarıştan çekilir.** C ne kazanıyordu ne de ikinciydi; B–D sıralaması açısından "ilgisiz" bir adaydır. Sıralamalardan C silinir: $4 \times A \succ D \succ B$, $4 \times B \succ D \succ A$, $3 \times D \succ B \succ A$, $2 \times A \succ B \succ D$.
3. **Üç aday için Borda (2, 1, 0):**
   - A: $4 \times 2 + 0 + 0 + 2 \times 2 = 12$.
   - B: $4 \times 0 + 4 \times 2 + 3 \times 1 + 2 \times 1 = 13$.
   - D: $4 \times 1 + 4 \times 1 + 3 \times 2 + 2 \times 0 = 14$.
4. **Sonuç:** D = 14 > B = 13. Toplumsal sıra döndü.

Hiçbir seçmenin B–D sırası değişmedi: B'yi D'ye tercih eden 6, D'yi B'ye tercih eden 7 seçmen vardır, önce de sonra da. Ama toplumsal sıralama B ≻ D'den D ≻ B'ye döndü. Borda IIA'yı ihlal eder.

aday	4 aday	C yok
A	18	12
B	**22**	13
C	17	—
D	21	**14**

Tablo 7.15: C çekilince Borda puanları (4 aday: 3, 2, 1, 0; 3 aday: 2, 1, 0).

Neden böyle olur? 7.2.3'teki yoruma göre bir adayın Borda puanı, seçmenlerin onun altına koyduğu adayların sayısıdır. C çekilince her aday, C'yi kendi altına koyan her seçmenden 1 puan kaybeder. C'yi B'nin altına koyan seçmenler G1, G3 ve G4'tür: *4 + 3 + 2 = 9* kişi; B *22 − 9 = 13*'e iner. C'yi D'nin altına koyanlar yalnızca G1 ve G3'tür: 7 kişi; D *21 − 7 = 14*'e iner. A'nın altına C'yi koyanlar G1 ve G4'tür: 6 kişi; A *18 − 6 = 12*'ye iner. C, B'ye D'den daha çok "ara puan" kazandırıyordu; C çekilince bu fark kaybolur. Borda puanı, iki aday arasındaki *mesafeyi* (aralarında kaç aday olduğunu) ölçer; bu mesafe de öteki adaylara bağlıdır. IIA ise yalnızca iki adayın göreli sırasına bakılmasını ister.

IIA, aday kümesi değişmeden de bozulabilir. 4 kişilik $A \succ D \succ B \succ C$ grubu yalnızca C ile B'nin yerini değiştirsin ve $A \succ D \succ C \succ B$ desin. Bu seçmenlerin B–D sırası değişmez (D hâlâ B'nin önündedir). Ama B'nin puanı her biri için 1'den 0'a düşer: B 22'den *22 − 4 = 18*'e iner. D 21'de kalır; C ise *17 + 4 = 21*'e çıkar. B–D toplumsal sırası yine döner. Hiçbir seçmenin B–D sırası değişmediği hâlde B ile D'nin toplumsal sırası değişmiştir.

7.5.3 Arrow teoreminin pratik anlamı

Arrow teoremi sık sık yanlış yorumlanır. Söyledikleri ile söylemedikleri ayrılmalıdır.

- **Söylediği:** toplu karar kuralları bütün makul ölçütleri aynı anda sağlayamaz; her kural bir koşuldan ödün verir.
- **Söylemediği:** oylamanın anlamsız olduğunu söylemez. Belirli bir kuralı yasaklamaz ya da önermez. Kuralların çoğu profilde makul sonuç verdiğini de inkâr etmez; yalnızca bir koşulun *bazı* profillerde bozulacağını söyler.
- **Kim neyden ödün verir?** Borda IIA'dan ödün verir (7.5.2). Condorcet ya da çoğunluk sıralaması geçişlilikten, dolayısıyla evrensel alanın "geçişli sonuç" şartından ödün verir (döngü). Diktatörlük ise adaletten ödün verir.
- **Kurumlar için:** hangi ölçüte öncelik verildiği açıkça belirlenmeli ve belgelenmelidir. Bir kural seçmek, hangi kusurun kabul edileceğini seçmektir.

Teoremden kaçış yolları koşulları gevşetmektir. Birincisi tercih alanını kısıtlamaktır: bütün profillere değil, yalnızca belirli bir yapıdaki profillere izin verilirse olumlu sonuçlar mümkündür. Bunun en önemli örneği tek tepeli tercihlerdir (7.5.6). İkincisi sıralama yerine yoğunluk bilgisi kullanmaktır. Arrow'un çerçevesinde seçmen yalnızca sıralama bildirir; "A'yı B'ye biraz tercih ederim" ile "A'yı B'ye çok tercih ederim" arasında fark yoktur. Puanlamalı oylama ya da para gibi yoğunluk bilgisi taşıyan araçlar bu çerçevenin dışındadır.

7.5.4 Gibbard–Satterthwaite teoremi

Arrow teoremi toplumsal sıralamalar hakkındadır ve stratejik davranışla doğrudan ilgilenmez. Stratejik oy sorusunu ise Allan Gibbard (1973) ve Mark Satterthwaite (1975) birbirinden bağımsız olarak yanıtladı. Teoremin ifadesi için iki koşul gerekir.

Tanım 7.16

Bir sosyal seçim fonksiyonu (SCF) f için:

- **Örtenlik** (*surjectivity, onto*): her aday, en az bir profilde kazanır.
- **Stratejiye dayanıklılık**: her seçmen i, her profil ve i'nin her yanlış pusulası $\succ'_i$ için $f(\succ_i, \succ_{-i})$ sonucu, i'nin gerçek tercihi $\succ_i$'ye göre $f(\succ'_i, \succ_{-i})$'den kötü değildir. Burada $\succ_{-i}$, öteki seçmenlerin pusulalarıdır.

Teorem 7.2 (Gibbard 1973, Satterthwaite 1975)

En az 3 aday varsa, her profilde tek kazanan seçen, örten ve stratejiye dayanıklı her sosyal seçim fonksiyonu bir diktatörlüktür.

Stratejiye dayanıklılık tanımı oyun kuramı diliyle şöyle okunur: dürüst pusula her seçmen için *baskın strateji*dir. Öteki seçmenler ne bildirirse bildirsin, seçmenin gerçeği söylemesi en az yalan söylemesi kadar iyidir. Teorem, bu güçlü özelliğin üç ya da daha fazla adayda yalnızca diktatörlükte mümkün olduğunu söyler. Diktatörlük gerçekten stratejiye dayanıklıdır: diktatörün birinci tercihi kazanır, diktatör yalan söylemekle bir şey kazanmaz, öteki seçmenlerin pusulası ise sonucu hiç etkilemez.

Örnek profil teoremi somutlaştırır. Çoğullukta 3. grubun C'ye verdiği taktik oy (7.4.1) ve Borda'da B'nin gömülmesi (7.4.2) bu iki kuralın stratejiye dayanıklı olmadığının kanıtlarıdır. Teoreme göre bu, yalnız bu iki kuralın kusuru değildir: diktatörlük dışındaki *her* örten kural bir profilde manipüle edilebilir.

Arrow ile ilişkisi: Arrow teoremi sıralamalar, Gibbard–Satterthwaite tek kazanan hakkındadır. İki teorem yakından bağlıdır; Gibbard–Satterthwaite teoremi Arrow teoreminden türetilebilir. Mekanizma tasarımı açısından anlamı şudur: parasız ortamda, üç ve daha fazla seçenek arasında seçim yapan genel bir teşvik uyumlu kural yoktur. Para kullanılabildiğinde durum değişir; ödemeler, yalan söylemenin bedelini ayarlamaya izin verir ve VCG mekanizması gibi stratejiye dayanıklı mekanizmalar mümkün olur.

Bu yüzden tasarımın hedefi stratejik davranışı yok etmek değil, sınırlandırmaktır. Bunun üç yolu sık kullanılır: seçmenlerin öteki oylar hakkında az şey bilmesi (manipülasyon için profil bilgisi gerekir), manipülasyonu hesaplamanın zor olması (7.6.1) ve tercih alanının kısıtlanması (7.5.6).

Sık yapılan hata

Teoremden “her seçmen her zaman yalan söyler” sonucunu çıkarmak. Teorem yalnızca, kuralın manipüle edilebildiği *en az bir* profil bulunduğunu söyler. Çoğu profilde ve çoğu seçmen için dürüst oy en iyi seçenek olabilir. Bir başka hata da teoremin iki adayda da geçerli olduğunu sanmaktır; iki adayda çoğunluk kuralı stratejiye dayanıklıdır.

7.5.5 May teoremi: iki aday için basit çoğunluk

Arrow ve Gibbard–Satterthwaite teoremlerinin ikisi de en az üç aday gerektirir. İki aday olduğunda durum çok daha iyidir. Kenneth May (1952), iki aday (x ve y) varken hangi kuralın "adil" olduğunu sordu ve üç koşulu sağlayan tek kuralın basit çoğunluk olduğunu gösterdi. **Basit çoğunlukta** (*simple majority*) daha çok oy alan aday kazanır; oylar eşitse sonuç beraberliktir.

Tanım 7.17

İki adaylı bir kural için:

- **Anonimlik** (*anonymity*): seçmenlerin isimleri önemsizdir; oylar seçmenler arasında yer değiştirse sonuç aynı kalır.
- **Tarafsızlık** (*neutrality*): adayların isimleri önemsizdir; her seçmen x ile y'yi yer değiştirirse sonuç da yer değiştirir.
- **Pozitif duyarlılık** (*positive responsiveness*): sonuç beraberlikse ya da x kazanıyorsa ve bir seçmen oyunu x lehine değiştirirse (ötekiler aynı kalırken), x kesin kazanır.

Teorem 7.3 (May, 1952)

İki aday için bir kural, ancak ve ancak basit çoğunluksa anonimlik, tarafsızlık ve pozitif duyarlılığı sağlar.

Teoremin bir yönü kolayca görülür: basit çoğunluk üç koşulu da sağlar. Anonimdir, çünkü sonuç yalnızca her adayın kaç oy aldığına bağlıdır; hangi seçmenin hangi oyu verdiği sayıları değiştirmez. Tarafsızdır, çünkü her seçmen x ile y'yi yer değiştirince iki adayın oy sayıları da yer değiştirir ve kazanan da yer değiştirir. Pozitif duyarlıdır, çünkü beraberlikte bir seçmenin x'e geçmesi x'i bir oy öne geçirir; x zaten öndeyse farkı büyütür. Teoremin asıl gücü öteki yöndedir: bu üç koşulu sağlayan başka bir kural yoktur. Her koşulun neden gerekli olduğu, onu bozan kurallarla görülür.

Koşul	Anlamı	İhlal eden kural
Anonimlik	Seçmenlerin isimleri önemsiz; oylar yer değiştirse sonuç aynı	Diktatör; ağırlıklı oy
Tarafsızlık	Adayların isimleri önemsiz; iki aday simetrik muamele görür	"Statüko ⅔ olmadan değişmez" (nitelikli çoğunluk)
Pozitif duyarlılık	Beraberlikte ya da x kazanırken bir seçmen x'e geçerse x kesin kazanır	"Her zaman yazı-tura"; "tek sayıda oy alan kazanır"

Tablo 7.16: May teoreminin koşulları ve onları ihlal eden kurallar.

Örnek 7.17

Bir kooperatifte 9 üye mevcut yönetmeliği (statüko, x) yeni bir yönetmelikle (y) değiştirmeyi oyluyor. Kural: "yönetmelik ancak oyların en az ⅔'siyle değişir." 9'un ⅔'si 6'dır.

- 5 üye y, 4 üye x der: y 6'ya ulaşamaz; x kalır.
- Şimdi her üye oyunu tersine çevirsin: 5 üye x, 4 üye y. x yine kalır.

Tarafsızlık, oylar tersine dönünce sonucun da tersine dönmesini (y'nin kazanmasını) isterdi. Kural, statükoyu kayırdığı için tarafsızlığı ihlal eder. Kural anonimdir (kimin oy verdiği önemsizdir).

Nitelikli çoğunluk (*qualified majority*) kuralları anayasa değişikliği gibi önemli kararlarda bilinçli olarak kullanılır; amaç, statükoyu kolayca değiştirilemez kılmaktır. May teoremi bunun bir bedeli olduğunu söyler: böyle bir kural iki seçeneğe eşit davranmaz. "Her zaman yazı-tura" kuralı ise oylara hiç tepki vermez; "tek sayıda oy alan kazanır" kuralında bir seçmenin x'e geçmesi x'in oy sayısını çiftten teke ya da tekten çifte çevirerek onu kaybettirebilir.

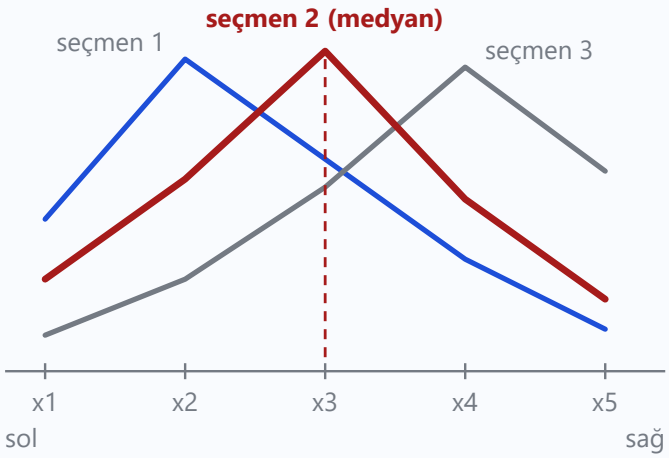
May teoreminin mesajı şudur: iki adayda kural tartışması biter, basit çoğunluk açık ara en doğal seçimdir. Zorluklar (döngü, Arrow, Gibbard–Satterthwaite) üç ve daha fazla adayla başlar.

7.5.6 Kaçış yolu: tek tepeli tercihler

Arrow ve Gibbard–Satterthwaite teoremlerinin ortak varsayımı evrensel alandır: her olası tercih profiline izin verilir. Ama pek çok gerçek durumda tercihler rastgele değildir; belirli bir yapıları vardır. Bu yapılardan en önemlisi tek tepeliliktir.

Tanım 7.18

Adaylar bir eksen üzerinde soldan sağa sıralı olsun (örneğin siyasi sol–sağ, bütçe miktarı ya da bir caddedeki konum). Seçmenin en çok sevdiği bir konum (**tepesi**) varsa ve aday bu tepeden hangi yöne uzaklaşırsa uzaklaşsın seçmenin faydası azalıyorsa, seçmenin tercihi **tek tepelidir** (*single-peaked*). Bütün seçmenlerin tercihi aynı eksene göre tek tepeliyse profil tek tepelidir.



Şekil 7.11: Sol–sağ ekseninde x1–x5 konumları ve üç seçmenin tek tepeli fayda eğrileri. Seçmen 1'in tepesi x2, seçmen 2'ninki x3, seçmen 3'ünkü x4'tür. Medyan tepe x3'tür (seçmen 2).

Şekil 7.11'de her eğrinin tek bir zirvesi vardır ve eğri zirveden iki yana doğru alçalır. Örneğin bir belediye meclisinin parka ayıracağı bütçeyi oyladığı düşünülsün. Her üyenin aklında ideal bir miktar vardır; ondan çok fazla ya da çok az olan teklifler onu daha az memnun eder. Bu tür tercihler doğal olarak tek tepelidir.

Teorem 7.4 (Medyan seçmen teoremi; Black, 1948)

Tercihler tek tepeli ve seçmen sayısı tekse, tepeleri sıralandığında ortadaki seçmenin (medyan seçmenin) tepesi Condorcet kazananıdır. Dolayısıyla Condorcet döngüsü oluşmaz.

İspat. Seçmen sayısı $n = 2k + 1$ ve medyan tepe t olsun. Tanım gereği tepesi t'de ya da t'nin solunda olan en az $k + 1$ seçmen, tepesi t'de ya da sağında olan da en az $k + 1$ seçmen vardır. t'nin sağındaki bir aday z ele alınsın. Tepesi t'de ya da solunda olan bir seçmen için hem t hem z tepenin sağındadır ve t tepeye daha yakındır. Tek tepelilik gereği bu seçmen t'yi z'ye tercih eder. Bu seçmenler en az $k + 1$ kişidir, yani çoğunluktur; t, z'yi yener. t'nin solundaki bir aday için aynı akıl yürütme tepesi t'de ya da sağında olan seçmenlerle yapılır. Demek ki t her rakibini yener ve Condorcet kazananıdır. □

Teoremin iki önemli sonucu vardır. Birincisi, tek tepeli profillerde Condorcet döngüsü olmaz; bu da gerçek tercihlerin çoğu zaman bir eksene göre düzenli olmasının döngüleri neden seyrek kıldığını açıklar. İkincisi, “tepeleri bildir, medyanı seç” kuralı stratejiye dayanıklıdır. Bir seçmen yanlış bir tepe bildirerek medyanı ancak kendi tepesinden *uzaklaştırabilir*. Tepesi medyanın solunda olan bir seçmen, tepesini daha sola bildirirse medyan değişmez; sağa, medyanın ötesine bildirirse medyan sağa, yani ondan uzağa kayar. Kendi tepesine yaklaştıracak hiçbir yalan yoktur.

Örnek 7.18

Beş seçmenin tepeleri eksende 2, 3, 7, 8, 9 olsun. Sıralı listenin ortasındaki (üçüncü) değer 7'dir; medyan tepe 7'dir.

- 7 ile solundaki bir konum (örneğin 5) karşılaştırılınca tepeleri 7, 8, 9 olan 3 seçmen 7'yi seçer; 7, 3 – 2 kazanır.
- 7 ile sağındaki bir konum (örneğin 8,5) karşılaştırılınca tepeleri 2, 3, 7 olan 3 seçmen 7'yi seçer; 7 yine 3 – 2 kazanır.

Tepelerin ortalaması *(2 + 3 + 7 + 8 + 9)/5 = 5,8*'dir, ama ortalama Condorcet kazananı değildir: 5,8 ile 7 karşılaştırılınca tepeleri 7, 8, 9 olan 3 seçmen 7'yi seçer. Tepesi 2 olan seçmen tepesini 0 diye bildirirse sıralı liste 0, 3, 7, 8, 9 olur ve medyan yine 7'dir; yalan bir şey kazandırmaz.

Medyan seçmen teoremi Gibbard–Satterthwaite teoremiyle çelişmez; onun varsayımlarından birini, evrensel alanı ortadan kaldırır. Gibbard–Satterthwaite teoremi bütün profiller için konuşur; alan tek tepeli profillerle kısıtlanınca teorem artık uygulanmaz. Mekanizma tasarımında tesis yerleşimi için kullanılan **medyan mekanizması** (*median mechanism*) bu fikrin uygulamasıdır: bir kütüphane ya da hastane bir caddeye yerleştirilecekse ve herkes en yakın konumu istiyorsa, bildirilen konumların medyanı seçilir ve kimse yanlış konum bildirerek kazanç sağlayamaz.

Sık yapılan hata

- **Medyan yerine ortalama almak:** ortalama uç değerlerden etkilenir ve stratejiye dayanıklı değildir; bir seçmen tepesini abartarak ortalamayı kendine çekebilir.
- **Her tercihi tek tepeli sanmak:** tek tepelilik bir varsayımdır. Örneğin "ya çok büyük bir bütçe ya hiç bütçe; orta yol en kötüsü" diyen bir seçmenin tercihi tek tepeli değildir. Böyle seçmenler varsa medyan teoremi uygulanamaz.

7.5.7 Dört sonuç yan yana

Bu kısımdaki dört sonuç Tablo 7.17'de karşılaştırılmıştır.

Teorem	Varsayımlar	Sonuç	Kaçış yolu
Arrow (1951)	≥ 3 aday, evrensel alan, Pareto, IIA	Kural diktatörlüktür	Tercih alanını kısıtlamak (tek tepeli); IIA'dan vazgeçmek; sıralama yerine kardinal bilgi (puan, yoğunluk)
Gibbard–Satterthwaite (1973, 1975)	≥ 3 aday, evrensel alan, örten (her aday kazanabilir), stratejiye dayanıklı	Kural diktatörlüktür	Tek tepeli tercihler; rastgele kurallar (ör. rastgele diktatör); para (VCG)
May (1952)	2 aday; anonimlik, tarafsızlık, pozitif duyarlılık	Kural basit çoğunluktur (ve tersi)	Gerekmez: imkânsızlık değil, karakterizasyon

Teorem	Varsayımlar	Sonuç	Kaçış yolu
Tek tepeli tercihler (Black, 1948)	Adaylar bir eksende, her seçmenin tek tepesi var, tek sayıda seçmen	Medyan seçmenin tepesi Condorcet kazananıdır; medyan kuralı stratejiye dayanıklıdır	Kendisi bir kaçış yoludur: evrensel alan varsayımını kısıtlar

Tablo 7.17: Dört sonucun varsayımları, sonuçları ve kaçış yolları.

Tablodaki “rastgele diktatör” kuralı açıklamaya değer: seçmenlerden biri kurayla seçilir ve onun birinci tercihi kazanır. Bu kural stratejiye dayanıklıdır (seçilen seçmen için yalan söylemek yalnızca kendi favorisini kaybettirir) ve hiçbir seçmen önceden diktatör değildir. Gibbard–Satterthwaite teoremi yalnızca belirlenimci (rastgelelik içermeyen) kurallar için geçerli olduğu için bu kural teoremin dışında kalır.

Arrow ve Gibbard–Satterthwaite teoremlerinde diktatörlük, varsayımların hepsi birlikte istendiğinde ortaya çıkar. Bir varsayım gevşeyince olumlu sonuçlar mümkün olur: iki aday (May), tek tepeli tercihler (Black), rastgelelik ya da para. Bu teoremlerin pratik değeri, tasarımcıya hangi kapıların kapalı, hangilerinin açık olduğunu göstermeleridir.

7.6 Hesaplamalı sosyal seçim ve uygulamalar

Sosyal seçim bilgisayar bilimiyle buluşur ve gerçek kurumlarda karşılık bulur. Kuralların hesaplama maliyeti, önermeler üzerinde toplu karar, gerçek seçimlerde kullanılan kurallar ve birden çok sandalyenin dağıtıldığı nispi temsil bu buluşmanın başlıca konularıdır.

7.6.1 Hesaplamalı sosyal seçim

Hesaplamalı sosyal seçim (*computational social choice*), bilgisayar bilimi, ekonomi ve sosyal seçimin kesişimidir. Bu alanda kurallar yalnızca hangi ölçütleri sağladıklarına göre değil, hesaplama maliyetine göre de incelenir. Üç tür soru öne çıkar: kazananı hesaplamak ne kadar zordur, manipülasyonu hesaplamak ne kadar zordur ve kazananı belirlemek için seçmenlerden ne kadar bilgi toplamak gerekir?

Kazananı hesaplamak. Çoğulluk ve Borda, n seçmen ve m aday için $O(nm)$ sürede hesaplanır; her pusula bir kez okunur (7.2.10). Copeland için önce ikili karşılaştırma tablosu doldurulur: $m(m-1)/2$ çiftin her biri için n pusula taranır; süre $O(nm^2)$'dir. Buna karşılık Kemeny kuralında ve Dodgson kuralında kazananı bulmak NP-zordur (Bartholdi, Tovey ve Trick, 1989). **Dodgson kuralını** (*Dodgson rule*) *Alice Harikalar Diyarında*'nın yazarı Lewis Carroll (gerçek adıyla Charles Dodgson) önermiştir: bir adayı Condorcet kazananı yapmak için pusulalarda en az sayıda *komşu yer değiştirmesi* (sıralamada yan yana iki adayın yerini değiştirmek) gereken aday kazanır. Condorcet kazananı varsa ona sıfır değişiklik gerekir ve o kazanır. Fikir doğaldır, ama bu en küçük sayıyı hesaplamak genel olarak çok zordur.

Burada bir ironi vardır: Condorcet ölçütlerini sağlayan ve döngüde de makul karar veren bazı kurallar (Kemeny, Dodgson), hesaplama açısından en pahalı kurallardır. Pratikte küçük aday sayılarında bu bir sorun değildir; büyük aday sayılarında ise yaklaşık algoritmalar ve tamsayılı programlama çözücüleri kullanılır.

Manipülasyonu hesaplamak. Gibbard–Satterthwaite teoremi manipülasyonun mümkün olduğunu söyler, ama kârlı bir yalan pusulayı *bulmanın* ne kadar zor olduğu ayrı bir sorudur. Tek bir manipülatör, çoğulluk ve Borda'da kârlı pusulayı açgözlü bir algoritmayla polinom sürede bulur: favorisini başa koyar, kalan sıraları da favorisini en az tehdit edecek biçimde doldurur. IRV'de ise bu sorun NP-zordur (Bartholdi ve Orlin, 1991). Bu zorluk, manipülasyona karşı bir "bariyer" sayılabilir. Ama bariyer yalnızca en kötü durum içindir: NP-zor bir problemin bazı örnekleri zor olsa da gerçek seçimlerde karşılaşılan örneklerin çoğu kolay çözülebilir.

Tercih çıkarımı. Bir seçmenin tam sıralamasını öğrenmek ne kadar bilgi gerektirir? m aday için $m!$ farklı sıralama vardır. Her evet/hayır sorusu (bir bit) olası sıralamaların sayısını en iyi durumda yarıya indirir. Bu yüzden tam sıralama için en az $\log_2(m!)$ bit gerekir. $m = 10$ için $10! = 3.628.800$ ve $\log_2(3.628.800) \approx 21{,}8$, yani yaklaşık 22 bittir. Büyük seçimlerde ya da çok adaylı çevrim içi sistemlerde bu iletişim maliyeti önemlidir. **Tercih çıkarımı** (*preference elicitation*), seçmenlere adım adım sorular sorup yalnızca kazananı belirlemeye yetecek bilgiyi toplamayı inceler. Örneğin çoğullukta yalnızca birinci tercih yeterlidir; tam sıralamayı sormak gereksizdir.

Alanın iki konusu daha vardır: **adil bölüşüm** (bölünebilir ya da bölünemez kaynakları kıskançlık yaratmadan paylaştırmak) ve **yargı birleştirme** (7.6.2).

7.6.2 Yargı birleştirme: doktrin paradoksu

Oylama yalnızca adaylar arasında seçim yapmak için kullanılmaz. Bir mahkeme heyeti, bir yönetim kurulu ya da bir bilim kurulu çoğu zaman *önermeler* hakkında karar verir: "sözleşme geçerli mi?", "sözleşme ihlal edildi mi?". Bireysel yargıları toplu bir yargıya dönüştürmeye **yargı birleştirme** (*judgment aggregation*) denir.

Üç yargıçlı bir mahkeme düşünülsün. Davalı, ancak p (sözleşme geçerli) ve q (sözleşme ihlal edildi) önermelerinin ikisi de doğruysa tazminat öder. Sonuç önermesi $p \wedge q$'dur ("p ve q").

Yargıç	p	q	Sonuç: p ∧ q
1. yargıç	evet	evet	evet
2. yargıç	evet	hayır	hayır
3. yargıç	hayır	evet	hayır
Çoğunluk	evet (2–1)	evet (2–1)	hayır (1–2)

Tablo 7.18: Doktrin paradoksu: üç yargıç, iki gerekçe, bir sonuç.

Her yargıç kendi içinde tutarlıdır: 1. yargıç iki gerekçeyi de kabul eder ve tazminata hükmeder; 2. ve 3. yargıçlar gerekçelerden birini reddeder ve tazminata karşı çıkar. Ama önermeler tek tek oylanınca çoğunluk "p doğru, q doğru, $p \wedge q$ yanlış" der. Bu mantıksal olarak tutarsız bir toplu yargıdır. Bu duruma **doktrin paradoksu** (*doctrinal paradox, discursive dilemma*; Kornhauser ve Sager, 1986) denir.

Paradoksun pratik sonucu, kararın usule bağlı olmasıdır. Mahkeme *gerekçelere göre* karar verirse (önce p ve q ayrı ayrı oylanır, sonuç bunlardan çıkarılır) davalı öder. *Sonuca göre* karar verirse (doğrudan $p \wedge q$ oylanır) davalı ödemez. Aynı yargılardan iki farklı karar çıkar. Bu durum, Condorcet döngüsünün yargılardaki karşılığıdır: bireyler tutarlıdır, çoğunluk tutarsızdır.

List ve Pettit (2002) bu gözlemi bir teoreme dönüştürdü: anonim, sistematik (her önerme aynı yöntemle ve yalnızca o önermeye verilen oylara göre karara bağlanır) ve her zaman tutarlı bir toplu yargı üreten bir birleştirme kuralı yoktur. Bu, Arrow teoreminin yargılar için karşılığıdır.

7.6.3 Gerçek dünyada oylama kuralları

Bu bölümdeki kuralların çoğu gerçek kurumlarda kullanılır:

- **İki turlu sistem:** Türkiye ve Fransa'da cumhurbaşkanlığı seçimleri. İlk turda mutlak çoğunluk sağlanamazsa ilk iki aday ikinci turda yarışır.
- **Anında ikinci tur:** Avustralya Temsilciler Meclisi seçimleri ve bazı ABD şehirleri. Seçmen tam sıralama verir.
- **Çoğulluk:** Birleşik Krallık ve ABD'de tek üyeli seçim bölgeleri. Her bölgede en çok oyu alan aday seçilir.
- **Borda benzeri puanlama:** Eurovision şarkı yarışmasında her ülke beğendiği şarkılara 12, 10, 8, 7, … puan verir; spor ödüllerinde (örneğin en değerli oyuncu oylamalarında) da benzer puanlama kullanılır.
- **Onay:** bazı bilimsel dernek seçimleri; "uygun olduğum saatler" anketleri.
- **Çevrim içi:** Reddit ve forum oylamaları, topluluk anketleri, sınıf ve kulüp kararları.

Eurovision'un puan vektörü (12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0, …) bir pozisyonel puanlama kuralıdır. Birinci sıraya ikinciden 2 puan fazla verilmesi, birinci tercihi Borda'dan biraz daha çok ödüllendirmek içindir. Bu tasarım seçimi de 7.2.5'teki tartışmanın bir örneğidir: puan vektörü, uzlaşmacı adaylar ile güçlü bir taban desteği olan adaylar arasındaki dengeyi belirler.

7.6.4 Birden çok sandalye: nispi temsil ve D'Hondt

Buraya kadarki kurallar tek bir kazanan seçer. Meclis seçimlerinde ise bir seçim çevresinden birden çok, örneğin S sandalye dağıtılır. Bu durumda amaç bir kazanan bulmak değil, sandalyeleri partiler arasında paylaştırmaktır. **Nispi temsil** (*proportional representation*) ilkesine göre her partinin sandalye payı oy payına yakın olmalıdır. Sandalyeler tamsayı olduğu için tam orantı genellikle mümkün değildir ve yuvarlamanın nasıl yapılacağı bir yöntem seçimidir.

Tanım 7.19

D'Hondt yöntemi (*D'Hondt method*): her partinin oyu 1, 2, 3, … ile bölünür. Bütün partilerin bütün bölümleri tek bir listede büyükten küçüğe sıralanır; en büyük S bölüm birer sandalye kazanır. Bir partinin sandalye sayısı, bu S bölüm arasındaki kendi bölümlerinin sayısıdır.

Türkiye'de milletvekili seçimlerinde her seçim çevresinde D'Hondt yöntemi uygulanır; buna ek olarak ülke genelinde bir baraj vardır.

Örnek 7.19

6 sandalye; oylar A 42.000, B 31.000, C 15.000, D 12.000 (toplam 100.000).

1. Her partinin oyu 1, 2, 3, 4'e bölünür (Tablo 7.19).
2. Bütün bölümler büyükten küçüğe dizilir: 42.000 (A ÷ 1), 31.000 (B ÷ 1), 21.000 (A ÷ 2), 15.500 (B ÷ 2), 15.000 (C ÷ 1), 14.000 (A ÷ 3), 12.000 (D ÷ 1), …
3. En büyük altı bölüm sandalye alır: A'nın üç, B'nin iki, C'nin bir bölümü. Yedinci sıradaki D'nin 12.000'i dışarıda kalır.

Sonuç: A 3, B 2, C 1, D 0.

Bölen	A	B	C	D
÷1	**42.000**	**31.000**	**15.000**	12.000
÷2	**21.000**	**15.500**	7.500	6.000
÷3	**14.000**	10.333	5.000	4.000
÷4	10.500	7.750	3.750	3.000
Sandalye	3	2	1	0

Tablo 7.19: D'Hondt bölümleri. Vurgulu hücreler en büyük altı bölümdür.

Bölme işleminin bir yorumu vardır: bir partinin k. bölümü, o partinin zaten $k-1$ sandalyesi varken bir sandalye daha alırsa “sandalye başına düşen oy” miktarıdır. Her sandalye, o an sandalye başına en çok oyu temsil edecek partiye verilir. Sonuçta A %42 oyla sandalyelerin %50'sini alır; D %12 oyla hiç sandalye alamaz. D'Hondt büyük partileri hafifçe kayırır.

Başka bir yöntem **en büyük kalan** (*largest remainder*; Hare) yöntemidir. Her partinin “hak ettiği” sandalye sayısı, oy payı × S olarak hesaplanır: A $0{,}42 \times 6 = 2{,}52$, B $0{,}31 \times 6 = 1{,}86$, C $0{,}15 \times 6 = 0{,}90$, D $0{,}12 \times 6 = 0{,}72$. Her parti önce bu sayının tam kısmı kadar sandalye alır: A 2, B 1, C 0, D 0; toplam 3. Kalan 3 sandalye, küsuratı en büyük olan partilere birer birer verilir: C (0,90), B (0,86), D (0,72). A'nın küsuratı (0,52) en küçüğüdür. Sonuç: A 2, B 2, C 1, D 1. Aynı oylarla D'Hondt A'ya 3, en büyük kalan yöntemi ise D'ye 1 sandalye verir.

Tek kazananlı kurallardaki gibi burada da her yöntem bir ödünleşimdir. D'Hondt hükümet kararlılığını, yani büyük partilerin çoğunluk kurabilmesini öne çıkarır; en büyük kalan yöntemi küçük partilerin temsilini öne çıkarır.

7.6.5 Hangi oylama kuralı adil?

Bölümün bütün sonuçları bu soruya tek bir cevap olmadığını gösterir. Adalet değerlendirmesi temsil, çoğunluk desteği, azınlıkların korunması ve stratejiye dayanıklılık gibi ölçütlere bağlıdır ve bu ölçütler birbiriyle çatışır.

- **Çoğulluk** basittir ve hızlıdır, ancak oy bölünmesine ve çoğunluk desteği olmayan bir kazanana yol açabilir (örnek profilde A, oyların 6/13'üyle).
- **Borda** daha fazla tercih bilgisi kullanır ve uzlaşmacı adayı öne çıkarır (B), ancak stratejik sıralamaya, özellikle gömmeye açıktır.
- **Anında ikinci tur** çoğunluk desteği arar, fakat monotonluk ve stratejik oy sorunlarını tamamen çözmez.
- **Condorcet yöntemleri** ikili çoğunluğa saygı gösterir, ama döngüde ek bir kural gerektirir; Kemeny gibi döngüde de karar verenlerin hesaplanması zordur.

Kural seçimi kararın bağlamına ve öncelikli ölçütlere göre yapılmalıdır. Bir kulüp başkanlığı için hızlı ve anlaşılır bir çoğulluk yeterli olabilir; aday sayısı fazla olan bir sınıf etkinliği seçimi için sıralamalı bir kural (Borda ya da bir Condorcet yöntemi) daha dengeli sonuç verebilir. Önemli olan, kuralın oylamadan önce açıkça belirlenmesi ve hangi kusurların kabul edildiğinin bilinmesidir.

7.7 Özet

- **Sosyal seçim**, bireysel tercihleri (tercih profili) toplu bir karara (SCF) ya da toplumsal bir sıralamaya (SWF) dönüştüren kuralları inceler. Oy pusulası seçmenin stratejisidir; oylama parasız bir mekanizma tasarımı problemidir.
- **Kurallar:** mutlak çoğunluk, çoğulluk, iki turlu sistem, Borda, onay, pozisyonel puanlama, anında ikinci tur, Condorcet, Copeland, maksimin ve Kemeny. 13 seçmenli örnek profilde bu kurallar dört farklı kazanan verir: çoğulluk A, iki tur ve IRV C, Borda ve 2-onay B, Condorcet ve Kemeny D.
- **Condorcet paradoksu:** geçişli bireysel tercihler döngüsel bir toplumsal tercih üretebilir ($A \succ B \succ C \succ A$). Döngüde gündemi kontrol eden sonucu seçer; sona bırakılan aday kazanır. Kemeny en zayıf çoğunluğu bozar, ama hesaplanması NP-zordur.
- **Ölçütler:** çoğulluk Condorcet kaybedenini seçebilir; Borda ve IRV Condorcet kazananını seçmeyebilir; IRV monoton değildir; Copeland ve Kemeny Condorcet tutarlıdır.
- **Stratejik oy:** çoğullukta taktik oy (oy bölünmesinden kaçmak), Borda'da gömme. Dürüstlük her kuralda en iyi strateji değildir; gömme geri de tepebilir.
- **Arrow:** en az 3 adayda evrensel alan, Pareto, IIA ve diktatör yok koşulları birlikte sağlanamaz. Borda IIA'yı bozar.

- **Gibbard–Satterthwaite:** en az 3 adayda stratejiye dayanıklı ve örten her kural diktatörlüktür.
- **May ve tek tepelilik:** 2 adayda basit çoğunluk, anonimlik, tarafsızlık ve pozitif duyarlılığı sağlayan tek kuraldır. Tek tepeli tercihlerde medyan seçmenin tepesi Condorcet kazananıdır ve medyan kuralı stratejiye dayanıklıdır.
- **Hesaplamalı sosyal seçim:** çoğulluk ve Borda $O(nm)$, Copeland $O(nm^2)$; Kemeny ve Dodgson NP-zor; IRV'de manipülasyon NP-zor. Doktrin paradoksu, döngünün yargılardaki karşılığıdır. D'Hondt büyük partileri hafifçe kayırır.

7.8 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

sosyal seçim *social choice* · tercih profili *preference profile* · oy pusulası *ballot* · geçişlilik *transitivity* · sosyal refah fonksiyonu *social welfare function (SWF)* · mutlak çoğunluk *majority rule* · çoğulluk *plurality rule* · iki turlu sistem *two-round (runoff) system* · Borda sayımı *Borda count* · onay oylaması *approval voting* · pozisyonel puanlama kuralı *positional scoring rule* · k-onay *k-approval* · 2-onay *2-approval* · veto kuralı *veto rule* · anında ikinci tur *instant-runoff voting (IRV)* · Condorcet kazananı *Condorcet winner* · Condorcet kaybedeni *Condorcet loser* · ikili çoğunluk çizgesi *pairwise majority graph* · turnuva *tournament* · Copeland kuralı *Copeland rule* · eşitlik bozma kuralı *tie-breaking rule* · Condorcet paradoksu / döngüsü *Condorcet paradox / cycle* · ikili elemeli oylama *sequential pairwise voting* · gündem (manipülasyonu) *agenda (manipulation)* · Kemeny kuralı *Kemeny rule* · sıralama birleştirme *rank aggregation* · monotonluk *monotonicity* · maksimin (Simpson) kuralı *maximin (Simpson) rule* · stratejik oy *strategic voting* · taktik oy *tactical voting (plurality)* · oy bölünmesi *vote splitting* · Duverger yasası *Duverger's law* · gömme *burying* · Arrow'un imkânsızlık teoremi *Arrow's impossibility theorem* · evrensel alan *unrestricted domain* · Pareto (oybirliği) koşulu *Pareto (unanimity) condition* · ilgisiz alternatiflerden bağımsızlık *independence of irrelevant alternatives (IIA)* · diktatör yok *non-dictatorship* · Gibbard–Satterthwaite teoremi *Gibbard–Satterthwaite theorem* · örtenlik *surjectivity (onto)* · May teoremi *May's theorem* · basit çoğunluk *simple majority* · anonimlik *anonymity* · tarafsızlık *neutrality* · pozitif duyarlılık *positive responsiveness* · nitelikli çoğunluk *qualified majority* · tek tepeli tercihler *single-peaked preferences* · medyan seçmen teoremi *median voter theorem* · medyan mekanizması *median mechanism* · hesaplamalı sosyal seçim *computational social choice* · Dodgson kuralı *Dodgson rule* · tercih çıkarımı *preference elicitation* · yargı birleştirme *judgment aggregation*

doktrin paradoksu *doctrinal paradox* | nispi temsil *proportional representation* | D'Hondt yöntemi *D'Hondt method* | en büyük kalan (Hare) yöntemi *largest remainder (Hare) method*

7.9 Kendini Dene

1. 11 seçmen: 5 × A ≻ B ≻ C, 4 × B ≻ C ≻ A, 2 × C ≻ B ≻ A. Mutlak çoğunluk, çoğulluk, IRV, Borda (2, 1, 0) ve Condorcet kazananları kimdir? Profilde Condorcet kaybedeni var mı?

   Cevap

   Birinci tercihler A 5, B 4, C 2. Eşik 6'dır; $5 < 6$ olduğundan mutlak çoğunluk kazananı yoktur. Çoğulluk A'yı seçer. IRV: C (2) elenir, oyları sıradaki B'ye geçer; B $4 + 2 = 6$, A 5; B kazanır. Borda: A $5 \times 2 = 10$, B $5 \times 1 + 4 \times 2 + 2 \times 1 = 15$, C $4 \times 1 + 2 \times 2 = 8$; B kazanır. İkili: B – A 6 – 5, B – C 9 – 2, C – A 6 – 5. B Condorcet kazananıdır. A her iki rakibine 5 – 6 yenilir: Condorcet kaybedenidir ve çoğulluk tam olarak onu seçmiştir.

2. Aynı profilde $s = (1, a, 0)$ puanlama kuralları ($0 \le a \le 1$) ele alınsın. Her adayın puanı a cinsinden nedir? Hangi a değerlerinde A kazanır? C hiç kazanabilir mi?

   Cevap

   A hiçbir sıralamada ikinci değildir: A = 5. B, 4 seçmenin birinci, $5 + 2 = 7$ seçmenin ikinci tercihidir: B = $4 + 7a$. C, 2 seçmenin birinci, 4 seçmenin ikinci tercihidir: C = $2 + 4a$. A'nın B'yi geçmesi için $5 > 4 + 7a$, yani $a < 1/7$ gerekir ($a = 1/7$'de A ile B eşittir). Bu aralıkta A, C'yi de geçer ($5 > 2 + 4a$). C hiçbir a için kazanamaz, çünkü her $a \ge 0$ için $4 + 7a > 2 + 4a$'dır.

3. Aynı profilde dürüst çoğulluk kazananı A'dır. (a) C ≻ B ≻ A diyen 2 seçmen çoğullukta nasıl oy vererek kendi tercihlerine göre daha iyi bir sonuç alır? (b) Borda (2, 1, 0) kullanılıyorsa, A seçmenleri asıl rakip B'yi gömüp A ≻ C ≻ B bildirirse ne olur?

   Cevap

   (a) C'nin (2 oy) kazanma şansı yoktur. Bu iki seçmen B'ye oy verirse B $4 + 2 = 6$, A 5 olur ve B kazanır. Gerçek sıralamaları C ≻ B ≻ A'ya göre B, A'dan iyidir: taktik oy kârlıdır. (b) Her A seçmeni için B 1 yerine 0, C 0 yerine 1 puan alır: B $15 - 5 = 10$, C $8 + 5 = 13$, A 10. C kazanır. C, A seçmenlerinin en kötü adayıdır; gömme geri tepmiştir.

4. 7 seçmen: 3 × A ≻ B ≻ C, 2 × B ≻ C ≻ A, 2 × C ≻ A ≻ B. İkili karşılaştırmaların sonuçları, Kemeny sıralaması ve maksimin kazananı nedir? İkili elemeli oylamada B'nin kazanması için gündem nasıl

kurulmalıdır?

Cevap

A – B: *3 + 2 = 5* – 2, A ≻ B. B – C: *3 + 2 = 5* – 2, B ≻ C. C – A: *2 + 2 = 4* – 3, C ≻ A. Bir döngü vardır. Uyum skorları: A ≻ B ≻ C *5 + 3 + 5 = 13*, B ≻ C ≻ A 11, C ≻ A ≻ B 11, A ≻ C ≻ B 10, B ≻ A ≻ C 10, C ≻ B ≻ A 8. Kemeny sıralaması A ≻ B ≻ C'dir; en zayıf çoğunluk C ≻ A (4 – 3) bozulur. Maksimin: A'nın en kötü karşılaştırması 3 (C'ye karşı), B'ninki 2, C'ninki 2; A kazanır. B'nin kazanması için B sona bırakılır: önce A – C (C 4 – 3 kazanır), sonra C – B (B 5 – 2 kazanır).

5. İki seçenekli (*x* ve *y*) kararlar için üç kural önerilir: (i) "Başkanın oyu iki sayılır, en çok oy alan kazanır"; (ii) "*y*'nin kabulü için oyların en az ⅗'ü gerekir, yoksa *x* kalır"; (iii) "*x*'e verilen oy sayısı çiftse *x*, tekse *y* kazanır". Her biri May teoreminin hangi koşulunu ihlal eder?

Cevap

(i) Anonimlik: başkan ile başka bir üyenin oyları yer değiştirince sonuç değişebilir. (ii) Tarafsızlık: iki seçenek simetrik değildir. Örneğin 11 üyede 6 üye *y*, 5 üye *x* derse *6/11 < 3/5* olduğundan *x* kalır. Oylar ters çevrilince (6 üye *x*, 5 üye *y*) yine *x* kalır; sonuç ters dönmez. (iii) Pozitif duyarlılık: *x* kazanırken bir seçmenin *x*'e geçmesi sayıyı tekleştirir ve *x*'i kaybettirir.

6. Tek tepeli tercihli 7 seçmenin tepeleri bir eksende 1, 1, 2, 4, 6, 9, 10'dur. Condorcet kazananı hangi konumdur? Tepelerin ortalaması nedir? "Tepeleri bildir, medyanı seç" kuralında tepesi 10 olan seçmen tepesini 20 ya da 3 diye bildirirse sonuç ne olur?

Cevap

Medyan, sıralı listenin dördüncü değeridir: 4. Medyan seçmen teoremine göre Condorcet kazananı 4'tür. Ortalama *33/7 ≈ 4,71*'dir; 4 ile 4,71 karşılaştırılınca tepeleri 1, 1, 2, 4 olan 4 seçmen 4'ü seçer. Seçmen 20 bildirirse liste 1, 1, 2, 4, 6, 9, 20 olur ve medyan yine 4'tür; abartma işe yaramaz. 3 bildirirse liste 1, 1, 2, 3, 4, 6, 9 olur ve medyan 3'e iner; sonuç bu seçmenin tepesinden daha da uzaklaşır. Yalan söylemek kazandırmaz.

7. 7 sandalye; oylar A 47.000, B 33.000, C 20.000. D'Hondt ve en büyük kalan (Hare) yöntemleriyle sandalye dağılımı nedir?

Cevap

D'Hondt bölümleri: A 47.000, 23.500, 15.667, 11.750; B 33.000, 16.500, 11.000; C 20.000, 10.000. En büyük yedi bölüm 47.000 (A), 33.000 (B), 23.500 (A), 20.000 (C), 16.500 (B), 15.667 (A), 11.750 (A): A 4, B 2, C 1. Hare: hak edilen sayılar A *0,47 × 7 = 3,29*, B 2,31, C 1,40; tam kısımlar 3, 2, 1 (toplam 6). Kalan 1 sandalye en büyük küsurata (C, 0,40) gider: A 3, B 2, C 2. D'Hondt büyük partiye bir sandalye fazla verir.

8. 8 adaylı bir Kemeny sayımı kaba kuvvetle yapılacaktır. Kaç sıralama denenir ve her sıralamada kaç çift incelenir? Bir seçmenin tam sıralamasını öğrenmek için en az kaç bit gerekir?

Cevap

$8! = 40.320$ sıralama denenir. Her sıralamada $8 \cdot 7 / 2 = 28$ çift vardır; toplam $40.320 \times 28 = 1.128.960$ çift değerlendirmesi (her biri n grup taraması) gerekir. Tam sıralama $\log_2(8!) = \log_2(40.320) \approx 15,3$, yani en az 16 bit gerektirir (bit sayısı tamsayı olmalıdır).

7.10 İleri okuma

- F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia (ed.), *Handbook of Computational Social Choice*, Cambridge University Press, 2016. Hesaplamalı sosyal seçimin temel kaynağıdır; Kemeny, Dodgson, manipülasyonun karmaşıklığı ve yargı birleştirme bölümleri bu notun konularını ayrıntılandırır.
- D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010, Bölüm 23 (Voting). Arrow teoreminin kısa ve okunabilir bir ispatını ve tek tepeli tercihleri az matematikle anlatır.
- A. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017. Sosyal seçim ve oylama bölümü Arrow ve Gibbard–Satterthwaite teoremlerinin ispatlarını verir.
- N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007, Bölüm 9. Sosyal seçimi mekanizma tasarımı çerçevesine yerleştirir ve Gibbard–Satterthwaite teoreminin ispatını içerir.

7.11 Kaynaklar

1. K. J. Arrow, *Social Choice and Individual Values*, Wiley, 1951.
2. M. Balinski, H. P. Young, *Fair Representation*, Yale University Press, 1982.
3. J. J. Bartholdi, J. B. Orlin, “Single transferable vote resists strategic voting”, *Social Choice and Welfare* 8(4), 1991.
4. J. J. Bartholdi, C. A. Tovey, M. A. Trick, “The computational difficulty of manipulating an election”, *Social Choice and Welfare* 6, 1989.
5. J. J. Bartholdi, C. A. Tovey, M. A. Trick, “Voting schemes for which it can be difficult to tell who won the election”, *Social Choice and Welfare* 6, 1989.
6. D. Black, “On the rationale of group decision-making”, *Journal of Political Economy* 56(1), 1948.
7. F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia (ed.), *Handbook of Computational Social Choice*, Cambridge University Press, 2016.
8. C. Dwork, R. Kumar, M. Naor, D. Sivakumar, “Rank aggregation methods for the web”, *WWW*, 2001.
9. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010 (Bölüm 23).

10. A. Gibbard, “Manipulation of voting schemes: a general result”, *Econometrica* 41(4), 1973.

11. G.-Th. Guilbaud, “Les théories de l'intérêt général et le problème logique de l'agrégation”, *Économie appliquée* 5(4), 1952.

12. A. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017.

13. J. G. Kemeny, “Mathematics without numbers”, *Daedalus* 88(4), 1959.

14. L. A. Kornhauser, L. G. Sager, “Unpacking the court”, *Yale Law Journal* 96(1), 1986.

15. C. List, P. Pettit, “Aggregating sets of judgments: an impossibility result”, *Economics and Philosophy* 18(1), 2002.

16. K. O. May, “A set of independent necessary and sufficient conditions for simple majority decision”, *Econometrica* 20(4), 1952.

17. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 9).

18. M. Regenwetter, B. Grofman, A. A. J. Marley, I. M. Tsetlin, *Behavioral Social Choice*, Cambridge University Press, 2006.

19. M. A. Satterthwaite, “Strategy-proofness and Arrow's conditions”, *Journal of Economic Theory* 10, 1975.

KISIM III

Ağlar ve Verimlilik

Bölüm 8–9

KISIM III · BÖLÜM 8

Ağ Tasarımı ve Yönlendirme

Bir yolun ne kadar hızlı olduğu, o yolu kaç kişinin seçtiğine bağlıdır. Bu basit gözlem, trafik ve iletişim ağlarındaki rota seçimini bir oyuna dönüştürür: her kullanıcı kendi süresini kısaltmaya çalışır, ama seçimi ötekilerin süresini de değiştirir. Bu not, ağ ve yönlendirme kavramlarından başlayarak bencil yönlendirmeyi, Wardrop dengesini ve sistem optimumunu, yeni bir yolun herkesi yavaşlatabildiğini gösteren Braess paradoksunu, tıkanıklık oyunlarını ve Rosenthal potansiyelini, oyuncuların ağın kendisini kurduğu ağ oluşturma oyunlarını ve ağlarda verimlilik ile adalet arasındaki gerilimi adım adım anlatır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum08`

ÖĞRENME ÇIKTILARI

Wardrop dengesini bulmak
İki yollu bir ağda gecikmeleri eşitleyerek dengeyi hesaplamak; sistem optimumunu marjinal maliyetlerle bulup dengeyle karşılaştırmak.

Braess paradoksunu açıklamak
Yeni bir yolun herkesin süresini neden artırabildiğini sayısal olarak göstermek.

Potansiyel ile düşünmek
Rosenthal potansiyelini hesaplamak; en iyi tepki dinamiğinin neden durduğunu kanıtlamak.

Ağ oluşturmayı modellemek
Bağlantı kuran oyuncuların hangi ağlarda dengeye geldiğini kontrol etmek.

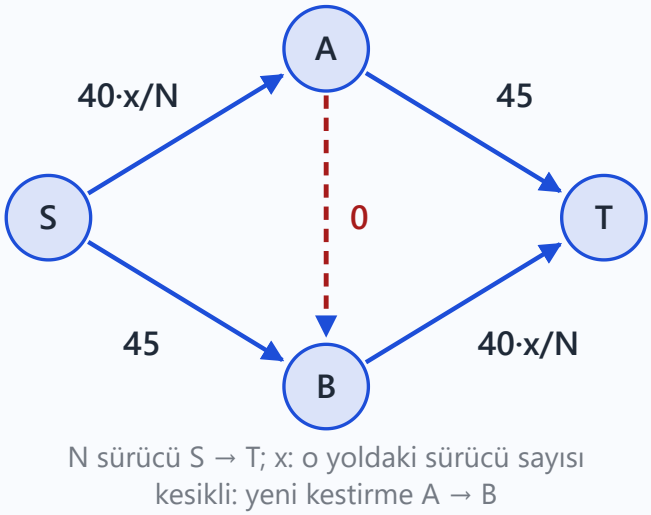
8.1 Ağlar ve yönlendirme

Bu bölümün bütün kavramları tek bir gerilimin etrafında döner: bir ağın kullanıcıları kendi yolculuklarını kısaltmak ister, ama her kullanıcının seçimi ötekilerin yolculuğunu da uzatır. Küçük bir oyun bu gerilimi somutlaştırır. Ağ, bağlantı, kapasite ve gecikme gibi temel kavramlar ise sonraki bütün hesapların dilidir.

8.1.1 Bir oyunla başlangıç: Braess paradoksu

N sürücü, S noktasından T noktasına gitmek ister. İki rota vardır: S→A→T (üst yol) ve S→B→T (alt yol). Her rotada bir dar yol ve bir geniş yol bulunur. Dar yolların süresi o yolu kullanan sürücü sayısıyla artar: dar yolda *x* sürücü varsa yolculuk *40 · x / N* dakika sürer. Bütün sürücüler aynı dar yola girerse süre 40 dakikaya çıkar; hiç kimse yoksa yol boştur. Geniş yolların süresi ise trafikten bağımsızdır, her zaman 45 dakikadır.

Oyunun kuralı basittir: her sürücü kendi en hızlı yolunu seçer. Kimse ötekilerin süresini önemsemez; herkes yalnız kendi yolculuğuna bakar. Bu, "bencil" bir davranıştır ama kötü niyetli değildir. Sabah işe giden bir sürücü de navigasyon uygulamasının önerdiği en hızlı rotayı seçer; başkalarının yolculuğunu düşünmesi beklenmez.



Şekil 8.1: Braess paradoksu oyununun ağı. S→A ve B→T dar yollardır (süre *40 · x / N*), A→T ve S→B geniş yollardır (sabit 45 dakika). Kesikli kırmızı ok, süresi 0 olan yeni A→B kestirmesidir.

Şekil 8.1 oyunun ağını gösterir. Kestirme yokken iki rota birbirinin aynası gibidir. Sürücüler iki rotaya eşit bölünürse her dar yolda *N/2* sürücü olur ve dar yol *40 · (N/2)/N = 20* dakika sürer. Her rota böylece 20 + 45 = 65 dakikadır. Oyun iki soru sorar:

- A'dan B'ye süresi 0 olan bir kestirme açılırsa yolculuklar kısalır mı?
- Herkes kendi en hızlı yolunu seçince ortalama süre en küçük olur mu?

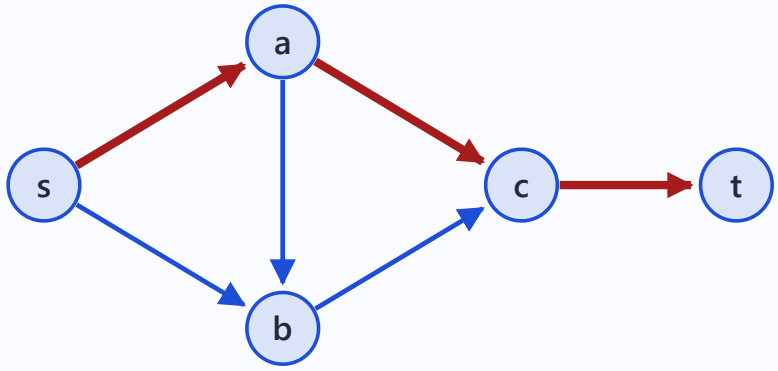
Sağduyu iki soruya da "evet" der. Yeni bir yol kapasiteyi artırır; daha çok kapasite daha az tıkanıklık demek gibi görünür. Herkesin en hızlı yolu seçmesi de "en hızlı" sonucu vermelidir. Bu bölümün en şaşırtıcı sonucu, iki cevabın da "hayır" olabileceğidir. Kestirme açıldığında her sürücü zikzak rotaya (S→A→B→T) geçer ve herkesin süresi 65 dakikadan 80 dakikaya çıkar. Hesabın bütün adımları 8.3.1'de verilir.

8.1.2 Ağ tasarımı ve yönlendirme

Bir ağla ilgili kararlar iki düzeyde verilir. Birinci düzey ağın kendisini kurmaktır: hangi noktalar birbirine bağlanacak, her bağlantı ne kadar trafik taşıyabilecek, kurulum ne kadara mal olacak? İkinci düzey kurulmuş ağı kullanmaktır: bir noktadan ötekine giden trafik hangi yollardan akacak?

Tanım 8.1

Ağ tasarımı (*network design*), ağın düğümlerini, bağlantılarını, bu bağlantıların kapasitelerini ve maliyetlerini belirleme problemidir. **Yönlendirme** (*routing*), verilen bir ağda akışın kaynak ile hedef arasında hangi yolları kullanacağını belirleme problemidir.



Şekil 8.2: Tasarım ile yönlendirmenin farkı. Tasarım, hangi kenarların hangi kapasiteyle var olacağını seçer; yönlendirme, kurulmuş ağda akışın izleyeceği yolu (kırmızı: s→a→c→t) seçer.

Şekil 8.2'de beş düğümlü küçük bir ağ görülür. Tasarımcı “a ile b arasında bir bağlantı olsun mu, olacaksa ne kadar geniş olsun?” diye sorar. Yönlendirme ise ağ kurulduktan sonra “s'den t'ye giden bir paket ya da araç hangi kenarlardan geçsin?” sorusudur. İki problem birbirine bağlıdır: iyi bir tasarım, kullanıcıların o ağda nasıl yol seçeceğini hesaba katmalıdır. Braess paradoksu tam olarak bu bağın ihmal edilmesinin bedelidir.

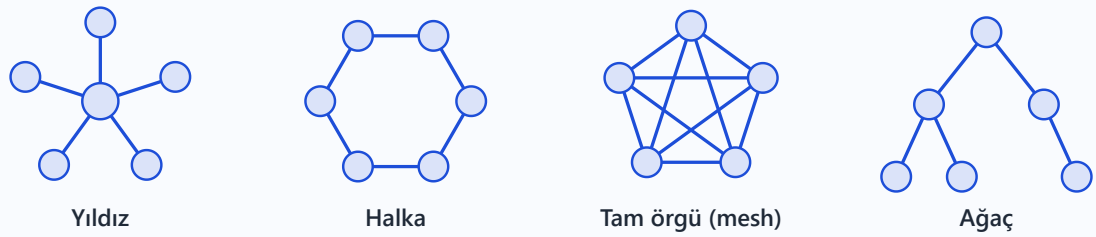
Algoritmik oyun kuramı bu problemlere yeni bir bakış ekler. Klasik ağ mühendisliğinde trafik “yönlendirilen” bir şeydir: bir merkez, paketlerin ya da araçların hangi yoldan gideceğine karar verir. Oysa gerçek ağların çoğunda kararları kullanıcıların kendisi verir. Şehirde her sürücü kendi rotasını seçer; internette her ağ işletmecisi kendi çıkarına göre trafik yönlendirir. Algoritmik oyun kuramı, kullanıcıların kendi maliyetlerini azaltmaya çalıştığı bu trafik ve iletişim ağlarını inceler. Temel uygulama alanları şehir trafiği, internet omurgaları ve enerji şebekeleridir.

8.1.3 Temel kavramlar: düğüm, bağlantı, kapasite, maliyet

Matematikte bir ağ, **çizge** (*graph*) ile modellenir. Çizge, noktalardan ve bu noktaları birleştiren çizgilerden oluşur. Bu bölümde dört temel kavram sürekli kullanılır.

- **Düğüm** (*node*): ağdaki bir nokta. Şehir trafiğinde bir kavşak, internette bir **yönlendirici** (*router*), elektrik şebekesinde bir trafo merkezidir.
- **Bağlantı** (*link*) ya da **kenar** (*edge*): iki düğümü birleştiren hat. Bir yol, bir fiber optik hat ya da bir iletim hattıdır. Yönlü bir kenar (s→a) yalnız bir yönde kullanılabilir; tek yönlü bir cadde gibi.
- **Kapasite** (*capacity*) ve **bant genişliği** (*bandwidth*): bir bağlantının birim zamanda taşıyabildiği akış. Bir yolun saatte geçirebildiği araç sayısı ya da bir hattın saniyede taşıyabildiği bit sayısıdır. Kapasite aşılınca gecikme hızla artar.
- **Maliyet**: iki türlüdür. *Kurulum maliyeti* ağı kurarken ödenir ve tasarım kararlarını etkiler. *Kullanım maliyeti* ise ağı kullanırken ödenir: gecikme, harcanan enerji ya da ücret. Kullanım maliyeti yönlendirme kararlarını etkiler.

Düğümlerin birbirine nasıl bağlandığına ağın **topolojisi** denir. Şekil 8.3 dört temel topolojiyi gösterir.



Şekil 8.3: Dört temel topoloji. Yıldızda her düğüm bir merkeze bağlıdır; halkada her düğümün iki komşusu vardır; tam örgüde her düğüm çifti doğrudan bağlıdır; ağaçta döngü yoktur ve her düğüme tek bir yoldan ulaşılır.

Her topolojinin artısı ve eksisi vardır. **Yıldız** az bağlantı ister (n düğüm için n − 1 bağlantı) ve her düğüm merkeze bir adım uzaklıktadır; ama merkez çökerse bütün ağ çöker. **Halka** da az bağlantı ister (n bağlantı) ve tek bir kopukluğa dayanır, ama uzak düğümler arasında çok adım vardır. **Tam örgü** (*mesh*) her çift arasında doğrudan bağlantı kurar; uzaklıklar en kısadır ama bağlantı sayısı n(n − 1)/2'dir ve hızla büyür. **Ağaç**, katmanlı bir düzen kurar ve döngü içermez. Veri merkezlerinde ağaç ya da katmanlı topolojiler yaygındır; şehir trafiğinde ise yol ağı örgüye yakındır, çünkü iki nokta arasında çoğu zaman birden çok seçenek vardır.

Bu topolojiler 8.5'te yeniden karşımıza çıkar. Orada sorulan soru, bağlantıları kendileri satın alan oyuncuların sonunda hangi topolojide karar kılacağıdır. Bağlantı ucuzsa tam örgüye, pahalıysa yıldıza yakın ağlar ortaya çıkar.

8.1.4 Klasik yönlendirme algoritmaları

Kenarların her birine bir sayı (ağırlık) atanmış bir ağda en kısa yolu bulmak, bilgisayar mühendisliğinin temel problemlerinden biridir. Ağırlık bir uzaklık, bir süre ya da bir maliyet olabilir. Tablo 8.1 yönlendirmede kullanılan dört klasik yöntemi özetler.

Yöntem	Ne yapar?	Not
Dijkstra	Negatif olmayan ağırlıklarla tek kaynaktan en kısa yollar	*O((V + E) log V)*; OSPF'nin (en kısa yolu önce açan bağlantı durumu protokolü) çekirdeği
Bellman–Ford	Negatif ağırlıklara izin verir, komşulardan öğrenerek günceller	*O(V · E)*; dağıtık sürümü RIP (uzaklık vektörü protokolü)
Dinamik yönlendirme	Bağlantı durumu ve maliyet değişince yolları yeniden hesaplar	Trafik ve arızalara uyum
Yük dengeleme	Trafiği birden fazla yol veya sunucu arasında dağıtır	Ör. ECMP (eşit maliyetli çok yol)

Tablo 8.1: Klasik yönlendirme algoritmaları.

Tablodaki V düğüm sayısını, E kenar sayısını gösterir. **Dijkstra algoritması** kaynaktan başlar ve her adımda "kesin uzaklığı henüz belirlenmemiş düğümler içinde en yakın olanı" kesinleştirir. Algoritmanın doğru çalışması için ağırlıkların negatif olmaması gerekir: bir kez kesinleşmiş bir uzaklık, sonradan negatif bir kenar yüzünden kısalamaz. İnternetteki **OSPF** (*Open Shortest Path First*) protokolünde her yönlendirici ağın tamamının haritasını tutar ve bu haritada Dijkstra çalıştırır.

Örnek 8.1

Şekil 8.2'deki ağın kenarlarına şu ağırlıklar verilsin: s→a 2, s→b 4, a→b 3, a→c 3, b→c 2, c→t 1. Dijkstra s'den başlar.

1. Başlangıçta s'nin uzaklığı 0'dır, ötekiler bilinmiyor (∞). s kesinleşir. Komşuları güncellenir: a için 0 + 2 = 2, b için 0 + 4 = 4.
2. Kesinleşmemiş düğümlerde en küçük tahmin a'nınkidir (2). a kesinleşir. Komşuları: b için 2 + 3 = 5, ama b'nin tahmini zaten 4'tür; 5 > 4 olduğu için değişmez. c için 2 + 3 = 5.
3. Sıradaki en küçük b'dir (4). b kesinleşir. Komşusu c için 4 + 2 = 6 > 5; c değişmez.
4. Sıradaki c'dir (5). c kesinleşir. Komşusu t için 5 + 1 = 6.
5. Son olarak t kesinleşir (6). Geriye doğru izlenince yol t ← c ← a ← s'dir: en kısa yol s→a→c→t, uzunluğu 2 + 3 + 1 = 6'dır. Bu, şekildeki kırmızı yoldur.

Bellman–Ford algoritması farklı çalışır: her düğüm, komşularının bildirdiği uzaklıklara bakarak kendi tahminini günceller ve bu işlem tahminler değişmeyene kadar sürer. En çok $V - 1$ tur yeter ve her tur bütün kenarlarda gezinir; süre bu yüzden $O(V \cdot E)$'dir. Dijkstra'dan yavaştır ama negatif ağır-

lıkları kabul eder. Asıl önemli özelliği *dağıtık* çalışabilmesidir: hiçbir düğümün ağın tamamını bilmesi gerekmez, herkes yalnız komşusundan öğrenir. İnternetteki **RIP** protokolü bu "komşudan öğrenme" fikrine dayanan bir uzaklık vektörü protokolüdür.

Dinamik yönlendirme, bir bağlantı koptuğunda ya da maliyeti değiştiğinde yolları yeniden hesaplar. **Yük dengeleme** ise trafiği tek bir yola yığmak yerine birden çok yola ya da sunucuya dağıtır. **ECMP** (*equal-cost multi-path*, eşit maliyetli çok yol) bunun basit bir biçimidir: uzunluğu eşit olan birden çok en kısa yol varsa trafik bunlar arasında bölünür.

Bu algoritmaların hepsi ortak bir varsayım yapar: kenar ağırlıkları *sabittir*. Dijkstra, a→c kenarının 3 dakika sürdüğünü kabul eder ve bu sayıyı kimin o kenarı kullandığından bağımsız sayar. Gerçekte ise gecikme, o yolu kaç kişinin seçtiğine bağlıdır. Yukarıdaki örnekte bütün sürücülere s→a→c→t önerilirse a→c kenarı tıkanır ve artık 3 dakika sürmez. Stratejik kullanıcılar önerilen rotadan sapabildiğinde ağın dengesi ile toplam gecikmeyi en küçük yapan dağılım birbirinden ayrılır. Yönlendirmeyi bir oyuna dönüştüren de budur: her kullanıcının "en kısa yol"u, ötekilerin seçimine bağlıdır.

8.1.5 Gecikme yüke bağlıdır

Yüke bağlı gecikmeyi modellemek için her kenara bir sayı yerine bir fonksiyon atanır.

> **Tanım 8.2**
>
> Bir e kenarının **gecikme fonksiyonu** (*latency function*) $\ell_e(x)$, kenar üzerindeki akış x iken bir kullanıcının o kenarda geçirdiği süredir. Gecikme fonksiyonu *azalmayan* bir fonksiyondur: akış artınca gecikme ya aynı kalır ya da artar, hiçbir zaman azalmaz.

"Azalmayan" koşulu sağduyuya dayanır: bir yola daha çok araç girdikçe yol daha hızlı olmaz. Bu bölümde dört tür gecikme fonksiyonu kullanılır.

- **Sabit**: $\ell(x) = b$. Trafikten etkilenmeyen geniş bir otoyol. Braess örneğindeki 45 dakikalık geniş yollar böyledir.
- **Doğrusal**: $\ell(x) = a \cdot x + b$. b boş yoldaki süredir, a ise her ek aracın süreyi ne kadar uzattığıdır. İki yollu ağ ve Braess örnekleri bu türü kullanır; örneğin $\ell(x) = 5 + x/100$ boş yolda 5 dakika sürer ve her 100 araç süreyi bir dakika uzatır.
- **BPR** (ABD Karayolları Bürosu, *Bureau of Public Roads*): $\ell(x) = t_0\,(1 + 0{,}15\,(x/c)^4)$. Burada t_0 boş yoldaki süre, c yolun kapasitesidir. Trafik planlamasında standart olarak kullanılır.
- **Kuyruk gecikmesi**: $\ell(x) = 1/(c - x)$, yalnızca $0 \le x < c$ için. Bu, tek sunuculu bir kuyruğun (**M/M/1 kuyruğu**) ortalama gecikmesidir. c saniyede işlenebilen, x saniyede gelen paket sayısıdır.



Şekil 8.4: Üç gecikme fonksiyonu. Yatay eksen akışın kapasiteye oranı (x/c), dikey eksen gecikmenin boş yol süresine oranıdır. Sabit gecikme düz kalır; doğrusal gecikme düzgün artar; BPR eğrisi kapasitenin altında neredeyse düzdür, kapasite aşılınca hızla yükselir.

Şekil 8.4 fonksiyonların biçimini karşılaştırır. BPR eğrisinin davranışı günlük deneyime uyar: yol kapasitesinin yarısıyla doluyken trafik neredeyse akıcıdır, kapasiteye yaklaşınca yavaşlama belirginleşir, kapasite aşılınca yolculuk süresi hızla uzar. Dördüncü kuvvet ($(x/c)^4$) bu "önce yavaş, sonra hızlı" artışı üretir.

Örnek 8.2

Boş yolda $t_0 = 10$ dakika süren ve kapasitesi saatte $c = 2.000$ araç olan bir yol düşünülsün.

1. $x = 1.000$: $x/c = 0{,}5$, $0{,}5^4 = 0{,}0625$. Gecikme $10 \cdot (1 + 0{,}15 \cdot 0{,}0625) \approx 10{,}09$ dakikadır; boş yoldan farkı yalnız 6 saniyedir.
2. $x = 2.000$: $x/c = 1$. Gecikme $10 \cdot 1{,}15 = 11{,}5$ dakikadır. Kapasitede yolculuk %15 uzar.
3. $x = 3.000$: $x/c = 1{,}5$, $1{,}5^4 = 5{,}0625$. Gecikme $10 \cdot (1 + 0{,}759) \approx 17{,}6$ dakikadır. Akış yalnız %50 arttı, ama gecikmedeki artış kapasitedekinin beş katından fazladır (1,5 dakikaya karşı 7,6 dakika).

Kuyruk gecikmesinde de aynı "duvar" görülür. $c = 100$ paket/saniye olsun. $x = 50$'de gecikme $1/50 = 0{,}02$ saniye (20 ms), $x = 90$'da $1/10 = 0{,}1$ saniye (100 ms), $x = 99$'da 1 saniyedir.

Kuyruk gecikmesinin bir özelliği ayrıca vurgulanmalıdır: x, c'ye yaklaştıkça $1/(c - x)$ sınırsız büyür. $x \geq c$ olduğunda ise formül anlamını yitirir; saniyede işlenebilenden çok paket geldiği için kuyruk durmadan uzar ve kararlı bir ortalama gecikme yoktur. Formül bu nedenle yalnız $x < c$ için geçerlidir.

Sık yapılan hata

- **Gecikmeyi sabit sanmak:** Bir yolun süresi kullanıcı sayısına bağlıdır. "Bu yol 20 dakika" demek, "bu yol şu kadar araçla 20 dakika" demektir.
- **Kenar gecikmesi ile yol gecikmesini karıştırmak:** ℓ_e tek bir kenarın gecikmesidir. Bir yolun gecikmesi, üzerindeki kenarların gecikmelerinin toplamıdır (8.2.1).
- **Kuyruk formülünü kapasitenin üstünde kullanmak:** $1/(c - x)$, $x \geq c$ için tanımsız ya da negatif çıkar; bu bir "hızlı yol" değil, tanımsız (sonsuz büyüyen) bir kuyruktur.

8.2 Bencil yönlendirme

Yüke bağlı gecikmeli bir ağda iki soru öne çıkar: kullanıcılar kendi başlarına seçim yapınca trafik nasıl dağılır ve bu dağılım toplam gecikme açısından ne kadar iyidir? İlk sorunun cevabı Wardrop dengesi, ikincisinin ölçütü sistem optimumudur.

8.2.1 Model: akış, yol gecikmesi, toplam gecikme

Tanım 8.3

Bencil yönlendirme (*selfish routing*), her kullanıcının kendi gecikmesini en aza indiren rotayı seçtiği yönlendirmedir. Kullanıcılar çok sayıda ve her biri çok küçükse akış **bölünebilir** (*nonatomic*, atomik olmayan) kabul edilir: tek bir kullanıcının akışa etkisi ihmal edilir.

Bölünebilir akış varsayımı bir yaklaşımdır. Şehirde binlerce sürücü vardır ve tek bir sürücünün bir yola girmesi o yolun süresini fark edilmeyecek kadar az değiştirir. Trafiği tek tek araçlar yerine akan bir sıvı gibi düşünmek hesapları kolaylaştırır. Sürücü sayısı az ve her birinin etkisi belirginse (örneğin dört sürücü) model **atomik** olur; bu durum 8.4'te ele alınır.

Modelin bileşenleri şunlardır:

- r birim trafik, kaynak s'den hedef t'ye gider. Örneğin $r = 3.000$ araç.
- f_P: P yolundaki akış, yani o yolu seçen trafik miktarı. Bütün yollardaki akışların toplamı r'dir.
- f_e: e kenarındaki toplam akış. Bir kenar birden çok yolun parçası olabilir; o kenarı içeren bütün yolların akışları toplanır: $f_e = \Sigma_{P \ni e} f_P$.

Bu bileşenlerle iki ölçü tanımlanır. Bir yolun gecikmesi, üzerindeki kenarların gecikmelerinin toplamıdır:

$$\ell_P(f) = \Sigma_{e \in P} \ell_e(f_e)$$

Ağın **toplam gecikmesi** ise her kenarda akış ile gecikmenin çarpımının toplamıdır:

$$C(f) = \Sigma_e f_e \cdot \ell_e(f_e)$$

Toplam gecikmenin neden böyle yazıldığı bir örnekle görülür. Bir kenarda 2.000 araç varsa ve her biri o kenarda 25 dakika geçiriyorsa, o kenarda harcanan toplam zaman 2.000 × 25 = 50.000 araç·dakikadır. "Araç·dakika" birimi, her aracın harcadığı dakikaların toplamını ifade eder. Bütün kenarlar için bu çarpımlar toplanınca ağda harcanan toplam zaman bulunur.

Aynı toplam yollar üzerinden de hesaplanabilir: her yolun akışı o yolun gecikmesiyle çarpılıp toplanır, $C(f) = \Sigma_P f_P \cdot \ell_P(f)$. İki yazım aynı sonucu verir. Bunun nedeni şudur: $f_P \cdot \ell_P$ çarpımında ℓ_P kenar gecikmelerinin toplamı olduğundan, her kenarın gecikmesi o kenardan geçen bütün yolların akışlarıyla çarpılmış olur; bu akışların toplamı da f_e'dir. Toplam gecikmeyi r'ye bölmek ortalama yolculuk süresini verir.

Bireysel rota seçimleri, toplam gecikmeyi en aza indiren dağılımdan uzaklaşabilir. Braess paradoksu bunun en uç biçimidir: yeni bir bağlantı eklemek denge gecikmesini artırabilir. Önce bu uzaklaşmanın nasıl ölçüleceği tanımlanmalıdır.

8.2.2 Wardrop ilkeleri

İngiliz ulaştırma mühendisi John Glen Wardrop, 1952'de trafik akışının iki farklı biçimde dağılabileceğini ileri sürdü (Wardrop, 1952). Bu iki ilke bugün de trafik planlamasının temelidir.

Tanım 8.4

Wardrop'un 1. ilkesi (kullanıcı dengesi). Bir f akışı, kullanılan her yolun gecikmesi herhangi bir yolun gecikmesinden büyük değilse bir **Wardrop dengesidir** (*Wardrop equilibrium*):

$$f_P > 0 \Rightarrow \ell_P(f) \le \ell_Q(f) \text{ her } Q \text{ yolu için.}$$

Wardrop'un 2. ilkesi (sistem optimumu). Bir f^* akışı, toplam gecikme $C(f)$'yi bütün olası akışlar içinde en küçük yapıyorsa **sistem optimumudur** (*system optimum*).

Birinci ilke üç eşdeğer biçimde okunabilir:

- Kullanılan bütün yolların gecikmesi eşittir ve en düşüktür.
- Kullanılmayan bir yol daha yavaştır (ya da kullanılanlarla eşit hızdadır).
- Tek bir kullanıcı yol değiştirerek yolculuk süresini azaltamaz.

Neden kullanılan yolların gecikmeleri eşit olmalıdır? Diyelim ki iki yol da kullanılıyor ama Yol 1 daha hızlı. O zaman Yol 2'deki bir sürücü Yol 1'e geçerek kazanır. Bu bir dengeyle bağdaşmaz; denge, kimsenin tek başına sapmak istemediği durumdur. Üçüncü okuma, Wardrop dengesinin aslında Nash

dengesi fikrinin bölünebilir trafiğe uyarlanmış hâli olduğunu gösterir: her kullanıcı ötekilerin seçimine karşı en iyi tepkiyi verir.

İkinci ilke ise bir planlayıcının bakış açısıdır. Bütün trafiği tek elden yönetebilen bir merkez, ortalama (toplam) gecikmeyi en küçük yapan dağılımı seçer. Bu dağılımda bazı kullanıcılar daha yavaş bir yola "feda edilebilir": toplam açısından onların daha yavaş yolda olması, ötekilerin hızlanmasıyla telafi edilir. Sistem optimumunu uygulamak bu yüzden zordur; feda edilen kullanıcıların uymaya istekli olması gerekir.

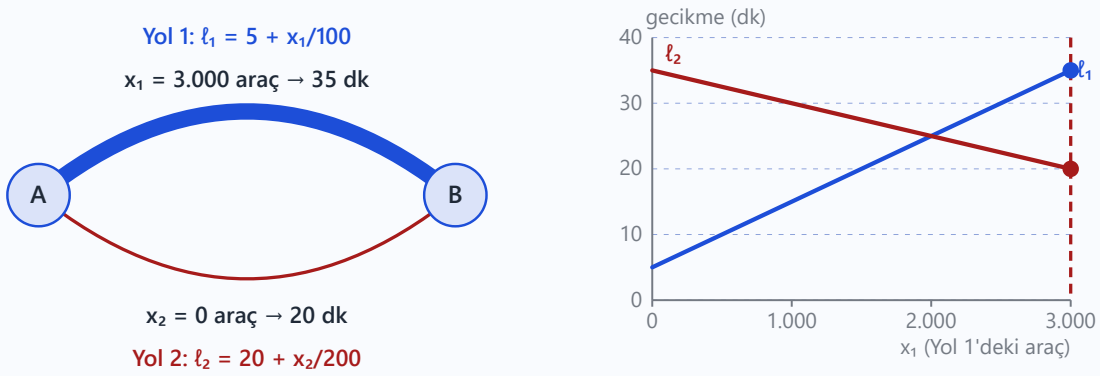
Özetle denge, kullanıcıların bireysel tercihlerini açıklar; ama sistem optimumuyla aynı olmak zorunda değildir. Doğrusal gecikmelerde (daha genel olarak sürekli ve azalmayan gecikmelerde) bir Wardrop dengesi her zaman vardır. Üstelik denge birden çok olsa bile her kenarın denge gecikmesi aynıdır; hangi denge bulunursa bulunsun kullanıcıların yaşadığı süreler değişmez. Bu sonuç 8.2.5'te Beckmann potansiyeliyle açıklanır.

8.2.3 İki yollu ağda Wardrop dengesi

Bölüm boyunca kullanılan iki yollu ağ şöyledir. 3.000 araç A'dan B'ye gider. Yol 1 kısa ama dardır: boşken 5 dakika sürer, ama her 100 araç süreyi bir dakika uzatır. Yol 2 uzun ama geniştir: boşken 20 dakika sürer, her 200 araç süreyi bir dakika uzatır.

> $$\ell_1(x_1) = 5 + x_1/100,\ \ell_2(x_2) = 20 + x_2/200,\ x_1 + x_2 = 3.000$$

Her yol tek bir kenardan oluştuğu için burada yol gecikmesi ile kenar gecikmesi aynıdır. Dengeyi bulmak için önce iki "deneme" yapılır; bu denemeler dengenin neden belirli bir noktada olması gerektiğini gösterir.



Şekil 8.5: Deneme 1: bütün araçlar Yol 1'de. Solda ağ (kalın çizgi çok akış), sağda gecikmelerin x_1'e göre grafiği: mavi doğru ℓ_1, kırmızı doğru ℓ_2. $x_1 = 3.000$'de Yol 1 35 dakika, boş Yol 2 20 dakikadır.

Şekil 8.5'teki grafik her denemeyi tek bakışta değerlendirmeyi sağlar. Yatay eksende Yol 1'deki araç sayısı x_1 vardır; Yol 2'deki araç sayısı $3.000 - x_1$'dir. Mavi doğru x_1 arttıkça yükselir (Yol 1 kalabalıklaşır), kırmızı doğru x_1 arttıkça alçalır (Yol 2 boşalır). İki doğrunun kesiştiği nokta, iki yolun eşit sürdü-

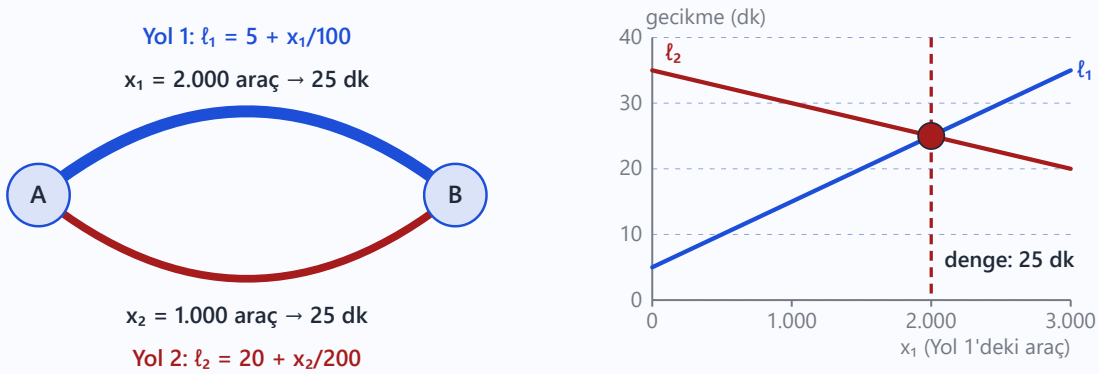
ğü noktadır.

Deneme	x_1 / x_2	ℓ_1 (dk)	ℓ_2 (dk)	Kim sapmak ister?
Herkes Yol 1'de	3.000 / 0	35	20	Yol 1'dekiler
Eşit bölünme	1.500 / 1.500	20	27,5	Yol 2'dekiler
Denge	2.000 / 1.000	25	25	Kimse

Tablo 8.2: İki yollu ağda üç deneme.

Örnek 8.3

1. **Deneme 1: herkes Yol 1'de.** $\ell_1(3.000) = 5 + 30 = 35$ dakika; boş Yol 2 ise $\ell_2(0) = 20$ dakika. Yol 1'deki her sürücü Yol 2'ye geçerse 15 dakika kazanır. Bu bir denge değildir.
2. **Deneme 2: 1.500 / 1.500.** $\ell_1 = 5 + 15 = 20$, $\ell_2 = 20 + 7{,}5 = 27{,}5$ dakika. Bu kez Yol 2'dekiler Yol 1'e kayar. Yine denge değildir.
3. **Denge koşulu.** İki deneme, dengenin 1.500 ile 3.000 arasında bir yerde olduğunu gösterir. İki yol da kullanılıyorsa gecikmeler eşit olmalıdır: $5 + x_1/100 = 20 + (3.000 - x_1)/200$.
4. **Çözüm.** Kesirlerden kurtulmak için iki taraf 200 ile çarpılır: $1.000 + 2x_1 = 4.000 + 3.000 - x_1$. x_1'li terimler bir tarafa toplanır: $3x_1 = 6.000$, yani $x_1 = 2.000$ ve $x_2 = 1.000$.
5. **Gecikmeler.** $\ell_1 = 5 + 20 = 25$, $\ell_2 = 20 + 5 = 25$ dakika. İki yol eşit sürer.
6. **Kontrol.** Yol değiştiren tek bir sürücü öteki yolda da yaklaşık 25 dakika yaşar (akış bölünebilir olduğundan tek sürücünün etkisi ihmal edilir). Kimse kazanmaz: bu bir Wardrop dengesidir. Toplam gecikme $3.000 \times 25 = 75.000$ araç·dakikadır.



Şekil 8.6: İki yollu ağın Wardrop dengesi. 2.000 araç Yol 1'de, 1.000 araç Yol 2'de; iki yol da 25 dakika sürer. Grafikte denge, iki gecikme doğrusunun kesiştiği noktadır.

Şekil 8.6 dengeyi gösterir. Denge noktasının solunda mavi doğru kırmızının altındadır: Yol 1 daha hızlıdır ve araçlar Yol 1'e kayar, yani nokta sağa gider. Sağında mavi doğru üstte kalır: Yol 1 daha yavaştır ve araçlar Yol 2'ye kayar, nokta sola gider. Her iki yönden gelen bu "itiş", akışı kesişme noktasına taşır.

Denge bu yüzden kararlıdır.

Eşitleme yöntemi yalnız iki yol da kullanılıyorsa geçerlidir. Bir yol, bütün trafiği taşırken bile öteki boş yoldan hızlı olabilir mi? Örneğin Yol 1, 3.000 araçla bile Yol 2'nin boş hâlinden hızlı kalacak kadar geniş olsaydı ($\ell_1(3.000) \le \ell_2(0)$), denge herkesin Yol 1'de olduğu durum olurdu. O zaman kullanılan tek yol Yol 1'dir ve kullanılmayan Yol 2 daha yavaştır; tanımın ikinci okuması tam bu durumu kapsar. Kısacası eşitleme denkleminin çözümü 0 ile r arasında çıkmazsa denge bir uçtadır.

Örnek 8.4

6.000 araç; Yol 1: $\ell_1 = 10 + x_1/200$, Yol 2: $\ell_2 = 25 + x_2/100$.

1. Önce uç durumlar denetlenir. Herkes Yol 1'deyse $\ell_1 = 10 + 30 = 40 > \ell_2(0) = 25$; Yol 1'dekiler kaçar. Herkes Yol 2'deyse $\ell_2 = 25 + 60 = 85 > \ell_1(0) = 10$; Yol 2'dekiler kaçar. Demek ki iki yol da kullanılır.
2. Eşitlik: $10 + x_1/200 = 25 + (6.000 - x_1)/100$. İki taraf 200 ile çarpılır: $2.000 + x_1 = 5.000 + 12.000 - 2x_1$.
3. Toplanınca $3x_1 = 15.000$, $x_1 = 5.000$, $x_2 = 1.000$.
4. Kontrol: $\ell_1 = 10 + 25 = 35$, $\ell_2 = 25 + 10 = 35$ dakika. İkisi de 35 dakikadır; toplam gecikme $6.000 \times 35 = 210.000$ araç·dakikadır.

8.2.4 Denge ile sistem optimumu aynı değil

Dengede herkes 25 dakika yolculuk eder ve toplam 75.000 araç·dakikadır. Bu, toplam gecikmenin olabilecek en küçük değeri midir? Bunu anlamak için toplam gecikme x_1'in bir fonksiyonu olarak yazılır:

$$C(x_1) = x_1 \cdot \ell_1(x_1) + x_2 \cdot \ell_2(x_2),\ x_2 = 3.000 - x_1$$

Bu fonksiyonun birkaç değeri Tablo 8.3'te verilmiştir. Tablo, eğrinin en alt noktasının dengede (2.000) değil, 1.500'de olduğunu gösterir.

x_1	0	500	1.000	1.500	2.000	2.500	3.000
C	105.000	86.250	75.000	**71.250**	75.000	86.250	105.000
Φ	82.500	69.375	60.000	54.375	**52.500**	54.375	60.000

Tablo 8.3: Toplam gecikme $C(x_1)$ ve Beckmann potansiyeli $\Phi(x_1)$ (araç·dakika).

Optimumu tabloyu taramadan kesin olarak bulmak için **marjinal maliyet** (*marginal cost*) kavramı kullanılır. Bir yolun marjinal maliyeti, o yola bir araç daha eklemenin toplam gecikmeye kattığı miktardır.

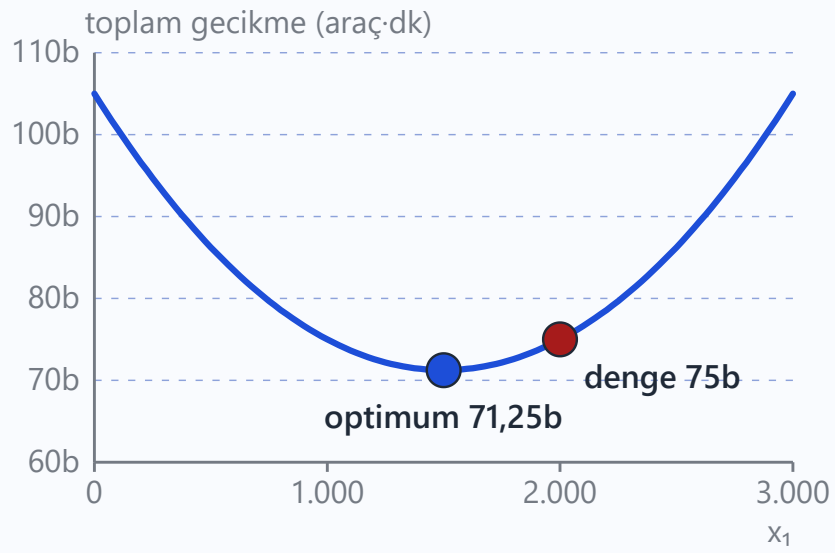
Bu miktarın neden yolun kendi gecikmesinden büyük olduğu şöyle görülür. Yol 1'de x_1 araç varken bir araç daha eklensin. Yeni araç kendisi ℓ_1 kadar gecikme yaşar. Ama ayrıca yolu biraz daha kalabalıklaştırır: gecikme her araç için $\ell_1' = 1/100$ dakika artar ve bu artışı yoldaki öteki x_1 aracın hepsi yaşar. Toplamda $x_1 \cdot \ell_1'$ kadar ek gecikme doğar. Marjinal maliyet bu iki parçanın toplamıdır:

$$(x \cdot \ell(x))' = \ell(x) + x \cdot \ell'(x)$$

Bu formül, çarpımın türevi kuralının ($(uv)' = u'v + uv'$) bu duruma uygulanmasıdır. Türev, burada "fonksiyonun girdisi bir birim artınca çıktısının ne kadar arttığı" olarak okunabilir. Doğrusal gecikmeler için hesap kolaydır:

- Yol 1: $x_1 \cdot \ell_1 = 5x_1 + x_1^2/100$. $5x_1$'in türevi 5'tir; $x_1^2/100$'ün türevi $2x_1/100 = x_1/50$'dir. Marjinal maliyet $5 + x_1/50$.
- Yol 2: $x_2 \cdot \ell_2 = 20x_2 + x_2^2/200$. Marjinal maliyet $20 + x_2/100$.

Sistem optimumunda iki yolun marjinal maliyetleri eşit olmalıdır. Nedeni, dengedeki akıl yürütmenin aynısıdır ama bu kez planlayıcının gözünden: Yol 1'in marjinal maliyeti daha yüksek olsaydı, bir aracı Yol 1'den Yol 2'ye almak toplam gecikmeyi azaltırdı. Planlayıcı böyle bir iyileştirme kalmayana kadar araç taşır; durduğu yerde marjinal maliyetler eşittir.



Şekil 8.7: Toplam gecikme $C(x_1)$ eğrisi (b = bin araç·dakika). Denge ($x_1 = 2.000$, 75 bin) eğrinin dibinde değildir; en alt nokta optimumdur ($x_1 = 1.500$, 71,25 bin).

Şekil 8.7 toplam gecikme eğrisini gösterir. Eğri bir çanak biçimindedir: uçlarda (bütün trafik tek yolda) toplam gecikme çok büyüktür, ortada bir yerde en küçük olur. Denge noktası çanağın dibinin sağında kalır.

Örnek 8.5

1. Marjinal maliyetler eşitlenir: $5 + x_1/50 = 20 + (3.000 - x_1)/100$.
2. İki taraf 100 ile çarpılır: $500 + 2x_1 = 2.000 + 3.000 - x_1$, yani $3x_1 = 4.500$.
3. $x_1 = x_2 = 1.500$. Trafik iki yola eşit bölünür.
4. Gecikmeler: $\ell_1 = 20$, $\ell_2 = 27{,}5$ dakika. Toplam: $1.500 \times 20 + 1.500 \times 27{,}5 = 30.000 + 41.250 = 71.250$ araç·dakika.
5. Kontrol: marjinal maliyetler $5 + 1.500/50 = 35$ ve $20 + 1.500/100 = 35$; eşittir.

	x_1 / x_2	ℓ_1 / ℓ_2 (dk)	Toplam (araç·dk)
Denge	2.000 / 1.000	25 / 25	75.000
Optimum	1.500 / 1.500	20 / 27,5	71.250

Tablo 8.4: İki yollu ağda denge ile optimum.

Tablo 8.4 iki dağılımı karşılaştırır. Dengede toplam gecikme optimumdan 3.750 araç·dakika fazladır; oran olarak $75.000 / 71.250 \approx 1{,}053$, yani yaklaşık %5 fazla. Bencil davranışın bu "bedeli", kuramda **anarşinin bedeli** (*price of anarchy*) adıyla incelenir.

Optimum neden kendiliğinden oluşmaz? Optimumda Yol 1'deki sürücü 20, Yol 2'deki sürücü 27,5 dakika yaşar. Yol 2'deki her sürücü Yol 1'e geçerek 7,5 dakika kazanacağını görür ve geçer. Bu yüzden optimum bir denge değildir: kendi hâline bırakılan trafik optimumda kalmaz, dengeye kayar. Optimumu korumak için Yol 2'deki 1.500 sürücünün "feda edilmeyi" kabul etmesi gerekir.

Dengedeki verimsizliğin kaynağı da marjinal maliyetle görülür. Dengede ($x_1 = 2.000$) Yol 1'in marjinal maliyeti $5 + 2.000/50 = 45$, Yol 2'ninki $20 + 1.000/100 = 30$'dur. Bir aracı Yol 1'den Yol 2'ye almak toplam gecikmeyi yaklaşık $45 - 30 = 15$ araç·dakika azaltır. Ama o aracın sürücüsü bunu görmez: o yalnız kendi süresine bakar ve iki yolda da 25 dakika yaşadığı için yerinden memnundur.

Örnek 8.6

Örnek 8.4'teki ağda (6.000 araç; $\ell_1 = 10 + x_1/200$, $\ell_2 = 25 + x_2/100$) sistem optimumu bulunur.

1. Marjinal maliyetler: Yol 1 için $10 + 2x_1/200 = 10 + x_1/100$, Yol 2 için $25 + 2x_2/100 = 25 + x_2/50$.
2. Eşitlik: $10 + x_1/100 = 25 + (6.000 - x_1)/50$. İki taraf 100 ile çarpılır: $1.000 + x_1 = 2.500 + 12.000 - 2x_1$, yani $3x_1 = 13.500$.
3. $x_1 = 4.500$, $x_2 = 1.500$. Gecikmeler $\ell_1 = 10 + 22{,}5 = 32{,}5$, $\ell_2 = 25 + 15 = 40$ dakika.
4. Toplam: $4.500 \times 32{,}5 + 1.500 \times 40 = 146.250 + 60.000 = 206.250$ araç·dakika. Dengedeki 210.000'den 3.750 azdır. Optimumda Yol 2'deki sürücüler 7,5 dakika daha yavaştır ve Yol 1'e geçmek ister.

8.2.5 İki ayrı en aza indirme: Beckmann potansiyeli

Sistem optimumu, bir fonksiyonun (C) en küçük noktasıdır. Wardrop dengesi de bir fonksiyonun en küçük noktası olarak yazılabilir mi? Martin Beckmann, C. B. McGuire ve C. B. Winsten 1956'da bunun mümkün olduğunu gösterdi (Beckmann, McGuire, Winsten, 1956).

Tanım 8.5

Bölünebilir bir yönlendirme probleminde **Beckmann potansiyeli** (*Beckmann potential*) şudur:

$$\Phi(f) = \sum_e \int_0^{f_e} \ell_e(t)\, dt$$

Formüldeki integral göründüğü kadar zor değildir. $\int_0^{f_e} \ell_e(t)\, dt$, gecikme eğrisinin altında 0 ile f_e arasında kalan alandır. Bunu bir toplam gibi düşünmek kolaydır: trafik kenara azar azar girsin. İlk gelen sürücü boş yolu, yani $\ell_e(0)$'ı görür; sonrakiler biraz daha kalabalık yolu görür; en son gelen $\ell_e(f_e)$'yi görür. Alan, her sürücünün *girdiği anda* gördüğü gecikmelerin toplamıdır. Toplam gecikme $f_e \cdot \ell_e(f_e)$ ise herkese en son durumdaki gecikmeyi yükler; bu yüzden potansiyel genellikle toplam gecikmeden küçüktür.

Doğrusal bir gecikme $\ell(t) = b + a \cdot t$ için alan bir yamuktur ve hesabı kolaydır: $\int_0^x (b + a\,t)\, dt = b\,x + a\,x^2/2$. İki yollu ağda:

$$\Phi(x_1) = 5x_1 + x_1^2/200 + 20x_2 + x_2^2/400$$

Bu fonksiyonun x_1'e göre türevi alınır. $x_2 = 3.000 - x_1$ olduğundan x_1 bir birim artınca x_2 bir birim azalır. Yol 1 parçasının türevi $5 + x_1/100 = \ell_1$'dir; Yol 2 parçası ise ℓ_2 kadar azalır. Sonuç:

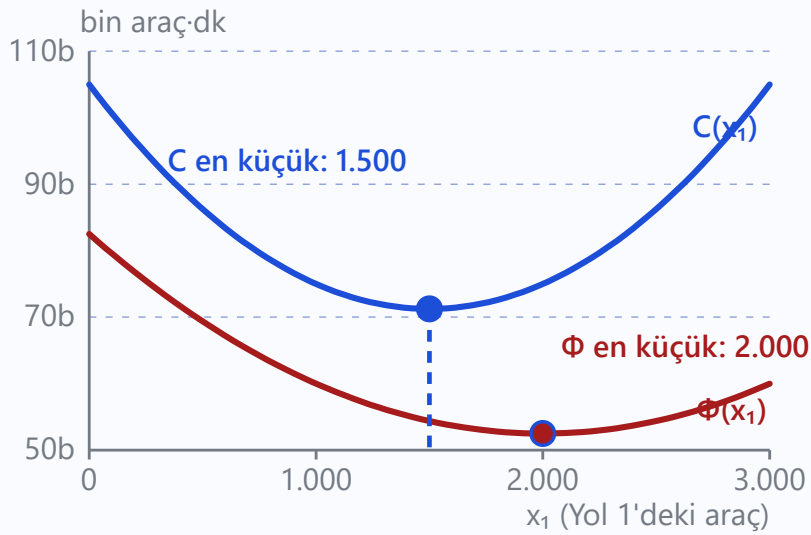
$$\Phi'(x_1) = \ell_1(x_1) - \ell_2(x_2)$$

Türevin sıfır olduğu nokta, $\ell_1 = \ell_2$ olan noktadır; bu da tam Wardrop koşuludur. Potansiyelin türevi, "son gelen sürücünün" gördüğü gecikmedir. Bencil sürücü de tam olarak buna bakar: yola girerken o anki gecikmeyi görür.

Teorem 8.1 (Beckmann, McGuire, Winsten)

Gecikme fonksiyonları sürekli ve azalmayan ise bir f akışı Wardrop dengesidir ancak ve ancak Beckmann potansiyeli Φ'yi bütün olası akışlar içinde en küçük yapar. Sistem optimumu ise toplam gecikme $C(f) = \sum_e f_e \cdot \ell_e(f_e)$'yi en küçük yapan akıştır.

Teoremin iki önemli sonucu vardır. Birincisi *varlık*: olası akışlar kümesi sınırlıdır ve sürekli bir fonksiyon böyle bir kümede en küçük değerini alır; demek ki Wardrop dengesi her zaman vardır. İkincisi *tekillik*: gecikmeler azalmadığı için Φ "çanak biçimli" (dışbükey) bir fonksiyondur. Çanak biçimli bir fonksiyonun birden çok en küçük noktası olsa bile bu noktalarda kenar gecikmeleri aynıdır. 8.2.2'deki "hangi denge bulunursa bulunsun kenar gecikmeleri aynıdır" cümlesinin dayanağı budur.



Şekil 8.8: İki ayrı en aza indirme. Mavi eğri toplam gecikme $C(x_1)$, kırmızı eğri Beckmann potansiyeli $\Phi(x_1)$. C'nin dibi $x_1 = 1.500$'de (optimum, 71.250), Φ'ninki $x_1 = 2.000$'de (denge, 52.500).

Şekil 8.8 iki eğriyi yan yana koyar. Tablo 8.3'teki sayılar da aynı şeyi söyler: Φ en küçük değerini 2.000'de (52.500), C en küçük değerini 1.500'de (71.250) alır. Denge potansiyelin dibidir, optimum toplam gecikmenin dibidir. İki dip farklı yerdedir, çünkü iki fonksiyonun türevleri farklıdır:

- Φ'nin bir kenara göre türevi ℓ_e'dir: son gelen sürücünün gördüğü gecikme.
- C'nin bir kenara göre türevi $\ell_e + f_e \cdot \ell_e'$'dir: marjinal maliyet.

Fark tek bir terimden gelir: $f_e \cdot \ell_e'$. Bu terim, bir sürücünün kendi varlığıyla ötekilere yüklediği ek gecikmedir. Sürücü bu **dışsallığı** (*externality*) hesaba katmaz. Dengede Yol 1'de bu terim $2.000 \times 1/100 = 20$ araç·dakikadır: Yol 1'e giren her yeni sürücü, oradaki 2.000 kişinin her birini 1/100 dakika yavaşlatır. Dışsallık kavramı 8.5.1'de ayrıca incelenir.

8.2.6 Kod izleme: denge_bul()

İki yollu ağda dengeyi denklem çözerek bulmak kolaydır. Gecikmeler doğrusal değilse (örneğin BPR) denklemi elle çözmek zorlaşır. O zaman sayısal bir yöntem kullanılır. Aşağıdaki fonksiyon dengeyi **ikiye bölme** (*bisection*) yöntemiyle arar. Girdi olarak iki gecikme fonksiyonu `l1`, `l2`, toplam trafik `r` ve istenen hassasiyet `eps` alınır.

```
def denge_bul(l1, l2, r, eps=50):
  lo, hi = 0, r
  while hi - lo > eps:
    x = (lo + hi) / 2
    if l1(x) < l2(r - x):
      lo = x      # Yol 1 hâlâ hızlı: artır
    else:
      hi = x      # Yol 1 yavaş: azalt
  return (lo + hi) / 2
```

Kod şöyle çalışır:

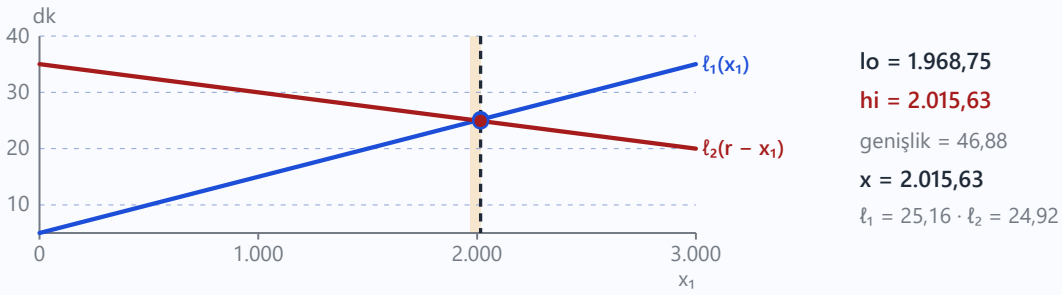
- `lo` ve `hi`, dengenin içinde bulunduğu bilinen aralığın uçlarıdır. Başta aralık *[0, r]*'dir: Yol 1'deki araç sayısı 0 ile *r* arasında olmak zorundadır.
- Döngü, aralığın genişliği `eps`'ten büyük olduğu sürece döner.
- Her turda aralığın orta noktası `x` denenir. Yol 1'de `x`, Yol 2'de `r - x` araç vardır.
- `l1(x) < l2(r - x)` ise Yol 1 hâlâ daha hızlıdır. Araçlar Yol 1'e geçmek ister; denge daha sağdadır. Aralığın sol ucu `x`'e taşınır.
- Aksi hâlde Yol 1 fazla kalabalıktır; denge daha soldadır. Aralığın sağ ucu `x`'e taşınır.
- Döngü bitince aralığın ortası döndürülür.

Yöntemin doğru çalışmasının nedeni şudur: $g(x) = \ell_1(x) - \ell_2(r - x)$ farkı x ile artar, çünkü x artınca Yol 1 yavaşlar, Yol 2 hızlanır. Artan bir fonksiyonun tek bir sıfırı vardır ve o sıfır dengedir. $x = 0$'da Yol 1 hızlıdır ($g < 0$), $x = r$'de yavaştır ($g > 0$); sıfır bu ikisinin arasındadır. Her tur aralığı yarıya indirdiği ve sıfırı hep aralığın içinde tuttuğu için arama sıfıra yaklaşır.

Tur	`x`	$\ell_1(x)$	$\ell_2(r-x)$	Karar	Yeni aralık	Genişlik
—	—	—	—	başlangıç	[0, 3.000]	3.000
1	1.500	20	27,5	`lo = x`	[1.500, 3.000]	1.500
2	2.250	27,5	23,75	`hi = x`	[1.500, 2.250]	750
3	1.875	23,75	25,63	`lo = x`	[1.875, 2.250]	375
4	2.062,5	25,63	24,69	`hi = x`	[1.875, 2.062,5]	187,5
5	1.968,75	24,69	25,16	`lo = x`	[1.968,75, 2.062,5]	93,75
6	2.015,63	25,16	24,92	`hi = x`	[1.968,75, 2.015,63]	46,88

Tablo 8.5: `denge_bul(l1, l2, 3000)` izlemesi ($\ell_1 = 5 + x/100$, $\ell_2 = 20 + x/200$, `eps = 50`).

Altıncı turdan sonra (Şekil 8.9) genişlik 46,88'dir ve $46{,}88 \le 50$ olduğundan döngü durur. Fonksiyon $(1.968{,}75 + 2.015{,}63)/2 \approx 1.992{,}19$ döndürür. Gerçek denge 2.000'dir; hata yaklaşık 8 araçtır ve `eps`'in yarısından küçüktür.



Şekil 8.9: `denge_bul` 'un altıncı turu. Mavi doğru $\ell_1(x_1)$, kırmızı doğru $\ell_2(r - x_1)$; turuncu şerit kalan aralık [1.968,75, 2.015,63], kesikli çizgi denenen $x = 2.015,63$. Aralık kesişme noktasını (2.000) hep içinde tutar.

Her tur aralığı yarıya indirdiği için k tur sonra aralığın genişliği $r/2^k$'dir ve hata en çok bu kadardır. 3.000 araçla 6 turda genişlik $3.000/64 \approx 46,9$'a iner. Daha hassas bir sonuç için her ek tur hatayı yarıya böler; 10 tur daha hatayı yaklaşık bin kat küçültür. Yöntem bu yüzden çok hızlıdır.

Örnek 8.7

Örnek 8.4'teki ağda ($r = 6.000$; $\ell_1 = 10 + x/200$, $\ell_2 = 25 + x/100$) `denge_bul` 'un ilk iki turu izlenir.

1. Başlangıç aralığı [0, 6.000]'dir. Tur 1'de $x = 3.000$: $\ell_1 = 10 + 15 = 25$, $\ell_2(3.000) = 25 + 30 = 55$. $25 < 55$, Yol 1 hızlıdır: `lo = 3000` , aralık [3.000, 6.000].
2. Tur 2'de $x = 4.500$: $\ell_1 = 10 + 22,5 = 32,5$, $\ell_2(1.500) = 25 + 15 = 40$. $32,5 < 40$: `lo = 4500` , aralık [4.500, 6.000].
3. Denge (5.000) bu aralığın içindedir. Arama yedinci turda durur ve yaklaşık 4.992 döndürür.

Bu yöntem yalnız iki yollu ağlarda çalışır, çünkü orada tek bir bilinmeyen (x_1) vardır. Çok yollu genel ağlarda denge, Beckmann potansiyelini en küçük yapan akıştır ve **dışbükey optimizasyon** yöntemleriyle bulunur. Trafik planlamasında en yaygın kullanılan yöntem Frank–Wolfe algoritmasıdır: her adımda mevcut gecikmelerle en kısa yolları (örneğin Dijkstra ile) bulur ve akışı bu yollara doğru biraz kaydırır.

8.2.7 Bencil yönlendirmede sık hatalar

Sık yapılan hata

- **Dengeyi “eşit bölünme” sanmak:** Denge trafiği yollara eşit bölmez, gecikmeleri eşitler. İki yollu ağda eşit bölünme (1.500 / 1.500) denge değil, optimumdur.
- **Eşitleme denklemini körü körüne kullanmak:** Denklem yalnız iki yol da kullanılıyorsa geçerlidir. Çözüm negatif ya da r'den büyük çıkarsa denge bir uçtadır.

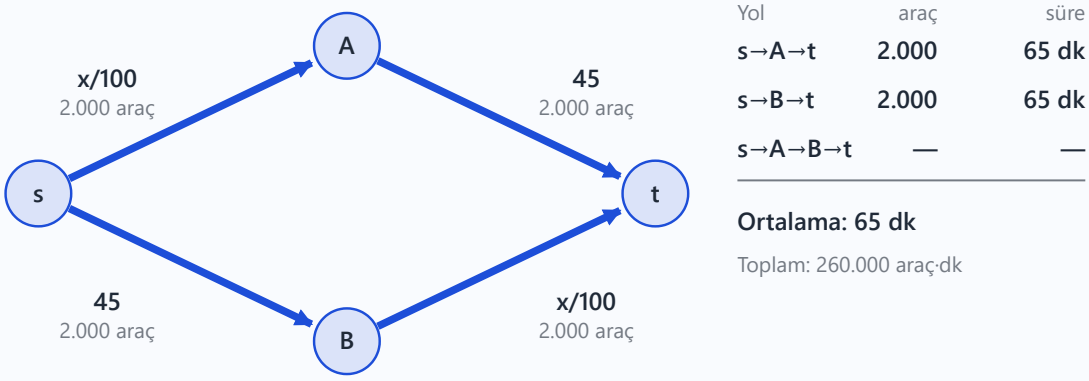
- **Optimumda gecikmeleri eşitlemek:** Optimumda eşitlenen *marjinal maliyetlerdir* ($\ell + x \cdot \ell'$), gecikmeler değil. Optimumda yolların süreleri çoğu zaman farklıdır (20 ve 27,5).
- **Türevde sabiti ikiye katlamak:** $(x \cdot (5 + x/100))' = 5 + x/50$'dir, $10 + x/50$ değil. Ek terim yalnız $x \cdot \ell'$'den gelir.
- **Toplam gecikmeyi ortalama sanmak:** 75.000 araç·dakika toplamdır; ortalama yolculuk süresi $75.000 / 3.000 = 25$ dakikadır.
- **Potansiyeli toplam gecikmeyle karıştırmak:** Φ bir "maliyet" değil, dengeyi bulmaya yarayan yardımcı bir fonksiyondur. Değeri (52.500) gerçek bir toplam süreyi göstermez.

8.3 Braess paradoksu

Alman matematikçi Dietrich Braess, 1968'de trafik planlamasıyla ilgili kısa bir makale yayımladı (Braess, 1968). Makalenin gösterdiği şey basit ama rahatsız ediciydi: bir ağa yeni bir yol eklemek, bencil rota seçimi yüzünden herkesin yolculuğunu uzatabilir.

8.3.1 Braess paradoksu adım adım

Kurulum 8.1.1'deki oyunun aynısıdır, yalnız sürücü sayısı somut olarak 4.000'dir. 4.000 araç s'den t'ye gider. s→A ve B→t dar yollardır: x araç varken $x/100$ dakika sürerler. A→t ve s→B geniş otoyollardır: trafikten bağımsız olarak 45 dakika sürerler. Oyundaki $40 \cdot x/N$ formülünde $N = 4.000$ yazılınca $x/100$ elde edilir; iki anlatım aynıdır.

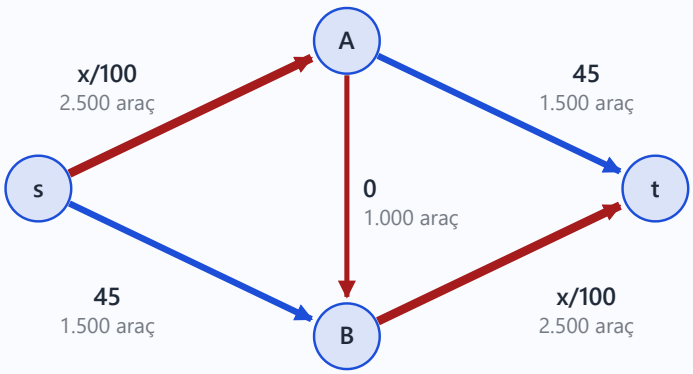


Şekil 8.10: Kestirme yokken Braess ağının dengesi. Araçlar iki rotaya 2.000 / 2.000 bölünür; her rota $2.000/100 + 45 = 65$ dakika sürer. Toplam gecikme 260.000 araç·dakikadır.

Örnek 8.8

1. **Kestirme yokken.** İki rota vardır: üst (s→A→t) ve alt (s→B→t). Ağ simetriktir; her rotanın bir dar ve bir geniş yolu vardır. Üst rotada x araç varsa üst rota $x/100 + 45$, alt rota $(4.000 - x)/100 + 45$ dakika sürer. Eşitlik $x = 2.000$'i verir. Her rota $20 + 45 = 65$ dakikadır. Rota değiştiren sürücü kazanmaz: bu bir Wardrop dengesidir. Üstelik simetri yüzünden aynı zamanda optimumdur.
2. **Kestirme açılır.** Belediye A'dan B'ye çok kısa, süresi 0 dakika olan bir bağlantı yolu açar. Başlangıçta kimse onu kullanmaz.
3. **İlk sürücü dener.** Tek bir sürücü zikzak rotayı (s→A→B→t) dener. s→A'da 2.000 araç vardır (20 dakika), A→B 0 dakika, B→t'de 2.000 araç vardır (20 dakika). Toplam $20 + 0 + 20 = 40$ dakika. Eski rotasında 65 dakika yaşıyordu; 25 dakika kazanır. Bireysel olarak çok kârlıdır ve sürücü geçer.
4. **Geçiş sürer.** Diyelim ki 1.000 sürücü zikzağa geçti; her eski rotada 1.500 sürücü kaldı. Dar yollarda artık $1.500 + 1.000 = 2.500$ araç vardır ve her biri 25 dakika sürer. Eski rotalar $25 + 45 = 70$, zikzak $25 + 0 + 25 = 50$ dakikadır. Zikzak hâlâ çok daha hızlıdır; geçiş sürer.
5. **Herkes zikzakta.** 4.000 sürücünün hepsi zikzaktadır. Dar yolların her birinde 4.000 araç vardır (40 dakika). Zikzak $40 + 0 + 40 = 80$ dakika sürer.
6. **Denge kontrolü.** Eski bir rotaya dönen sürücü bir dar yol (40 dakika) ve bir geniş yol (45 dakika) kullanır: $40 + 45 = 85$ dakika. 85 > 80 olduğundan kimse dönmez. Bu yeni Wardrop dengesidir.

Sonuç: yol eklendi ve herkes 15 dakika kaybetti. Toplam gecikme 260.000'den $4.000 \times 80 = 320.000$ araç·dakikaya çıktı.

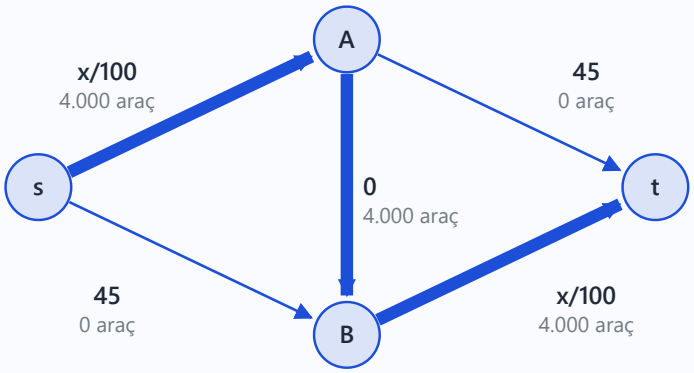


Yol	araç	süre
s→A→t	1.500	70 dk
s→B→t	1.500	70 dk
s→A→B→t	1.000	50 dk

Zikzak hâlâ en hızlı

50 dk < 70 dk

Şekil 8.11: Geçiş sürerken: 1.000 sürücü zikzakta. Dar yollarda 2.500 araç (25 dakika); eski rotalar 70, zikzak 50 dakika. Zikzak hâlâ en hızlı rotadır.



Yol	araç	süre
s→A→t	0	85 dk
s→B→t	0	85 dk
s→A→B→t	4.000	80 dk

65 → 80 dk

260 bin → 320 bin araç·dk

Şekil 8.12: Kestirmeli ağın dengesi: 4.000 sürücünün hepsi zikzakta, herkes 80 dakika. Eski rotalar 85 dakikaya çıktığı için boştur. Ortalama süre 65'ten 80 dakikaya, toplam gecikme 260 binden 320 bin araç·dakikaya çıkmıştır.

Şekil 8.10, Şekil 8.11 ve Şekil 8.12 bu süreci üç anda gösterir. Tablo 8.6 aynı süreci sayılarla özetler.

z	Dar yoldaki araç	Dar yol (dk)	Eski rota (dk)	Zikzak (dk)
0	2.000	20	65	40
1.000	2.500	25	70	50
2.000	3.000	30	75	60
3.000	3.500	35	80	70
4.000	4.000	40	85	80

Tablo 8.6: Kestirme açıldıktan sonra geçiş (z: zikzaktaki araç sayısı; kalan trafik iki eski rotaya eşit bölünür).

Tablonun her satırında zikzak eski rotalardan daha hızlıdır (fark 25 dakikadan 5 dakikaya iner). Ama *z* büyüdükçe iki süre de artar: her geçen sürücü, hem zikzaktakileri hem eski rotalardakileri yavaşlatır. Sürücüler her adımda "daha hızlı" rotaya geçer, ama hep birlikte geçtikleri için sonunda herkes başlangıçtakinden daha yavaştır. Kapasite artışı ile bencil seçim birleşince daha kötü bir dengeye varılmıştır.

Örnek 8.9

Aynı ağda (dar yollar *x/100*, geniş yollar 45 dakika) araç sayısı 6.000 olsun. Yeni yoldan önce ve sonra denge süresi bulunur.

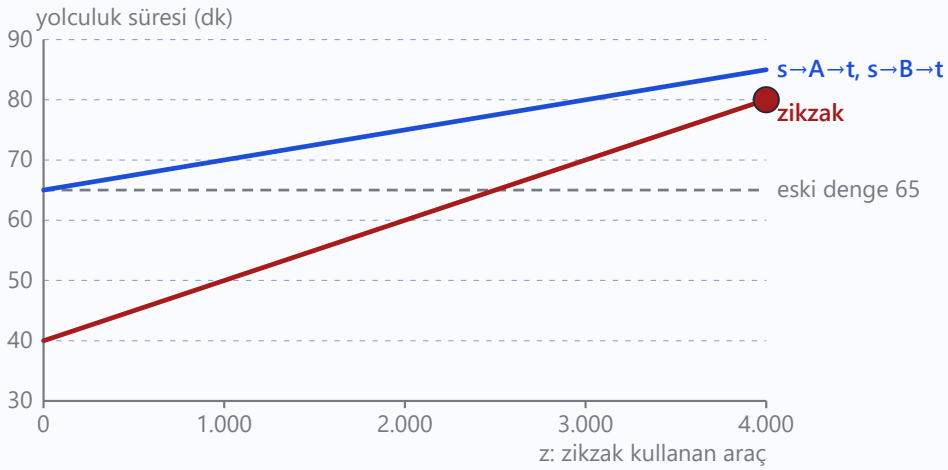
1. **Önce:** simetriyle 3.000 / 3.000. Her rota *30 + 45 = 75* dakika.
2. **Sonra, herkes zikzakta olabilir mi?** Dar yollarda 6.000 araç: zikzak *60 + 0 + 60 = 120* dakika. Eski rotaya dönen *60 + 45 = 105* dakika yaşar. 105 < 120 olduğundan dönmek kârlıdır; bu bir denge değildir.

3. **Trafiğin bölündüğü denge.** Zikzakta z, eski rotaların her birinde y araç olsun. s→A yolunu hem üst rota hem zikzak kullanır; üzerindeki akış $y + z$'dir. B→t için de aynısı geçerlidir. Kullanılan üç rotanın süresi eşit olmalıdır: zikzak $2(y + z)/100$, eski rota $(y + z)/100 + 45$. Eşitlik $(y + z)/100 = 45$, yani $y + z = 4.500$ verir.
4. **Akış korunumu.** Bütün araçlar sayılmalıdır: $2y + z = 6.000$. İki denklemden $y = 1.500$, $z = 3.000$ bulunur.
5. **Süre.** Dar yollarda 4.500 araç (45 dakika). Bütün rotalar $45 + 45 = 90$ dakikadır. Paradoks yine görülür: 75'ten 90'a. Bu kez bazı sürücüler eski rotalarda kalır, ama herkesin süresi 15 dakika uzar.

8.3.2 Neden herkes zikzağa geçer?

4.000 araçlı örnekte herkesin zikzağa geçmesi rastlantı değildir. Zikzak rota, kalan trafik nasıl dağılırsa dağılsın eski rotalardan hızlıdır. Bunu görmek için zikzaktaki araç sayısı z ile gösterilir ve kalan $4.000 - z$ araç iki eski rotaya eşit bölünür.

- Her eski rotada $(4.000 - z)/2 = 2.000 - z/2$ araç vardır.
- s→A'daki akış, üst rotadakiler ile zikzaktakilerin toplamıdır: $2.000 - z/2 + z = 2.000 + z/2$. Süre $20 + z/200$ dakika. B→t için de aynısı geçerlidir.
- Eski rota: bir dar yol ve bir geniş yol, $20 + z/200 + 45 = 65 + z/200$ dakika.
- Zikzak: iki dar yol ve kestirme, $2 \cdot (20 + z/200) + 0 = 40 + z/100$ dakika.



Şekil 8.13: Zikzaktaki araç sayısı z'ye göre yolculuk süreleri. Mavi doğru eski rotalar ($65 + z/200$), kırmızı doğru zikzak ($40 + z/100$). Kırmızı her z için mavinin altındadır; $z = 4.000$'de herkes 80 dakika yaşar. Kesikli çizgi eski denge (65).

Şekil 8.13 iki doğruyu gösterir. Zikzak, eski rotalardan ancak $40 + z/100 \geq 65 + z/200$ olduğunda yavaş olurdu; bu da $z/200 \geq 25$, yani $z \geq 5.000$ demektir. Oysa yalnız 4.000 araç vardır. Demek ki zikzak her durumda daha hızlıdır.

Aynı sonuca daha kısa bir yoldan da varılır. Zikzağın ikinci yarısı B→t'dir; bu yol en kalabalık hâlinde bile en çok *4.000/100 = 40* dakika sürer. Eski alt rotanın ilk yarısı s→B ise her zaman 45 dakikadır. Yani s'den B'ye geçmek için A üzerinden gitmek (s→A en çok 40 dakika) her zaman doğrudan otoyoldan (45 dakika) hızlıdır. Simetrik olarak A'dan t'ye B üzerinden gitmek de her zaman hızlıdır. Zikzak, her sürücü için **baskın bir seçimdir**: ötekiler ne yaparsa yapsın daha iyidir.

Mahkûmun İkilemi ile benzerlik açıktır. O oyunda ihanet her oyuncu için baskın stratejidir, ama herkes ihanet edince herkes kötü duruma düşer. Burada da zikzak herkes için baskındır, ama her geçen sürücü dar yollara yük bindirir. *z = 4.000*'de herkes 80 dakika yaşar; kestirme hiç olmasaydı herkes 65 dakika yaşayacaktı.

8.3.3 Kestirmeli ağın sistem optimumu

Kestirme yolun kendisi kötü değildir. Kötü olan, onun bencil biçimde kullanılmasıdır. Bunu görmek için kestirmeli ağın sistem optimumu hesaplanır: bir planlayıcı trafiği nasıl dağıtırsa ortalama süre en küçük olur?

Örnek 8.10

1. Simetri yüzünden iki eski rotaya eşit sayıda araç (y) verilir; zikzakta z araç olur, $2y + z = 4.000$. Dar yollardaki akış $a = y + z$'dir.
2. Toplam gecikme kenar kenar yazılır. Her dar yolda a araç $a/100$ dakika geçirir: iki dar yol birlikte $2 \cdot a \cdot a/100 = a^2/50$. Geniş yollarda toplam $2y$ araç 45 dakika geçirir: $90y$. Kestirme 0 katkı yapar. $y = 4.000 - a$ olduğundan $C(a) = a^2/50 + 90\,(4.000 - a)$.
3. Türev sıfırlanır: $C'(a) = a/25 - 90 = 0$, yani $a = 2.250$. Buradan $y = 1.750$, $z = 500$.
4. Toplam: $2.250^2/50 + 90 \cdot 1.750 = 101.250 + 157.500 = 258.750$ araç·dakika. Ortalama $258.750 / 4.000 \approx 64{,}7$ dakika.
5. Bu dağılımda dar yollar 22,5 dakika sürer. Zikzaktaki 500 sürücü $22{,}5 + 22{,}5 = 45$ dakika, eski rotalardaki 3.500 sürücü $22{,}5 + 45 = 67{,}5$ dakika yaşar.

Optimumda zikzağa küçük bir pay verilir (500 araç) ve ortalama süre 65'ten 64,7'ye biraz düşer. Kestirme, doğru kullanılırsa küçük de olsa bir iyileşme sağlar. Ama bu dağılım bir denge değildir: eski rotalardaki sürücüler 67,5, zikzaktakiler ise 45 dakika yaşar; eski rotalardakilerin hepsi zikzağa geçmek ister. Tablo 8.7 dört durumu karşılaştırır.

	Bencil denge	**Optimum**
Kestirme yok	20 + 45 = 65	65
Kestirme var	40 + 0 + 40 = 80	≈ 64,7

Tablo 8.7: Braess ağında ortalama yolculuk süresi (dakika).

Tablo, oyunun başındaki iki soruyu cevaplar. Kestirme açılınca yolculuklar kısalmaz, uzar (65 → 80). Herkes kendi en hızlı yolunu seçince ortalama süre en küçük olmaz (80 > 64,7). Kuramın cevabı iki soruya da "hayır"dır.

8.3.4 Paradoksun dersi ve gerçek örnekler

Tanım 8.6

Braess paradoksu (*Braess's paradox*): bir ağa yeni bir bağlantı eklemenin, bencil rota seçimi nedeniyle denge gecikmesini, hatta bütün kullanıcıların gecikmesini artırabilmesidir (Braess, 1968).

Paradoksun mantığı şudur: yeni yol bireysel olarak çekici görünür, ancak kullanıcıların birlikte oluşturduğu trafik deseni daha kötü bir dengeye taşınır. Bunun tersi de doğrudur: bir yolu kapatmak bazen herkesi hızlandırabilir. Braess ağında A→B'yi kapatmak herkesin süresini 80'den 65 dakikaya indirir.

Paradoks yalnız kâğıt üzerinde kalmamıştır. Sıkça anılan birkaç örnek vardır:

- **Stuttgart (1969).** Şehir merkezine yapılan yeni yol yatırımından sonra trafik bozulmuş, bir cadde kapatılınca düzelmiştir (Knödel, 1969).
- **New York (1990).** Dünya Günü'nde 42. Cadde trafiğe kapatıldığında beklenen tıkanıklığın oluşmadığı bildirilmiştir (Kolata, 1990).
- **Seul.** Cheonggyecheon deresi üzerindeki yükseltilmiş otoyol 2000'lerin başında kaldırıldı ve bir kent parkına dönüştürüldü. Bundan sonra şehir trafiğinin beklenen ölçüde kötüleşmediği sıkça anılır.
- **Mekanik ve elektrik ağları.** Yaylardan ve dirençlerden kurulan ağlarda da benzer bir paradoks gösterilmiştir: bir yayı kesmek asılı bir ağırlığı yukarı kaldırabilir (Cohen ve Horowitz, 1991).
- **Laboratuvar deneyleri.** Denekler bilgisayar ortamında rota seçtiğinde, kestirme eklenince zamanla paradoksun öngördüğü daha yavaş dengeye yöneldiler (Rapoport vd., 2009).

Bu örnekler dikkatle yorumlanmalıdır. Gerçek ağlar çok daha karmaşıktır; bir cadde kapatılınca bazı sürücüler toplu taşımaya geçer, bazıları yolculuk saatini değiştirir. Örnekler bu nedenle paradoksun kesin kanıtı değil, olası olduğunun göstergesidir. Ama ders açıktır: yol yapmak ya da kapatmak, yalnız kapasiteye göre değil, sürücülerin varacağı dengeye göre değerlendirilmelidir. Paradoksun etkisini azaltmanın yolları da vardır: ücretlendirme, erişim kısıtı ya da merkezî yönlendirme. Bunlar 8.6'da ele alınır.

8.3.5 Kod izleme: en_iyi_alt_ag()

Braess paradoksu, ağ tasarımcısına yeni bir soru sordurur: verilen ağdan hangi kenarlar kapatılırsa denge süresi en çok düşer? Aşağıdaki fonksiyon bu sorunun basit bir sürümünü çözer: tek bir kenar kapatmayı dener.

```
def en_iyi_alt_ag(ag, r):
  eniyi, sil = denge_suresi(ag, r), None
  for e in ag:
    alt = [k for k in ag if k != e]
    d = denge_suresi(alt, r)   # yol yoksa inf
    if d < eniyi:
      eniyi, sil = d, e
  return sil, eniyi
```

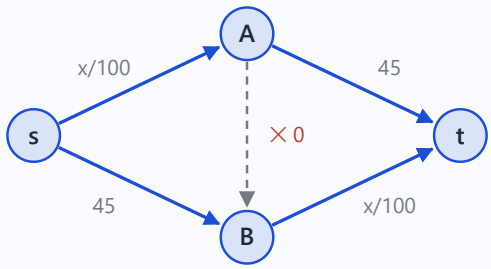
Fonksiyon, `denge_suresi(ag, r)` adlı yardımcı bir fonksiyonu kullanır. Bu fonksiyon, verilen alt ağdaki Wardrop dengesinde kullanılan yolların ortak süresini döndürür. Betikte bu süre Beckmann potansiyeli en aza indirilerek hesaplanmıştır (8.2.5). *s*'den *t*'ye hiç yol kalmazsa sonsuz (`inf`) döner. Kod satır satır şöyle çalışır:

- `eniyi`, o ana kadar bulunan en küçük denge süresidir; başta hiçbir şey silinmemiş ağın süresidir. `sil` hangi kenarın silindiğini tutar; başta `None` (hiçbiri).
- Döngü ağdaki her `e` kenarını sırayla dener.
- `alt`, `e` dışındaki bütün kenarlardan oluşan alt ağdır.
- `d`, bu alt ağın denge süresidir.
- `d` o ana kadarki en iyiden *kesin* küçükse aday güncellenir.
- Sonunda silinecek kenar ve yeni denge süresi döndürülür.

Silinen kenar	Kalan yollar ve denge	`d` (dk)	`d < eniyi` ?	`eniyi, sil`
— (başlangıç)	Herkes zikzakta	80	—	80, None
s→A	Yalnız s→B→t: *40 + 45*	85	hayır	80, None
A→t	Zikzak ve alt; zikzak baskın	80	hayır (eşit)	80, None
s→B	Üst ve zikzak; zikzak baskın	80	hayır (eşit)	80, None
B→t	Yalnız s→A→t: *40 + 45*	85	hayır	80, None
A→B	Braess öncesi ağ: 2.000 / 2.000, *20 + 45*	65	evet	65, A→B

Tablo 8.8: `en_iyi_alt_ag(braess, 4000)` izlemesi.

Tablonun iki satırı ayrıca açıklanmalıdır. A→t kapatılınca iki rota kalır: zikzak (s→A→B→t) ve alt (s→B→t). İkisi de B→t'yi kullanır; bu yüzden B→t'de 4.000 araç vardır ve 40 dakika sürer. Zikzağın ilk kısmı s→A en çok 40 dakika, alt rotanın ilk kısmı s→B ise 45 dakikadır; zikzak yine baskındır. Herkes zikzağa geçer ve süre *40 + 0 + 40 = 80* dakika olur. s→B kapatılınca da simetrik durum ortaya çıkar. Bu iki satırda `d` eski en iyiye eşittir; kod kesin küçüklük aradığı için aday değişmez.



silinen	denge süresi
—	80 dk
s→A	85 dk
A→t	80 dk
s→B	80 dk
B→t	85 dk
A→B	65 dk

eniyi
65 dk
sil = A→B

Şekil 8.14: `en_iyi_alt_ag`'ın sonucu. Solda ağ (A→B kesikli ve çarpı işaretli: kapatılacak kenar), ortada her silme denemesinin denge süresi, sağda sonuç: `eniyi = 65`, `sil = A→B`.

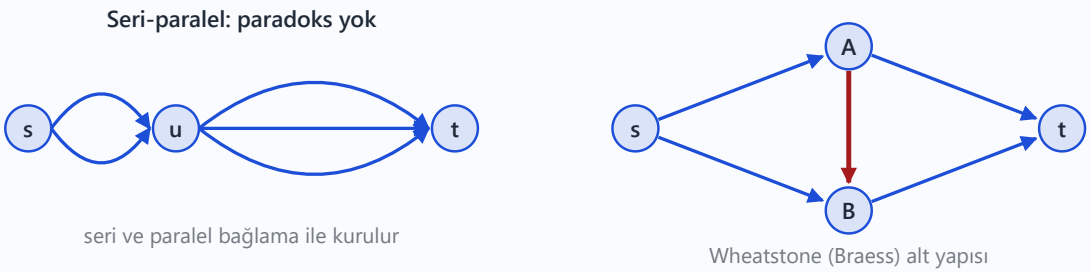
Şekil 8.14 sonucu gösterir: A→B kapatılmalıdır ve denge süresi 80'den 65 dakikaya iner. Kodun bir sınırlılığı vardır: yalnız *tek* kenar silmeyi dener. Genel durumda birden çok kenarı birlikte silmek daha iyi olabilir. m kenarlı bir ağda bütün alt ağları denemek 2^m denge hesabı ister. 30 kenarlı bir ağda bu bir milyardan fazladır; gerçek şehir ağlarında binlerce kenar vardır. Tasarımcının "hangi kenarları açık tutayım?" sorusu, genel ağda bu yüzden zordur.

8.3.6 Bencil kullanıcılar için ağ tasarımı

Bu soru biçimsel olarak şöyle yazılır: verilen ağın hangi alt ağı açık kalırsa Wardrop dengesindeki süre en küçük olur? Burada yeni yol yapmak yerine yol kapatmak da bir tasarım kararıdır. İki sonuç bu problemin zorluğunu ve sınırlarını çizer.

Zorluk. Tim Roughgarden 2006'da bu problemin hesaplama açısından zor olduğunu gösterdi (Roughgarden, 2006). Doğrusal gecikmelerde hiçbir kenarı kapatmamak bile en iyi seçimden en çok *4/3* kat kötüdür. Ama bundan daha iyisini verimli biçimde garanti etmek NP-zordur: verimli bir algoritmanın her ağda "hiçbir şeyi kapatma" seçeneğinden belirgin biçimde iyi bir alt ağ bulması beklenmez. Braess örneğinde oran bu sınırın altındadır: $80/65 \approx 1{,}23 < 4/3 \approx 1{,}33$.

Topoloji. Igal Milchtaich 2006'da hangi ağlarda paradoksun hiç görülemeyeceğini belirledi (Milchtaich, 2006). **Seri-paralel ağlarda** (*series-parallel network*) Braess paradoksu, gecikmeler nasıl seçilirse seçilsin hiç görülmez. Seri-paralel ağ, kenarları ardışık (seri) ya da yan yana (paralel) bağlayarak kurulan ağdır. İki yollu örnek böyledir: iki kenar s ile t arasında paralel bağlanmıştır. Diğer ağlarda ise uygun gecikmelerle paradoks görülebilir.



Şekil 8.15: Solda bir seri-paralel ağ: s ile u arasında iki paralel kenar, u ile t arasında üç paralel kenar, iki grup seri bağlı. Sağda Wheatstone yapısı: A→B köprüsü, ağı seri ve paralel bağlamayla kurulamaz hâle getirir.

Şekil 8.15 iki yapıyı karşılaştırır. Braess ağı, sağdaki **Wheatstone yapısıdır** (*Wheatstone network*); adını elektrik devrelerindeki Wheatstone köprüsünden alır. A→B köprüsü iki paralel rotayı "çaprazlar". Seri ve paralel bağlama işlemleriyle böyle bir çapraz kurulamaz. Milchtaich'in sonucuna göre iki uçlu bir ağın paradokstan bağışık olması için gerekli ve yeterli koşul, Wheatstone yapısını içermemesidir; bu da ağın seri-paralel olmasıyla aynı şeydir. Döngülü ağlar ya da tam çizgeler genellikle Wheatstone yapısı içerir ve böyle bir güvence vermez.

8.3.7 Braess paradoksunda sık hatalar

Sık yapılan hata

- **Kestirmeyi "kötü yol" sanmak:** Kestirmenin kendisi kötü değildir; optimumda kullanılır ve ortalamayı biraz düşürür (64,7). Sorun bencil kullanımdır.
- **Akış korunumunu unutmak:** s→A'daki akış yalnız üst rotadakiler değildir; zikzaktakiler de o yolu kullanır ($y + z$).
- **Her zaman herkesin zikzağa geçeceğini sanmak:** 6.000 araçta zikzak baskın değildir; denge trafiği böler (3.000 zikzakta, 1.500'er eski rotalarda).
- **Paradoksun her ağda ve her trafik düzeyinde görüleceğini sanmak:** Seri-paralel ağlarda hiç görülmez. Braess ağında bile trafik çok azsa kestirme gerçekten yararlı olabilir; paradoks belirli bir trafik aralığında ortaya çıkar.
- **Kapatma kararını tek kenarla sınırlamak:** En iyi alt ağ birden çok kenarın kapatılmasını gerektirebilir; tek kenar denemek yalnız bir sezgisel yöntemdir.

8.4 Tıkanıklık oyunları

Bölünebilir trafik modeli binlerce küçük kullanıcı için uygundur. Ama bazı durumlarda oyuncu sayısı azdır ve her birinin etkisi belirgindir: birkaç büyük taşımacılık şirketi, birkaç veri merkezi ya da bir ağda paket gönderen birkaç uygulama. Oyuncuların tek tek sayılabildiği bu modele **atomik** model

denir. Bu modelde de bencil seçimlerin bir dengeye varacağını gösteren güçlü bir araç vardır: Rosenthal potansiyeli.

8.4.1 Tıkanıklık oyunu: atomik ve bölünebilir

Tanım 8.7

Bir **tıkanıklık oyunu** (*congestion game*) şu bileşenlerden oluşur:

- n oyuncu ve bir kaynak kümesi E (örneğin bir ağın kenarları).
- Her oyuncunun stratejisi bir kaynak alt kümesidir (örneğin bir s–t yolu, yani o yolun kenarları).
- Her e kaynağının maliyeti $c_e(n_e)$ yalnızca onu kullanan oyuncu sayısı n_e'ye bağlıdır ve bu yükle artar.
- Bir oyuncunun maliyeti, seçtiği kaynakların maliyetleri toplamıdır: $c_i(s) = \sum_{e \in s_i} c_e(n_e)$.

Tanımdaki önemli nokta, bir kaynağın maliyetinin *kimin* kullandığına değil, *kaç kişinin* kullandığına bağlı olmasıdır. Bir yolda Ali ile Ayşe ya da Can ile Deniz olması fark etmez; iki kişi varsa maliyet $c_e(2)$'dir. Oyuncular kendi gecikmelerini azaltmaya çalışırken toplam ağ maliyeti yükselebilir; Braess paradoksu atomik modelde de görülür (8.4.3).

Atomik model ile bölünebilir model aynı fikrin iki yüzüdür. Tablo 8.9 farkları özetler.

	Atomik	**Bölünebilir**
Oyuncu	sonlu, n kişi	sürekli akış
Etki	tek kişi yükü 1 artırır	tek kişi ihmal edilir
Denge	saf Nash	Wardrop
Potansiyel	Rosenthal (toplam)	Beckmann (integral)

Tablo 8.9: Atomik ve bölünebilir tıkanıklık modelleri.

Atomik modelde bir oyuncu yol değiştirince yeni yolundaki her kenarın kullanıcı sayısı bir artar. Bu yüzden oyuncu, yeni yolun maliyetini hesaplarken kendisini de saymalıdır: şu an 2 kişinin kullandığı bir kenara geçen oyuncu $c_e(2)$ değil $c_e(3)$ öder. Bu küçük ayrıntı, atomik hesaplarda en sık yapılan hatanın kaynağıdır.

Robert Rosenthal 1973'te tıkanıklık oyunlarının çok önemli bir özelliğini gösterdi: sonlu her tıkanıklık oyununda bir potansiyel fonksiyonu vardır, bu yüzden **saf Nash dengesi** her zaman bulunur (Rosenthal, 1973). Bu dikkat çekici bir sonuçtur. Genel oyunlarda saf denge olmayabilir; Taş-Kağıt-Makas'ta yalnız karışık denge vardır. Tıkanıklık oyunlarında ise karışık stratejiye gerek kalmaz.

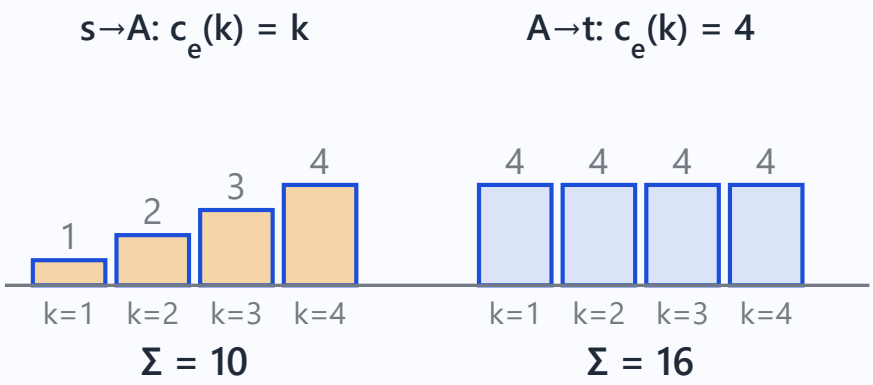
8.4.2 Rosenthal potansiyel fonksiyonu

Tanım 8.8

Bir s strateji profilinde n_e, e kaynağını kullanan oyuncu sayısı olsun. **Rosenthal potansiyeli** (*Rosenthal potential*) şudur:

$$\Phi(s) = \Sigma_e \left[c_e(1) + c_e(2) + \ldots + c_e(n_e) \right]$$

Her kenarda 1'den kullanıcı sayısı n_e'ye kadar maliyetler toplanır. Bu, toplam maliyet değildir. Toplam maliyette o kenardaki n_e kullanıcının her biri $c_e(n_e)$ öder ve kenarın katkısı $n_e \cdot c_e(n_e)$ olur. Potansiyelde ise kullanıcılar kenara sırayla giriyormuş gibi düşünülür: birinci gelen $c_e(1)$, ikinci gelen $c_e(2)$, son gelen $c_e(n_e)$ "öder". Beckmann potansiyelindeki "eğrinin altındaki alan"ın atomik karşılığı budur: integral yerine toplam gelir.



Şekil 8.16: Rosenthal potansiyelinin iki kenara katkısı. s→A kenarında $c_e(k) = k$: dört kullanıcıyla katkı $1 + 2 + 3 + 4 = 10$. A→t kenarında $c_e(k) = 4$: katkı $4 + 4 + 4 + 4 = 16$.

Şekil 8.16 bu toplamı çubuklarla gösterir. Maliyeti kullanıcı sayısıyla artan s→A kenarında çubuklar merdiven gibi yükselir; maliyeti sabit A→t kenarında hepsi aynı boydadır.

Örnek 8.11

Braess ağının atomik sürümünde 4 sürücü vardır. Kenar maliyetleri: s→A ve B→t'de $c_e(k) = k$ (k: kullanıcı sayısı), s→B ve A→t'de 4, A→B'de 0. 4 sürücünün hepsi s→A→t (üst) yolundadır.

1. s→A'da 4 kullanıcı vardır: katkı $1 + 2 + 3 + 4 = 10$.
2. A→t'de 4 kullanıcı vardır: katkı $4 + 4 + 4 + 4 = 16$.
3. s→B, B→t ve A→B boştur: katkı 0.
4. $\Phi = 10 + 16 = 26$.

5. Karşılaştırma için toplam maliyet: her sürücü $c(s \to A) + c(A \to t) = 4 + 4 = 8$ öder; toplam $4 \times 8 = 32$. Potansiyel (26) toplam maliyetten (32) farklıdır.

Potansiyeli değerli kılan şu özdeşliktir.

Teorem 8.2 (Rosenthal)

Bir tıkanıklık oyununda tek bir i oyuncusu stratejisini değiştirirse (ötekiler sabitken) potansiyeldeki değişim, tam olarak o oyuncunun maliyetindeki değişime eşittir:

$$\Delta\Phi = c_i(\text{yeni}) - c_i(\text{eski})$$

Bu yüzden sonlu her tıkanıklık oyununda saf Nash dengesi vardır ve iyileştiren hamleler dizisi her zaman sonlu sayıda adımda durur.

İspat. i oyuncusu eski stratejisi P'den yeni stratejisi P''ye geçsin. Kenarlar üç gruba ayrılır ve her grupta Φ'nin nasıl değiştiğine bakılır.

(1) Hem P'de hem P''de olan kenarlar: i bu kenarları kullanmayı sürdürür, kullanıcı sayısı değişmez. Bu kenarların Φ'ye katkısı aynı kalır. i'nin bu kenarlarda ödediği maliyet de aynı kalır.

(2) Yalnız P'de olan kenarlar: i bu kenarlardan çıkar ve kullanıcı sayısı n_e'den $n_e - 1$'e düşer. Bu kenarın toplamındaki son terim $c_e(n_e)$ silinir. Bu terim, i'nin eski durumda o kenarda ödediği maliyettir.

(3) Yalnız P''de olan kenarlar: i bu kenarlara girer ve kullanıcı sayısı n_e'den $n_e + 1$'e çıkar. Toplama yeni bir terim $c_e(n_e + 1)$ eklenir. Bu terim, i'nin yeni durumda o kenarda ödeyeceği maliyettir.

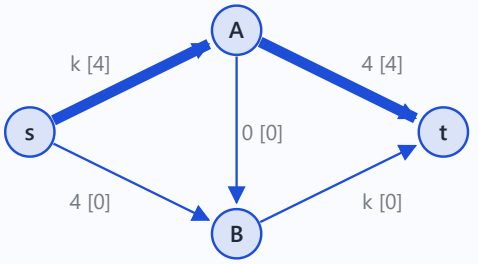
Üç grup birleştirilince Φ'deki değişim, "eklenen terimler eksi silinen terimler"dir; bu da i'nin yeni maliyeti eksi eski maliyetidir. Ortak kenarlar iki tarafta da aynı olduğu için farkta birbirini götürür. Böylece $\Delta\Phi = c_i(\text{yeni}) - c_i(\text{eski})$ elde edilir.

Bir oyuncu maliyetini düşüren bir hamle yaparsa ($c_i(\text{yeni}) < c_i(\text{eski})$), Φ kesin olarak azalır. Oyun sonlu olduğu için strateji profillerinin sayısı da sonludur; Φ bu yüzden sonlu sayıda değer alır. Kesin azalan bir dizi sonlu sayıda değer arasında sonsuza kadar süremez. Demek ki iyileştiren hamleler bir noktada durur. Durduğu yerde hiçbir oyuncu tek başına maliyetini düşüremez; bu tanım gereği bir saf Nash dengesidir. ■

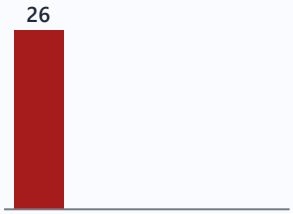
İspatın bir sonucu daha vardır: Φ'nin bütün profiller içindeki en küçük değerini veren profil de bir saf Nash dengesidir. Çünkü orada herhangi bir oyuncunun maliyetini düşüren bir hamle Φ'yi daha da küçültürdü; bu, en küçük değerde olmakla çelişir.

8.4.3 En iyi tepki dinamiği ve Φ adım adım

Teoremi işlerken görmek için Örnek 8.11'deki atomik Braess ağına **en iyi tepki dinamiği** uygulanır. Sürücüler 1, 2, 3, 4 sırasıyla hamle yapar. Sırası gelen sürücü, ötekiler sabitken maliyeti en küçük olan yolu seçer. Üç yol vardır: üst (s→A→t), alt (s→B→t) ve zik (s→A→B→t). Eşitlik olursa sürücü bu sıradaki ilkini seçer; ama yalnız kesin iyileşme varsa yerini değiştirir.



sürücü	yol	maliyet
1	üst	8
2	üst	8
3	üst	8
4	üst	8



Şekil 8.17: Başlangıç: dört sürücü de üst yolda. Kenar etiketlerinde maliyet fonksiyonu ve köşeli parantez içinde kullanıcı sayısı. Herkesin maliyeti *4 + 4 = 8*, *Φ = 26*.

Şekil 8.17 başlangıç durumunu gösterir. Dinamik dört hamlede şöyle ilerler.

Örnek 8.12

1. **Sürücü 1** (üst, 8). Üst yolda kalırsa 8 öder. Alt yola geçerse s→B 4 ve B→t'de tek kullanıcı (kendisi) olur: *4 + 1 = 5*. Zik'e geçerse s→A'da kullanıcı sayısı yine 4'tür (o zaten oradadır): *4 + 0 + 1 = 5*. Alt ve zik eşittir; sıradaki ilk olan alt seçilir. Maliyet 8'den 5'e iner. *Φ*: 26 → 23 (*Δ = −3*).
2. **Sürücü 2** (üst, artık 7: s→A'da 3 kişi). Üst 7. Alt: s→B 4, B→t'de sürücü 1 ile birlikte 2 kişi: *4 + 2 = 6*. Zik: s→A 3, A→B 0, B→t 2: *3 + 0 + 2 = 5*. En iyisi zik. Maliyet 7'den 5'e iner. *Φ*: 23 → 21 (*Δ = −2*).
3. **Sürücü 3** (üst, 7). Üst 7. Alt: *4 + 3 = 7*. Zik: *3 + 0 + 3 = 6*. En iyisi zik. Maliyet 7'den 6'ya iner. *Φ*: 21 → 20 (*Δ = −1*).
4. **Sürücü 4** (üst, 7). Alt: *4 + 4 = 8*. Zik: *3 + 0 + 4 = 7*. Daha iyisi yoktur; kalır.
5. **İkinci tur.** Sürücü 1 (alt, 7): üst *4 + 4 = 8*, zik *4 + 0 + 3 = 7* (s→A'ya girince orada 4 kişi olur); kesin iyileşme yok. Sürücü 2 ve 3 (zik, 6): üst 7, alt 7; kalırlar. Sürücü 4 (üst, 7): yine kalır. Kimse iyileştiremez: saf Nash dengesi, *Φ = 20*.

Her adımda *Φ*'nin düşüşü, hamle yapan sürücünün kazancına eşittir: 3, 2, 1. Bu, teoremin özdeşliğinin sayılarla doğrulanmasıdır. Örneğin sürücü 1'in hamlesinde s→A ve A→t kenarlarından birer terim silinir (4 ve 4, toplam 8: onun eski maliyeti); s→B ve B→t kenarlarına birer terim eklenir (4 ve 1, toplam 5: onun yeni maliyeti). *ΔΦ = 5 − 8 = −3*.

Hamle	Mevcut	üst / alt / zik	Seçim	Φ
Başlangıç	herkes üst, 8	—	—	26
Sürücü 1	üst, 8	8 / 5 / 5	alt	26 → 23
Sürücü 2	üst, 7	7 / 6 / 5	zik	23 → 21
Sürücü 3	üst, 7	7 / 7 / 6	zik	21 → 20
Sürücü 4	üst, 7	7 / 8 / 7	kalır	20
2. tur	(alt, zik, zik, üst)	kimse iyileştiremez	—	20

Tablo 8.10: En iyi tepki dinamiğinin izlemesi (seçenek maliyetleri: üst / alt / zik).



Şekil 8.18: Dinamiğin sonu: (alt, zik, zik, üst) dengesi. Maliyetler 7, 6, 6, 7; sağdaki çubuklar Φ'nin 26 → 23 → 21 → 20 düşüşünü gösterir.

Şekil 8.18 dengeyi gösterir. Toplam maliyet *7 + 6 + 6 + 7 = 26*'dır. A→B kenarı hiç olmasaydı sürücüler 2 + 2 bölünürdü: üst yolda 2 kişi (s→A 2, A→t 4: maliyet 6), alt yolda 2 kişi (s→B 4, B→t 2: maliyet 6). Toplam 24 olurdu. Kestirme, toplam maliyeti 24'ten 26'ya çıkarmıştır. Bu, Braess paradoksunun atomik sürümüdür.

Bir ayrıntı daha önemlidir: dinamiğin vardığı denge, oyunun tek dengesi değildir. Bütün profiller taranınca, sürücülerin adları göz ardı edilirse, dört saf denge bulunur: (alt, zik, zik, üst), (alt, zik, zik, zik), (zik, zik, zik, üst) ve dördünün de zik'te olduğu profil. Toplam maliyetleri sırasıyla 26, 29, 29 ve 32'dir. Dördünde de $\Phi = 20$'dir ve bu, potansiyelin en küçük değeridir. Hangi dengeye varılacağı, hamle sırasına ve eşitlik kuralına bağlıdır.

Örnek 8.13

Dengede (alt, zik, zik, üst) Φ doğrudan hesaplanır ve sürücü 4'ün zik'e geçmesinin etkisine bakılır.

1. Kullanıcı sayıları: s→A'yı iki zik ve bir üst sürücüsü kullanır (3 kişi); B→t'yi alt ve iki zik kullanır (3 kişi); A→t'yi yalnız üst (1); s→B'yi yalnız alt (1); A→B'yi iki zik (2).
2. $\Phi = (1 + 2 + 3) + (1 + 2 + 3) + 4 + 4 + (0 + 0) = 20$.
3. Sürücü 4 zik'e geçerse: s→A'da zaten oradadır (3 kişi, 3); A→B 0; B→t'de kullanıcı sayısı 4'e çıkar (4). Yeni maliyet *3 + 0 + 4 = 7*. Eski maliyeti de 7'dir.

4. Maliyet değişmediği için $\Delta\Phi = 0$. Kontrol: A→t'den 4 silinir, A→B'ye 0 ve B→t'ye 4 eklenir; fark 0'dır.

8.4.4 Potansiyel oyunlar: ne kazandık?

Rosenthal'ın fikri tıkanıklık oyunlarının ötesine genellenir. Dov Monderer ve Lloyd Shapley 1996'da bu genellemeye ad verdi (Monderer ve Shapley, 1996).

Tanım 8.9

Her oyuncunun her tek taraflı sapmasındaki maliyet değişimini aynen veren bir Φ fonksiyonu varsa oyuna **(tam) potansiyel oyun** (*potential game*) denir: her i, her s_i, s_i' ve her s_{-i} için

$$\Phi(s_i', s_{-i}) - \Phi(s_i, s_{-i}) = c_i(s_i', s_{-i}) - c_i(s_i, s_{-i})$$

Φ'ye **potansiyel fonksiyonu** (*potential function*) denir.

Potansiyel fonksiyonu, bütün oyuncuların bencil hamlelerini tek bir "ortak ölçü"de toplar. Her oyuncu kendi maliyetini düşürmek ister; potansiyel oyunda bunu yapan her hamle aynı ortak ölçüyü de düşürür. Oyuncular farkında olmadan aynı fonksiyonu küçültmek için çalışıyor gibidir. Bu bakış birkaç kazanç sağlar:

- **Varlık.** Φ'nin en küçük değerini veren durum bir saf Nash dengesidir; denge her zaman vardır.
- **Yakınsama.** İyileştiren hamleler dizisi sonsuza gidemez. Bu özelliğe **sonlu iyileşme özelliği** (*finite improvement property*) denir. Her en iyi tepki dinamiği durur.
- **Öğrenme.** Oyuncuların dengeyi "bilmesi" gerekmez. Sırayla en iyi tepki vermeleri dengeye ulaşmak için yeter. Bu, dengenin gerçek hayatta neden ortaya çıkabileceğine dair güçlü bir açıklamadır.
- **Verimlilik.** Φ'yi en küçük yapan denge, dengeler içinde iyi bir denge seçmenin bir yoludur; çünkü Φ toplam maliyetle yakından ilişkilidir. Ama bu bir garanti değildir: Örnek 8.13'teki ağda dört dengenin hepsinde $\Phi = 20$'dir, toplam maliyetleri ise 26 ile 32 arasında değişir.
- **Genellik.** Her (tam) potansiyel oyun bir tıkanıklık oyununa denktir (Monderer ve Shapley, 1996). Yani tıkanıklık oyunları potansiyel oyunların "tipik" örneği değil, tam karşılığıdır.

Bir sınır da vardır: **hız**. Dinamiğin durması garantidir, hızlı durması değil. Φ her adımda en az bir miktar azalır, ama başlangıç değeri çok büyük olabilir ve adım sayısı oyuncu sayısıyla üstel olarak büyüyebilir. Alex Fabrikant, Christos Papadimitriou ve Kunal Talwar 2004'te genel tıkanıklık oyunlarında saf denge bulmanın **PLS-tam** (*PLS-complete*) olduğunu gösterdi (Fabrikant, Papadimitriou, Talwar, 2004). PLS, "yerel en iyi" bir çözüm bulma problemlerinin karmaşıklık sınıfıdır. PLS-tam olmak, bazı tıkanıklık oyunlarında dengeyi bulmak için hızlı bir yöntemin bilinmediği anlamına gelir. Basit durumlarda (örneğin bütün oyuncuların aynı kaynaktan aynı hedefe gittiği simetrik ağ oyunlarında) ise denge, bir akış problemine indirgenerek hızlı bulunabilir.

8.4.5 Potansiyelde sık hatalar

Sık yapılan hata

- **Potansiyeli toplam maliyet sanmak:** $c_e(k) = k$ olan bir kenarda 4 kullanıcı için Φ katkısı $1 + 2 + 3 + 4 = 10$, toplam maliyet katkısı $4 \cdot 4 = 16$'dır.
- **Yeni yolda kendini saymamak:** Bir kenara geçen oyuncu o kenarın kullanıcı sayısını bir artırır. Şu an 3 kişinin kullandığı kenara geçen $c_e(4)$ öder.
- **Ortak kenarı iki kez değiştirmek:** Oyuncu eski ve yeni yolunda ortak olan bir kenarda kalır; o kenarın kullanıcı sayısı değişmez (sürücü 1 alt yerine zik'e geçseydi s→A'da 4 kişi kalırdı).
- **Dengeyi Φ'nin en büyük noktası sanmak:** Maliyet azaltan hamleler Φ'yi azaltır; denge, Φ'nin (yerel) en küçük noktasıdır.
- **Durmayı hız sanmak:** Sonlu iyileşme özelliği dinamiğin durduğunu söyler, kaç adımda durduğunu söylemez; genel durumda bu sayı üstel olabilir.

8.5 Ağ oluşturma

Buraya kadar ağ verili kabul edildi; kullanıcılar yalnız hangi yoldan gideceklerine karar veriyordu. Ağ oluşturma oyunlarında ise ağın kendisini oyuncular kurar. İnternette ağ işletmecileri birbirleriyle hangi bağlantıları kuracaklarına kendileri karar verir; sosyal ağlarda insanlar kiminle arkadaş olacaklarını seçer; eşler arası (P2P) dosya paylaşım ağlarında her bilgisayar hangi bilgisayarlara bağlanacağını seçer. Bu kararların arkasındaki temel kavram dışsallıktır.

8.5.1 Ağ dışsallıkları: olumsuz ve olumlu

Tanım 8.10

Dışsallık (*externality*): bir kullanıcının kararı başkalarının maliyetini ya da faydasını değiştirir, ama kullanıcı bu etkiyi kendi hesabına katmaz. Etki başkalarına zarar veriyorsa *olumsuz*, yarar sağlıyorsa *olumlu* dışsallıktır.

Olumsuz dışsallık: tıkanıklık. Bir yola giren her ek sürücü, aynı yoldaki diğer x kişinin her birini $\ell'(x)$ kadar yavaşlatır; toplamda $x \cdot \ell'(x)$ gecikme yükler. İki yollu örnekte Yol 1'de 2.000 araç varken ($\ell_1' = 1/100$) bu yük $2.000 \times 1/100 = 20$ araç·dakikadır. Yeni sürücü kendi 25 dakikasını görür, ötekilere yüklediği 20 araç·dakikayı görmez. Sonuç: kalabalık yol gereğinden çok kullanılır. Bu, 8.2.4'te dengenin optimumdan sapmasının nedenidir. Çare **tıkanıklık ücreti**dir: sürücüye ötekilere yüklediği gecikme kadar bir ücret ödetilirse sürücü bu etkiyi de hesaba katar (8.6.2).

Olumlu dışsallık: ağ etkisi. Bazı ağlara katılmanın değeri, katılan sayısıyla artar. Tek kişinin telefonu işe yaramaz; iki kişi birbirini arayabilir; yüz kişilik bir ağda her kullanıcı 99 kişiye ulaşır. Mesajlaşma uygulamaları ve sosyal ağlar da böyledir. n kullanıcılı bir ağda $n(n-1)/2$ olası bağlantı vardır: 10 kullanıcıda 45, 100 kullanıcıda 4.950 bağlantı. Ağın değerinin bu yüzden n^2 ile büyüdüğü iddiasına **Metcalfe yasası** denir. Bu iddia kesin bir yasa değil, kaba bir tahmindir: bütün bağlantılar eşit değerli değildir ve çoğu kullanıcı ağdakilerin küçük bir kısmıyla iletişim kurar.

Ağ etkisinde de dışsallık vardır: yeni bir kullanıcı ağa katılınca ötekilere de değer katar (onlar artık bir kişiye daha ulaşabilir), ama katılıp katılmamaya karar verirken yalnız kendi faydasına bakar. Bu yüzden benimseme, **kritik kitle** aşılana kadar yavaş kalır: az kişinin kullandığı bir ağa katılmak kimseye çekici gelmez.

Ağ oluşturma oyununda iki dışsallık bir aradadır. Bir bağ yalnız kuranın değil, başkalarının da uzaklığını kısaltır: bağı kuranın yanında, o bağ üzerinden geçen herkes yararlanır (olumlu dışsallık). Ama bağın bedelini yalnız onu satın alan öder. Oyuncular bu yüzden toplum açısından en iyi olandan daha az bağ kurabilir.

8.5.2 Ağ oluşturma oyunları

Ağ oluşturma modellerinde bağlantıya (*link*) kısaca **bağ** da denir. Her bağın bir kurulum maliyeti vardır; iyi bağlı olmak, yani ötekilere kısa uzaklıkta olmak ise değerlidir. Modelin temel gerilimi budur: bağ maliyeti ile erişim arasında bir denge kurmak. İki temel model vardır.

Tanım 8.11

Tek taraflı ağ oluşturma oyunu (*network formation game*; Fabrikant vd., 2003). n oyuncu vardır. Her i oyuncusu, istediği düğümlere bağ satın alır; her bağın fiyatı α'dır. Kim satın alırsa alsın, bir bağı herkes iki yönde kullanır. Oyuncunun maliyeti:

$$\text{maliyet}_i = \alpha \cdot (i\text{'nin aldığı bağ sayısı}) + \Sigma_j \, \text{uzaklık}(i, j)$$

Uzaklık, iki düğüm arasındaki en kısa yoldaki bağ sayısıdır; ağ kopuksa ulaşılamayan düğüme uzaklık sonsuzdur. Denge kavramı Nash dengesidir. **Sosyal maliyet** (*social cost*), bütün oyuncuların maliyetlerinin toplamıdır.

İki taraflı model (Jackson ve Wolinsky, 1996). Bir bağ ancak iki ucundaki oyuncu da isterse kurulur; biri isterse kopar. Denge kavramı **ikili kararlılıktır** (*pairwise stability*).

İki model gerçek hayattaki farklı durumlara karşılık gelir. Tek taraflı model, bir web sitesinin başka sitelere bağlantı vermesine ya da bir ağ işletmecisinin tek başına bir hat kiralamasına benzer: bağı kuran öder, karşı tarafın onayı gerekmez. İki taraflı model ise arkadaşlığa ya da iki ağ işletmecisinin karşılıklı **internet eşlemesi** (*peering*) anlaşmasına benzer: iki taraf da razı olmalıdır.

Her iki modelde de üç soru sorulur. Hangi ağlar dengedir? Sosyal olarak en iyi ağ hangisidir? İkisi ne kadar farklı olabilir? Bu sorular, bencil yönlendirmedeki "denge mi, optimum mu?" sorusunun ağ kurmadaki karşılığıdır.

8.5.3 İki taraflı bağ: ikili kararlılık

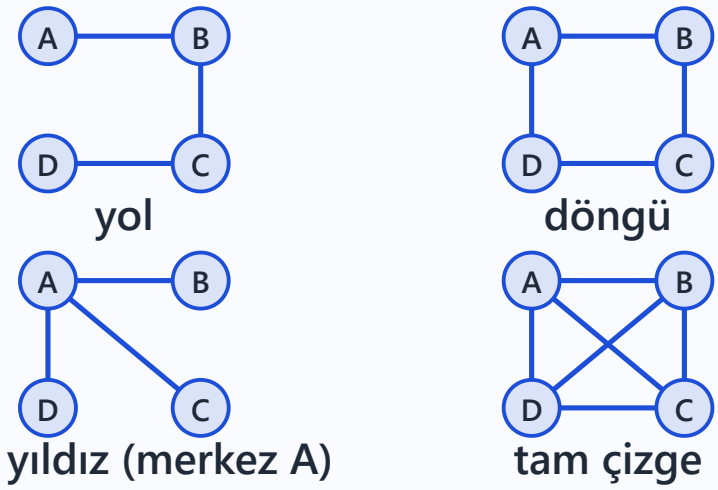
Tanım 8.12

Aşağıdaki iki koşulu sağlayan ağa **ikili kararlı** (*pairwise stable*) denir:

1. hiçbir oyuncu kendi bağlarından birini koparınca kazanmaz;
2. olmayan hiçbir bağ iki ucunu birlikte kazançlı çıkarmaz (biri kesin kazanıp öteki kaybetmiyorsa bağ kurulur).

Tanım Nash dengesinden biraz farklıdır. Nash dengesinde tek bir oyuncunun sapması incelenir. İkili kararlılıkta bağ koparmak tek kişinin kararıdır, ama bağ kurmak iki kişinin ortak kararıdır. Kurulmamış bir bağ değerlendirilirken iki ucun da kazancına bakılır.

Somut bir örnek için $n = 4$ oyuncu (A, B, C, D) alınır. Bir oyuncunun faydası, ötekilere uzaklığına göre şöyle hesaplanır: uzaklığı 1 olan her oyuncudan 4, uzaklığı 2 olandan 2,2, uzaklığı 3 olandan 1 fayda. Her bağın bedeli, iki uçtaki oyuncuların her birine 2'dir. Oyuncunun net faydası, uzaklık faydalarının toplamından bağ bedellerinin çıkarılmasıyla bulunur. Şekil 8.19 karşılaştırılacak dört ağı gösterir.



Şekil 8.19: Dört oyunculu dört ağ: yol (A–B–C–D), döngü, yıldız (merkez A) ve tam çizge.

Örnek 8.14

1. **Yol A–B–C–D.** A'nın uzaklıkları: B 1, C 2, D 3. Fayda *4 + 2,2 + 1 = 7,2*; bir bağı vardır, bedel 2; net 5,2. B'nin uzaklıkları: A 1, C 1, D 2. Fayda *4 + 4 + 2,2 = 10,2*; iki bağ, bedel 4; net 6,2. Simetriyle C 6,2, D 5,2. Toplam 22,8.
2. **Yol ikili kararlı mı?** A–D bağı eklensin. A için D'nin uzaklığı 3'ten 1'e iner: fayda *4 − 1 = 3* artar, bedel 2'dir; net +1. D için de aynısı geçerlidir. İki uç da kazanır: bağ kurulur. Yol ikili kararlı değildir.
3. **Döngü.** Herkesin iki komşusu (uzaklık 1) ve bir karşı düğümü (uzaklık 2) vardır. Fayda *4 + 4 + 2,2 = 10,2*, iki bağ bedeli 4; net 6,2. Toplam 24,8. Bir oyuncu bağ koparırsa döngü bir yola dönüşür ve o oyuncu yolun ucu olur: net 5,2'ye düşer; kaybeder. Bir köşegen eklenirse (A–C) iki uçta uzaklık 2'den 1'e iner: fayda *4 − 2,2 = 1,8* artar, bedel 2'dir; net −0,2. Kimse eklemek istemez. Döngü ikili kararlıdır.
4. **Yıldız (merkez A).** Merkezin üç komşusu vardır: fayda 12, bedel 6; net 6. Her yaprağın bir komşusu (merkez) ve iki uzaklığı 2 olan yaprağı vardır: fayda *4 + 2,2 + 2,2 = 8,4*, bedel 2; net 6,4. Toplam *6 + 3 × 6,4 = 25,2*. Merkez bir bağı koparırsa o yaprak ağdan kopar ve merkez 4 fayda kaybederken yalnız 2 kazanır. Yaprak bağını koparırsa bütün faydasını kaybeder. İki yaprak bağ kursa her biri 1,8 kazanır ama 2 öder: eklemezler. Yıldız ikili kararlıdır.
5. **Tam çizge.** Herkes herkese bağlıdır: fayda 12, üç bağ bedeli 6; net 6. Toplam 24. Bir oyuncu bir bağı koparırsa o komşuya uzaklığı 1'den 2'ye çıkar (ötekiler üzerinden): 1,8 fayda kaybeder ama 2 bedel kazanır; net +0,2. Tam çizge ikili kararlı değildir.

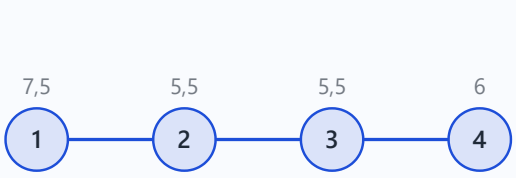
Ağ	Net faydalar	Toplam	İkili kararlı?
Yol	5,2 / 6,2 / 6,2 / 5,2	22,8	Hayır: A–D eklenir (her biri +3 fayda, −2 bedel)
Döngü	6,2 (hepsi)	24,8	Evet
Yıldız	6 / 6,4 / 6,4 / 6,4	25,2	Evet; toplamı en yüksek ağ
Tam çizge	6 (hepsi)	24	Hayır: bağ koparan 2 kazanır, 1,8 kaybeder

Tablo 8.11: Dört oyunculu ağlarda net faydalar ve ikili kararlılık.

Tablo 8.11 iki dersi birlikte verir. Birincisi, ikili kararlı ağlar birden fazla olabilir: hem döngü hem yıldız kararlıdır. Dört oyunculu bütün ağlar (64 olası bağ kümesi) taranınca yalnız dört yıldız (merkezi farklı) ve üç döngü (düğüm sırası farklı) ikili kararlı çıkar. İkincisi, kararlı bir ağ toplamı en yüksek ağla örtüşmek zorunda değildir. Yıldız, bütün ağlar içinde toplamı en yüksek olandır (25,2). Döngü kararlıdır ama yıldızdan 0,4 geridedir. Oyuncular döngüye varırsa oradan kendiliğinden çıkmazlar; tek bir oyuncunun ya da tek bir çiftin hamlesi döngüyü yıldıza dönüştüremez. Ağ Oluşturma oyununun “Social” senaryosu (4 düğüm) bu sayıları kullanır.

8.5.4 Tek taraflı bağ: ağ oluşturma oyunu adım adım

Tek taraflı modelde örnek şöyledir: $n = 4$ oyuncu, bağ fiyatı $\alpha = 1{,}5$. Her oyuncu hangi bağları kendisi satın alacağını seçer; bağı kim alırsa alsın herkes kullanır. Bir oyuncunun maliyeti, aldığı bağ sayısının 1,5 katı ile ötekilere uzaklıklarının toplamıdır. Ağ kopuksa maliyet sonsuzdur; bu yüzden hiçbir oyuncu ağı koparan bir hamle yapmaz.



Yol 1–2–3–4 (i, i+1'i alır)

maliyetler 7,5 · 5,5 · 5,5 · 6

sosyal maliyet 24,5

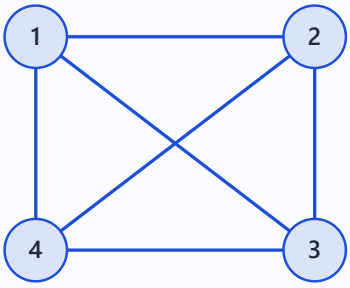
Şekil 8.20: Yol ağı 1–2–3–4; her *i* oyuncusu *(i, i + 1)* bağını alır, 4 hiç bağ almaz. Düğümlerin üstünde maliyetler: 7,5 · 5,5 · 5,5 · 6; sosyal maliyet 24,5.

Şekil 8.20 başlangıç noktası olan yol ağını gösterir. Bu ağdan başlayarak birkaç aday ağ sırayla denetlenir.

Örnek 8.15

1. **Yol ağı.** Oyuncu 1 bir bağ alır (1,5); uzaklıkları 2'ye 1, 3'e 2, 4'e 3, toplam 6. Maliyeti $1{,}5 + 6 = 7{,}5$. Oyuncu 2: bir bağ (1,5), uzaklıklar $1 + 1 + 2 = 4$, maliyet 5,5. Oyuncu 3: aynı biçimde 5,5. Oyuncu 4: bağ almaz, uzaklıklar $3 + 2 + 1 = 6$, maliyet 6. Sosyal maliyet $7{,}5 + 5{,}5 + 5{,}5 + 6 = 24{,}5$. Uçtaki oyuncular çok uzak kalır.
2. **Kârlı sapma.** Oyuncu 1, 4'e ikinci bir bağ alırsa ağ bir döngüye dönüşür. Uzaklıkları 2'ye 1, 4'e 1, 3'e 2 olur; toplam 4. Maliyeti $2 \cdot 1{,}5 + 4 = 7 < 7{,}5$. Yol ağı bir Nash dengesi değildir.
3. **Tam çizge.** 6 bağ vardır ve her oyuncu ötekilere 1 uzaklıktadır. Sosyal maliyet $6 \cdot 1{,}5 + 4 \cdot 3 = 9 + 12 = 21$. Ama bağ almış bir oyuncu bir bağını bırakırsa 1,5 kazanır; o komşuya uzaklığı yalnız 1'den 2'ye çıkar (başka bir düğüm üzerinden). Net kazanç $1{,}5 - 1 = 0{,}5$; sapma kârlıdır. Tam çizge denge değildir.
4. **Yıldız.** Merkez 1 üç bağı alır: maliyeti $3 \cdot 1{,}5 + 3 = 7{,}5$. Her yaprağın maliyeti $0 + 1 + 2 + 2 = 5$. Bir yaprak başka bir yaprağa bağ alırsa o yaprağa uzaklığı 2'den 1'e iner: 1 kazanır, 1,5 öder; maliyeti 5,5 olur ($1 < 1{,}5$, kârsız). İki yaprağa birden bağ alırsa 3 öder, 2 kazanır; yine kârsız. Merkez bir bağı bırakırsa ağ kopar ve maliyeti sonsuz olur. Kimse sapmaz: yıldız bir Nash dengesidir. Sosyal maliyet $7{,}5 + 3 \cdot 5 = 22{,}5$.
5. **Döngü.** Her oyuncu bir bağ alır (örneğin 1, 2'ye; 2, 3'e; 3, 4'e; 4, 1'e). Her biri $1{,}5 + 1 + 1 + 2 = 5{,}5$ öder; sosyal maliyet 22. Bağını bırakan oyuncu yolun ucuna düşer: uzaklıkları $1 + 2 + 3 = 6$, maliyeti 6. Köşegen ekleyen $3 + 3 = 6$ öder. İkisi de 5,5'ten kötüdür.

Döngü de bir Nash dengesidir.



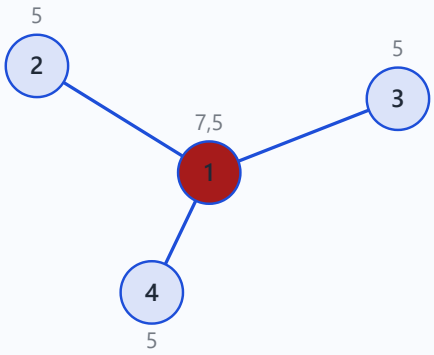
Tam çizge: 6 bağ

sosyal: 6 · 1,5 + 12 = 21

Bağını bırakan: −1,5 + 1

→ kazanır, denge değil

Şekil 8.21: Tam çizge: 6 bağ, sosyal maliyet *6 · 1,5 + 12 = 21*. Bağını bırakan oyuncu 1,5 kazanır, yalnız 1 uzaklık kaybeder; denge değildir.



Merkez 1 üç bağı alır

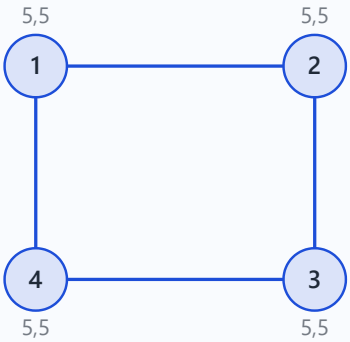
merkez 4,5 + 3 = 7,5; yaprak 5

Yaprak bağ alırsa 1,5 + 4 = 5,5

Merkez bırakırsa ağ kopar

→ Nash dengesi (22,5)

Şekil 8.22: Yıldız: merkez 1 üç bağı alır (maliyet 7,5), yapraklar 5'er öder. Yaprak bağ alırsa 5,5'e çıkar; merkez bağ bırakırsa ağ kopar. Nash dengesi, sosyal maliyet 22,5.



4-döngü (her biri bir bağ): 5,5

sosyal maliyet 22

Betik: tüm 4.096 profil tarandı

dengeler: yıldız 22,5 · döngü 22

optimum: tam çizge 21

Şekil 8.23: 4-döngü: her oyuncu bir bağ alır ve 5,5 öder; sosyal maliyet 22. Bütün 4.096 strateji profili taranınca $\alpha = 1{,}5$'te dengeler yıldız (22,5) ve döngü (22) çıkar; optimum olan tam çizge (21) denge değildir.

Şekil 8.21, Şekil 8.22 ve Şekil 8.23 üç ağı karşılaştırır. Sonucun doğruluğu bir kaba kuvvet taramasıyla denetlenmiştir. Her oyuncunun stratejisi, öteki üç oyuncunun hangilerine bağ alacağıdır: $2^3 = 8$ seçenek. Dört oyuncu için $8^4 = 4.096$ strateji profili vardır. Her profilde her oyuncunun bütün sapmaları denenmiş ve $\alpha = 1{,}5$'te denge olan ağların yalnız yıldız (sosyal maliyet 22,5) ve döngü (22) olduğu görülmüştür. Sosyal maliyeti en küçük olan tam çizge (21) bir denge değildir.

Bu sonucun anlamı, bencil yönlendirmedeki sonuçla aynıdır: denge ile optimum ayrışır. Tam çizgede her bağ, iki oyuncu arasındaki uzaklığı kısaltarak *iki* oyuncuya yarar (her biri ötekine 1 adım yakındır, sosyal kazanç 2). Ama bağın sahibi yalnız kendi kazancına (1) bakar ve bunu bağın fiyatıyla (1,5) karşılaştırır. Karşı tarafın kazancı dışsallıktır.

8.5.5 Kod izleme: maliyet()

Bir oyuncunun maliyetini hesaplamak için ötekilere uzaklıkları bilinmelidir. Bağların hepsi eşit uzunlukta (1) olduğu için en kısa uzaklıklar **genişlik öncelikli arama** (BFS, *breadth-first search*) ile bulunur. BFS, başlangıç düğümünden başlar; önce bütün komşuları (uzaklık 1), sonra onların henüz görülmemiş komşularını (uzaklık 2) ziyaret eder ve böyle sürer.

```
def maliyet(i, bag, alpha):
  uz = {i: 0}; kuyruk = [i]
  while kuyruk:
    u = kuyruk.pop(0)
    for v in komsu(bag, u):
      if v not in uz:
        uz[v] = uz[u] + 1
        kuyruk.append(v)
  aldigi = sum(1 for (a, b) in bag if a == i)
  return alpha * aldigi + sum(uz.values())
```

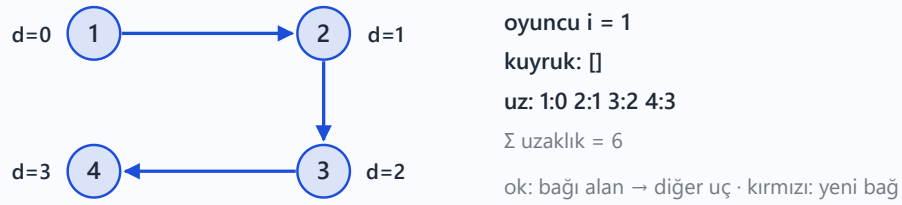
Fonksiyon, `komsu(bag, u)` adlı yardımcı bir fonksiyon kullanır. Bu fonksiyon, `u`'nun komşularını kimin aldığına bakmadan döndürür: bağı kim alırsa alsın herkes kullanır. `bag` listesi `(a, b)` çiftlerinden oluşur; `(a, b)`, bağı `a`'nın satın aldığı anlamına gelir. Kod satır satır şöyle çalışır:

- `uz` sözlüğü bilinen uzaklıkları tutar; başta yalnız `i` vardır (uzaklık 0). `kuyruk`, ziyaret edilecek düğümlerin listesidir.
- Kuyruk boşalana kadar baştaki düğüm `u` çıkarılır.
- `u`'nun her `v` komşusu için: `v` daha önce görülmemişse uzaklığı `u`'nunkinden 1 fazla yazılır ve `v` kuyruğun sonuna eklenir.
- `aldigi`, `i`'nin satın aldığı bağ sayısıdır: ilk elemanı `i` olan çiftler sayılır.
- Maliyet, `alpha` çarpı aldığı bağ sayısı artı bütün uzaklıkların toplamıdır.

Kod ağın bağlı olduğunu varsayar; bağlı değilse ulaşılamayan düğümler `uz`'a hiç girmez ve toplam yanlış biçimde küçük çıkar. Gerçek bir uygulamada `len(uz) < n` durumunda sonsuz döndürmek gerekir.

Adım	u	Yeni komşular	kuyruk	uz
başlangıç	—	—	[1]	1:0
1	1	2 → 1	[2]	1:0, 2:1
2	2	3 → 2	[3]	1:0, 2:1, 3:2
3	3	4 → 3	[4]	1:0, 2:1, 3:2, 4:3
4	4	yok (hepsi ziyaretli)	[]	aynı
sonuç	`aldigi = 1` , uzaklık toplamı 6, maliyet *1,5 · 1 + 6 = 7,5*			

Tablo 8.12: `maliyet(1, bag, 1.5)` izlemesi; yol ağı 1–2–3–4, oyuncu 1 yalnız (1, 2) bağını almış.



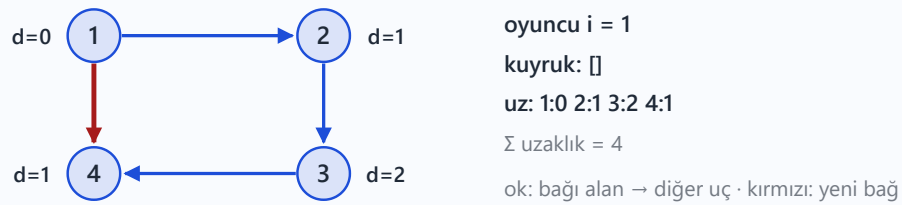
oyuncu i = 1
kuyruk: []
uz: 1:0 2:1 3:2 4:3
Σ uzaklık = 6
ok: bağı alan → diğer uç · kırmızı: yeni bağ

Şekil 8.24: İlk BFS'in sonu: yol ağında oyuncu 1'in uzaklıkları *d = 0, 1, 2, 3*; toplam 6. Oklar bağı alandan diğer uca doğrudur.

Şekil 8.24 ilk çağrının sonunu gösterir. Ardından bir sapma denenir: oyuncu 1 ayrıca (1, 4) bağını satın alır. Aynı fonksiyon yeniden çağrılır.

Adım	u	Yeni komşular	kuyruk	uz
başlangıç	—	—	[1]	1:0
1	1	2 → 1, 4 → 1	[2, 4]	1:0, 2:1, 4:1
2	2	3 → 2	[4, 3]	1:0, 2:1, 3:2, 4:1
3	4	yok	[3]	aynı
4	3	yok	[]	aynı
sonuç	`aldigi = 2` , uzaklık toplamı 4, maliyet *1,5 · 2 + 4 = 7*			

Tablo 8.13: Sapmadan sonra `maliyet(1, bag, 1.5)` ; oyuncu 1, (1, 2) ve (1, 4) bağlarını almış.



oyuncu i = 1
kuyruk: []
uz: 1:0 2:1 3:2 4:1
Σ uzaklık = 4
ok: bağı alan → diğer uç · kırmızı: yeni bağ

Şekil 8.25: Sapmadan sonraki BFS'in sonu: kırmızı ok yeni (1, 4) bağıdır. Oyuncu 1'in uzaklıkları 2 ve 4'e 1, 3'e 2; toplam 4.

Şekil 8.25 yeni bağla kısalan uzaklıkları gösterir. İki çağrının karşılaştırılması dengeyi denetler: $7 < 7{,}5$ olduğundan sapma kârlıdır ve yol ağı bir Nash dengesi değildir. İkinci izlemede kuyruğun sırası önemlidir: 4, 3'ten önce kuyruğa girdiği için önce o çıkar. BFS'te bir düğüm ilk kez görüldüğü anda uzaklığı kesinleşir; sonradan daha kısa bir yol bulunamaz, çünkü düğümler uzaklık sırasıyla ziyaret edilir.

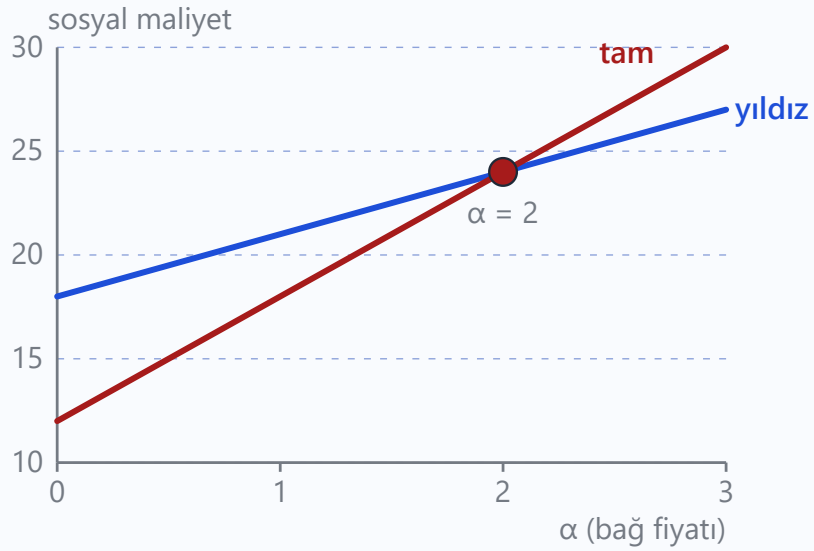
Bir oyuncunun bütün sapmalarını denemek için bu fonksiyon, oyuncunun her olası bağ kümesi için bir kez çağrılır. n oyunculu bir oyunda her oyuncunun 2^{n-1} bağ kümesi vardır; bu sayı hızla büyür. Fabrikant ve arkadaşları, bir oyuncunun en iyi tepkisini bulmanın genel durumda NP-zor olduğunu da göstermiştir (Fabrikant vd., 2003). Küçük örneklerde kaba kuvvet yeter; büyük ağlarda yaklaşık yöntemler gerekir.

8.5.6 Bağ fiyatı değişince denge ve optimum

Bağ fiyatı α değişince hem optimum hem dengeler değişir. Önce optimum incelenir. Dört oyunculu iki aday ağın sosyal maliyeti şöyle hesaplanır:

- **Yıldız**: 3 bağ. Uzaklık toplamı: merkez $1 + 1 + 1 = 3$, her yaprak $1 + 2 + 2 = 5$; toplam $3 + 3 \cdot 5 = 18$. Sosyal maliyet $3\alpha + 18$.
- **Tam çizge**: 6 bağ. Herkes ötekilere 1 uzaklıkta: $4 \cdot 3 = 12$. Sosyal maliyet $6\alpha + 12$.

İki doğru $3\alpha + 18 = 6\alpha + 12$, yani $\alpha = 2$'de kesişir (Şekil 8.26). $\alpha < 2$ iken tam çizge, $\alpha > 2$ iken yıldız daha ucuzdur. Nedeni açıktır: tam çizgede yıldıza göre 3 bağ fazladır (maliyet 3α) ve bu bağlar uzaklık toplamını 6 azaltır (her biri iki yaprağı birbirine 2 adım yakınlaştırır). Fazladan bağlar $3\alpha < 6$, yani $\alpha < 2$ iken kârlıdır. Genel n'de de optimum $\alpha \le 2$ için tam çizge, $\alpha \ge 2$ için yıldızdır (Fabrikant vd., 2003).



Şekil 8.26: Dört oyunculu ağda sosyal maliyetin bağ fiyatı α'ya göre değişimi. Kırmızı doğru tam çizge ($6\alpha + 12$), mavi doğru yıldız ($3\alpha + 18$); $\alpha = 2$'de kesişirler.

α	Optimum	Dengeler
0,5	tam (15)	tam (15)
1,5	tam (21)	döngü 22, yıldız 22,5
2,5	yıldız (25,5)	yıldız 25,5 · yol 27,5

Tablo 8.14: Dört oyunculu ağ oluşturma oyununda optimum ve dengeler (sosyal maliyetler; dengeler kaba kuvvet taramasıyla).

Tablo 8.14 üç bölgeyi gösterir.

- **$\alpha < 1$: bağ ucuz.** Bir bağ en az 1 uzaklık kazandırır ve bedeli 1'den azdır. Bu yüzden herkes doğrudan bağlanır: tam çizge hem denge hem optimumdur. Yıldız ise denge değildir: $\alpha = 0{,}5$'te yaprak başka bir yaprağa bağ alırsa 0,5 öder ve 1 kazanır.
- **$1 < \alpha < 2$: arada.** Optimum hâlâ tam çizgedir, ama tam çizge artık denge değildir: bağ bırakan oyuncu α kazanır, yalnız 1 kaybeder. $\alpha = 1{,}5$'teki dengeler (yıldız 22,5, döngü 22) optimumdan (21) daha pahalıdır.
- **$\alpha > 2$: bağ pahalı.** Optimum yıldızdır ve yıldız aynı zamanda dengedir. Ama yanında daha pahalı dengeler de çıkabilir. $\alpha = 2{,}5$'te yol ağı ($3 \cdot 2{,}5 + 20 = 27{,}5$) da bir dengedir. Bunun için bağları ortadaki iki oyuncunun alması gerekir: uçtaki bir oyuncu bağını kendisi almış olsaydı, bağını ortaya daha yakın bir düğüme taşıyarak uzaklığını azaltabilirdi.

Genel bir sonuç: $\alpha \geq 1$ olduğu sürece yıldız her zaman bir dengedir. Merkez bağ bırakırsa ağ kopar; yaprak yeni bir bağ alırsa en çok 1 uzaklık kazanır ve en az 1 öder. Ama denge çoğu zaman tek değildir ve bazı dengeler optimumdan belirgin biçimde kötüdür. Bencil davranışın bu kayıp oranı, ağ oluşturma oyunlarında da anarşinin bedeli adıyla incelenir.

8.5.7 Ağ oluşturmada sık hatalar

Sık yapılan hata

- **Bağın bedelini iki tarafa da yazmak:** Tek taraflı modelde bağın bedelini yalnız satın alan öder. İki taraflı (Jackson–Wolinsky) örnekte ise bedel her iki uca da ayrı ayrı yazılır. Modeller karıştırılmamalıdır.
- **Sosyal maliyette uzaklıkları bir kez saymak:** Her oyuncu kendi uzaklıklarını toplar; bu yüzden sosyal maliyette her çiftin uzaklığı iki kez sayılır (tam çizgede 6 çift, uzaklık toplamı 12).
- **Bağ bırakınca uzaklığı sonsuz sanmak:** Tam çizgede bir bağ bırakılınca o komşuya uzaklık yalnız 2'ye çıkar; başka yollar vardır. Ağ ancak tek yol koparsa kopar.
- **Yalnız bağ eklemeyi denemek:** Sapmalar bağ eklemeyi, bırakmayı ve bağ değiştirmeyi (birini bırakıp başkasını almayı) kapsar.

- **İkili kararlılıkta tek ucun kazancına bakmak:** Bağ kurmak için iki ucun da kazanması (ya da en azından kaybetmemesi) gerekir.
- **Optimumun denge olduğunu varsaymak:** *α = 1,5*'te optimum (tam çizge) denge değildir.

8.6 Kontrol ve teşvikler

Önceki kesimlerin ortak sonucu, bencil kullanıcıların vardığı dengenin toplum açısından en iyi durumdan ayrışabildiğidir. Bu ayrışmaya karşı üç yol vardır: kararları bir merkeze toplamak, kullanıcıların teşviklerini değiştirmek ya da kaynakları adil bir ölçütle paylaştırmak.

8.6.1 Merkezî mi, dağıtık mı?

Bir ağda yönlendirme kararlarını kimin vereceği temel bir tasarım seçimidir. Tablo 8.15 iki uç yaklaşımı karşılaştırır.

	Merkezî yönlendirme	Dağıtık (bencil) yönlendirme
Kim karar verir?	Trafik kontrol merkezi, SDN (yazılım tanımlı ağ) denetleyicisi, internet servis sağlayıcısı (İSS) trafik mühendisliği	Her sürücü / her yönlendirici (navigasyon uygulamaları, BGP: ağlar arası yönlendirme protokolü)
Hedef	Toplam gecikme *C(f)* en küçük (Wardrop 2. ilke)	Kendi gecikmesi en küçük (Wardrop 1. ilke)
İki yollu örnek	1.500 / 1.500 → 71.250 araç·dk	2.000 / 1.000 → 75.000 araç·dk
Gereken	Tüm talep ve gecikme bilgisi; kullanıcıların uyması	Yalnızca yerel bilgi; uyum sorunu yok
Zayıf yan	Optimumda bazıları daha yavaş yolda → sapma isteği; tek nokta hatası	Braess gibi verimsiz dengeler

Tablo 8.15: Merkezî ve dağıtık (bencil) yönlendirme.

Merkezî yönlendirmenin örnekleri gerçek ağlarda vardır. **Yazılım tanımlı ağda** (SDN, *software-defined networking*) yönlendiriciler kendi başlarına karar vermez; bir denetleyici yazılım bütün ağın durumunu görür ve her yönlendiriciye ne yapacağını söyler. Bir **internet servis sağlayıcısı** da kendi ağı içinde trafik mühendisliği yaparak trafiği hatlara dengeli dağıtır. Bu durumlarda merkez, kendi ağının sahibidir ve yönlendirdiği paketler itiraz etmez.

Dağıtık yönlendirmenin en önemli örneği ise internetin kendisidir. İnternet, binlerce bağımsız ağın birleşmesinden oluşur ve bu ağlar arasındaki rotalar **BGP** (*Border Gateway Protocol*, ağlar arası yönlendirme protokolü) ile seçilir. Her ağ, rotalarını kendi ticari çıkarına göre seçer. Şehir trafiğinde de her sürücü kendi navigasyon uygulamasını izler.

İki yaklaşımın güçlü ve zayıf yanları birbirini tamamlar. Merkezî yönlendirme optimumu hedefleyebilir, ama tüm talep ve gecikme bilgisini gerektirir ve kullanıcıların uymasına bağlıdır. Optimumda bazı kullanıcılar daha yavaş yoldadır (iki yollu örnekte Yol 2'dekiler 27,5 dakika yaşarken Yol 1'dekiler 20

dakika yaşar); bu kullanıcılar sapmak ister. Ayrıca merkez çökerse bütün sistem etkilenir. Dağıtık yönlendirme yalnız yerel bilgiyle çalışır ve uyum sorunu yoktur, çünkü herkes zaten kendi istediğini yapar; ama Braess paradoksundaki gibi verimsiz dengelere varabilir.

Bu yüzden ağlarda *ara çözümler* yaygındır. Merkez her şeyi kontrol etmez; yalnız bir kısmını etkiler:

- **Fiyat:** Merkez kullanıcılara ne yapacaklarını söylemez, ama bazı yolları daha pahalı yapar (tıkanıklık ücreti).
- **Bilgi:** Merkez rota önerir; kullanıcı öneriye uyup uymamakta serbesttir.
- **Kısmi kontrol:** Merkez trafiğin bir kısmını kendisi yönlendirir, gerisini kullanıcılar seçer. Buna **Stackelberg yönlendirme** (*Stackelberg routing*) denir. Ad, önce liderin hamle yaptığı, sonra izleyicilerin ona tepki verdiği Stackelberg oyunlarından gelir: merkez kendi trafiğini, bencil kullanıcıların vardığı dengeyi iyileştirecek biçimde yerleştirir.

8.6.2 Teşvikler, yük dengeleme ve uygulamalar

Doğru teşvikler, kullanıcıları ağ kaynaklarını dengeli kullanmaya yönlendirir ve verimliliği artırır. Trafikte ücretler ya da uyarılar yoğunluğu azaltır. Akıllı elektrik şebekelerinde zamana bağlı tarife, pik saatlerde yükü dengeler: elektrik akşam saatlerinde pahalıysa, çamaşır makinesi gece çalıştırılır. Teşviklerin en temiz örneği, 8.5.1'de anılan tıkanıklık ücretidir.

Fikir şudur: Wardrop dengesinin optimumdan sapmasının nedeni, sürücünün ötekilere yüklediği $x \cdot \ell'(x)$ gecikmesini görmemesidir. Her yola, optimumdaki bu dışsallık kadar bir ücret konursa sürücü artık "süre + ücret" toplamına bakar. Bu toplam, o yolun marjinal maliyetine eşit olur. Sürücüler bu toplamları eşitlediğinde, planlayıcının marjinal maliyetleri eşitlediği optimuma varılır.

Örnek 8.16

İki yollu ağda ($\ell_1 = 5 + x_1/100$, $\ell_2 = 20 + x_2/200$, 3.000 araç) ücretler süre birimiyle (dakika karşılığı) ölçülsün.

1. Optimum 1.500 / 1.500'dür. Optimumdaki dışsallıklar: Yol 1'de $x_1 \cdot \ell_1' = 1.500 / 100 = 15$, Yol 2'de $x_2 \cdot \ell_2' = 1.500 / 200 = 7{,}5$.
2. Yol 1'e 15, Yol 2'ye 7,5 dakika karşılığı ücret konur. Sürücü artık $5 + x_1/100 + 15$ ile $20 + x_2/200 + 7{,}5$'i karşılaştırır.
3. Yeni denge: $20 + x_1/100 = 27{,}5 + (3.000 - x_1)/200$. İki taraf 200 ile çarpılır: $4.000 + 2x_1 = 5.500 + 3.000 - x_1$, yani $3x_1 = 4.500$ ve $x_1 = 1.500$.
4. Ücretli dengede trafik tam optimumdaki gibi bölünür. Her sürücünün "süre + ücret" toplamı 35'tir ($20 + 15 = 27{,}5 + 7{,}5$). Toplam yolculuk süresi 71.250 araç·dakikaya iner.

Toplanan ücretler toplumdan kaybolmaz; yol yapımına ya da toplu taşımaya aktarılabilir. Ama sürücünün kendi bakışından süre + ücret (35) eski denge süresinden (25) yüksektir. Ücretlendirmenin siyasi olarak zor olmasının bir nedeni budur.

Yanlış teşvikler ise ters etki yapar. Stratejik, bencil davranış artar ve sonuç tıkanıklık ile düşük verimliliktir. İki örnek sık görülür. Birincisi, herkese aynı "en hızlı" rotayı öneren bir navigasyon uygulaması yan yolları doldurabilir: uygulama her sürücüyü tek tek en hızlı yola gönderir, ama herkes aynı yola gidince o yol yavaşlar. İkincisi, ücret yanlış seçilirse trafik başka bir darboğaza kayar: bir köprü ücretli yapılınca bütün araçlar ücretsiz ama dar bir yan köprüye yönelebilir.

Bu bölümün kavramları üç büyük uygulama alanında kullanılır:

- **İnternet.** İSS'ler veri paketlerini yönlendirir; ağlar arası rota seçimi BGP ile yapılır. Yük dengeleme (ECMP) ve trafik mühendisliği tıkanıklığı azaltır.
- **Trafik yönetimi.** Rota optimizasyonu, akıllı sinyalizasyon ve navigasyon uygulamaları. Trafik planlamacıları yeni bir yolun etkisini, sürücülerin varacağı Wardrop dengesini hesaplayarak değerlendirir.
- **Enerji.** Akıllı elektrik şebekelerinde enerji dağıtımı ve yük kaydırma. Tüketicilerin elektriği ne zaman kullanacağı, tıkanıklık oyununa benzer bir yapı oluşturur: herkes aynı saatte kullanırsa şebeke zorlanır.

8.6.3 Verimlilik ile adalet: bant genişliği paylaşımı

Toplam gecikmeyi en küçük yapmak tek ölçüt değildir. Bir ağın kaynakları kullanıcılar arasında paylaştırılırken *adalet* de önemlidir. Bunu görmek için küçük bir örnek kurulur. İki bağlantının kapasitesi 1'dir. Akış A ikisini de kullanır; akış B yalnız 1. bağlantıyı, akış C yalnız 2. bağlantıyı kullanır. Her akışa bir hız (x_A, x_B, x_C) verilir. Kapasite kısıtları şöyledir:

$$x_A + x_B \le 1, \; x_A + x_C \le 1$$

Hangi paylaşım seçilmelidir? Üç ölçüt üç farklı cevap verir (Tablo 8.16).

Ölçüt	Ne en iyilenir?	(x_A, x_B, x_C)	Toplam
En büyük toplam akış	$\Sigma\, x_i$	(0, 1, 1)	2 (A aç kalır)
Maksimin adalet (eşitlikçi ölçüt)	Önce en küçük pay olabildiğince büyütülür, sonra bir sonraki	(1/2, 1/2, 1/2)	1,5
Orantılı adalet (Nash sosyal refahı)	Payların çarpımı maksimize edilir (Kelly, 1997)	(1/3, 2/3, 2/3)	≈ 1,67

Tablo 8.16: Bant genişliği paylaşımında üç ölçüt.

Örnek 8.17

1. **En büyük toplam akış.** A iki bağlantıdan kapasite kullanır; ona verilen her birim, B'den bir birim ve C'den bir birim eksiltir. Toplam açısından A'ya hiç vermemek en iyisidir: $x_A = 0$, $x_B = x_C = 1$, toplam 2. Ama A tamamen aç kalır.

2. **Maksimin adalet.** Bütün paylar birlikte, eşit hızla artırılır. Paylar 1/2'ye ulaşınca 1. bağlantı ($x_A + x_B = 1$) ve 2. bağlantı ($x_A + x_C = 1$) aynı anda dolar. Artık hiçbir payı, bir başkasınınkini azaltmadan büyütmek mümkün değildir. Sonuç (1/2, 1/2, 1/2), toplam 1,5.
3. **Orantılı adalet.** Payların çarpımı $x_A \cdot x_B \cdot x_C$ en büyük yapılır. Çarpımı büyütmek için kapasiteler boşa harcanmaz: $x_B = x_C = 1 - x_A$. Çarpım $x_A\,(1 - x_A)^2$ olur.
4. Çarpımı maksimize etmek, logaritmasını maksimize etmekle aynıdır, çünkü logaritma artan bir fonksiyondur: $\log x_A + 2\log(1 - x_A)$. $\log x$'in türevi $1/x$'tir. Türev sıfıra eşitlenir: $1/x_A - 2/(1 - x_A) = 0$, yani $1/x_A = 2/(1 - x_A)$.
5. Çapraz çarpımla $1 - x_A = 2x_A$, $x_A = 1/3$. Paylaşım (1/3, 2/3, 2/3), toplam 5/3 ≈ 1,67.

Üç ölçüt arasındaki fark bir değer yargısıdır. En büyük toplam akış ölçütü toplamı önemser ve bir kullanıcıyı tamamen dışarıda bırakmaya razıdır. Maksimin adalet en kötü durumdaki kullanıcıyı önemser; herkesi eşitler, ama toplamdan fedakârlık eder. **Orantılı adalet** (*proportional fairness*) ikisinin arasında durur. İki bağlantıyı birden kullanan A daha az alır, çünkü o iki kaynaktan pay tüketir; ama kimse sıfıra düşmez, çünkü sıfır pay çarpımı sıfırlar. Toplam (≈ 1,67), maksiminden yüksek, en büyük toplam akıştan düşüktür.

Orantılı adaletin Nash sosyal refahıyla aynı şey olması rastlantı değildir: iki kavram da payların çarpımını maksimize etmeyi ister. Logaritma biçimindeki yazım, bu ölçütün "bir kullanıcının payını %10 artırmak, kimin olursa olsun aynı değerdedir" anlamına geldiğini gösterir. Frank Kelly ve arkadaşları 1998'de, internetin tıkanıklık denetimini (TCP benzeri protokolleri) bu açıdan yorumladı (Kelly vd., 1998). Her bağlantı, üzerindeki yük arttıkça bir *tıkanıklık fiyatı* bildirir; her akış, kendi yolundaki fiyatlara bakarak hızını ayarlar. Bu dağıtık algoritma, hiçbir merkez olmadan orantılı adil paylaşıma yakınsar.

8.6.4 Tartışma soruları

Bu bölümün kavramları, kesin cevabı olmayan ama düşünmeye değer üç soruya ışık tutar.

Navigasyon uygulamaları herkese "en hızlı" rotayı önerdiğinde Wardrop dengesine mi, sistem optimumuna mı yaklaşılır? Neden? Uygulama her sürücüye o anki gecikmelere göre en hızlı rotayı öneriyorsa, her sürücü kendi en iyi tepkisini daha kolay bulur. Bu, Wardrop dengesine yaklaşmayı hızlandırır; sistem optimumuna değil. Uygulama, sürücülerin bir kısmını bilerek daha yavaş rotalara gönderirse optimuma yaklaşılabilir; ama o sürücüler öneriye uymayabilir. Tek bir uygulamanın çok sayıda kullanıcısı varsa uygulama bir tür Stackelberg lideri gibi davranabilir.

Bir şehir yeni bir köprü planlıyor. Braess etkisini önceden görmek için köprü açıldığında sürücülerin varacağı denge nasıl hesaplanır? Önce ağ modellenir: her yol için bir gecikme fonksiyonu (örneğin BPR) ve kaynak–hedef çiftleri arasındaki talepler belirlenir. Sonra köprülü ve köprüsüz iki ağda Wardrop dengesi hesaplanır; bu, Beckmann potansiyelini Frank–Wolfe gibi bir yöntemle en aza indirmek demektir. İki dengenin toplam gecikmeleri karşılaştırılır. Köprülü dengenin toplamı daha yüksekse köprü bencil kullanımla zarar verebilir; çare ücretlendirme ya da erişim kısıtı olabilir.

İnternette yönlendiriciler (BGP) bencil ajan mıdır? Kimin maliyetini en aza indirirler? Tek tek yönlendiriciler bir program izler; bencil olan, onları işleten ağlardır. Her ağ (örneğin bir İSS), rotalarını kendi ticari çıkarına göre seçer: ücret ödediği bir ağ yerine ücret aldığı ya da ücretsiz eşleme yaptığı ağları tercih eder. İnternet rotalarının çoğu zaman en kısa ya da en hızlı olmamasının nedeni budur. BGP'de minimize edilen, paketin gecikmesi değil, ağ işletmecisinin maliyetidir.

8.7 Özet

- **Gecikme yüke bağlıdır.** Her kenarın gecikmesi, üzerindeki akışın azalmayan bir fonksiyonudur: sabit, doğrusal, BPR ya da kuyruk ($1/(c - x)$). Klasik en kısa yol algoritmaları (Dijkstra, Bellman–Ford) sabit ağırlık varsayar; yüke bağlı gecikme yönlendirmeyi bir oyuna dönüştürür.
- **Bencil yönlendirmede** her kullanıcı kendi gecikmesini küçültür. Bir yolun gecikmesi kenar gecikmelerinin toplamıdır; toplam gecikme $C(f) = \Sigma\, f_e\, \ell_e(f_e)$'dir.
- **Wardrop dengesinde** kullanılan bütün yolların gecikmesi eşit ve en düşüktür. İki yollu örnekte (3.000 araç) denge 2.000 / 1.000, herkes 25 dakika, toplam 75.000 araç·dakikadır.
- **Sistem optimumunda** marjinal maliyetler ($\ell + x \cdot \ell'$) eşitlenir: 1.500 / 1.500, toplam 71.250. Optimum bir denge değildir; fark, sürücünün hesaba katmadığı $x \cdot \ell'$ dışsallığından gelir.
- **Beckmann potansiyeli** $\Phi = \Sigma \int \ell_e$'nin en küçük noktası Wardrop dengesidir; denge her zaman vardır ve kenar gecikmeleri tektir. İkiye bölme (`denge_bul`) iki yollu ağda dengeyi her turda aralığı yarıya indirerek bulur.
- **Braess paradoksu**: 4.000 araçlı ağda 0 dakikalık kestirme herkesin süresini 65'ten 80 dakikaya çıkarır; zikzak herkes için baskındır. Kestirmeli ağın optimumu ≈ 64,7'dir. Seri-paralel ağlarda paradoks görülmez; hangi yolların kapatılacağını seçmek genel ağlarda zordur.
- **Tıkanıklık oyunlarında** Rosenthal potansiyeli $\Phi = \Sigma_e\, \Sigma_{k \le n_e}\, c_e(k)$, tek oyuncunun her sapmasında onun maliyet farkı kadar değişir. Bu yüzden saf Nash dengesi vardır ve en iyi tepki dinamiği durur ($26 \to 23 \to 21 \to 20$); ama genel durumda hızlı durmayabilir (PLS-tam).
- **Ağ oluşturmada** bağ maliyeti ile erişim arasında gerilim vardır. İki taraflı modelde yıldız ve döngü ikili kararlıdır; tek taraflı modelde $\alpha = 1{,}5$'te yıldız ve döngü dengedir, optimum olan tam çizge değildir. Optimum $\alpha \le 2$ için tam çizge, $\alpha \ge 2$ için yıldızdır.
- **Kontrol ve teşvikler**: merkezî yönlendirme optimumu hedefler ama bilgi ve uyum ister; ara çözümler (tıkanıklık ücreti, rota önerisi, Stackelberg yönlendirme) yaygındır. Bant genişliği paylaşımında en büyük toplam akış, maksimin adalet ve orantılı adalet farklı paylaşımlar seçer.

Kavram	Özü	Örnek
Bencil yönlendirme	Herkes kendi gecikmesini küçültür	3.000 araç, iki yol
Wardrop dengesi	Kullanılan yollar eşit ve en hızlı	2.000 / 1.000, 25 dk
Sistem optimumu	Marjinal maliyetler eşit	1.500 / 1.500, 71.250 araç·dk
Braess paradoksu	Yeni yol → herkes daha yavaş	65 → 80 dk
Rosenthal potansiyeli	ΔΦ = hamle yapanın maliyet farkı	Φ: 26 → 23 → 21 → 20

Kavram	Özü	Örnek
Ağ oluşturma	Bağ maliyeti ↔ erişim	$\alpha = 1{,}5$: yıldız / döngü denge, tam çizge optimum

Tablo 8.17: Bölümün kavramları bir bakışta.

Denge ile optimum çoğu zaman ayrışır: sürücü kendi gecikmesine, planlayıcı marjinal maliyete bakar.

8.8 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

ağ tasarımı *network design* · yönlendirme *routing* · düğüm *node* · bağlantı *link* · kenar *edge* · çizge *graph* · kapasite *capacity* · bant genişliği *bandwidth* · yönlendirici *router* · yıldız / halka / ağaç / tam örgü *star / ring / tree / mesh topology* · OSPF *Open Shortest Path First* · eşit maliyetli çok yol (ECMP) *equal-cost multi-path* · gecikme fonksiyonu *latency function* · BPR gecikme fonksiyonu *BPR latency function* · M/M/1 kuyruğu *M/M/1 queue* · bencil yönlendirme *selfish routing* · bölünebilir (atomik olmayan) *nonatomic* · atomik *atomic* · Wardrop dengesi *Wardrop equilibrium* · kullanıcı dengesi *user equilibrium* · sistem optimumu *system optimum* · marjinal maliyet *marginal cost* · Beckmann potansiyeli *Beckmann potential* · Braess paradoksu *Braess's paradox* · seri-paralel ağ *series-parallel network* · Wheatstone yapısı *Wheatstone network* · tıkanıklık oyunu *congestion game* · Rosenthal potansiyeli *Rosenthal potential* · potansiyel fonksiyonu *potential function* · potansiyel oyun *potential game* · sonlu iyileşme özelliği *finite improvement property* · PLS-tam *PLS-complete* · dışsallık *externality* · Metcalfe yasası *Metcalfe's law* · ağ oluşturma oyunu *network formation game* · ikili kararlılık *pairwise stability* · internet eşlemesi *peering* · tam çizge *complete graph* · sosyal maliyet *social cost* · yazılım tanımlı ağ (SDN) *software-defined networking* · BGP (ağlar arası yönlendirme protokolü) *Border Gateway Protocol* · internet servis sağlayıcısı (İSS) *Internet service provider* · Stackelberg yönlendirme *Stackelberg routing* · maksimin adalet *max-min fairness* · orantılı adalet *proportional fairness*

8.9 Kendini Dene

1. 5.000 araç A'dan B'ye gider. Yol 1: $\ell_1 = 10 + x_1/100$, Yol 2: $\ell_2 = 20 + x_2/400$ (dakika). Wardrop den-

gesi, gecikme ve toplam gecikme nedir?

Cevap

Uç durumlar denge değildir: herkes Yol 1'de ise $\ell_1 = 60 > 20$; herkes Yol 2'de ise $\ell_2 = 32{,}5 > 10$. Eşitlik: $10 + x_1/100 = 20 + (5.000 - x_1)/400$. 400 ile çarpılınca $4.000 + 4x_1 = 8.000 + 5.000 - x_1$, $5x_1 = 9.000$, $x_1 = 1.800$, $x_2 = 3.200$. Gecikmeler $10 + 18 = 28$ ve $20 + 8 = 28$ dakika. Toplam $5.000 \times 28 = 140.000$ araç·dakika.

2. Aynı ağda sistem optimumu ve toplam gecikmesi nedir? Bencil yönlendirme kaç araç·dakika fazla gecikme üretir?

Cevap

Marjinal maliyetler: Yol 1 $10 + x_1/50$, Yol 2 $20 + x_2/200$. Eşitlik: $10 + x_1/50 = 20 + (5.000 - x_1)/200$; 200 ile çarpılınca $2.000 + 4x_1 = 4.000 + 5.000 - x_1$, $5x_1 = 7.000$, $x_1 = 1.400$, $x_2 = 3.600$. Gecikmeler 24 ve 29 dakika. Toplam $1.400 \times 24 + 3.600 \times 29 = 33.600 + 104.400 = 138.000$. Fark $140.000 - 138.000 = 2.000$ araç·dakika. Optimumda Yol 2'dekiler 5 dakika daha yavaştır ve Yol 1'e geçmek ister.

3. Braess ağında 3.000 araç olsun; dar yollar $x/100$, geniş yollar 50 dakika. Kestirme (A→B, 0 dakika) açılmadan önce ve açıldıktan sonra denge süresi nedir? Paradoks görülür mü?

Cevap

Önce: 1.500 / 1.500, her rota $15 + 50 = 65$ dakika. Sonra: herkes zikzakta olsun; dar yollarda 3.000 araç (30 dakika), zikzak $30 + 0 + 30 = 60$. Eski rotaya dönen $30 + 50 = 80$ yaşar; kimse dönmez. Denge 60 dakikadır. Kestirme herkesi 5 dakika hızlandırır; paradoks görülmez. Dar yolun en kalabalık hâli (30) geniş yoldan (50) çok hızlı olduğu için kestirme gerçekten yararlıdır. Paradoks yalnız belirli bir trafik aralığında ortaya çıkar.

4. Atomik Braess ağında (s→A ve B→t: $c_e(k) = k$; s→B ve A→t: 4; A→B: 0) 3 sürücü (üst, alt, zik) profilindedir. Φ ve maliyetler nedir? Üst yoldaki sürücü zik'e geçerse Φ nasıl değişir?

Cevap

Kullanıcı sayıları: s→A 2 (üst, zik), A→t 1, s→B 1, B→t 2 (alt, zik), A→B 1. $\Phi = (1 + 2) + 4 + 4 + (1 + 2) + 0 = 14$. Maliyetler: üst $2 + 4 = 6$, alt $4 + 2 = 6$, zik $2 + 0 + 2 = 4$. Üst sürücü zik'e geçerse s→A'da kalır (2), A→B 0, B→t'de 3 kişi olur (3): yeni maliyet $2 + 0 + 3 = 5$. Kazancı 1'dir; bu yüzden Φ 14'ten 13'e iner. Kontrol: A→t'den 4 silinir, A→B'ye 0 ve B→t'ye 3 eklenir; $-4 + 3 = -1$. Bu oyunun tek saf dengesi üç sürücünün de zik'te olduğu profildir (her biri 6, toplam 18); toplam maliyeti en küçük profil ise (üst, alt, zik)'tir (16).

5. Tek taraflı ağ oluşturma oyununda $n = 5$ oyuncu ve $\alpha = 3$ olsun. Yıldızın ve tam çizgenin sosyal ma-

liyeti nedir? Hangisi optimumdur? Yıldız bir Nash dengesi midir?

Cevap

Yıldız: 4 bağ, maliyet $4 \cdot 3 = 12$. Uzaklıklar: merkez $4 \cdot 1 = 4$, her yaprak $1 + 3 \cdot 2 = 7$; toplam $4 + 4 \cdot 7 = 32$. Sosyal maliyet 44. Tam çizge: 10 bağ ($5 \cdot 4/2$), maliyet 30; uzaklık toplamı $5 \cdot 4 = 20$; sosyal maliyet 50. $\alpha = 3 > 2$ olduğundan optimum yıldızdır. Yıldız dengedir: yaprak yeni bir bağla yalnız 1 uzaklık kazanır ve 3 öder; merkez bağ bırakırsa ağ kopar.

6. İki taraflı dört oyunculu örnekte (fayda: uzaklık 1 için 4, uzaklık 2 için 2,2, uzaklık 3 için 1) bağın her uca bedeli c olsun. c hangi aralıktayken tam çizge ikili kararlıdır? Yıldız?

Cevap

Tam çizge: bağ koparan c kazanır, o komşuya uzaklığı 1'den 2'ye çıkar ve $4 - 2{,}2 = 1{,}8$ kaybeder. Kararlılık için $c \le 1{,}8$ gerekir; eklenecek bağ yoktur. Yıldız: iki yaprak bağ kurarsa her biri 1,8 kazanır ve c öder; kurulmaması için $c \ge 1{,}8$. Merkez bir bağı koparırsa o yaprağı tamamen kaybeder (4) ve c kazanır; $c \le 4$ gerekir. Yaprak bağını koparırsa 8,4 kaybeder. Yıldız $1{,}8 \le c \le 4$ aralığında kararlıdır. $c = 1{,}8$'de ikisi de kararlıdır.

7. İki yollu ağda ($\ell_1 = 5 + x_1/100$, $\ell_2 = 20 + x_2/200$, 3.000 araç) yalnız Yol 1'e bir ücret τ konacaktır (dakika karşılığı). Hangi τ değerinde ücretli denge sistem optimumuyla aynı olur?

Cevap

Optimum 1.500 / 1.500'dür; orada $\ell_1 = 20$, $\ell_2 = 27{,}5$. Ücretli dengenin 1.500'de olması için sürücünün iki yolu eşit görmesi gerekir: $20 + \tau = 27{,}5$, yani $\tau = 7{,}5$. Daha küçük ücrette Yol 1 hâlâ fazla kalabalık, daha büyükte fazla boş kalır. Bu tek ücret, iki yola ayrı ayrı marjinal ücret (15 ve 7,5) koymakla aynı ücret farkını (7,5) verir; sürücünün kararı yalnız farka bağlıdır.

8. `denge_bul(l1, l2, 3000)` `eps = 10` ile çağrılırsa döngü kaç tur döner ve son aralığın genişliği ne olur?

Cevap

Her tur genişliği yarıya indirir: k tur sonra genişlik $3.000/2^k$'dir. Döngü genişlik 10'dan büyük olduğu sürece döner. $2^8 = 256$ için genişlik $\approx 11{,}7 > 10$; $2^9 = 512$ için $\approx 5{,}86 \le 10$. Döngü 9 tur döner; son genişlik yaklaşık 5,86 araçtır. `eps = 50` 'ye göre yalnız 3 tur fazladır.

8.10 İleri okuma

- T. Roughgarden, *Selfish Routing and the Price of Anarchy* (2005): bencil yönlendirme, Wardrop dengesi, Braess paradoksu ve ağ tasarımının zorluğunu tek bir kitapta, adım adım ispatlarla anlatır.
- N. Nisan, T. Roughgarden, É. Tardos ve V. Vazirani (ed.), *Algorithmic Game Theory* (2007), Bölüm 18–19: yönlendirme oyunlarını ve potansiyel fonksiyon yöntemiyle ağ oluşturma oyunlarını bilgisayar bilimi bakışıyla sunar.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 8 ve 17: trafik ağlarını, Braess paradoksunu ve ağ etkilerini az matematikle, çok örnekle anlatır.
- A. R. Karlin ve Y. Peres, *Game Theory, Alive* (2017): trafik yönlendirmesini ve potansiyel oyunları öğrenci düzeyinde, alıştırmalarla verir.

8.11 Kaynaklar

1. M. Beckmann, C. B. McGuire, C. B. Winsten, *Studies in the Economics of Transportation*, Yale University Press, 1956.
2. D. Braess, "Über ein Paradoxon aus der Verkehrsplanung", *Unternehmensforschung* 12, 258–268, 1968.
3. J. E. Cohen, P. Horowitz, "Paradoxical behaviour of mechanical and electrical networks", *Nature* 352, 699–701, 1991.
4. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010 (Bölüm 8: trafik ağları ve Braess; Bölüm 17: ağ etkileri).
5. A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, S. Shenker, "On a network creation game", *ACM PODC*, 2003.
6. A. Fabrikant, C. Papadimitriou, K. Talwar, "The complexity of pure Nash equilibria", *ACM STOC*, 2004.
7. M. Jackson, A. Wolinsky, "A strategic model of social and economic networks", *Journal of Economic Theory* 71(1), 44–74, 1996.
8. A. R. Karlin, Y. Peres, *Game Theory, Alive*, American Mathematical Society, 2017.
9. F. Kelly, "Charging and rate control for elastic traffic", *European Transactions on Telecommunications* 8, 33–37, 1997.
10. F. Kelly, A. Maulloo, D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability", *Journal of the Operational Research Society* 49, 237–252, 1998.
11. W. Knödel, *Graphentheoretische Methoden und ihre Anwendungen*, Springer, 1969.
12. G. Kolata, "What if they closed 42d Street and nobody noticed?", *The New York Times*, 25 Aralık 1990.
13. I. Milchtaich, "Network topology and the efficiency of equilibrium", *Games and Economic Behavior* 57, 321–346, 2006.
14. D. Monderer, L. Shapley, "Potential games", *Games and Economic Behavior* 14, 124–143, 1996.

15. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 18–19).

16. A. Rapoport, T. Kugler, S. Dugar, E. J. Gisches, "Choice of routes in congested traffic networks: Experimental tests of the Braess paradox", *Games and Economic Behavior* 65(2), 538–571, 2009.

17. R. W. Rosenthal, "A class of games possessing pure-strategy Nash equilibria", *International Journal of Game Theory* 2, 65–67, 1973.

18. T. Roughgarden, *Selfish Routing and the Price of Anarchy*, MIT Press, 2005.

19. T. Roughgarden, "On the severity of Braess's paradox: Designing networks for selfish users is hard", *Journal of Computer and System Sciences* 72, 922–953, 2006.

20. J. G. Wardrop, "Some theoretical aspects of road traffic research", *Proceedings of the Institution of Civil Engineers*, Part II, 1, 325–378, 1952.

KISIM III · BÖLÜM 9

Anarşinin Bedeli

Trafikte her sürücü kendi yolunu seçer, bir sunucu kümesinde her iş kendi makinesini seçer, bir ağda her kullanıcı kendi bağlantısını kiralar. Hiçbiri sistemin toplam maliyetini düşünmez. Bu bencil seçimlerin sonucu, merkezî bir planlayıcının kuracağı en iyi düzenden ne kadar kötü olabilir? Anarşinin Bedeli bu kaybı tek bir oranla ölçer; İstikrarın Bedeli aynı soruyu iyimser yönden sorar. Bu not, iki kavramı tanımlarından başlayarak Pigou ağına, doğrusal gecikmede 4/3 sınırına ve pürüzsüzlük yöntemine, Braess ağına, yük dengeleme oyununa, adil maliyet paylaşımındaki H_n sınırına ve kaybı azaltan tıkanıklık ücretine kadar adım adım anlatır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum09`

ÖĞRENME ÇIKTILARI

Tanımlamak
Anarşinin Bedeli (PoA) ve İstikrarın Bedeli'ni (PoS) tanımlamak; aralarındaki farkı örnekle göstermek.

Hesaplamak
Pigou, Braess ve yük dengeleme örneklerinde denge ve optimum maliyetini bulup oranı hesaplamak.

Sınırı açıklamak
Doğrusal gecikmede PoA ≤ 4/3 sınırının sezgisel ispatını ve sıkılığını açıklamak.

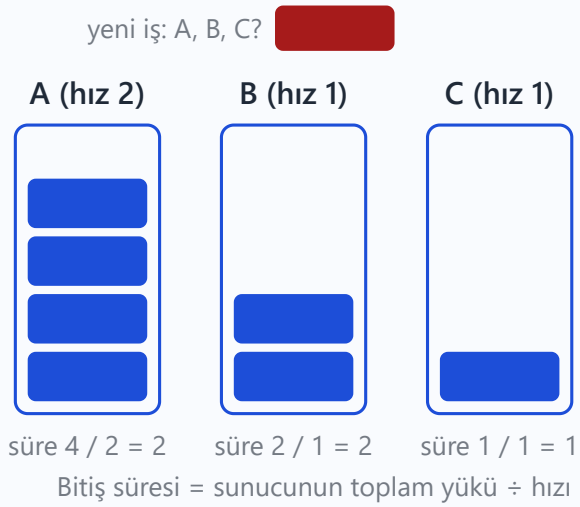
İyileştirmek
Tıkanıklık ücretinin optimumu nasıl dengeye dönüştürdüğünü iki yollu ağda ve Pigou ağında göstermek.

9.1 Temel kavramlar

Oyun kuramının temel denge kavramı olan Nash dengesi bir *kararlılık* tanımıdır: dengede kimse tek başına seçimini değiştirmek istemez. Tanım, dengenin iyi bir sonuç olduğunu söylemez. Mahkûmun İkilemi'nde iki oyuncu da ihanet eder ve birlikte sessiz kalsalar alacakları sonuçtan daha kötüsünü alır. Bu bölüm “Denge ne kadar kötü olabilir?” sorusunu bir sayıya çevirir. Soru önce küçük bir oyunda somutlaşır, ölçüt ardından tanımlanır.

9.1.1 Bir oyunla başlangıç: yük dengeleme

Bir sunucu kümesinde üç sunucu vardır: A, B ve C. Sunucuların hızları farklıdır; A'nın hızı 2, B ile C'nin hızı 1'dir. İşler sırayla gelir ve her iş sunuculardan birine yerleşir. Bir işin **bitiş süresi**, yerleştiği sunucunun toplam yükünün sunucu hızına bölümüdür. Örneğin A'da 4 birim iş varsa A'daki her iş *4 / 2 = 2* birim sürede biter. B'de 2 birim iş varsa B'deki işler *2 / 1 = 2* sürede, C'de 1 birim iş varsa C'deki iş *1 / 1 = 1* sürede biter.



Şekil 9.1: Yük dengeleme oyununun bir anı. A'da 4, B'de 2, C'de 1 birim iş vardır; bitiş süreleri sırasıyla 4/2 = 2, 2/1 = 2 ve 1/1 = 1'dir. Yeni gelen iş üç sunucudan birini seçecektir.

Şekil 9.1'de yeni bir iş gelmiştir. İş yalnız kendi bitiş süresini düşünür. A'ya giderse A'nın yükü 5 olur ve iş *5/2 = 2,5* sürede biter. B'ye giderse *3/1 = 3*, C'ye giderse *2/1 = 2* sürede biter. Bencil iş C'yi seçer. Bu seçim o iş için en iyisidir, ama sistemin toplam maliyeti için en iyisi olmayabilir. Oyunun sorduğu iki soru şunlardır:

- Her iş kendi bitiş süresini en aza indirirse sistemin toplam maliyeti de en küçük olur mu?
- Olmuyorsa, bencil işlerin vardığı denge en iyi atamadan en çok ne kadar kötü olabilir?

Birinci sorunun cevabı genelde “hayır”dır. İkinci sorunun cevabı oyundan oyuna değişen bir sayıdır; bölümün asıl konusu da bu sayıdır. Aynı oyun 9.4.8'de ayrıntılı çözülür: en kötü dengenin toplam süresi 13, en iyi atamanınki 12,5 çıkar ve oran 1,04 olur.

9.1.2 Anarşinin Bedeli: bencilliğin faturası

Anarşinin Bedeli (*Price of Anarchy, PoA*), bencil davranışın oluşturduğu en kötü denge ile sosyal optimum arasındaki verimlilik kaybını ölçer. Bu tanımdaki iki kavram şunlardır:

- **Sosyal optimum** (*social optimum*), sistemin ortak hedefi açısından en iyi sonuçtur. Maliyet problemlerinde sosyal optimum *toplam maliyeti en aza indiren* durumdur. Trafikte bu, bütün sürücülerin toplam yolculuk süresinin en küçük olduğu dağılımdır.

- **Denge**, herkesin kendi maliyetine bakarak seçim yaptığı ve kimsenin tek başına seçimini değiştirmek istemediği durumdur.

Trafik ağı iki kavram arasındaki gerilimi iyi gösterir. Her sürücü kendi yolculuk süresini azaltmaya çalışır. Ancak bir sürücünün yola girmesi o yoldaki herkesi biraz yavaşlatır ve sürücü bu etkiyi hesaba katmaz. Sonuçta herkes kendi açısından akıllıca davranırken toplam gecikme gereğinden büyük olabilir.

Kavramın tarihi kısadır. Elias Koutsoupias ve Christos Papadimitriou 1999'da paralel makinelere iş dağıtan bencil oyuncuları incelerken en kötü dengenin optimuma oranını "koordinasyon oranı" (*coordination ratio*) adıyla tanımladılar (Koutsoupias ve Papadimitriou, 1999). "Anarşinin bedeli" adını Papadimitriou 2001'de verdi (Papadimitriou, 2001). Ad bir karşılaştırmayı anlatır: her şeye karar veren merkezî bir "diktatör" yerine herkesin kendi başına karar vermesinin, yani anarşinin, bedeli.

Bu bölümde iki sayısal örnek sık sık karşımıza çıkar. Birincisi iki yollu bir ağdır: 3.000 araç iki yol arasında bölünür; bencil dengede toplam gecikme 75.000 araç·dk, en iyi dağılımda 71.250 araç·dk'dır. Oran *75.000 / 71.250 ≈ 1,053*'tür; yani bencillik toplam gecikmeyi yaklaşık %5 artırır. İkincisi Braess ağıdır: 4.000 aracın kullandığı bir ağa yeni bir yol eklenince herkesin yolculuk süresi 65 dakikadan 80 dakikaya çıkar. Anarşinin Bedeli bu tür farkları tek bir sayıyla ölçer ve daha önemlisi, bütün bir oyun sınıfı için bu sayıya bir üst sınır koyar.

9.1.3 Nash dengesi ve sosyal optimum

Oran iki ayrı sonucu karşılaştırır. Tablo 9.1 bu iki sonucu yan yana koyar.

	Nash dengesi (bireysel)	**Sosyal optimum (sistem)**
Tanım	Hiçbir oyuncu tek taraflı strateji değiştirerek kendi sonucunu iyileştiremez.	Toplam maliyeti en aza indirir ya da sosyal refahı en yükseğe çıkarır.
Temsil ettiği	Bireysel teşvikler	Sistemin ortak hedefi
Kaç tane?	Birden fazla olabilir (iyi ve kötü dengeler).	Değeri tektir; aynı değeri veren birden çok durum olabilir.
Kararlı mı?	Evet, tanım gereği.	Çoğu zaman hayır: birileri sapmak ister.

Tablo 9.1: Nash dengesi ile sosyal optimumun karşılaştırması.

Nash dengesinin tanımını kısaca hatırlayalım. n oyuncu vardır. Her oyuncu i bir s_i stratejisi seçer; seçimlerin tamamı $s = (s_1, \ldots, s_n)$ bir **strateji profili**dir. Oyuncu i'nin bu profildeki maliyeti $C_i(s)$ ile gösterilir. s_{-i}, i dışındaki oyuncuların stratejileridir. Bir s profili, her oyuncu i ve her alternatif s_i' için $C_i(s) \le C_i(s_i', s_{-i})$ sağlanıyorsa bir **Nash dengesidir**. Sözle: ötekiler yerinde dururken hiçbir oyuncu kendi seçimini değiştirerek maliyetini düşüremez. Bu bölümde oyuncular kazanç değil maliyet taşır; bu yüzden eşitsizlik "küçük ya da eşit" yönündedir.

Dengenin optimumdan ayrışmasının üç tipik nedeni vardır:

1. **Bireysel rasyonellik.** Herkes kendi faydasına bakar. Sistemin toplam maliyeti hiçbir oyuncunun amaç fonksiyonunda bulunmaz.
2. **Dışsallıklar.** Bir oyuncunun seçimi başkalarının maliyetini değiştirir, ama oyuncu bu etkiyi hesaba katmaz. Yola giren sürücü ötekileri yavaşlatır; bu yavaşlama sürücünün kendi faturasında görünmez. Başkasına yüklenen ve karar verenin hesaba katmadığı bu maliyete **dışsallık** (*externality*) denir.
3. **Bilgi eksikliği.** Oyuncular sistemin bütününü görmeyebilir; en iyi düzeni bilseler bile ona uymak için güvence yoktur.

Anarşinin Bedeli bu iki sonucu birbirine oranlar: oranın payında denge, paydasında optimum bulunur.

9.1.4 Tanımlar: PoA ve PoS

Bir oyunda birden çok Nash dengesi olabilir. Bu durumda "dengenin maliyeti" tek bir sayı değildir. İki doğal seçim vardır: en kötü dengeye bakmak ya da en iyi dengeye bakmak. İlki Anarşinin Bedeli'ni, ikincisi İstikrarın Bedeli'ni verir. Gösterim şöyledir: NE, oyunun Nash dengelerinin kümesidir; $C(s)$, s durumundaki toplam maliyettir; OPT, toplam maliyeti en küçük durumdur (sosyal optimum).

Tanım 9.1

Bir maliyet oyununun **Anarşinin Bedeli** (*Price of Anarchy*), en kötü Nash dengesinin maliyetinin optimumun maliyetine oranıdır:

$$PoA = \max_{s \in NE} C(s) / C(OPT)$$

İstikrarın Bedeli (*Price of Stability*), en iyi Nash dengesinin maliyetinin optimumun maliyetine oranıdır:

$$PoS = \min_{s \in NE} C(s) / C(OPT)$$

İki tanım yalnız bir sözcükte ayrılır: *max* ya da *min*. PoA **kötümser bir garanti**dir: oyuncular hangi dengeye giderse gitsin kayıp en çok bu kadardır. PoS ise **iyimser bir senaryo**dur: biri oyuncuları en iyi dengeye yönlendirebilirse kayıp bu kadara iner. PoS'un adındaki "istikrar", sonucun kararlı (dengede) olma şartından gelir: optimumun kendisi kararlı değilse, kararlı sonuçların en iyisi optimumdan ne kadar uzaktır?



Şekil 9.2: Maliyet ekseni üzerinde optimum ve dengeler. Optimum (OPT) en soldadır; en iyi Nash dengesinin optimuma oranı PoS'u, en kötü Nash dengesinin optimuma oranı PoA'yı verir.

Şekil 9.2 tanımı bir eksen üzerinde gösterir. Maliyet sağa doğru artar. Optimum tanım gereği en soldadır, çünkü hiçbir durum ondan ucuz olamaz. Dengeler optimumun sağında ya da tam üstündedir. En soldaki denge PoS'u, en sağdaki denge PoA'yı belirler. Bu resimden doğrudan şu sonuç okunur:

Teorem 9.1 (Oranların sırası)

Dengesi olan her maliyet oyununda $1 \le PoS \le PoA$'dır. Oyunun tek bir dengesi varsa $PoS = PoA$'dır.

İspat. Her denge bir durumdur ve optimum bütün durumların en ucuzudur; bu yüzden her denge s için $C(s) \ge C(OPT)$'dır. Maliyetler pozitif olduğundan iki yan $C(OPT)$'ye bölünebilir ve her dengenin oranı en az 1 olur; en iyi dengeninki de dahil, yani $PoS \ge 1$. Bir kümenin en küçük elemanı en büyük elemanından büyük olamaz; bu yüzden $\min C(s) \le \max C(s)$ ve $PoS \le PoA$'dır. Tek denge varsa en küçük ve en büyük eleman aynı elemandır ve iki oran eşittir. □

Örnek 9.1

İki özdeş makineye boyları 2, 2, 1 ve 1 olan dört iş yerleşecek; sosyal maliyet en büyük yüktür. Bu oyunun dengeleri 9.4.4'te tek tek bulunur. Burada sonucu kullanalım: dengelerin en büyük yükleri 3 ve 4'tür, optimum ise 3'tür.

1. **En kötü denge:** en büyük yükü 4 olan denge. $PoA = 4 / 3 \approx 1{,}333$.
2. **En iyi denge:** en büyük yükü 3 olan denge. $PoS = 3 / 3 = 1$.
3. **Denetim:** $1 \le 1 \le 4/3$; sıralama teoremi sağlanır. PoS = 1 olması, optimumun kendisinin de bir denge olduğunu söyler.

Fayda problemlerinde oran ters çevrilir. Bazı oyunlarda oyuncular maliyet değil fayda taşır ve sistem hedefi **sosyal refahı** (*social welfare*, kısaca SW) en büyük yapmaktır. Bu durumda en kötü denge, refahı en küçük olan dengedir. Oran $SW(OPT) / SW(denge)$ olarak yazılır; yani optimum paya geçer. Böylece oran yine 1'den büyük ya da 1'e eşit olur ve "büyük değer daha kötü verimlilik demektir" kuralı korunur. Örneğin optimumun refahı 100, en kötü dengenin refahı 80 ise PoA $100/80 = 1{,}25$'tir.

9.1.5 Verimlilik ölçütleri ve oranın okunması

Oranın anlamı, sosyal maliyetin nasıl tanımlandığına bağlıdır. Bu bölümde iki ölçüt kullanılır:

- **Toplam maliyet**: tüm kullanıcıların gecikme ya da kaynak kullanım maliyetlerinin toplamı. Trafik örneklerinde ölçüt budur; birimi araç·dk'dır (her aracın dakika cinsinden süresi toplanır).
- **En büyük yük** (*makespan*): bir iş dağıtımında en yüklü makinenin yükü, yani tüm işlerin bitiş anı. Yük dengelemede kullanılan klasik ölçüt budur.

Maliyet problemlerinde PoA, en kötü denge maliyetinin sosyal optimum maliyetine oranıdır. PoA'nın 1 olması verimlilik kaybı bulunmadığını gösterir: en kötü denge bile optimum kadar iyidir. Oran büyüdükçe stratejik davranışın oluşturduğu kayıp artar. Oran yüzdeye kolayca çevrilir: PoA = 1,25 ise en kötü denge optimumdan %25 daha pahalıdır.

PoA	Anlamı	Örnek
1	kayıp yok	Braess ağı, yeni yol açılmadan önce
4/3	%33 fazla maliyet	Pigou ağı; doğrusal gecikmenin sınırı
2 − 2/(m + 1)	2 kattan az	yük dengeleme, m özdeş makine
sınırsız	garanti yok	dik gecikme (x^p)

Tablo 9.2: PoA değerlerinin okunması. Örneklerin hepsi bu notta ayrıntılı çözülür.

Neden en kötü durum? İlk bakışta en kötü dengeye bakmak fazla karamsar görünebilir. Ama PoA'nın değeri tam da buradan gelir. Oyuncuların hangi dengeye gideceğini çoğu zaman bilemeyiz: başlangıç durumuna, hamle sırasına ve rastlantıya bağlıdır. PoA bu bilgiye gerek duymayan bir garanti verir: "Ne olursa olsun kayıp en fazla bu kadar." Bir ağ tasarımcısı PoA'nın küçük olduğunu biliyorsa merkezî bir denetime yatırım yapmadan da sistemin makul çalışacağından emin olabilir. Bu fikir, algoritmalardaki yaklaşım oranı fikrinin oyun kuramındaki karşılığıdır.

9.1.6 İki kötü durum oranı: PoA ve yaklaşım oranı

Algoritma derslerinde NP-zor problemler için **yaklaşım algoritmaları** incelenir. Bu problemlerde optimumu polinom zamanda bulan bir yöntem bilinmez; onun yerine hızlı bir algoritma iyi bir çözüm arar. Algoritmanın kalitesi **yaklaşım oranı** (*approximation ratio*) ile ölçülür: algoritmanın bulduğu çözümün maliyeti ile optimumun maliyeti arasındaki en kötü durum oranı. Anarşinin Bedeli de bir en kötü durum oranıdır. İkisinin farkı, optimuma ulaşmayı neyin engellediğindedir.

Oran	Optimuma ulaşmayı engelleyen	Pay / payda
Yaklaşım oranı	hesaplama gücü: polinom zamanda çözülemeyen (NP-zor) problem	hızlı bir algoritmanın bulduğu çözüm ÷ optimum
Anarşinin Bedeli	koordinasyon: merkezî otorite yok, herkes kendi çıkarına bakar	en kötü denge ÷ optimum

Tablo 9.3: İki kötü durum oranı. İkisi de "bulunan / optimum" biçimindedir; eksik olan şey farklıdır.

İki oran aynı problem üzerinde karşılaştırılabilir. Özdeş m makineye işleri dağıtıp en büyük yükü küçültmek NP-zor bir problemdir. Ronald Graham'ın 1966'da incelediği **açgözlü liste algoritması** işleri verilen sırayla alır ve her işi o anda en az yüklü makineye koyar (Graham, 1966). Bu algoritmanın bulduğu en büyük yük optimumun en çok $2 - 1/m$ katıdır. Aynı problemde işler bencil oyuncular olursa, saf Nash dengelerinin en büyük yükü optimumun en çok $2 - 2/(m + 1)$ katıdır (Finn ve Horowitz, 1979). $m = 2$ için iki sayı $3/2$ ve $4/3$'tür.

Örnek 9.2

İki özdeş makine ve boyları sırasıyla 1, 1, 2 olan üç iş olsun. Liste algoritması işleri bu sırayla yerleştirir.

1. Birinci 1'lik iş: iki makine de boş, M1'e gider. Yükler (1, 0).
2. İkinci 1'lik iş: en az yüklü makine M2'dir. Yükler (1, 1).
3. 2'lik iş: iki makine eşit yüklü, M1'e gider. Yükler (3, 1); en büyük yük 3.
4. Optimum: 2'lik iş tek başına bir makinede, iki 1'lik iş ötekinde: (2, 2); en büyük yük 2.

Oran $3 / 2 = 2 - 1/2$'dir; Graham sınırı bu örnekte tam olarak gerçekleşir. Üstelik (3, 1) ataması bir Nash dengesi değildir: M1'deki 1'lik iş M2'ye geçerse $1 + 1 = 2 < 3$ bekler ve kazanır. Bencil işler bu kötü atamada kalmaz. Liste algoritmasının kaybı hesaplamanın kısıtından, bencil dengenin kaybı ise koordinasyon eksikliğinden gelir.

Analizin sınırları. PoA, oyuncuları tamamen bencil ve rasyonel kabul eder. Gerçek oyuncular yalnız bencil değildir: güven, sosyal normlar, itibar ve tekrarlanan etkileşim işbirliğini artırabilir. Bu yüzden PoA bir *tahmin* değil, kötümser bir *garanti*dir. Gözlenen kayıp çoğu zaman PoA'nın verdiği sınırdan küçüktür; ama sınırın aşılmayacağı bilinir.

9.1.7 Tanımlarda sık hatalar

Sık yapılan hata

- **PoA ile PoS'u karıştırmak:** PoA en kötü dengeye, PoS en iyi dengeye bakar. “Bir dengenin optimuma oranı” her zaman PoA değildir; yalnız en kötü dengeninki PoA'dır.
- **Oranı ters yazmak:** Maliyet probleminde optimum paydadadır; oran hiçbir zaman 1'den küçük çıkmaz. 0,75 gibi bir değer bulunduysa oran ters çevrilmiştir.
- **Fark ile oranı karıştırmak:** “En kötü denge − optimum” bir farktır ve birime bağlıdır (araç·dk, saniye). PoA birimsiz bir orandır; farklı ölçekteki sistemler bu sayede karşılaştırılabilir.
- **Tek dengede PoS = 1 sanmak:** Tek denge kötü olabilir. Pigou ağında tek denge vardır ve $PoA = PoS = 4/3$'tür.

- **PoA'yı bir tahmin sanmak:** PoA, belirli bir örnekte ne olacağını değil, en kötü durumda ne olabileceğini söyler.

9.2 Pigou ağı

Anarşinin Bedeli'ni gösteren en küçük örnek, İngiliz iktisatçı Arthur Cecil Pigou'nun 1920'de verdiği iki yollu ağdır (Pigou, 1920). Örnek o kadar sadedir ki bütün hesap bir kâğıda sığar. Yine de doğrusal gecikmeli ağlardaki en kötü durumu tam olarak yakalar.

9.2.1 Kurulum: bölünebilir trafik ve Wardrop dengesi

Trafik ağları bir **çizge** ile modellenir: düğümler kavşaklar, kenarlar (bağlantılar) yollardır. Her kenarın bir **gecikme fonksiyonu** (*latency function*) vardır: $\ell_e(x)$, e kenarından x miktarda trafik geçerken o kenarı kullanan her sürücünün harcadığı süredir. Gecikme trafikle artar ya da sabit kalır; yoğun yol yavaştır. Sürücüler bir başlangıç düğümü s'den bir hedef düğümü t'ye gider ve her biri bir yol (ardışık kenarlar dizisi) seçer. Bir yolun gecikmesi üzerindeki kenarların gecikmelerinin toplamıdır.

Pigou ağında ve trafik örneklerinin çoğunda sürücü sayısı çok büyüktür ve tek bir sürücünün etkisi yok sayılacak kadar küçüktür. Bu durumda trafik **bölünebilir** (*nonatomic*) kabul edilir: 1 birimlik trafik, sayısız çok küçük sürücüden oluşan bir "akışkan" gibi düşünülür ve istenen oranda bölünebilir. Örneğin trafiğin 0,3'ü bir yolda, 0,7'si ötekinde olabilir. Tek bir sürücü yol değiştirdiğinde yolların gecikmeleri değişmez; çünkü bir sürücünün payı sıfıra yakındır.

Tanım 9.2

Bölünebilir trafikte bir akış dağılımı, trafik taşıyan her yolun gecikmesi kullanılabilecek her yolun gecikmesinden büyük değilse bir **Wardrop dengesi**dir (*Wardrop equilibrium*). Eşdeğer olarak: kullanılan bütün yolların gecikmesi eşittir ve kullanılmayan hiçbir yol daha kısa değildir.

Bu tanım, Nash dengesinin bölünebilir trafikteki karşılığıdır. Bir sürücü daha yavaş bir yoldaysa daha hızlı yola geçerek kazanırdı; dengede böyle bir sürücü yoktur. Tanım, İngiliz ulaştırma mühendisi John Glen Wardrop'un 1952'de ifade ettiği ilkeden adını alır. Wardrop'un ikinci ilkesi ise **sistem optimumu**dur: toplam gecikmenin en küçük olduğu dağılım. Bu bölümde sistem optimumu ile sosyal optimum aynı şeydir.

Toplam gecikme şöyle hesaplanır. f_e, e kenarındaki trafik olsun. Bu kenardaki her sürücü $\ell_e(f_e)$ süre harcar ve kenarda f_e kadar trafik vardır; kenarın toplam maliyete katkısı "trafik × gecikme"dir:

$$C(f) = \Sigma_e \, \ell_e(f_e) \cdot f_e$$

9.2.2 Pigou ağında denge

Pigou ağında s'den t'ye 1 birim trafik gider ve iki yol vardır. Trafiğin üst yoldaki payı x, alt yoldaki payı $1 - x$ olsun.

- **Üst yol:** gecikme $\ell(x) = x$. Yol boşken hızlıdır, doldukça yavaşlar; herkes üstteyse gecikme 1'dir. Dar bir şehir içi yolu düşünülebilir.
- **Alt yol:** gecikme her zaman 1. Ne kadar araç girerse girsin süre değişmez. Geniş ama uzun bir çevre yolu düşünülebilir.

Dengeyi bulmak için bir sürücünün gözünden bakalım. Alt yolda süre 1'dir. Üst yolda süre x'tir ve x bir pay olduğundan hiçbir zaman 1'i aşmaz. Demek ki üst yol hiçbir zaman alttan yavaş değildir. Alt yolda bir sürücü varsa ve $x < 1$ ise o sürücü üste geçip kazanır. Bu geçişler $x = 1$'e kadar sürer. $x = 1$'de üst yolun gecikmesi 1 olur ve alt yolla eşitlenir; artık kimse geçerek kazanamaz.

Örnek 9.3

1. **Aday: x = 1/2.** Üstte 1/2 dk, altta 1 dk. Alttaki bir sürücü üste geçerse yaklaşık 1/2 dk bekler ve kazanır. Denge değil.
2. **Aday: x = 0,9.** Üstte 0,9 dk, altta 1 dk. Alttakiler yine kazanır. Denge değil.
3. **Aday: x = 1.** Herkes üstte, süre 1 dk. Üstteki bir sürücü alta geçse yine 1 dk bekler; kazancı yoktur. Alt yol boştur ama üst yoldan daha kısa değildir. Wardrop koşulu sağlanır: denge.
4. **Maliyet:** $C(\text{denge}) = 1 \cdot 1 + 0 \cdot 1 = 1$. Herkes 1 dk bekler.

Dengenin tek olduğu da bu akıl yürütmeden çıkar: $x < 1$ olan her dağılımda alttaki sürücüler kaçar.

9.2.3 Sosyal optimum: toplam maliyetin en küçüğü

Merkezî bir planlayıcı ise soruya başka bir gözle bakar. Planlayıcı x'i seçer ve toplam gecikmeyi en küçük yapmak ister. Üst yolda x kadar trafik, her biri x dk; alt yolda $1 - x$ kadar trafik, her biri 1 dk bekler:

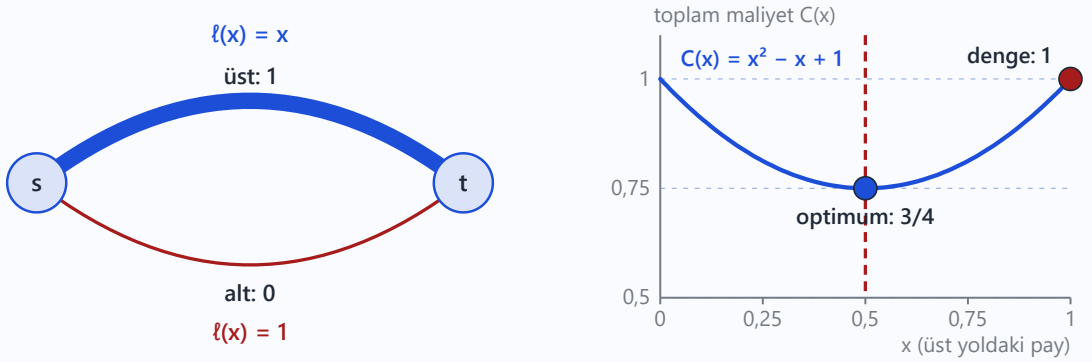
$$C(x) = x \cdot x + (1 - x) \cdot 1 = x^2 - x + 1$$

Bu fonksiyonun en küçük değeri **türev** yardımıyla bulunur. Türev, bir eğrinin her noktadaki eğimidir. Bir çukurun en alt noktasında eğim sıfırdır: noktanın solunda eğri iner (eğim eksi), sağında çıkar (eğim artı). Bu bölümde gereken türev kuralları şunlardır: x^k'nın türevi $k \cdot x^{k-1}$'dir (örneğin x^2'nin türevi $2x$); $c \cdot x$'in türevi c'dir; sabit sayının türevi 0'dır; toplamın türevi türevlerin toplamıdır. Türev $C'(x)$ ile gösterilir.

Örnek 9.4

1. **Türevi al.** $C(x) = x^2 - x + 1$; x^2'nin türevi $2x$, $-x$'in türevi -1, 1'in türevi 0. Sonuç: $C'(x) = 2x - 1$.
2. **Sıfıra eşitle.** $2x - 1 = 0 \Rightarrow x = 1/2$.
3. **Çukur mu, tepe mi?** $x < 1/2$ iken $C'(x) < 0$ (eğri iniyor), $x > 1/2$ iken $C'(x) > 0$ (eğri çıkıyor). Nokta bir çukurdur.
4. **Değeri hesapla.** $C(1/2) = 1/4 - 1/2 + 1 = 3/4$. Aynı sonuç "trafik × gecikme" toplamından da çıkar: üstte $1/2 \cdot 1/2 = 1/4$, altta $1/2 \cdot 1 = 1/2$; toplam $3/4$.
5. **Uçları denetle.** $C(0) = 1$ (herkes altta) ve $C(1) = 1$ (herkes üstte). İkisi de $3/4$'ten büyüktür ✓.

Optimumda trafiğin yarısı alt yola "feda edilir". Bu sürücüler 1 dk bekler; üsttekiler ise yalnız 1/2 dk bekler. Optimum *adil değildir* ve *kararlı değildir*: alttaki sürücüler üste geçmek ister, çünkü orada yarım dakika beklerlerdi. Merkezî bir zorlama olmadan bu dağılım kendiliğinden ortaya çıkmaz.



Şekil 9.3: Pigou ağı. Solda ağ: üst yolun gecikmesi $\ell(x) = x$, alt yolunki 1; çizgi kalınlığı dengedeki trafiği gösterir (herkes üstte). Sağda toplam maliyet $C(x) = x^2 - x + 1$: denge $x = 1$'de maliyet 1, optimum $x = 1/2$'de maliyet 3/4.

Şekil 9.3 iki sonucu aynı grafikte gösterir. Sağdaki parabolün sağ ucu dengedir (kırmızı nokta, maliyet 1); en alt noktası optimumdur (mavi nokta, maliyet 3/4). Denge, eğrinin en kötü noktalarından birine oturmuştur.

9.2.4 Oran: PoA = 4/3

Pigou ağının tek dengesi vardır; bu yüzden en kötü denge ile en iyi denge aynıdır ve PoA ile PoS eşittir:

$$PoA = C(\text{denge}) / C(\text{optimum}) = 1 / (3/4) = 4/3 \approx 1{,}33$$

Bencil yönlendirme, merkezî planlamaya göre %33 daha fazla toplam gecikme üretir. Bu sayı küçük bir örnekten çıktı, ama önemi büyüktür: doğrusal gecikmeli *her* ağda, *her* trafik miktarında oranın 4/3'ü aşmadığı ispatlanmıştır (9.3.4). Pigou ağı bu sınıfın en kötü örneğidir.

9.2.5 Özel maliyet ve marjinal maliyet

Denge ve optimum neden farklı yerlere düşer? Cevap, sürücünün baktığı maliyet ile toplumun baktığı maliyetin farklı olmasıdır.

- **Sürücü özel maliyetine bakar.** Üst yolda kendi gecikmesi x'tir; bunu alt yolun 1'iyle karşılaştırır. $x < 1$ olduğu sürece üstü seçer. Sonuç $x = 1$'dir.
- **Toplum marjinal maliyete bakar.** Üst yolun toplam maliyeti $x \cdot x = x^2$'dir. Üst yola küçük bir trafik eklenince bu toplam ne kadar artar? Türevi kadar: $2x$. Alt yolun toplam maliyeti $(1 - x) \cdot 1$'dir ve oraya eklenen her birim toplamı tam 1 artırır. Planlayıcı iki artışı eşitler: $2x = 1 \Rightarrow x = 1/2$.

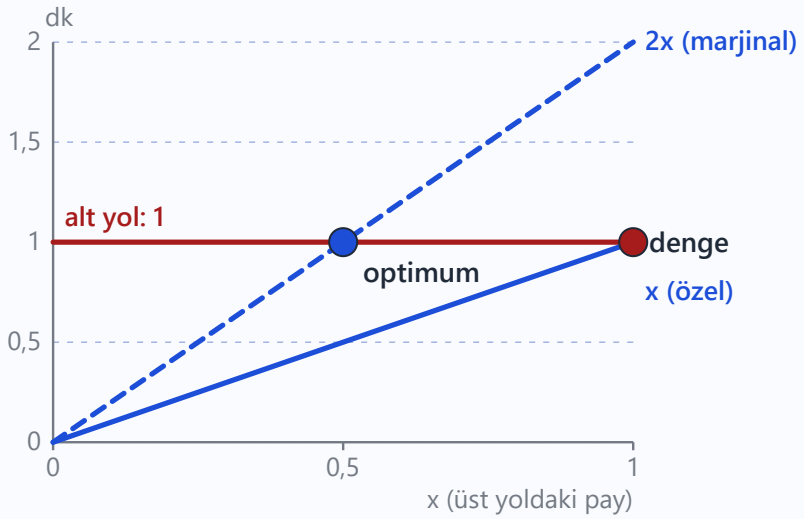
Tanım 9.3

Gecikme fonksiyonu $\ell(f)$ olan ve f trafik taşıyan bir kenarın **marjinal maliyeti** (*marginal cost*), kenarın toplam maliyeti $f \cdot \ell(f)$'nin türevidir:

$$(f \cdot \ell(f))' = \ell(f) + f \cdot \ell'(f)$$

Birinci terim, eklenen sürücünün kendi gecikmesidir (özel maliyet). İkinci terim $f \cdot \ell'(f)$, o sürücünün kenardaki diğer herkese eklediği gecikmedir (dışsallık).

Tanımdaki formül çarpımın türevi kuralından gelir, ama sözle de anlaşılabilir. Bir kenarda f araç varken bir araç daha girer. Bu araç kendisi $\ell(f)$ kadar bekler. Ayrıca kenarın gecikmesi $\ell'(f)$ kadar artar ve bu artış kenardaki f aracın her birine yansır; toplam ek yük $f \cdot \ell'(f)$'dir.



Şekil 9.4: Pigou ağında özel ve marjinal maliyet. Düz mavi doğru üst yolun özel maliyeti x, kesikli mavi doğru marjinal maliyeti $2x$, kırmızı yatay doğru alt yolun maliyeti 1'dir. Özel maliyet 1'le $x = 1$'de (denge), marjinal maliyet 1'le $x = 1/2$'de (optimum) kesişir.

Şekil 9.4 iki kesişimi gösterir. Kırmızı doğru ile düz mavi doğrunun kesiştiği yer dengedir; kırmızı doğru ile kesikli doğrunun kesiştiği yer optimumdur. İki mavi doğru arasındaki dikey uzaklık $x \cdot \ell'(x) = x \cdot 1 = x$'tir: bir sürücünün diğerlerine yüklediği gecikme. Sürücü bu farkı görmez; bu yüzden üst yola gereğinden çok sürücü girer.

Bu gözlem genelleşir:

- **Dengede** kullanılan yolların gecikmeleri eşitlenir.
- **Optimumda** kullanılan yolların marjinal maliyetleri eşitlenir; yani "kendi gecikmem + benim yüzümden ötekilere eklenen gecikme" eşitlenir.

İki koşul arasındaki fark tam olarak $f \cdot \ell'(f)$ terimidir. PoA'yı azaltmanın en doğrudan aracı olan tıkanıklık ücreti bu farka dayanır. Sürücüye bu dışsallık kadar ücret ödetilirse sürücünün gördüğü maliyet marjinal maliyete eşit olur ve bencil seçim optimumla çakışır (9.6.2).

9.2.6 İki yollu ağ: 3.000 araç

Aynı iki koşul daha gerçekçi bir ağda da uygulanabilir. 3.000 araç iki yol arasında bölünür. Yol 1 kısa ama dardır: $\ell_1 = 5 + x_1/100$ dk. Yol 2 uzun ama geniştir: $\ell_2 = 20 + x_2/200$ dk. Burada $x_1 + x_2 = 3.000$'dir. İki gecikme de $ax + b$ biçiminde, yani doğrusaldır: b boş yoldaki süre, a her ek aracın getirdiği yavaşlamadır.

Örnek 9.5

Denge (gecikmeler eşit).

1. $x_2 = 3.000 - x_1$ yazılır: $5 + x_1/100 = 20 + (3.000 - x_1)/200$.

2. Kesirlerden kurtulmak için iki yan 200 ile çarpılır: $1.000 + 2x_1 = 4.000 + 3.000 - x_1$.
3. $3x_1 = 6.000 \Rightarrow x_1 = 2.000$, $x_2 = 1.000$.
4. Gecikmeler: $\ell_1 = 5 + 20 = 25$, $\ell_2 = 20 + 5 = 25$ ✓. Herkes 25 dk bekler: $C(\text{denge}) = 3.000 \cdot 25 = 75.000$ araç·dk.

Optimum (marjinal maliyetler eşit). Yol 1'in marjinal maliyeti $\ell_1 + x_1 \cdot \ell_1' = 5 + x_1/100 + x_1/100 = 5 + x_1/50$; Yol 2'ninki $20 + x_2/100$'dür.

1. $5 + x_1/50 = 20 + (3.000 - x_1)/100$. İki yan 100 ile çarpılır: $500 + 2x_1 = 2.000 + 3.000 - x_1$.
2. $3x_1 = 4.500 \Rightarrow x_1 = 1.500$, $x_2 = 1.500$.
3. Gecikmeler: $\ell_1 = 5 + 15 = 20$, $\ell_2 = 20 + 7{,}5 = 27{,}5$. Optimum yine adil değildir: Yol 2'dekiler 7,5 dk daha fazla bekler.
4. Toplam: $1.500 \cdot 20 + 1.500 \cdot 27{,}5 = 30.000 + 41.250 = 71.250$ araç·dk.

Oran: $PoA = 75.000 / 71.250 = 20/19 \approx 1{,}053$. Bencillik toplam gecikmeyi yaklaşık %5 artırır; bu, 4/3 sınırının çok altındadır.

Bu örnek iki şey gösterir. Birincisi, gerçekçi ağlarda kayıp çoğu zaman en kötü durum sınırından küçüktür. İkincisi, optimum dağılım her zaman bazı kullanıcıları daha uzun beklemeye zorlar. Bu sürücülerin optimuma gönüllü uyması beklenemez; onları yönlendirmek için ücret gibi bir teşvik gerekir. Aynı ağdaki ücret hesabı 9.6.1'de yapılır.

9.2.7 Pigou ağında sık hatalar

Sık yapılan hata

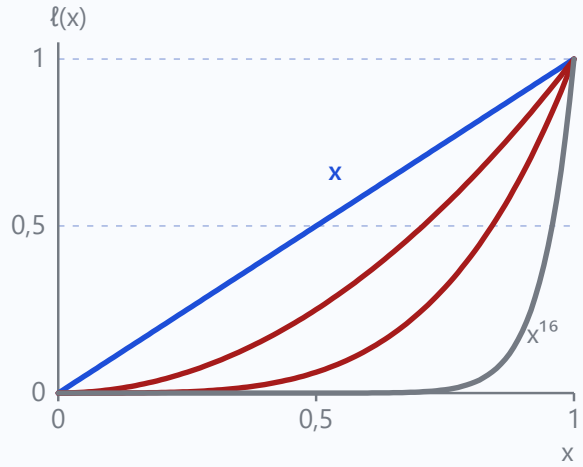
- **Dengede iki yolu eşit paylaştırmak:** "İki yol var, trafik yarı yarıya bölünür" düşüncesi yanlıştır. Denge, gecikmelerin eşitlendiği yerdir; Pigou'da bu herkesin üstte olduğu $x = 1$'dir. Yarı yarıya bölünme optimumdur.
- **Sürücünün marjinal maliyete baktığını sanmak:** Sürücü yalnız kendi gecikmesini görür. Marjinal maliyete bakan, toplamı düşünen planlayıcıdır.
- **Toplam maliyette yalnız gecikmeleri toplamak:** $C = \ell_1 + \ell_2$ yazmak yanlıştır. Her yolun gecikmesi o yoldaki trafikle çarpılır: $C = x_1\ell_1 + x_2\ell_2$. Pigou dengesinde alt yol boştur ve katkısı 0'dır, 1 değil.
- **Optimumu denge sanmak:** $x = 1/2$'de alttakiler üste geçmek ister; optimum kararlı değildir.

9.3 Doğrusal gecikme ve Braess

Pigou ağında oran 4/3 çıktı. Bu sayı bir tesadüf mü, yoksa genel bir yasanın parçası mı? Soru üç alt soruya ayrılır. Gecikme fonksiyonu değişince oran nasıl değişir? Doğrusal gecikmede oran neden hiçbir ağda 4/3'ü aşmaz? Aynı ispat fikri başka oyunlara nasıl taşınır?

9.3.1 Gecikme dikleşince PoA büyür

Pigou ağında üst yolun gecikmesini $\ell(x) = x$ yerine $\ell(x) = x^p$ alalım; $p \geq 1$ bir tamsayıdır. Alt yolun gecikmesi yine 1'dir. p büyüdükçe eğri "dikleşir": x 1'den küçükken x^p çok küçüktür, x 1'e yaklaşınca birden 1'e çıkar. Örneğin $0{,}5^1 = 0{,}5$, ama $0{,}5^4 = 0{,}0625$. Bu tür gecikme, kapasitesine yaklaşana kadar neredeyse boş gibi davranan, sonra aniden tıkanan bir yolu anlatır.



Şekil 9.5: Üst yolun gecikmesi $\ell(x) = x^p$; mavi doğru $p = 1$, kırmızı eğriler orta değerler, gri eğri $p = 16$'dır. Bütün eğriler $x = 1$'de 1 değerine ulaşır, ama p büyüdükçe o noktaya kadar çok düşük kalır.

Denge değişmez. $0 \leq x \leq 1$ iken $x^p \leq 1$'dir; üst yol yine hiçbir zaman alttan yavaş değildir. Herkes üste geçer ve herkes 1 dk bekler: $C(\text{denge}) = 1$.

Optimum iyileşir. Toplam maliyet $C(x) = x \cdot x^p + (1 - x) \cdot 1 = x^{p+1} + 1 - x$'tir. Optimumu bulmak için türev sıfıra eşitlenir:

1. x^{p+1}'in türevi $(p + 1)x^p$, $1 - x$'in türevi -1'dir: $C'(x) = (p + 1)x^p - 1$.
2. $(p + 1)x^p = 1 \Rightarrow x^p = 1/(p + 1)$.
3. İki yanın p'inci kökü alınır: $x^{OPT} = (p + 1)^{-1/p}$. Negatif üs "bir bölü" demektir; kesirli üs $1/p$ ise p'inci köktür. Örneğin $p = 2$ için $3^{-1/2} = 1/\sqrt{3} \approx 0{,}577$.
4. Optimum maliyet sadeleşir: $x^p = 1/(p + 1)$ olduğundan $x^{p+1} = x/(p + 1)$ ve $C(OPT) = x/(p + 1) + 1 - x = 1 - x \cdot p/(p + 1)$.

p	x^{OPT}	C(OPT)	PoA = 1 / C(OPT)
1	0,5	0,75	1,333
2	0,577	0,615	1,626
4	0,669	0,465	2,151
8	0,76	0,325	3,081
16	0,838	0,212	4,727

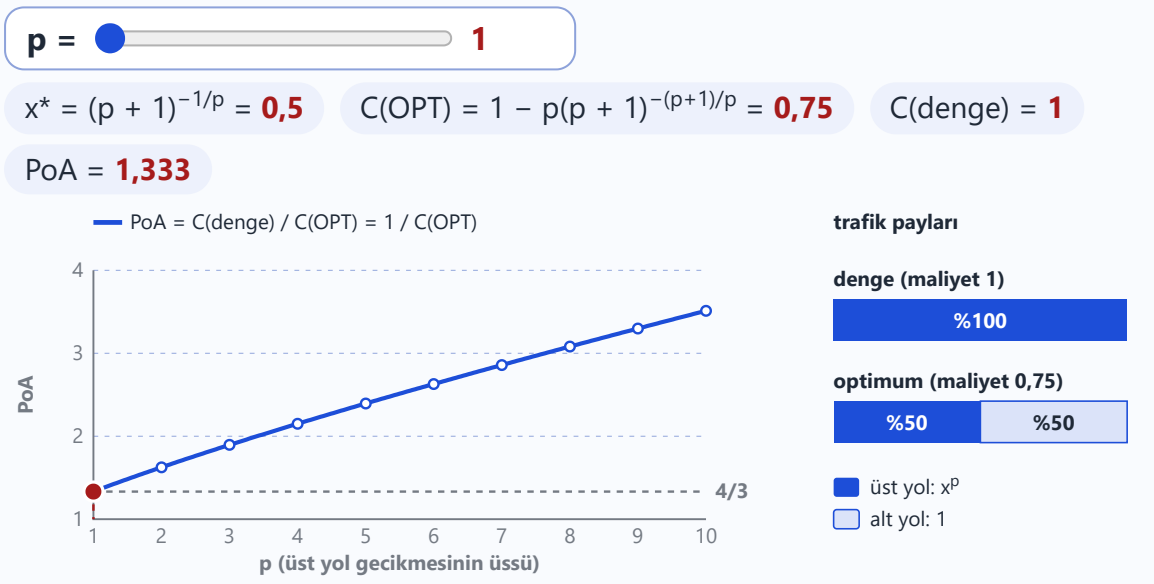
Tablo 9.4: Pigou ağında $\ell = x^p$ iken optimum pay, optimum maliyet ve PoA. Denge maliyeti her satırda 1'dir.

Tablo 9.4 bir eğilimi gösterir. Gecikme dikleştikçe planlayıcı üst yola daha çok trafik koyabilir, çünkü üst yol kapasitesine yaklaşana kadar çok hızlıdır. Optimumda üst yoldaki pay artar (0,5'ten 0,838'e), optimum maliyet düşer (0,75'ten 0,212'ye). Denge ise yerinde durur: herkes üstte, herkes 1 dk. Oran bu yüzden büyür.

Büyüme yavaştır ama sınırsızdır. p sonsuza giderken x^{OPT} 1'e yaklaşır ve $C(OPT)$ sıfıra iner; payda sıfıra yaklaşınca oran sınırsız büyür. Sayısal olarak $p = 10$'da PoA yaklaşık 3,51, $p = 100$'de yaklaşık 18,3'tür. Oran kabaca $p / \ln p$ hızında büyür. Buradan çıkan ders şudur: **4/3 evrensel bir sabit değildir**; doğrusal gecikmeye özgüdür. Sınırı belirleyen ağın biçimi değil, gecikme fonksiyonlarının ne kadar "dik" olabileceğidir.

9.3.2 Etkileşimli: gecikme üssü ve PoA

Şekil 9.6 aynı hesabı canlı yapar. Kaydırıcı üs p'yi 1'den 10'a kadar değiştirir. Soldaki grafik PoA'nın p'ye göre nasıl büyüdüğünü, sağdaki çubuklar dengede ve optimumda trafiğin iki yola nasıl bölündüğünü gösterir. Altta dört sayı yazılıdır: $x^* = (p+1)^{-1/p}$, $C(OPT)$, $C(denge) = 1$ ve PoA.



Şekil 9.6: Etkileşimli Pigou ağı. Üst yol x^p, alt yol 1; $C(x) = x^{p+1} + (1 - x)$, $C'(x) = 0 \Rightarrow x^* = (p+1)^{-1/p}$, $PoA = 1 / C(x^*)$. Varsayılan değer $p = 1$'dir: optimumda trafiğin %50'si üstte, $C(OPT) = 0{,}75$, PoA = 4/3.

Grafikte 4/3 düzeyi kesikli bir çizgiyle işaretlidir. Yalnız $p = 1$ noktası tam bu çizginin üstüne düşer; öteki bütün noktalar çizginin yukarısında kalır. Kaydırıcı sağa çekildikçe dengedeki çubuk değişmez (trafiğin %100'ü üstte), optimumdaki çubukta üst yolun payı büyür. Dengenin "katılığı" ile optimumun "esnekliği" arasındaki fark, oranın büyümesinin kaynağıdır.

9.3.3 Kod izleme: pigou_poa(p)

Tablodaki sayılar kısa bir Python fonksiyonuyla üretilir. Fonksiyon p'yi alır ve Pigou ağının PoA'sını döndürür.

```
def pigou_poa(p):          # üst yol x**p, alt yol 1
  x = (p + 1) ** (-1 / p)  # C'(x) = 0 noktası
  opt = x ** (p + 1) + (1 - x)
  return 1 / opt           # denge maliyeti 1

for p in [1, 2, 4, 8, 16]:
  print(p, round(pigou_poa(p), 3))
```

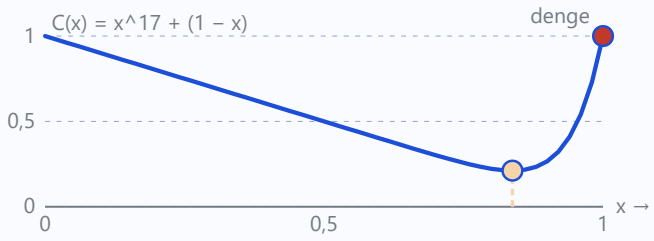
Kod satır satır şöyle çalışır:

- `x = (p + 1) ** (-1 / p)` : türevin sıfır olduğu nokta, yani optimumdaki üst yol payı. Python'da `**` üs alma işlecidir; `-1 / p` kesirli bir üstür.
- `opt = x ** (p + 1) + (1 - x)` : bu noktadaki toplam maliyet. Birinci terim üst yolun katkısı ($x \cdot x^p$), ikinci terim alt yolun katkısıdır.
- `return 1 / opt` : denge maliyeti her p için 1 olduğundan oran $1 / C(OPT)$'dir.
- Son iki satır fonksiyonu $p = 1, 2, 4, 8, 16$ için çağırır ve sonucu üç basamağa yuvarlayarak yazar.

p	$x = (p+1)^{-1/p}$	x^{p+1}	$1 - x$	opt	çıktı
1	$2^{-1} = 0{,}5$	0,25	0,5	0,75	1.333
2	$3^{-1/2} \approx 0{,}5774$	0,1925	0,4226	0,6151	1.626
4	$5^{-1/4} \approx 0{,}6687$	0,1337	0,3313	0,4650	2.151
8	$9^{-1/8} \approx 0{,}7598$	0,0844	0,2402	0,3246	3.081
16	$17^{-1/16} \approx 0{,}8377$	0,0493	0,1623	0,2116	4.727

Tablo 9.5: `pigou_poa(p)` izlemesi. Ara değerler dört basamağa yuvarlanmıştır.

Her satırda $x^{p+1} = x/(p + 1)$ özdeşliği de denetlenebilir. Örneğin $p = 4$ için $0{,}6687 / 5 \approx 0{,}1337$ ✓. Çıktı sütununda ondalık ayırıcı noktadır, çünkü Python sayıları İngilizce yazımla basar.



p = 16
x^{OPT} = 0,838
C(OPT) = 0,212
PoA = 1 / C(OPT) = 4,727

Şekil 9.7: *p = 16* için toplam maliyet eğrisi $C(x) = x^{17} + (1 - x)$. Eğri uzun süre $1 - x$ gibi iner, $x \approx 0{,}838$'de en küçük değeri 0,212'ye ulaşır, sonra dengeye ($x = 1$, maliyet 1) hızla çıkar. PoA = 4,727.

9.3.4 Doğrusal gecikmede 4/3 üst sınırı

Pigou ağında oranın tam 4/3 olduğunu gördük. Tim Roughgarden ve Éva Tardos 2002'de bunun bir rastlantı olmadığını gösterdiler: doğrusal gecikmeli hiçbir ağda oran 4/3'ü aşmaz (Roughgarden ve Tardos, 2002).

Teorem 9.2 (Roughgarden–Tardos, 2002)

Gecikmeler doğrusal ise, yani her kenar için $\ell_e(x) = a_e x + b_e$ ve $a_e, b_e \geq 0$ ise, her ağda ve her trafik talebinde bölünebilir trafiğin dengesi için

$$C(\text{denge}) \leq 4/3 \cdot C(\text{optimum})$$

sağlanır. Sınır sıkıdır: Pigou ağında eşitlik olur.

İspat üç adımdır ve her adım küçük bir fikre dayanır. Gösterim şöyledir: f_e dengede e kenarındaki trafik, f_e^{OPT} optimumdaki trafiktir. Toplam gecikme her kenarda "trafik × gecikme" toplamıdır: $C(f) = \Sigma_e \ell_e(f_e) \cdot f_e$.

1. adım: denge koşulu. Dengede herkes o anki gecikmelere göre en kısa yolu kullanır. Bir düşünce deneyi yapalım: gecikmeleri dengedeki değerlerinde "donduralım" ve sürücüleri optimumdaki yollara taşıyalım. Dondurulmuş gecikmelerle hiçbir sürücü dengedeki yolundan daha kısa bir yol bulamaz; çünkü dengedeki yolu zaten en kısasıdır. Bu yüzden optimumun dağılımı, dondurulmuş gecikmelerle ölçülünce dengenin kendisinden ucuz olamaz:

$$\Sigma_e \ell_e(f_e) \cdot f_e \leq \Sigma_e \ell_e(f_e) \cdot f_e^{OPT}$$

Sol yan dengenin gerçek maliyeti $C(f)$'dir. Sağ yan bir düşünce deneyidir: yollar dengedeki kadar yavaş, ama trafik optimumdaki gibi dağılmış. Geriye bu sağ yanı optimumun gerçek maliyeti $C(f^{OPT})$ ile sınırlamak kalır. Bunu 2. adım her kenar için ayrı ayrı yapar.

Örnek 9.6

Pigou ağında denge $f = (1, 0)$ (üst, alt), optimum $f^{OPT} = (1/2, 1/2)$'dir. Dengedeki gecikmeler: üst yol $\ell(1) = 1$, alt yol 1.

1. **Sol yan:** $1 \cdot 1 + 1 \cdot 0 = 1$, dengenin maliyeti.
2. **Sağ yan:** dondurulmuş gecikmeler (1 ve 1) optimumdaki trafikle çarpılır: $1 \cdot 1/2 + 1 \cdot 1/2 = 1$.
3. $1 \le 1$ ✓. Dengede iki yolun gecikmesi eşit olduğundan, trafiğin nasıl dağıldığı dondurulmuş maliyeti değiştirmez.

9.3.5 Kenar eşitsizliği ve toplama

2. adım: kenar eşitsizliği. Doğrusal gecikmeli her kenar için şu eşitsizlik doğrudur:

$$\ell_e(f_e) \cdot f_e^{OPT} \le \ell_e(f_e^{OPT}) \cdot f_e^{OPT} + \tfrac{1}{4} \cdot \ell_e(f_e) \cdot f_e$$

Sözle: düşünce deneyindeki bedel (sol yan), optimumun o kenardaki kendi bedelini en çok dengenin o kenardaki bedelinin dörtte biri kadar aşar.

Neden doğru? Kenar indisini bırakıp kısaca f ve f^{OPT} yazalım ve $\ell(x) = ax + b$'yi yerine koyalım. İfadeler iki gruba ayrılır:

- **b'li terimler.** Solda $b \cdot f^{OPT}$, sağda $b \cdot f^{OPT} + \tfrac{1}{4} \cdot b \cdot f$ vardır. Sağdaki ek terim negatif olamaz ($b, f \ge 0$); sağ yan küçük değildir.
- **a'lı terimler.** Solda $a \cdot f \cdot f^{OPT}$, sağda $a \cdot (f^{OPT})^2 + a \cdot f^2/4$ vardır. Gereken eşitsizlik $a \cdot f \cdot f^{OPT} \le a \cdot (f^{OPT})^2 + a \cdot f^2/4$'tür. Her şey sağa toplanırsa $0 \le a \cdot ((f^{OPT})^2 - f \cdot f^{OPT} + f^2/4) = a \cdot (f^{OPT} - f/2)^2$ çıkar. Bir karenin negatif olmaması ve $a \ge 0$ yeterlidir.

Son adımdaki tam kare açılımla denetlenebilir: $(u - v)^2 = u^2 - 2uv + v^2$; burada $u = f^{OPT}$, $v = f/2$ alınırsa $2uv = f \cdot f^{OPT}$ ve $v^2 = f^2/4$ olur. Eşitlik ancak $f^{OPT} = f/2$ iken (ya da $a = 0$ ve $b \cdot f = 0$ iken) sağlanır. ¼ sabiti bu tam karenin katsayısından gelir.

Örnek 9.7

Pigou'nun üst yolu ($\ell = x$, yani $a = 1, b = 0$): $f = 1, f^{OPT} = \tfrac{1}{2}$.

1. Sol: $\ell(1) \cdot \tfrac{1}{2} = 1 \cdot \tfrac{1}{2} = \tfrac{1}{2}$.
2. Sağ: $\ell(\tfrac{1}{2}) \cdot \tfrac{1}{2} + \tfrac{1}{4} \cdot \ell(1) \cdot 1 = \tfrac{1}{2} \cdot \tfrac{1}{2} + \tfrac{1}{4} = \tfrac{1}{2}$.
3. İki yan eşittir, çünkü $f^{OPT} = f/2$'dir. Pigou ağı eşitsizliği tam sınırında kullanır; 4/3'ün sıkı olmasının nedeni budur.

> **Eşitlik olmayan bir kenar** ($a = 1$, $b = 0$, $f = 2$, $f^{OPT} = ½$): sol $2 \cdot ½ = 1$; sağ $½ \cdot ½ + ¼ \cdot 2 \cdot 2 = ¼ + 1 = 5/4$. $1 \le 5/4$ ✓; fark $(½ - 1)^2 = ¼$'tür.

3. adım: topla. 2. adımdaki eşitsizlik bütün kenarlar için toplanır. Sol yanların toplamı 1. adımdaki sağ yandır. Sağ yanların toplamı $C(f^{OPT}) + ¼ \cdot C(f)$'dir. 1. adımla birleştirilince:

$$C(f) \le \Sigma_e \ell_e(f_e) f_e^{OPT} \le C(f^{OPT}) + ¼ \cdot C(f)$$

¼ · C(f) sola geçirilir: $¾ \cdot C(f) \le C(f^{OPT})$. İki yan $¾ \cdot C(f^{OPT})$'ye bölünür: $C(f) / C(f^{OPT}) \le 4/3$. □

Pigou ağında eşitlik olduğu için sınır **sıkıdır** (*tight bound*): 4/3'ten küçük bir sayı yazılamaz. Roughgarden 2003'te daha güçlü bir sonuç gösterdi: sabit fonksiyonları da içeren bir gecikme sınıfında en kötü örnek her zaman Pigou gibi iki paralel bağlantılı bir ağdır (Roughgarden, 2003). Yani ağın karmaşıklığı PoA'yı büyütmez; PoA'yı yalnız gecikme fonksiyonlarının sınıfı belirler. Doğrusal sınıf için bu 4/3, x^p gibi polinomlar için Tablo 9.4'teki değerlerdir.

9.3.6 Sınırın sayısal sınanması

Bir sınırın ne söylediği en iyi, bilinen örneklerle karşılaştırılınca anlaşılır. Tablo 9.6 bu bölümün örneklerini bir araya getirir.

Örnek	Denge	Optimum	Bu örnekte gözlenen	Kuramsal üst sınır
Pigou (x ve 1)	1	3/4	4/3 (sınıra eşit)	4/3
Braess (4.000 araç)	320.000	258.750	80 / 64,69 ≈ 1,237	4/3
Braess, normalleştirilmiş (x, 1, 1, x, 0)	2	3/2	4/3 (sınıra eşit)	4/3
İki yol, 3.000 araç	75.000	71.250	1,053	4/3
20.000 rastgele doğrusal paralel ağ	—	—	en büyük 1,33	4/3
Yük Dengeleme Oyunu, Seviye 2 (toplam süre)	13	12,5	1,04	5/2 (atomik oyuncular)

Tablo 9.6: Gözlenen oranlar ve kuramsal üst sınırlar.

Kuramsal üst sınır, bir sınıftaki (örneğin doğrusal gecikmeli bütün ağlardaki) en kötü durumdur. Belirli bir örnekte gözlenen oran çoğu zaman daha küçüktür. Tablodaki rastgele deneyde bir betik 2–4 paralel bağlantılı, gecikme katsayıları a, b ve trafik talebi rastgele seçilmiş 20.000 ağda dengeyi ve optimumu hesapladı. Oran hiçbir örnekte 4/3'ü aşmadı; Pigou'ya benzeyen örneklerde (bir bağlantı neredeyse sabit, öteki sıfırdan başlayan doğrusal) 4/3'e yaklaştı. Bu notun hesaplarında yinelenen deney de en büyük oranı yaklaşık 1,333 buldu.

Son satırdaki örnek farklı bir sınıftandır. Oradaki oyuncular bölünebilir trafik değil, tek tek işlerdir; bu tür oyunların sınırı 5/2'dir ve 9.3.9'da ele alınır.

9.3.7 Tek bir reçete: pürüzsüzlük

4/3 ispatı yalnız iki malzeme kullanır: dengenin tanımı ve her kenarda geçerli bir eşitsizlik. Roughgarden 2009'da bu yapıyı bir reçeteye dönüştürdü (Roughgarden, 2009). Reçetenin adı **pürüzsüzlük**tür (*smoothness*) ve birçok oyunun PoA sınırı aynı kalıpla ispatlanabilir.

Düşünce deneyi. s bir durum olsun (her oyuncunun bir stratejisi var), s^{OPT} optimum. Ötekiler s'de kalırken yalnız oyuncu i, s^{OPT}'deki stratejisine geçseydi ne öderdi? Bu, i'nin **deneme maliyeti**dir.

Gösterim. s_{-i}: i dışındaki oyuncuların stratejileri. $C_i(s_i^{OPT}, s_{-i})$: i'nin deneme maliyeti. $C(s) = \Sigma_i\, C_i(s)$: toplam maliyet.

Tanım 9.4

$\lambda > 0$ ve $0 \le \mu < 1$ iki sabit olsun. Her s ve s^{OPT} için

$$\Sigma_i\, C_i(s_i^{OPT}, s_{-i}) \le \lambda \cdot C(s^{OPT}) + \mu \cdot C(s)$$

sağlanıyorsa oyun **(λ, μ)-pürüzsüzdür** (*(λ, μ)-smooth*).

Sözle: deneme maliyetlerinin toplamı, optimumun maliyetinin λ katı ile s'nin maliyetinin μ katının toplamını aşmaz. Eşitsizlik *her* durum s için istenir; yalnız dengeler için değil. Koşul bu yüzden güçlüdür, ama ispatta tam da bu genellik işe yarar. λ ve μ küçüldükçe elde edilen sınır iyileşir.

4/3 ispatındaki kenar eşitsizliği tam olarak bir pürüzsüzlük koşuludur. Sol yanı "dengedeki gecikmelerle, optimumdaki trafiğin bedeli"dir; bölünebilir trafikte bu, deneme maliyetlerinin toplamıdır. Sağ yanı $1 \cdot C(f^{OPT}) + \tfrac{1}{4} \cdot C(f)$'dir. Demek ki doğrusal gecikmeli bölünebilir yönlendirme $(1, \tfrac{1}{4})$-pürüzsüzdür.

9.3.8 Pürüzsüzlükten PoA sınırına

Teorem 9.3 (Pürüzsüzlük sınırı)

(λ, μ)-pürüzsüz bir maliyet oyununda her Nash dengesi s için $C(s) \le \lambda / (1 - \mu) \cdot C(s^{OPT})$; yani $PoA \le \lambda / (1 - \mu)$.

İspat. **1. Denge.** s bir Nash dengesidir: kimse tek başına sapmakla kazanmaz. Özel olarak hiçbir oyuncu optimumdaki stratejisine geçerek de kazanmaz: her i için $C_i(s) \le C_i(s_i^{OPT}, s_{-i})$.

2. Topla. Bu eşitsizlikler bütün oyuncular için toplanır. Sol yanların toplamı toplam maliyettir: $C(s) \le \Sigma_i\, C_i(s_i^{OPT}, s_{-i})$.

3. Pürüzsüzlüğü uygula. Sağ yan, tanım gereği en çok $\lambda \cdot C(s^{OPT}) + \mu \cdot C(s)$'dir. Böylece $C(s) \le \lambda \cdot C(s^{OPT}) + \mu \cdot C(s)$.

4. Düzenle. $\mu \cdot C(s)$ sola geçirilir: $(1 - \mu) \cdot C(s) \le \lambda \cdot C(s^{OPT})$. $\mu < 1$ olduğundan $1 - \mu$ pozitiftir ve iki yan $(1 - \mu) \cdot C(s^{OPT})$'ye bölünebilir: $C(s) / C(s^{OPT}) \le \lambda / (1 - \mu)$. s herhangi bir denge olduğundan sınır en kötü denge için de geçerlidir: $PoA \le \lambda / (1 - \mu)$. □

Tanıdık örnekler. 4/3 ispatının kenar eşitsizliği $\lambda = 1, \mu = ¼$ demektir: $1 / (1 - ¼) = 1 / (3/4) = 4/3$. Atomik (bölünemez oyunculu) doğrusal tıkanıklık oyunlarında $\lambda = 5/3, \mu = 1/3$ alınabilir: $(5/3) / (2/3) = 5/2$.

> **Örnek 9.8**
>
> Bir oyunun $(1, ½)$-pürüzsüz olduğu biliniyor. Sınır $1 / (1 - ½) = 2$'dir. Aynı λ ile μ = ¼ olsaydı sınır 4/3 olurdu. μ'nün etkisi büyüktür: μ 1'e yaklaştıkça $1 - \mu$ sıfıra yaklaşır ve sınır patlar. Bu yüzden tanımda $\mu < 1$ şartı vardır; $\mu = 1$ olsaydı 4. adımda sıfıra bölmek gerekirdi ve hiçbir sınır çıkmazdı.

Neden önemli? İspat yalnız "kimse tek başına sapmaz" koşulunu, üstelik yalnız tek bir sapma için (optimumdaki stratejiye geçiş) kullanır. Bu koşul daha gevşek denge türlerinde de ortalama olarak sağlanır. Örneğin oyuncuların olasılıklı seçim yaptığı karışık dengelerde 1. adım beklenen değerlerle yazılabilir ve aynı sınır çıkar. Pürüzsüzlükten çıkan ve bu yolla geniş denge türlerine taşınan sınıra **sağlam PoA** (*robust Price of Anarchy*) denir.

9.3.9 Atomik tıkanıklık oyunları: PoA ≤ 5/2

Bölünebilir trafikte tek bir sürücünün etkisi yok sayılır. Oysa birçok sistemde oyuncu sayısı azdır ve her oyuncu kaynağı fark edilir ölçüde yükler: birkaç büyük veri aktarımı, birkaç sunucu işi. Bu tür oyunlara **atomik** (*atomic*) tıkanıklık oyunları denir. Model şöyledir:

- Her oyuncu bir birim yük taşır ve bir kenar kümesi (yol) seçer.
- Kenar maliyeti doğrusaldır: kenarı x oyuncu kullanıyorsa her biri $c_e(x) = a_e x + b_e$ öder.
- z_e: dengede e kenarını kullanan kişi sayısı; y_e: optimumda kullanan kişi sayısı.

Fark tek bir noktadadır. Bölünebilir trafikte yol değiştiren sürücü yeni yolun gecikmesini değiştirmez. Atomik oyunda tek başına optimumdaki yoluna geçen bir oyuncu, e kenarında en çok $z_e + 1$ kişi bulur: dengede orada olan z_e kişi ve kendisi. Bu "+1", oyuncunun kendi eklediği gecikmedir.

Anahtar eşitsizlik. $c(x) = x$ olan bir kenar için deneme maliyetlerinin toplamı en çok $y(z + 1)$'dir: optimumda kenarı kullanan y oyuncunun her biri en çok $z + 1$ öder. Optimumun o kenardaki maliyeti $y \cdot y = y^2$, dengenin maliyeti z^2'dir. Gereken pürüzsüzlük eşitsizliği, her $y, z \ge 0$ tamsayısı için:

$$y(z + 1) \le 5/3 \cdot y^2 + 1/3 \cdot z^2$$



Şekil 9.8: Anahtar eşitsizliğin tablosu. Satırlar *y* (optimumdaki yük), sütunlar *z* (dengedeki yük); hücre *3 × (sağ yan − sol yan)*'dır, kesirsiz olsun diye. Negatif hücre yoktur; turuncu 0'larda iki yan eşittir.

Şekil 9.8 eşitsizliği küçük değerlerde tek tek sınar. Hücre değeri $3 \cdot (5/3 \cdot y^2 + 1/3 \cdot z^2 - y(z+1)) = 5y^2 + z^2 - 3y(z+1)$'dir. Hiçbir hücre negatif değildir; bir betik $y, z \le 60$ için bütün değerleri denedi ve negatif değer bulmadı. Turuncu 0'lar eşitlik noktalarıdır: $(y, z) = (0, 0)$, $(1, 1)$ ve $(1, 2)$.

Örnek 9.9

1. **(y, z) = (1, 2):** sol $1 \cdot 3 = 3$; sağ $5/3 \cdot 1 + 1/3 \cdot 4 = 5/3 + 4/3 = 3$. Eşitlik; hücre değeri 0.
2. **(y, z) = (2, 3):** sol $2 \cdot 4 = 8$; sağ $5/3 \cdot 4 + 1/3 \cdot 9 = 20/3 + 3 = 29/3 \approx 9{,}67$. Sağlanır; fark $5/3$, hücre değeri $3 \cdot 5/3 = 5$.
3. **(y, z) = (1, 0):** sol $1 \cdot 1 = 1$; sağ $5/3$. Sağlanır; hücre değeri $5 - 3 = 2$.
4. **Sonuç:** eşitsizlik her kenarda geçerlidir; kenarlar toplanınca oyun $(5/3, 1/3)$-pürüzsüz olur ve $PoA \le (5/3)/(2/3) = 5/2$.

Genel doğrusal maliyet $a\,x + b$ için de aynı sonuç çıkar: a'lı kısım yukarıdaki eşitsizliğin a katıdır, b'li kısım ise (sabit maliyet kalabalıktan etkilenmediği için) $b \cdot y \le 5/3 \cdot b \cdot y$ biçiminde kendiliğinden sağlanır. 5/2 sınırı da sıkıdır; bunu Awerbuch, Azar ve Epstein ile Christodoulou ve Koutsoupias 2005'te birbirinden bağımsız gösterdiler (Awerbuch vd., 2005; Christodoulou ve Koutsoupias, 2005).

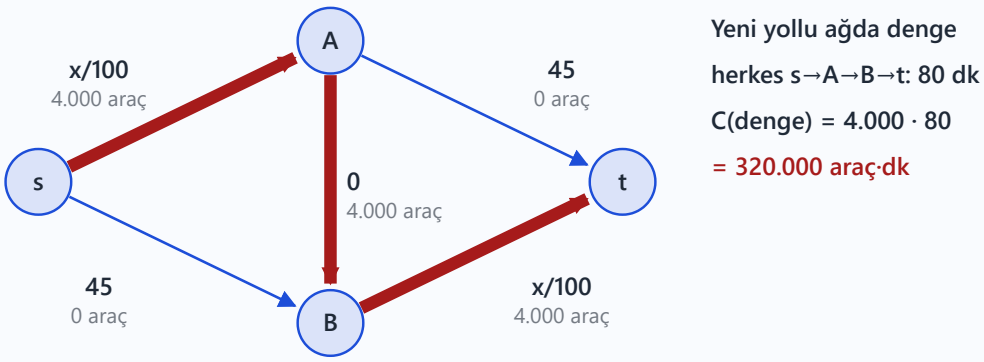
Karşılaştırma öğreticidir: bölünebilir trafikte sınır 4/3, bölünemez oyunculara 5/2'dir. Aradaki farkın kaynağı $z + 1$'deki "+1"dir: oyuncunun kendi eklediği gecikme. $(y, z) = (1, 1)$'de sol yan 2'dir; daha küçük bir λ ile, örneğin $(1, 1/3)$ ile, sağ yan $1 + 1/3 = 4/3 < 2$ olurdu ve eşitsizlik bozulurdu. Bu yüzden büyük oyuncuların oyunlarında kayıp daha büyük olabilir.

9.3.10 Braess ağında PoA

Braess paradoksu bencil yönlendirmenin en şaşırtıcı sonucudur: bir ağa yeni bir yol eklemek herkesin yolculuk süresini artırabilir. Alman matematikçi Dietrich Braess 1968'de bu olguyu gösterdi. Burada paradoksa PoA gözüyle bakılır.

Ağda s'den t'ye 4.000 araç gider. Dört eski bağlantı vardır: $s \to A$ ve $B \to t$ trafiğe bağlıdır ($x/100$ dk), $A \to t$ ve $s \to B$ sabittir (45 dk). Yeni yol açılmadan önce iki rota vardır ve simetri nedeniyle trafik eşit bölünür: her rotada 2.000 araç, süre $2.000/100 + 45 = 65$ dk. Bu dağılım hem denge hem optimumdur; PoA = 1'dir.

A ile B arasına gecikmesi 0 olan bir yol ($A \to B$) açılsın. Böylece üçüncü bir rota doğar: $s \to A \to B \to t$ (zikzak).



Şekil 9.9: Yeni yollu Braess ağında denge. 4.000 aracın hepsi zikzak rotayı ($s \to A \to B \to t$, kırmızı) kullanır; kişi başı süre $40 + 0 + 40 = 80$ dk, toplam 320.000 araç·dk.

Denge. Şekil 9.9'da herkes zikzaktadır. Denetleyelim: $s \to A$ ve $B \to t$'de 4.000'er araç vardır, her biri $4.000/100 = 40$ dk'dır. Zikzak $40 + 0 + 40 = 80$ dk sürer. Eski rotaya geçen bir sürücü $40 + 45 = 85$ dk bekler; 85 > 80 olduğundan kimse geçmez. Denge maliyeti $4.000 \cdot 80 = 320.000$ araç·dk'dır. Yeni yol herkesin süresini 65'ten 80 dakikaya çıkarmıştır.

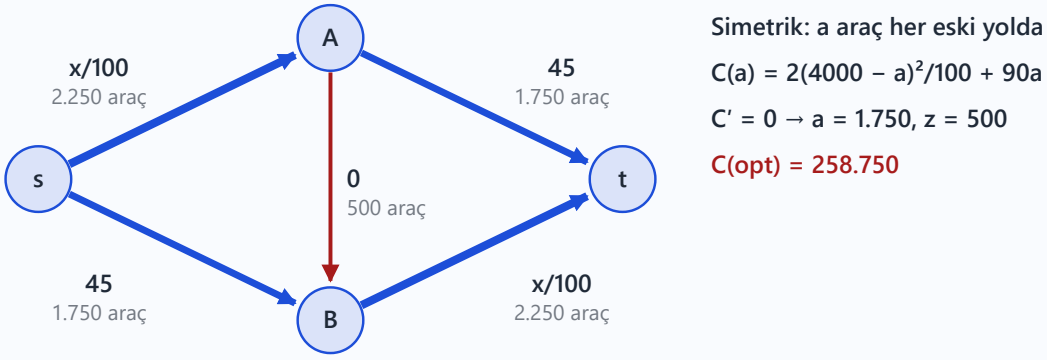
Sık yapılan kıyas: 80/65 ≈ 1,23. Bu oran yeni ağın dengesini *eski* ağın dengesiyle karşılaştırır ve Braess kaybını ölçer. Ama PoA'nın paydası *aynı ağın* optimumudur. Soru şudur: yeni yollu ağda planlayıcı 65 dakikadan daha iyi bir ortalama elde edebilir mi?

Planlayıcı zikzağı kullanır mı? Eski dağılımda (her eski rotada 2.000, zikzak boş) marjinal maliyetlere bakılır. $s \to A$'da 2.000 araç vardır ve gecikme 20 dk'dır. Marjinal maliyet $\ell + x \cdot \ell' = 20 + 2.000 \cdot (1/100) = 40$ dk'dır. Zikzak bu türden iki kenar kullanır: $40 + 0 + 40 = 80$. Eski rota ise $40 + 45 = 85$'tir (sabit kenarın marjinal maliyeti kendi gecikmesine eşittir). $80 < 85$: planlayıcı bile birkaç aracı zikzağa yollar.

Örnek 9.10

Simetri nedeniyle optimumda her eski rotada a araç, zikzakta $z = 4.000 - 2a$ araç bulunduğunu varsayalım.

1. **Kenar yükleri.** $s \to A$ kenarını $s \to A \to t$ rotasındaki a araç ve zikzaktaki z araç kullanır: $a + z = 4.000 - a$. Simetriden $B \to t$ de öyledir.
2. **Toplam maliyet.** İki değişken kenarın her biri $(4.000 - a)$ araç × $(4.000 - a)/100$ dk katkı verir; iki sabit kenarın her biri $a \cdot 45$; yeni yolun katkısı 0'dır: $C(a) = 2(4.000 - a)^2/100 + 90a$.
3. **Türev.** $(4.000 - a)^2$'nin a'ya göre türevi $-2(4.000 - a)$'dır. $C'(a) = -4(4.000 - a)/100 + 90 = 0 \Rightarrow 4.000 - a = 2.250 \Rightarrow a = 1.750, z = 500$.
4. **Maliyet.** $C(1.750) = 2 \cdot 2.250^2/100 + 90 \cdot 1.750 = 101.250 + 157.500 = 258.750$ araç·dk. Ortalama süre $258.750 / 4.000 \approx 64{,}69$ dk; 65'ten biraz iyidir.
5. **Denetim: marjinal maliyetler eşit mi?** $s \to A$'da 2.250 araç, gecikme 22,5 dk, marjinal maliyet $22{,}5 + 22{,}5 = 45$. Zikzak: $45 + 0 + 45 = 90$. Eski rota: $45 + 45 = 90$. Eşit ✓. Bir betik simetrik olmayanlar dahil bütün dağılımları taradı ve aynı optimumu buldu.
6. **Oran.** $PoA = 320.000 / 258.750 \approx 1{,}237 \le 4/3$.



Şekil 9.10: Yeni yollu Braess ağında optimum: her eski rotada 1.750, zikzakta 500 araç. $C(OPT) = 258.750$ araç·dk.

Şekil 9.10 optimum dağılımı gösterir. Yeni yol optimumu biraz iyileştirmiştir (260.000'den 258.750'ye), ama dengeyi çok kötüleştirmiştir (260.000'den 320.000'e). PoA bu iki etkiyi aynı ağın içinde ölçer.

Normalleştirilmiş Braess ağı. Ders kitaplarında ağ genellikle 1 birim trafikle ve sade gecikmelerle verilir: $s \to A$: x, $A \to t$: 1, $s \to B$: 1, $B \to t$: x, $A \to B$: 0. Dengede herkes zikzaktadır: $1 + 0 + 1 = 2$; eski rotaya geçen $1 + 1 = 2$ bekler ve kazanmaz. Optimum, zikzağı kapatıp trafiği yarı yarıya bölmektir: her sürü-

cü *1/2 + 1 = 3/2* bekler. Oran *2 / (3/2) = 4/3*'tür: bu ağ da 4/3 sınırına tam olarak ulaşır. 4.000 araçlı ağda sabit ve değişken kenarların süreleri bu kadar uyumlu değildir; bu yüzden oran sınırın altında, 1,237'de kalır.

9.3.11 Sınırlarda sık hatalar

Sık yapılan hata

- **4/3'ü evrensel sanmak:** Sınır doğrusal gecikme ve bölünebilir trafik içindir. x^p gecikmesinde PoA sınırsız büyür; atomik oyuncularda sınır 5/2'dir.
- **Karmaşık ağda oranın büyüyeceğini sanmak:** Roughgarden'ın sonucuna göre en kötü örnek her zaman iki paralel bağlantılıdır; ağ yapısı oranı büyütmez.
- **Braess'te paydaya eski ağı koymak:** 80/65 Braess kaybıdır, PoA değildir. PoA'nın paydası aynı ağın optimumudur (258.750).
- **Pürüzsüzlüğü yalnız dengede denetlemek:** Tanım her *s* durumu için istenir; ispat bu genelliği kullanır.
- **μ'yü 1 ya da daha büyük almak:** $1 - \mu \le 0$ olursa bölme yapılamaz ya da işaret değişir; sınır çıkmaz.

9.4 Yük dengeleme

Anarşinin Bedeli ilk kez trafik ağlarında değil, paralel makinelere iş dağıtan bir modelde tanımlandı (Koutsoupias ve Papadimitriou, 1999). Bu model bugün sunucu kümelerinde, bulutta görev yerleştirmede ve işlemci çekirdeklerine iş dağıtmada karşımıza çıkar. Trafikten iki farkı vardır: oyuncular bölünemez (her iş tek bir makineye gider) ve sosyal maliyet çoğu zaman toplam değil, en büyük yüktür.

9.4.1 Yük dengeleme oyunu

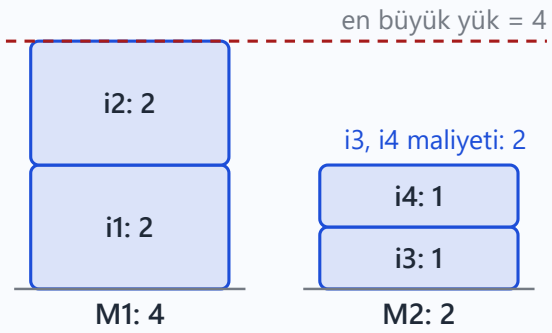
Tanım 9.5

Yük dengeleme oyununda (*load balancing game*) n iş (oyuncu) ve m özdeş makine vardır; iş j'nin boyu w_j'dir. Her iş bir makine seçer. Bir makinenin **yükü**, ona yerleşen işlerin boyları toplamıdır. Bir işin maliyeti, seçtiği makinenin yüküdür (o makinedeki işlerin hepsi makinenin yükü kadar sürede biter). **Sosyal maliyet**, makinelerin en büyük yüküdür (*makespan*), yani tüm işlerin bitiş anıdır.

Bu modelde iş j, k makinesinden h makinesine geçerse yeni maliyeti $yük(h) + w_j$ olur; çünkü h'deki mevcut yüke kendi boyunu ekler. Bu sayı şimdiki maliyet $yük(k)$'dan küçükse geçiş kârlıdır. Bir atama, hiçbir iş için kârlı bir geçiş yoksa Nash dengesidir:

> *her j ve her h için: yük(h) + w_j ≥ yük(k_j)*

Burada k_j, *j*'nin şu anki makinesidir. Kendi makinesine "geçiş" anlamsızdır: $h = k_j$ iken sol yan *yük(k_j) + w_j* olur ve sağ yandan hep büyüktür.



Şekil 9.11: İki özdeş makinede dört iş: M1'de boyu 2 olan i1 ve i2 (yük 4), M2'de boyu 1 olan i3 ve i4 (yük 2). En büyük yük 4'tür; M2'deki işlerin maliyeti 2'dir.

Bu oyunda saf Nash dengesi her zaman vardır. Bu, her oyunda doğru değildir; örneğin Taş-Kâğıt-Makas'ta saf denge yoktur. Yük dengelemede ise şu akıl yürütme dengenin varlığını gösterir. Herhangi bir atamadan başlayalım. Atama denge değilse kârlı bir geçiş vardır: iş *j*, *k*'den *h*'ye geçer ve *yük(h) + w_j < yük(k)*'dır. Geçişten sonra *k*'nin yükü azalır, *h*'nin yükü artar; ama *h*'nin yeni yükü *k*'nin eski yükünden küçüktür. Makine yükleri büyükten küçüğe sıralanıp bir liste yapılırsa, bu liste geçişten sonra "küçülür": iki liste soldan karşılaştırıldığında ilk farklı yerde yeni listede daha küçük bir sayı vardır. Sözlükteki sıralamaya benzeyen bu karşılaştırmada liste sonsuza dek küçülemez, çünkü işlerin makinelere dağılımı sonlu sayıdadır ve hiçbir liste tekrar edemez. Demek ki kârlı geçişler bir yerde biter; bittiği yer bir Nash dengesidir.

Örnek 9.11

Boyları 2, 2, 1, 1 olan işler başta hepsi M1'de olsun. Yükler (6, 0), sıralı liste [6, 0].

1. i1 (boy 2) M2'ye geçer: *0 + 2 = 2 < 6*. Yükler (4, 2), sıralı liste [4, 2]. İlk eleman 6'dan 4'e indi: liste küçüldü.
2. i3 (boy 1) M2'ye geçer: *2 + 1 = 3 < 4*. Yükler (3, 3), liste [3, 3]. Yine küçüldü.
3. Artık hiçbir iş kazanamaz: 2'lik iş öteki makinede *3 + 2 = 5*, 1'lik iş *3 + 1 = 4* bekler. Süreç (3, 3) dengesinde durur.

9.4.2 Adım adım: {2, 2, 1, 1} örneği

Boyları 2, 2, 1 ve 1 olan dört iş iki özdeş makineye dağılacaktır. Her iş kendi bitiş süresini, yani makinesinin toplam yükünü küçültmek ister. Sosyal maliyet en büyük yüktür.

Örnek 9.12

1. **Sosyal optimum.** Toplam iş *2 + 2 + 1 + 1 = 6*'dır. İki makine olduğundan en büyük yük en az *6/2 = 3*'tür. {2, 1} | {2, 1} ataması yükleri 3 ve 3 yapar ve bu alt sınıra ulaşır. OPT = 3.
2. **Başka bir atama.** İki büyük iş M1'de (yük 4), iki küçük iş M2'de (yük 2). En büyük yük 4. Bu bir denge olabilir mi?
3. **Büyük işin sapması.** i1 (boy 2) M2'ye geçerse *2 + 2 = 4* bekler; şimdi de 4 bekliyor. Kazanç yok. i2 için de aynısı geçerlidir.
4. **Küçük işin sapması.** i3 (boy 1) M1'e geçerse *4 + 1 = 5* bekler; şimdi 2 bekliyor. Daha kötü. i4 için de aynısı.
5. **Sonuç.** Kimse kazanmaz: {2, 2} | {1, 1} bir Nash dengesidir ve en büyük yükü 4'tür.
6. **Oranlar.** Optimum da bir dengedir (3 | 3'te 2'lik iş geçerse 5, 1'lik iş geçerse 4 bekler). En kötü denge 4, en iyi denge 3: *PoA = 4/3*, *PoS = 1*.



Şekil 9.12: {2, 2} | {1, 1} atamasında sapma denetimi. i1 M2'ye geçerse *2 + 2 = 4* bekler, şimdi de 4 bekliyor: kazanmaz. i3 M1'e geçerse *4 + 1 = 5 > 2* bekler.

Bu örnekte kötü dengenin oluşma nedeni Şekil 9.12'de görülür. M1 fazla yüklüdür, ama oradaki işler büyüktür; M2'ye geçince kendi boyları yüzünden yine 4 beklerler. M2'deki küçük işler ise M1'e gitmek için hiçbir neden görmez. Kimse "yükü dengelemek" için kendini feda etmez. Eşitlikte (4 < 4 yanlış) iş yerinde kalır; bu kural tanımdaki kesin küçüklükten gelir.

9.4.3 Kod izleme: nash_mi()

Bir atamanın denge olup olmadığını denetleyen fonksiyon şöyledir. `boy` iş boylarının listesi, `atama` her işin makine indisidir (0 = M1, 1 = M2), `m` makine sayısıdır.

```
def nash_mi(boy, atama, m):
  yuk = [0] * m
  for j, k in enumerate(atama):
    yuk[k] += boy[j]
  for j, k in enumerate(atama):
    for h in range(m):
      if yuk[h] + boy[j] < yuk[k]:
        return False      # j kaçar
  return True             # kimse kaçmaz
```

Fonksiyon iki aşamada çalışır:

- **Yükleri hesapla.** `yuk = [0] * m`, `m` tane sıfırdan oluşan bir liste kurar. İlk döngüde `enumerate(atama)` her iş için indisi `j` ve makinesini `k` verir; işin boyu o makinenin yüküne eklenir.
- **Sapmaları dene.** İkinci döngü her iş `j` ve her makine `h` için `yuk[h] + boy[j] < yuk[k]` koşulunu sınar. Koşul doğruysa iş `j` `h`'ye geçerek kazanır; fonksiyon hemen `False` döndürür. Hiçbir sapma kârlı değilse `True` döner.

`nash_mi([2, 2, 1, 1], (0, 0, 1, 1), 2)` çağrısının izlemesi Tablo 9.7'de verilmiştir. $h = k$ (kendi makinesi) için koşul hep yanlıştır, çünkü `yuk[k] + boy[j] < yuk[k]` olamaz; tabloda yalnız $h \neq k$ adımları gösterilir.

Adım	Değişkenler	İşlem	Sonuç
1	—	`yuk = [0, 0]`	[0, 0]
2	j = 0, k = 0	i1 (boy 2) M1'e	[2, 0]
3	j = 1, k = 0	i2 (boy 2) M1'e	[4, 0]
4	j = 2, k = 1	i3 (boy 1) M2'ye	[4, 1]
5	j = 3, k = 1	i4 (boy 1) M2'ye	[4, 2]
6	j = 0, h = 1	2 + 2 < 4?	hayır
7	j = 1, h = 1	2 + 2 < 4?	hayır
8	j = 2, h = 0	4 + 1 < 2?	hayır
9	j = 3, h = 0	4 + 1 < 2?	hayır
10	—	döngü bitti	`True`

Tablo 9.7: `nash_mi([2, 2, 1, 1], (0, 0, 1, 1), 2)` izlemesi.

Hiçbir iş kârlı bir makineye kaçamadı; fonksiyon `True` döndürür: {2, 2} | {1, 1} bir Nash dengesidir. Aynı fonksiyon (0, 0, 0, 1) atamasıyla çağrılsaydı yükler [5, 1] olurdu ve ilk sapma denemesinde (i1 için $1 + 2 = 3 < 5$) `False` dönerdi. Fonksiyonun maliyeti n iş ve m makine için $O(nm)$ karşılaştırmadır.

9.4.4 Kaba kuvvet: 16 atamanın hepsi

Dört işin her biri iki makineden birini seçebildiğinden $2^4 = 16$ atama vardır. Makineler özdeş olduğu için M1 ile M2'nin yer değiştirmesi aynı bölüşümü verir; bu yüzden 16 atama beş bölüşüme ayrılır.

Bölüşüm (M1 | M2)	Atama sayısı	En büyük yük	Nash?
{2, 1} | {2, 1}	4	3	✓ denge
{2, 1, 1} | {2}	4	4	✗
{2, 2} | {1, 1}	2	4	✓ denge
{2, 2, 1} | {1}	4	5	✗
{2, 2, 1, 1} | { }	2	6	✗

Tablo 9.8: {2, 2, 1, 1} işlerinin iki makineye bütün bölüşümleri (M1 ile M2'nin yer değiştirmesi aynı satırda sayılır).

Atama sayıları şöyle bulunur. {2, 1} | {2, 1} için 2'lik işlerden biri (2 yol) ve 1'lik işlerden biri (2 yol) M1'e gider: $2 \cdot 2 = 4$. {2, 2} | {1, 1} için büyükler ya M1'de ya M2'de: 2. Toplam $4 + 4 + 2 + 4 + 2 = 16$ ✓.

Dengesiz bölüşümlerin her birinde kaçan bir iş vardır. {2, 1, 1} | {2}'de M1'deki 1'lik iş M2'ye geçerse $2 + 1 = 3 < 4$ bekler. {2, 2, 1} | {1}'de 2'lik iş geçerse $1 + 2 = 3 < 5$. Hepsi tek makinede ise herhangi bir iş boş makineye geçerek kazanır. Dengeler en büyük yükü 3 ve 4 olanlardır: $PoA = 4/3$ (en kötü denge), $PoS = 1$ (en iyi denge = optimum).

En iyi tepki dinamiği. Oyuncuların sırayla kârlı geçiş yaptığı bu sürece **en iyi tepki dinamiği** (*best-response dynamics*) denir. Bu oyunda dinamik her zaman bir dengede durur, çünkü her kârlı geçiş sıralanmış yük listesini küçültür (9.4.1). Ama varılan denge başlangıca ve hamle sırasına bağlıdır. Örnek 9.11'deki sıra (3, 3)'e vardı; başta büyük işler M1'de, küçükler M2'deyse süreç hiç başlamaz ve (4, 2)'de kalır.

9.4.5 Kod izleme: poa_pos()

Tablo 9.8 elle kuruldu. Aynı işi bütün atamaları tarayan bir fonksiyon da yapabilir; fonksiyon her atama için `nash_mi`'yi çağırır.

```
from itertools import product
def poa_pos(boy, m):
  opt, dengeler = float("inf"), []
  for atama in product(range(m), repeat=len(boy)):
    yuk = [0] * m
    for j, k in enumerate(atama):
      yuk[k] += boy[j]
    opt = min(opt, max(yuk))
    if nash_mi(boy, atama, m):       # denge testi
      dengeler.append(max(yuk))
  return max(dengeler) / opt, min(dengeler) / opt
r = poa_pos([2, 2, 1, 1], 2)
print(round(r[0], 3), r[1])
```

- `product(range(m), repeat=len(boy))`, her işe bir makine indisi veren bütün demetleri sırayla üretir: (0, 0, 0, 0), (0, 0, 0, 1), …, (1, 1, 1, 1). Dört iş ve iki makinede 16 demet vardır.
- Her demet için yükler hesaplanır ve `opt`, şimdiye kadar görülen en küçük en büyük yük olarak güncellenir. `float("inf")` sonsuz büyük bir başlangıç değeridir; ilk karşılaştırmada mutlaka yerini gerçek bir sayıya bırakır.
- Atama dengeyse en büyük yükü `dengeler` listesine eklenir.
- Sonunda en kötü dengenin ve en iyi dengenin optimuma oranı döndürülür: `(PoA, PoS)`.

Atama	Yükler	Kaçan	Denge?	opt
(0, 0, 0, 0)	[6, 0]	i1: 0 + 2 = 2 < 6	hayır	6
(0, 0, 0, 1)	[5, 1]	i1: 1 + 2 = 3 < 5	hayır	5
(0, 0, 1, 0)	[5, 1]	i1: 1 + 2 = 3 < 5	hayır	5
(0, 0, 1, 1)	[4, 2]	yok	evet, 4	4
(0, 1, 0, 0)	[4, 2]	i3: 2 + 1 = 3 < 4	hayır	4
(0, 1, 0, 1)	[3, 3]	yok	evet, 3	3
(0, 1, 1, 0)	[3, 3]	yok	evet, 3	3
(0, 1, 1, 1)	[2, 4]	i3: 2 + 1 = 3 < 4	hayır	3
(1, 0, 0, 0)	[4, 2]	i3: 2 + 1 = 3 < 4	hayır	3
(1, 0, 0, 1)	[3, 3]	yok	evet, 3	3
(1, 0, 1, 0)	[3, 3]	yok	evet, 3	3
(1, 0, 1, 1)	[2, 4]	i3: 2 + 1 = 3 < 4	hayır	3
(1, 1, 0, 0)	[2, 4]	yok	evet, 4	3
(1, 1, 0, 1)	[1, 5]	i1: 1 + 2 = 3 < 5	hayır	3
(1, 1, 1, 0)	[1, 5]	i1: 1 + 2 = 3 < 5	hayır	3
(1, 1, 1, 1)	[0, 6]	i1: 0 + 2 = 2 < 6	hayır	3

Tablo 9.9: `poa_pos([2, 2, 1, 1], 2)` izlemesi. "Kaçan" sütunu, `nash_mi`'nin bulduğu ilk kârlı sapmadır.

Tarama bitince `dengeler = [4, 3, 3, 3, 3, 4]` ve `opt = 3`'tür. Fonksiyon *(4/3, 3/3) = (1.333…, 1.0)* döndürür; son satır bunu `1.333 1.0` olarak yazar. Yöntemin bedeli yüksektir: n iş ve m makinede m^n atama vardır. 30 iş ve 2 makinede bu sayı bir milyarı aşar. Kaba kuvvet küçük örnekleri doğrulamak için iyidir; büyük oyunlarda PoA'yı analizle sınırlamak gerekir.

9.4.6 Genel sonuç: 2 − 2/(m + 1)

{2, 2, 1, 1} örneğinde PoA 4/3 çıktı. Bu, iki makine için olabilecek en kötü değerdir. Genel sonuç şudur (Finn ve Horowitz, 1979; Koutsoupias ve Papadimitriou, 1999):

Teorem 9.4 (Özdeş makinelerde saf PoA)

m özdeş makinede sosyal maliyet en büyük yük ise saf Nash dengelerinin Anarşinin Bedeli şudur:

$$PoA = 2 - 2/(m + 1)$$

Bu değer sıkıdır.

Değerler: $m = 1$ için 1 (tek makinede seçim yoktur), $m = 2$ için 4/3, $m = 3$ için 3/2, $m = 9$ için 1,8. m büyüdükçe oran 2'ye yaklaşır ama hiçbir zaman 2'ye ulaşmaz. Bencil iş dağılımı, makine sayısı ne olursa olsun, en iyi dağılımın iki katından kötü olamaz.

Sınırın neden 2'nin altında kaldığı kısa bir akıl yürütmeyle görülür. Bir dengede en yüklü makinenin yükü L olsun. O makinede tek iş varsa L bu işin boyudur ve optimum da bu işi bir makineye koymak zorundadır; oran 1'dir. Birden çok iş varsa oradaki en küçük işin boyu w olsun. Dengede bu iş başka bir makineye geçmek istemez: her makine h için $yük(h) + w \geq L$, yani her makinenin yükü en az $L - w$'dir. Toplam iş miktarı bu yüzden en az $L + (m - 1)(L - w)$'dir. Optimum, toplam işin makine sayısına bölümünden (ortalama yükten) küçük olamaz: $OPT \geq L - w(m - 1)/m$. Ayrıca en yüklü makinede en az iki iş ve en küçüğü w olduğundan $w \leq L/2$'dir. İkinci bilgi birinciye konunca $OPT \geq L - (L/2)(m - 1)/m = L \cdot (m + 1)/(2m)$ çıkar. İki yan düzenlenince $L / OPT \leq 2m/(m + 1) = 2 - 2/(m + 1)$ elde edilir. Örnek 9.13 bu sınıra ulaşıldığını, yani sınırın sıkı olduğunu gösterir.

Örnek 9.13

Üç özdeş makineye boyları 3, 3, 3, 1, 1, 1 olan altı iş dağılacaktır. Sınır $2 - 2/4 = 3/2$'dir. Bu örnek sınıra ulaşır.

1. **Optimum.** Toplam iş 12, üç makine: en büyük yük en az 4. {3, 1} | {3, 1} | {3, 1} bunu sağlar: OPT = 4.
2. **Kötü atama.** {3, 3} | {3} | {1, 1, 1}: yükler 6, 3, 3.
3. **M1'deki 3'lük iş:** M2'ye geçerse $3 + 3 = 6$, M3'e geçerse $3 + 3 = 6$ bekler; şimdi de 6 bekliyor. Kazanmaz.
4. **M2'deki 3'lük iş:** şimdi 3 bekliyor; M3'e geçse 6, M1'e geçse 9. Kazanmaz.
5. **M3'teki 1'lik iş:** şimdi 3 bekliyor; M2'ye geçse 4, M1'e geçse 7. Kazanmaz.
6. **Sonuç.** Atama bir dengedir ve oran $6/4 = 3/2$'dir. Bütün $3^6 = 729$ atamayı tarayan bir betik, dengelerin en büyük yüklerinin 4 ve 6 olduğunu doğruladı.

Karışık dengeler. İşler makineyi olasılıkla seçebilirse (karışık strateji) durum değişir. Bu durumda sosyal maliyet olarak beklenen en büyük yük kullanılır ve PoA artık sabit bir sayıyla sınırlanamaz: m ile çok yavaş ama sınırsız büyür. Büyüme hızı $\log m / \log \log m$ düzeyindedir (Koutsoupias vd., 2003;

Czumaj ve Vöcking, 2002). Bu, “toplar ve kutular” problemine benzer: *m* top *m* kutuya rastgele atılınca en dolu kutuda beklenenden fazla top birikir. Rastgelelik, dengede bile şanssız çakışmalara yol açar.

9.4.7 Farklı hızlı (ilişkili) makineler

Gerçek sunucular aynı hızda değildir. **İlişkili makineler** (*related machines*) modelinde makine *i*'nin hızı s_i'dir ve bir işin bitiş süresi, makinesinin toplam yükünün s_i'ye bölümüdür. Hızlı makine aynı yükü daha kısa sürede bitirir. Bu modelde de saf Nash dengesi her zaman vardır (aynı “sıralı liste küçülür” fikri bitiş süreleriyle çalışır).

Örnek 9.14

Hızlar A = 2, B = 1, C = 1; iş boyları 2, 1, 1. Sosyal maliyet en büyük bitiş süresidir.

1. **Optimum.** A {2}, B {1}, C {1}: süreler *2/2 = 1*, *1/1 = 1*, *1/1 = 1*. En büyük süre 1. Bu en iyisidir, çünkü 2'lik iş en hızlı makinede bile 1 sürer.
2. **Kötü atama.** A {1, 1}, B {2}, C { }: süreler *2/2 = 1*, *2/1 = 2*, 0. En büyük süre 2.
3. **2'lik işin sapması.** Şimdi 2 bekliyor. A'ya geçerse *(2 + 2)/2 = 2*, C'ye geçerse *2/1 = 2*. Kazanmaz.
4. **1'lik işin sapması.** Şimdi 1 bekliyor. C'ye geçerse *1/1 = 1*, B'ye geçerse *3/1 = 3*. Kazanmaz.
5. **Sonuç.** İkinci atama da bir dengedir. Oran *2/1 = 2*.

Atama	Süreler (A, B, C)	En büyük	Nash?
A {2}, B {1}, C {1}	1, 1, 1	1	evet (optimum)
A {1, 1}, B {2}, C { }	1, 2, 0	2	evet

Tablo 9.10: İlişkili makinelerde iki denge (hızlar A = 2, B = 1, C = 1; işler 2, 1, 1).

Oran 2'dir ve bu, özdeş üç makinenin sınırı *2 − 2/4 = 1,5*'ten büyüktür. Özdeş makinelerde saf dengenin kaybı hep 2 katın altında kalır. Hızlar farklıysa kayıp, makine sayısı arttıkça (çok yavaş da olsa) sınırsız büyür; büyüme *log m / log log m* düzeyindedir (Czumaj ve Vöcking, 2002). Yük Dengeleme oyununun 2. ve 3. seviyeleri bu modeli kullanır.

9.4.8 Oyuna dönüş: toplam bitiş süresi

Bölümün başındaki Yük Dengeleme oyununun ölçüsü farklıdır. Oyunda sosyal maliyet en büyük yük değil, **bütün işlerin bitiş sürelerinin toplamıdır**. Seviye 2'de (oyundaki adı *CPU Scheduling*) boyu 1 olan 7 iş vardır ve sunucu hızları A = 2, B = 1, C = 1'dir. Bir sunucuda *L* iş varsa her biri *L / s* sürede biter; o sunucunun toplama katkısı *L · L / s*'dir.

Yükler (A, B, C)	Toplam süre	Nash?
(3, 2, 2)	3 · 1,5 + 2 · 2 + 2 · 2 = 12,5	evet (optimum)
(4, 2, 1)	4 · 2 + 2 · 2 + 1 · 1 = 13	evet (en kötü)

Tablo 9.11: Seviye 2'nin iki dengesi (7 birim iş, hızlar 2, 1, 1).

Örnek 9.15

1. **(4, 2, 1) denge mi?** A'daki bir iş 2 bekler; C'ye geçerse *(1 + 1)/1 = 2*, B'ye geçerse 3 bekler. Kazanmaz. B'deki bir iş 2 bekler; C'ye geçerse 2, A'ya geçerse *5/2 = 2,5*. Kazanmaz. C'deki iş 1 bekler; en iyisi zaten odur. Denge ✓.
2. **(3, 2, 2) denge mi?** A'daki iş 1,5 bekler; B ya da C'ye geçse 3. B'deki iş 2 bekler; A'ya geçse *4/2 = 2* (eşit, kazanmaz), C'ye geçse 3. Denge ✓.
3. **Oran.** *PoA = 13 / 12,5 = 1,04*. Simetriği (4, 1, 2) de aynı maliyetle (13) bir dengedir. Bütün yük dağılımlarını tarayan bir betik, dengelerin toplam sürelerinin yalnız 12,5 ve 13 olduğunu ve en küçük toplamın 12,5 olduğunu doğruladı.

Bencil yerleşim nereye varır? Oyunda işler sırayla gelir. Her iş, geldiği anda kendi bitiş süresi en kısa olacak sunucuyu seçerse (eşitlikte alfabede önce gelen) ne olur? Tablo 9.12 bu süreci adım adım izler.

İş	A'ya giderse	B'ye giderse	C'ye giderse	Seçim	Yükler
1	1/2 = 0,5	1	1	**A**	(1, 0, 0)
2	2/2 = 1	1	1	**A** (eşitlik)	(2, 0, 0)
3	3/2 = 1,5	1	1	**B** (eşitlik)	(2, 1, 0)
4	1,5	2	1	**C**	(2, 1, 1)
5	1,5	2	2	**A**	(3, 1, 1)
6	4/2 = 2	2	2	**A** (eşitlik)	(4, 1, 1)
7	5/2 = 2,5	2	2	**B** (eşitlik)	(4, 2, 1)

Tablo 9.12: Açgözlü bencil yerleşim. Her satırda yeni işin üç sunucudaki bitiş süresi; seçilen sunucu kalın yazılmıştır.

Sıra A, A, B, C, A, A, B olur ve sonuç tam olarak en kötü dengelerden biri olan (4, 2, 1)'dir. Oyunun bencil denge gösterimi (*Selfish equilibrium*) de bu açgözlü yerleşimle kurulur. Açgözlü yerleşim her adımda yeni iş için en iyisini seçer, ama sonradan gelen işlerin etkisini öngöremez; 6. işin A'ya gitmesi, 7. iş için A'yı pahalı hâle getirir.

Seviye 1'de (6 iş, özdeş üç sunucu) tek denge (2, 2, 2)'dir ve bu optimumdur: *PoA = 1*. Kayıp, sunucular farklılaşınca ortaya çıkar. Son olarak bir karşılaştırma: {2, 2, 1, 1} örneğindeki 4/3 en büyük yükü ölçer, oyun ise toplam süreyi ölçer. **PoA'nın değeri ölçülen sosyal maliyete bağlıdır**; "bu oyunun PoA'sı" denirken hangi maliyetin kastedildiği her zaman belirtilmelidir.

9.4.9 Yük dengelemede sık hatalar

Sık yapılan hata

- **Sapmada kendi boyunu eklememek:** İş h'ye geçince *yük(h)* değil, *yük(h)* + w_j bekler. {2, 2} | {1, 1}'de büyük iş M2'yi 2 yüklü görüp "2 < 4, kaçar" demek yanlıştır; geçince 4 bekler.
- **Eşitlikte kaçmak:** Tanımdaki koşul kesin küçüklüktür (<). Eşit maliyetli bir geçiş kârlı değildir; iş yerinde kalır.
- **Optimumu her zaman bir denge sanmak:** Bu örnekte optimum bir dengedir (PoS = 1), ama her oyunda böyle değildir; Pigou ağında optimum denge değildir.
- **Ölçütü karıştırmak:** En büyük yük ile toplam süre farklı PoA değerleri verir. Hesaba başlamadan sosyal maliyet açıkça yazılmalıdır.
- **Karışık dengelerin sınırını saf dengelere taşımak:** $2 - 2/(m + 1)$ saf dengeler içindir; karışık dengelerde oran m ile yavaşça büyür.

9.5 İstikrarın Bedeli

Şimdiye kadarki örneklerin çoğunda PoA ile PoS aynıydı ya da PoS 1'di. İki oranın çok farklı olabildiği bir oyun ailesi de vardır: oyuncuların bağlantı kiralayıp bedeli kullananlar arasında eşit böldüğü ağ tasarımı oyunları. Bu oyunlarda en kötü denge çok kötü olabilir, ama her zaman iyi bir denge de vardır. İyi dengenin ne kadar iyi olduğunu gösteren araç, potansiyel fonksiyonudur.

9.5.1 Aynı oyunda PoA ≠ PoS

Tanım 9.6

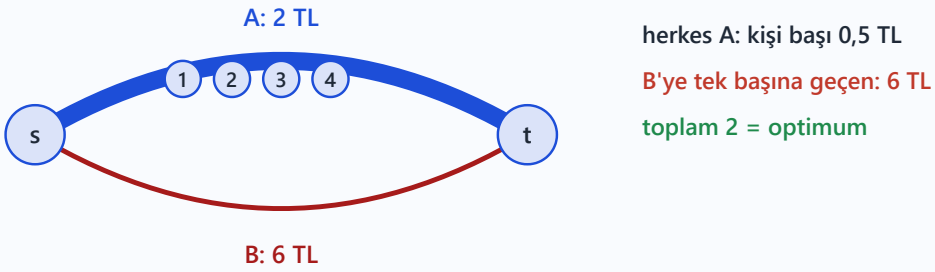
Adil maliyet paylaşımlı (*fair cost sharing*) ağ tasarımı oyununda her bağlantının sabit bir bedeli c_e vardır. Her oyuncu kendi başlangıç ve hedef düğümleri arasında bir yol seçer. Bir bağlantıyı n_e oyuncu kullanıyorsa her biri c_e / n_e öder. Toplam maliyet, kullanılan bağlantıların bedelleri toplamıdır.

Bu oyunda tıkanıklığın tersi bir etki vardır: bir bağlantıyı ne kadar çok kişi kullanırsa kişi başı bedel o kadar düşer. Oyuncular birbirini itmez, çeker. Bu yüzden oyuncuların aynı bağlantıda toplanması iyidir; sorun hangi bağlantıda toplanacaklarıdır.

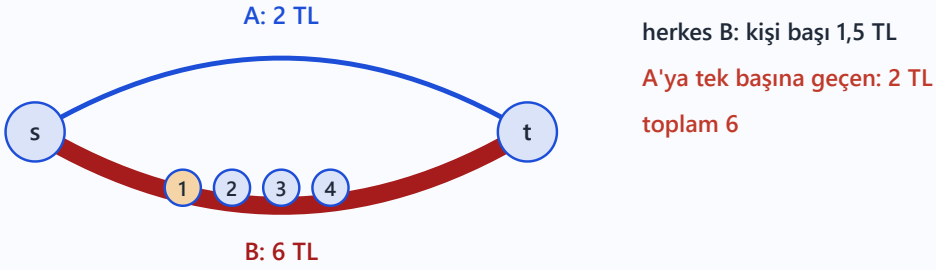
Örnek 9.16

4 oyuncu s'den t'ye bağlantı kiralar. İki seçenek vardır: A bağlantısı 2 TL, B bağlantısı 6 TL. Bedeli kullananlar eşit böler.

1. **Herkes A'da.** Her biri *2/4 = 0,5* TL öder. B'ye tek başına geçen 6 TL öderdi. Kimse geçmez: denge. Toplam 2 TL, bu aynı zamanda optimumdur (en ucuz bağlantı kullanılmıştır).
2. **Herkes B'de.** Her biri *6/4 = 1,5* TL öder. A'ya tek başına geçen 2 TL öderdi; *2 > 1,5* olduğundan kimse geçmez. Bu da bir dengedir; toplam 6 TL.
3. **Bölünmüş profil (1 A, 3 B).** A'daki oyuncu 2 TL, B'dekiler *6/3 = 2* TL öder. B'deki bir oyuncu A'ya geçerse A'yı iki kişi paylaşır ve *2/2 = 1* TL öder; *1 < 2*, kaçar. Denge değil.
4. **Bütün profiller.** $2^4 = 16$ profilin hepsi tarandığında yalnız "hepsi A" ve "hepsi B" denge çıkar. 2 A, 2 B profilinde B'deki oyuncu 3 TL öder, A'ya geçerse *2/3* TL öderdi; 3 A, 1 B'de B'deki 6 TL öder, A'ya geçerse 0,5.
5. **Oranlar.** *PoA = 6/2 = 3* (en kötü denge), *PoS = 2/2 = 1* (en iyi denge).



Şekil 9.13: İyi denge: dört oyuncunun hepsi A bağlantısında (2 TL), kişi başı 0,5 TL. B'ye tek başına geçen 6 TL öderdi. Toplam 2 = optimum.



Şekil 9.14: Kötü denge: dört oyuncunun hepsi B bağlantısında (6 TL), kişi başı 1,5 TL. A'ya tek başına geçen 2 TL öderdi; kimse geçmez. Toplam 6.

Şekil 9.13 ile Şekil 9.14 aynı oyunun iki dengesidir. Kötü dengede herkes birlikte A'ya geçse herkes kazanırdı (1,5 yerine 0,5 TL). Ama tek başına geçen 2 TL öder ve kaybeder. Sorun bir **koordinasyon sorunu**dur: oyuncuların hepsi aynı anda hareket edebilse kayıp ortadan kalkar. PoS, bu iyimser senaryonun ölçüsüdür. Biri oyunculara "hepiniz A'yı kullanın" derse ve bu öneri bir denge olduğu için kimse ondan sapmak istemezse kayıp yoktur. PoA ise öneri yapılmazsa ne kadar kaybedilebileceğini söyler.

PoA ne kadar büyük olabilir? Aynı yapı daha uç sayılarla kurulabilir. n oyuncu, bedeli 1 olan bir bağlantı ve bedeli n olan bir bağlantı olsun. Herkes pahalı bağlantıdaysa her biri $n/n = 1$ öder. Ucuz bağlantıya tek başına geçen de 1 öderdi; geçiş kârlı değildir. Bu profil bir dengedir ve toplam maliyeti n'dir; optimum 1'dir. Demek ki adil paylaşım oyunlarında PoA n kadar büyük olabilir. Oyuncu sayısı arttıkça en kötü denge sınırsız kötüleşir.

9.5.2 Adım adım: H_n örneği

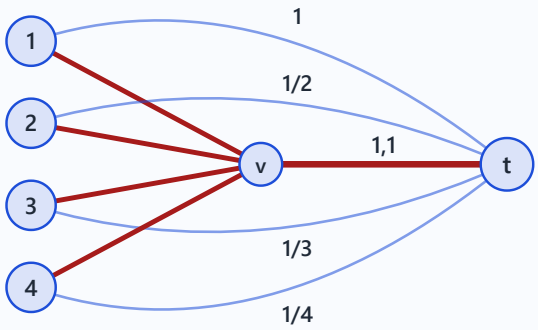
PoA kötü olabildiğine göre soru PoS'a döner: en iyi denge her zaman optimuma yakın mıdır? Cevap "evet, ama tam değil"dir. Aşağıdaki örnekte en iyi denge bile optimumdan belli bir oranda kötüdür. Örnekte **harmonik sayı** (*harmonic number*) ortaya çıkar:

> $$H_n = 1 + 1/2 + 1/3 + \ldots + 1/n$$

Örneğin $H_1 = 1$, $H_2 = 3/2$, $H_3 = 11/6 \approx 1{,}83$, $H_4 = 25/12 \approx 2{,}08$'dir. H_n çok yavaş büyür ve yaklaşık $\ln n$'dir; $H_{100} \approx 5{,}19$'dur.

Kurulum. 4 oyuncu vardır ve her biri t'ye iki yoldan biriyle gider: yalnız kendisinin kullandığı, bedeli $1/i$ olan yol (oyuncu 1 için 1, oyuncu 2 için 1/2, oyuncu 3 için 1/3, oyuncu 4 için 1/4) ya da herkesin ortak kullanabildiği, bedeli $1 + \varepsilon = 1{,}1$ olan yol. Burada ε küçük bir ek bedeldir (0,1). Ortak yolun bedeli kullananlarca eşit bölünür.

Optimum. Herkes ortak yolu kullanırsa toplam 1,1'dir. Herkes kendi yolunu kullanırsa toplam $1 + 1/2 + 1/3 + 1/4 = 25/12 \approx 2{,}083$'tür. Bölünmüş seçimler de denenebilir, ama ortak yol kullanılıyorsa bedeli bir kez ödenir ve kendi yollarını ekleyenler toplamı yalnızca artırır. Optimum herkesin ortak yolda olduğu durumdur: toplam 1,1, kişi başı 0,275.



oyuncu	yol	öder
1	ortak	0,275
2	ortak	0,275
3	ortak	0,275
4	ortak	0,275
toplam		1,1

Şekil 9.15: H_n örneğinde optimum: dört oyuncu ortak yolu (bedel 1,1) kullanır, her biri 0,275 öder. İnce çizgiler kullanılmayan kendi yollarıdır (bedelleri 1, 1/2, 1/3, 1/4).

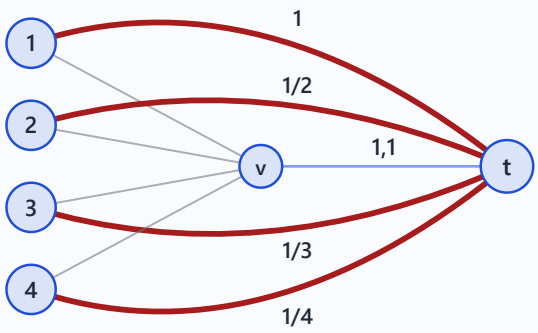
Örnek 9.17

Optimumdan başlanır ve en iyi tepki dinamiği izlenir: her adımda kendi yoluna geçerek kazanacak bir oyuncu varsa geçer.

1. **Oyuncu 4.** Ortak yolda 0,275 öder; kendi yolu *1/4 = 0,25 < 0,275*. Ayrılır. Kalan üç kişi artık *1,1/3 ≈ 0,367* öder.
2. **Oyuncu 3.** Kendi yolu *1/3 ≈ 0,333 < 0,367*. Ayrılır. Kalan iki kişi *1,1/2 = 0,55* öder.
3. **Oyuncu 2.** Kendi yolu *1/2 = 0,5 < 0,55*. Ayrılır. Oyuncu 1 ortak yolu tek başına öder: 1,1.
4. **Oyuncu 1.** Kendi yolu *1 < 1,1*. Ayrılır.
5. **Son durum.** Herkes kendi yolunda. Kimse ortak yola dönmek istemez: tek başına dönen 1,1 öderdi ve 1,1 her oyuncunun kendi yolundan pahalıdır. Toplam *1 + 1/2 + 1/3 + 1/4 = 25/12 ≈ 2,083*.

Tek denge. Ortak yolda *k ≥ 1* kişi varsa bunların en büyük numaralısının numarası en az *k*'dir; onun kendi yolu *1/i ≤ 1/k < 1,1/k* olduğundan ayrılır. Demek ki ortak yolu kullanan hiçbir profil denge değildir ve tek denge "herkes kendi yolunda"dır. 16 profili tarayan bir betik de bunu doğruladı.

Oran. Tek denge olduğundan *PoS = PoA = (25/12) / 1,1 = 125/66 ≈ 1,894*.



oyuncu	yol	öder
1	kendi	1
2	kendi	0,5
3	kendi	0,333
4	kendi	0,25
toplam		2,083

Şekil 9.16: H_n örneğinde tek denge: herkes kendi yolunda; ödemeler 1, 0,5, 0,333 ve 0,25, toplam *25/12 ≈ 2,083*.

Örnekteki çözülme bir zincir tepkimedir. Ortak yoldan en ucuz alternatifi olan oyuncu ayrılır; bu, kalanların payını artırır ve bir sonrakini ayrılmaya iter. ε sıfıra yaklaşırken *n* oyunculu aynı yapıda PoS $H_n = 1 + 1/2 + \dots + 1/n$'ye yaklaşır: optimum 1'e, tek denge H_n'ye gider. Dört oyuncuda bu sınır *25/12 ≈ 2,08*'dir; ε = 0,1 ile 1,894 bulduk.

9.5.3 Rosenthal potansiyeli

H_n örneği PoS'un en az H_n'ye yaklaşabildiğini gösterdi. Bunun bir üst sınır olduğunu, yani hiçbir adil paylaşım oyununda PoS'un H_n'yi aşmadığını göstermek için bir yardımcı fonksiyon gerekir.

Gösterim. c_e: e bağlantısının bedeli; n_e: e'yi kullanan oyuncu sayısı (her biri c_e/n_e öder). $H(k) = 1 + 1/2 + \dots + 1/k$ ve $H(0) = 0$; $H_n = H(n)$. Toplam maliyet $C(s)$: kullanılan bağlantıların bedelleri toplamı.

Tanım 9.7

Adil paylaşım oyununda **Rosenthal potansiyeli** (*Rosenthal potential*) şöyle tanımlanır:

$$\Phi(s) = \sum_{e} c_e \cdot H(n_e)$$

Tanım bir hayalî sayımdan gelir. Oyuncular bir bağlantıya tek tek eklenmiş gibi düşünülür: 1. kişi tek başına olduğu için c_e yazar, 2. kişi $c_e/2$, …, k'ıncı kişi c_e/k yazar. Bu sayıların toplamı $c_e \cdot (1 + 1/2 + \dots + 1/k) = c_e \cdot H(k)$'dir. Her kişi "geldiği andaki" payını yazmıştır; potansiyel bu payların bütün bağlantılar üzerinden toplamıdır.

Potansiyelin anahtar özelliği. Bir oyuncu yolunu değiştirdiğinde, kendi maliyetindeki değişim potansiyeldeki değişime tam olarak eşittir. Nedeni şudur: oyuncu bir e bağlantısından ayrılınca o bağlantının sayısı n_e'den $n_e - 1$'e iner ve potansiyelden tam olarak c_e/n_e düşer (toplamın son terimi), bu da oyuncunun orada ödediği paydır. Oyuncu yeni bir e' bağlantısına katılınca o bağlantının sayısı bir artar ve potansiyele $c_{e'}/(n_{e'} + 1)$ eklenir; bu da oyuncunun orada ödeyeceği yeni paydır. Bu yüzden sapan oyuncu kazanıyorsa (maliyeti düşüyorsa) potansiyel de aynı miktarda düşer. Böyle bir fonksiyonu olan oyunlara **potansiyel oyun** (*potential game*) denir.

Teorem 9.5 (Potansiyel ve maliyet)

Her s profili için $C(s) \le \Phi(s) \le H_n \cdot C(s)$.

İspat. Kullanılan bir bağlantıda en az bir, en çok n kullanıcı vardır; bu yüzden $1 = H(1) \le H(n_e) \le H(n) = H_n$'dir. Her terim $c_e \ge 0$ ile çarpılır: $c_e \le c_e \cdot H(n_e) \le H_n \cdot c_e$. Kullanılmayan bağlantılarda $H(0) = 0$ olduğundan katkıları hem maliyette hem potansiyelde sıfırdır. Bütün bağlantılar üzerinden toplanınca sol taraf $C(s)$, orta $\Phi(s)$, sağ taraf $H_n \cdot C(s)$ olur. □

Örnek 9.18

H_n örneğinde (dört oyuncu, $\varepsilon = 0{,}1$):

1. **Herkes ortak yolda.** Kullanılan tek bağlantı ortak yoldur ve 4 kullanıcısı vardır. $C = 1{,}1$, $\Phi = 1{,}1 \cdot H(4) = 1{,}1 \cdot 25/12 \approx 2{,}29$. $C \le \Phi$; ayrıca $\Phi = H_4 \cdot C$: üst sınıra eşit, çünkü bağlantıyı dört oyuncunun hepsi kullanır.

2. **Herkes kendi yolunda.** Her yolu tek kişi kullanır; *H(1) = 1*. *Φ = C = 25/12 ≈ 2,08*: alt sınıra eşit.
3. **Bir hamlenin etkisi.** Optimumdan oyuncu 4 ayrılınca maliyeti *0,275 − 0,25 = 0,025* düşer. Potansiyel: ortak yol *1,1 · H(3) = 1,1 · 11/6 ≈ 2,017*, oyuncu 4'ün yolu *1/4 · H(1) = 0,25*; toplam ≈ 2,267. Eski potansiyel ≈ 2,292; düşüş ≈ 0,025 ✓.

9.5.4 PoS ≤ H_n: optimumdan bir dengeye yürümek

Teorem 9.6 (Anshelevich vd., 2004)

n oyunculu her adil maliyet paylaşımlı ağ tasarımı oyununda *PoS* ≤ H_n'dir ve sınır sıkıdır.

İspat. **1. Optimumdan başla.** s^{OPT} profilinde en iyi tepki dinamiği çalıştırılır: her adımda bir oyuncu kendisi için daha ucuz bir yola geçer.

2. Φ her hamlede azalır. Potansiyel oyunda sapan oyuncunun maliyet değişimi Φ'nin değişimine eşittir; oyuncu kazanıyorsa Φ düşer. Φ her adımda kesin olarak düştüğü için aynı profile iki kez dönülemez. Profil sayısı sonlu olduğundan dinamik bir *s* profilinde durur. Durduğu yerde kimse kazançlı bir geçiş bulamaz; *s* bir Nash dengesidir ve $\Phi(s) \le \Phi(s^{OPT})$'dir.

3. Zinciri kur.

$$C(s) \le \Phi(s) \le \Phi(s^{OPT}) \le H_n \cdot C(s^{OPT})$$

Birinci eşitsizlik Teorem 9.5'in sol yanıdır (potansiyel maliyetten küçük değil). İkincisi 2. adımdır. Üçüncüsü yine Teorem 9.5'in sağ yanıdır (potansiyel maliyetin en çok H_n katı). Demek ki en az bir dengenin maliyeti optimumun en çok H_n katıdır: *PoS* ≤ H_n. Sıkılık H_n örneğinden gelir. □

H_n örneğinde zincirin sayıları şöyledir: *C(s) ≈ 2,08*, *Φ(s) ≈ 2,08*, *Φ(OPT) ≈ 2,29*, $H_4 \cdot C(OPT) = 25/12 \cdot 1{,}1 \approx 2{,}29$. Yani *2,08 ≤ 2,08 ≤ 2,29 ≤ 2,29*. Birinci ve üçüncü eşitsizlik eşitlik olarak sağlanır; örnek sınıra bu yüzden bu kadar yaklaşır.

İspatın gücü, hangi dengeye varılacağını bilmeye gerek duymamasıdır. Yalnız "potansiyel düşer" bilgisi yeter. Aynı yöntem PoA için çalışmaz: en kötü denge, optimumdan dinamikle ulaşılamayan bir yerde olabilir ve potansiyeli optimumunkinden büyük olabilir. Nitekim bu oyunlarda PoA *n*'ye kadar çıkabilirken PoS en çok $H_n \approx \ln n$'dir. 100 oyuncuda bu fark 100'e karşı yaklaşık 5,19'dur.

9.5.5 İstikrarın Bedeli'nde sık hatalar

Sık yapılan hata

- **Optimumun hep denge olduğunu sanmak:** A/B örneğinde optimum bir dengeydi; H_n örneğinde değildir. PoS = 1 ancak optimum bir dengeyse olur.
- **Adil paylaşımı tıkanıklıkla karıştırmak:** Burada kalabalık kişi başı bedeli düşürür. Tıkanıklık oyununda kalabalık gecikmeyi artırır. Sapma hesabında payda $n_e + 1$ (yeni gelen dahil) olmalıdır.
- **Potansiyeli maliyetle karıştırmak:** Φ bir hesap aracıdır, kimsenin ödediği bir bedel değildir. Toplam maliyet her bağlantı için c_e, potansiyel $c_e \cdot H(n_e)$ katkı verir.
- **PoS sınırını PoA'ya taşımak:** H_n yalnız en iyi denge için bir garantidir. En kötü denge optimumun n katı olabilir.

9.6 PoA'yı azaltma

PoA'yı ölçmek ilk adımdır; ikinci adım onu küçültmektir. Kayıp dışsallıktan geliyorsa çözüm, dışsallığı oyuncunun kendi hesabına sokmaktır. Bunun en doğrudan aracı tıkanıklık ücretidir; başka araçlar da vardır.

9.6.1 Tıkanıklık ücreti dengeyi kaydırır

9.2.6'daki iki yollu ağı yeniden ele alalım: $\ell_1 = 5 + x_1/100$, $\ell_2 = 20 + x_2/200$, toplam 3.000 araç. Ücret yokken denge $x_1 = 2.000$, toplam 75.000 araç·dk; optimum $x_1 = 1.500$, toplam 71.250 araç·dk'dır.

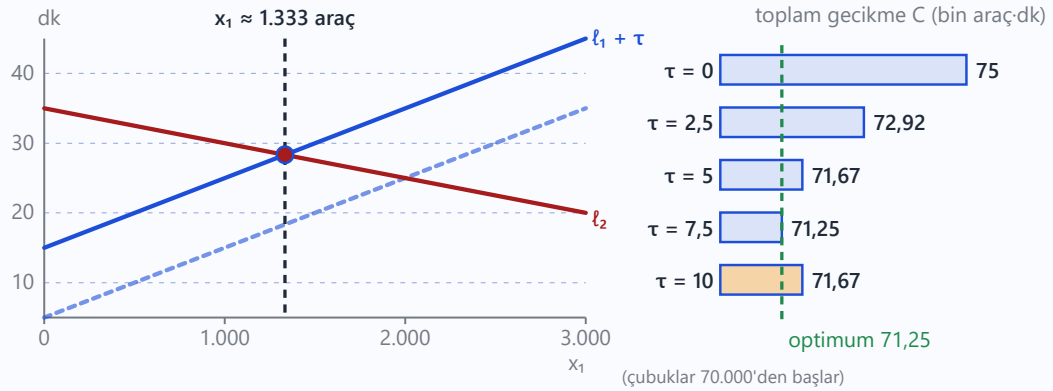
Yol 1'e bir **tıkanıklık ücreti** (*congestion toll*) τ konsun. Hesap basit kalsın diye ücret, sürücü için dakika cinsinden bir ek maliyet sayılır: τ = 2,5'lik ücret, sürücünün gözünde 2,5 dakikalık ek gecikmeyle aynıdır. Sürücü artık Yol 1'de $\ell_1 + \tau$ görür ve dengede $\ell_1 + \tau = \ell_2$ olur:

1. $5 + \tau + x_1/100 = 20 + (3.000 - x_1)/200$.
2. İki yan 200 ile çarpılır: $1.000 + 200\tau + 2x_1 = 7.000 - x_1$.
3. $3x_1 = 6.000 - 200\tau \Rightarrow x_1 = (6.000 - 200\tau)/3$.

τ	x_1	x_2	Toplam gecikme (araç·dk)
0	2.000	1.000	75.000
2,5	≈ 1.833	≈ 1.167	≈ 72.917
5	≈ 1.667	≈ 1.333	≈ 71.667
7,5	1.500	1.500	71.250 (optimum)

τ	x_1	x_2	Toplam gecikme (araç·dk)
10	≈ 1.333	≈ 1.667	≈ 71.667

Tablo 9.13: Yol 1'deki ücret τ (dk) arttıkça denge ve toplam gecikme. Toplam gecikmeye ücret eklenmez.



Şekil 9.17: İki yollu ağda ücretin etkisi. Solda Yol 1'in sürücüye görünen maliyeti $\ell_1 + \tau$ (mavi) ile ℓ_2 (kırmızı) kesişir; τ arttıkça kesişim sola kayar. Sağda her τ için toplam gecikme; en küçük değer τ = 7,5'te (71.250), τ = 10'da toplam yeniden artar.

Şekil 9.17'de ücretin etkisi iki biçimde görülür. Soldaki grafikte Yol 1'in eğrisi τ kadar yukarı kayar ve iki eğrinin kesişimi (denge) sola gider: daha az sürücü Yol 1'i seçer. Sağdaki çubuklar toplam gecikmeyi gösterir. Ücret arttıkça bazı sürücüler geniş Yol 2'ye geçer ve toplam düşer.

τ = 7,5'te optimum bir denge olur. $x_1 = (6.000 - 1.500)/3 = 1.500$, tam optimumun dağılımıdır. 7,5 sayısı rastgele değildir. Marjinal maliyet ücreti her yola o yoldaki dışsallığı koyar: Yol 1'e $x_1 \cdot \ell_1' = 1.500 \cdot 1/100 = 15$ dk, Yol 2'ye $x_2 \cdot \ell_2' = 1.500 \cdot 1/200 = 7,5$ dk. Bu iki ücret konursa sürücü Yol 1'de *20 + 15 = 35*, Yol 2'de *27,5 + 7,5 = 35* görür; eşittir, yani optimum dengedir. Ama sürücünün kararını yalnız iki yol arasındaki fark etkiler: *15 − 7,5 = 7,5*. Bu yüzden yalnız Yol 1'e konan 7,5 dk'lık ücret aynı sonucu verir.

Ücret bir transferdir. Ücret sürücüden toplanıp başka bir yere (belediyeye, yol yapımına) gider; kimsenin süresini uzatmaz. Bu yüzden tablodaki toplam gecikmeye eklenmez. Ama sürücülerin gözünden durum farklıdır: τ = 7,5'te Yol 1'deki 1.500 sürücü toplam *1.500 · 7,5 = 11.250* dk'lık ücret öder. Sürücülerin "gecikme + ücret" toplamı *71.250 + 11.250 = 82.500*'dür ve ücretsiz dengedeki 75.000'den büyüktür. Toplanan gelir sürücülere geri dağıtılmazsa sürücüler bir bütün olarak kaybeder; kazanç topluma (gelirin kullanıldığı yere) geçer.

Fazla ücret aşırıya kaçar. τ = 10'da $x_1 \approx 1.333$'e iner ve Yol 2 gereğinden fazla dolar; toplam gecikme yine yaklaşık 71.667'ye çıkar. Doğru ücret, dışsallığa eşit olandır; "ne kadar çok o kadar iyi" değildir.

9.6.2 Marjinal maliyet ücreti Pigou'yu düzeltir

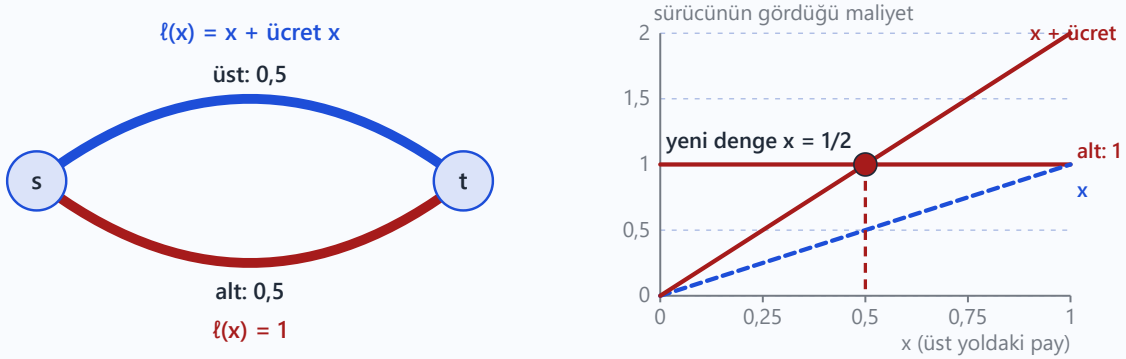
Pigou ağında amaç, optimumu ($x = 1/2$) bir denge yapmaktır. Ücretsizken herkes üst yoldadır ve toplam gecikme 1'dir.

Tanım 9.8

Marjinal maliyet ücreti (*marginal-cost toll*), her kenara o kenarın dışsallığı kadar ücret koyar: $\tau_e = f_e \cdot \ell_e'(f_e)$. Sürücünün gördüğü maliyet $\ell_e(f_e) + \tau_e$ olur; bu, kenarın marjinal maliyetidir.

Örnek 9.19

1. **Ücretleri hesapla.** Üst yolda $\ell(x) = x$, $\ell' = 1$; ücret $\tau = x \cdot 1 = x$. Alt yolda $\ell = 1$ sabittir, $\ell' = 0$; ücret yoktur.
2. **Sürücünün gördüğü.** Üstte $x + x = 2x$, altta 1.
3. **Yeni denge.** Gördükleri maliyetler eşitlenir: $2x = 1 \Rightarrow x = 1/2$. Bencil sürücüler kendiliğinden optimum dağılımı seçer. Gecikme toplamı $3/4$'tür.
4. **Toplanan ücret.** Üstteki 1/2 trafik, her biri $\tau = 1/2$ öder: $1/2 \cdot 1/2 = 1/4$.
5. **Sürücülerin bedeli.** Gecikme + ücret = $3/4 + 1/4 = 1$. Sürücüler ücretsiz dengedekiyle (1) aynı toplamı öder; gelir geri dağıtılmazsa sürücüler kazanmaz, kazanç topluma geçer.



Şekil 9.18: Marjinal maliyet ücretli Pigou ağı. Üst yolda sürücünün gördüğü maliyet $x + \text{ücret } x = 2x$ (kırmızı eğik doğru), alt yolunki 1 (kırmızı yatay doğru). Kesişim $x = 1/2$: yeni denge optimumla aynıdır.

Sabit ücret de yeter. Ücretin trafikle değişmesi gerekmez. Üst yola sabit $\tau = 1/2$ konursa sürücü üstte $x + 1/2$ görür; denge $x + 1/2 = 1 \Rightarrow x = 1/2$. Sabit ücretin doğru değeri, optimumdaki marjinal maliyet ücretinin değeridir. Bu sonuç genelleşir:

Teorem 9.7 (Marjinal maliyet ücretlendirmesi)

Bölünebilir trafikli bir ağda gecikmeler artan ve türevlenebilir, her kenarın toplam maliyeti $x \cdot \ell_e(x)$ de konveks (çukur biçimli) ise (doğrusal ve x^p gecikmeler böyledir), her kenara optimumdaki dışsallığı kadar ücret ($\tau_e = f_e^{OPT} \cdot \ell_e'(f_e^{OPT})$) konduğunda sistem optimumu bir Wardrop dengesi olur.

İspat. Optimumda trafik taşıyan bütün yolların marjinal maliyetleri eşittir ve kullanılmayan yolların marjinal maliyeti daha küçük değildir (9.2.5). Ücretler optimumdaki değerlerinde sabitlenince sürücünün optimumda bir kenarda gördüğü maliyet $\ell_e(f_e^{OPT}) + f_e^{OPT} \cdot \ell_e'(f_e^{OPT})$'dir; bu tam olarak kenarın marjinal maliyetidir. Bir yolun gördüğü maliyet kenarlarının toplamı olduğundan, sürücülerin gördüğü yol maliyetleri optimumdaki marjinal yol maliyetleriyle aynıdır. Demek ki optimumda kullanılan bütün yollar sürücünün gözünde eşit ve en ucuzdur: Wardrop koşulu sağlanır. □

Ücretin adı da Pigou'dan gelir: dışsallığı fiyatlandıran vergiye **Pigou vergisi** denir. Uygulamada zorluk ücretin hesaplanmasında değil, dışsallığın ölçülmesindedir: gecikme fonksiyonlarını ve trafiği bilmek gerekir.

9.6.3 PoA'yı azaltmanın diğer yolları

Ücret tek araç değildir. Literatürde ve uygulamada kullanılan başlıca araçlar şunlardır:

Tıkanıklık fiyatlandırması
: Kullanıcıların başkalarına yüklediği dışsal maliyeti karara dahil eder. Londra (2003), Stockholm (2006'da deneme, 2007'den beri kalıcı) ve Singapur (1975'ten beri) şehir merkezlerine girişi ücretlendirir.

Merkezî (Stackelberg) yönlendirme
: Trafiğin bir kısmını bir merkez yönlendirir, gerisi bencil seçer. Merkez önce hareket eder, bencil kullanıcılar onun seçimini görerek tepki verir; bu "önce lider, sonra izleyiciler" yapısı Stackelberg oyunlarından adını alır. Kontrol edilen pay arttıkça oran 1'e yaklaşır (Roughgarden, 2004).

Kapasite planlama
: Roughgarden ve Tardos şunu da gösterdi: r birim trafiğin bencil dengesi, $2r$ birim trafiğin en iyi yönlendirilmesinden daha pahalı değildir. Sonuç doğrusal olmayanlar dahil her artan gecikme için geçerlidir. Başka bir deyişle, her bağlantıyı iki kat hızlandırmak koordinasyon eksikliğini telafi eder; kapasite artırımı ücretin işini görebilir.

Rota önerisi ve bilgi
: Navigasyon uygulamalarının önerileri dengeyi daha düşük toplam maliyete yönlendirebilir; ama uyum gönüllüdür. Öneri bir dengeyse (kimse sapmak istemiyorsa) uyum kendiliğinden olur.

Mekanizma tasarımı
: Kuralları, iyi sonuç tek denge olacak biçimde kurmak. Örneğin A/B bağlantı oyununda B'nin kiralanmasını yasaklamak ya da A'ya katılana küçük bir indirim vermek kötü dengeyi ortadan

kaldırır.

Maliyeti hesaba katmak
: Müdahalenin bilgi, uygulama ve adalet maliyetleri, elde edilen verimlilik kazancıyla birlikte değerlendirilmelidir. %5'lik bir kaybı gidermek için kurulan ücret sistemi kendisi %5'ten pahalıysa anlamsızdır.

Örnek 9.20

Kapasite sonucu Pigou ağında sınanabilir. 1 birim trafiğin dengesi 1'dir. 2 birim trafiğin optimumu için üst yoldaki miktar x ($0 \le x \le 2$) seçilir: $C(x) = x \cdot x + (2 - x) \cdot 1$, $C'(x) = 2x - 1 = 0 \Rightarrow x = 1/2$, $C(1/2) = 1/4 + 3/2 = 7/4$. $1 \le 7/4$ ✓: bir birimlik bencil denge, iki birimin en iyi yönlendirilmesinden ucuzdur.

9.6.4 Gerçek hayat ve tartışma

Trafik ağlarında ücretli şeritler, internette ağlar arası rota seçimi, enerji şebekesinde zamana bağlı tarifeler: üçünde de soru aynıdır. Bencil denge ne kadar kötü, hangi müdahale onu optimuma yaklaştırır? İnternette her ağ işletmecisi trafiği kendi maliyetine göre yönlendirir; bu, bencil yönlendirmenin büyük ölçekli bir örneğidir. Enerji şebekesinde tüketiciler elektriği en ucuz saatte kullanır; zamana bağlı tarife, şebekenin yoğun saatlerdeki dışsallığını fiyata yansıtır.

Bu bölümün kavramları birkaç tartışma sorusuyla pekişir:

- **Bireysel çıkarlar neden her zaman sosyal optimuma ulaşmaz? Pigou örneğinde hangi dışsallık var?** Pigou'da üst yola giren sürücü oradaki herkesi $\ell' = 1$ oranında yavaşlatır; bu ek gecikme ($x \cdot \ell' = x$) onun hesabında yoktur. Optimumun kendisi de kararlı değildir: alttakiler üste geçmek ister.
- **PoA'yı azaltmak için hangi teşvikler uygulanabilir? Hangisi daha adil?** Ücret verimli ama gerileyici olabilir: aynı ücret düşük gelirli sürücüye daha ağır gelir. Bonus (alt yolu kullanana ödeme) aynı dağılımı sağlayabilir, ama bir bütçe gerektirir. Merkezî yönlendirme bazılarını uzun yola "zorlar". Adalet tartışması, verimlilik kazancının kimin cebine gittiği sorusuna dayanır.
- **Braess paradoksu günlük hayatta nerede görülebilir? Bir yolu kapatmak ne zaman mantıklıdır?** Yeni bir bağlantının herkesi aynı darboğaza yönlendirdiği her ağda görülebilir. Seul'de Cheonggyecheon üzerindeki otoyolun 2003'te başlayan yıkımından sonra kent trafiğinin beklendiği kadar kötüleşmemesi sık anılan bir örnektir. Bir yolu kapatmak, dengeyi optimuma yaklaştırıyorsa mantıklıdır; bunu anlamak için denge ve optimum birlikte hesaplanmalıdır.

9.6.5 Ücretlendirmede sık hatalar

Sık yapılan hata

- **Ücreti toplam gecikmeye eklemek:** Ücret bir transferdir; toplumun toplam gecikmesini değiştirmez. Sürücülerin "gecikme + ücret" toplamı ise ayrı bir sayıdır.
- **"Ne kadar çok o kadar iyi" sanmak:** $\tau = 10$, $\tau = 7{,}5$'ten kötüdür. Doğru ücret dışsallığa eşittir.
- **Marjinal ücretlerin farkını unutmak:** İki yollu ağda iki yolun ücretleri 15 ve 7,5'tir; tek yola konacak ücret farklarıdır (7,5).
- **Toplanan geliri ücret sanmak:** Pigou'da kişi başı ücret $\tau = x = 1/2$'dir; 1/4 ise toplanan toplam gelirdir.
- **Sabit gecikmeli yola ücret koymak:** Alt yolda $\ell' = 0$'dır; orada dışsallık yoktur ve marjinal maliyet ücreti sıfırdır.

9.7 Çözümlü problemler

Bu kesimdeki problemler bölümün bütün yöntemlerini bir arada kullanır: Pigou tipi ağlarda türevle optimum, yük dengelemede sapma denetimi, Braess ağında denge, H_n örneği, pürüzsüzlük ve kod izleme. Her çözüm ara adımlarıyla verilir.

9.7.1 Örnek problemler

Örnek 9.21

Problem. Pigou ağında üst yolun gecikmesi x^2 olsun (alt yol 1). Denge maliyeti, optimum ve PoA nedir?

Çözüm. $x^2 \le 1$ olduğundan üst yol alttan hiçbir zaman yavaş değildir; denge herkesin üstte olduğu durumdur ve maliyeti 1'dir. Optimum için $C(x) = x \cdot x^2 + (1 - x) = x^3 - x + 1$, $C'(x) = 3x^2 - 1 = 0 \Rightarrow x^{OPT} = 1/\sqrt{3} \approx 0{,}577$. $C(OPT) = 1 - 2/(3\sqrt{3}) \approx 0{,}615$. $PoA \approx 1/0{,}615 \approx 1{,}626$; Tablo 9.4'teki $p = 2$ satırıyla aynıdır.

Örnek 9.22

Problem. İşler {3, 3, 2, 2, 2}, 2 özdeş makine. Optimum en büyük yük nedir? En büyük yükü 7 olan bir Nash dengesi var mıdır?

Çözüm. Toplam 12, optimum en az 6; {3, 3} | {2, 2, 2} bunu sağlar: OPT = 6. {3, 2, 2} | {3, 2} atamasında yükler 7 ve 5'tir. M1'deki 2'lik iş M2'ye geçerse $5 + 2 = 7$ bekler; şimdiki 7'den küçük değil, kazanmaz. M1'deki 3'lük iş geçerse 8 bekler. M2'deki işler M1'e geçerse 9 ya da 10 bekler. Kimse kazanmaz: denge, oran $7/6$.

Örnek 9.23

Problem. Normalleştirilmiş Braess ağında ($s\to A$: x, $A\to t$: 1, $s\to B$: 1, $B\to t$: x, $A\to B$: 0; 1 birim trafik) PoA nedir?

Çözüm. Denge: herkes zikzakta, $1 + 0 + 1 = 2$; eski rotaya geçen de 2 bekler, kazanmaz. Optimum: zikzak kapalı, trafik yarı yarıya; her sürücü $1/2 + 1 = 3/2$. $PoA = 2 / (3/2) = 4/3$.

Örnek 9.24

Problem. H_n örneği $n = 3$ için kurulursa (ε küçük) PoS kaça yaklaşır?

Çözüm. Kendi yolları 1, 1/2, 1/3; ortak yol $1 + \varepsilon$. Aynı çözülme zinciriyle tek denge herkesin kendi yolunda olduğu durumdur: $H_3 = 1 + 1/2 + 1/3 = 11/6 \approx 1{,}83$. Optimum $1 + \varepsilon$ olduğundan $\varepsilon \to 0$ iken $PoS \to 11/6$.

Örnek 9.25

Problem. Üst yol $2x$, alt yol 1 olan Pigou ağında PoA nedir ve optimumu denge yapan sabit ücret kaçtır?

Çözüm. Denge: $2x = 1 \Rightarrow x = 1/2$; iki yolda da gecikme 1, $C = 1$. Optimum: $C(x) = 2x^2 + 1 - x$, $C' = 4x - 1 = 0 \Rightarrow x = 1/4$, $C = 1/8 + 3/4 = 7/8$. $PoA = 8/7$. Ücret: optimumda üstteki sürücü $2 \cdot 1/4 = 1/2$ bekler; $1/2 + \tau = 1 \Rightarrow \tau = 1/2$. Marjinal maliyet ücreti de aynıdır: $x \cdot \ell' = 1/4 \cdot 2 = 1/2$.

9.7.2 İzlemeler ve yeni konular

Örnek 9.26

Problem. `pigou_poa(3)` kaç döndürür?

Çözüm. $x = 4^{-1/3} \approx 0{,}63$; $x^4 \approx 0{,}1575$, $1 - x \approx 0{,}37$; $C(OPT) \approx 0{,}528$. Fonksiyon $1/0{,}528 \approx 1{,}896$ döndürür; değer $p = 2$ (1,626) ile $p = 4$ (2,151) arasındadır.

Örnek 9.27

Problem. Boyları 4, 3, 3, 2 olan dört işin hepsi başta M1'dedir. Her adımda kârlı sapan ilk iş en az yüklü makineye geçer. Dinamik hangi en büyük yükte durur? Optimum kaçtır?

Çözüm. Yükler (12, 0). 4'lük iş M2'ye geçer (*0 + 4 < 12*): (8, 4). Sonra 3'lük iş geçer (*4 + 3 = 7 < 8*): (5, 7). Artık M2'deki 4'lük iş M1'e geçse 9, 3'lük iş geçse 8 bekler (ikisi de 7'den büyük); M1'dekiler M2'ye geçse 10 ya da 9 bekler. Dinamik en büyük yük 7'de durur. Optimum {4, 2} | {3, 3} = 6 | 6; oran *7/6*.

Örnek 9.28

Problem. Örnek 9.16'daki oyunda A bağlantısı 3 TL, B bağlantısı 8 TL olsun (4 oyuncu). Dengeler hangileri, PoA ve PoS kaçtır?

Çözüm. Hepsi A: kişi başı 0,75; B'ye geçen 8 öder, denge. Hepsi B: kişi başı 2; A'ya geçen 3 öder, denge. Bölünmüş profillerde B'dekiler A'ya geçerek kazanır (örneğin 1 A, 3 B'de B'deki oyuncu 8/3 öder, A'ya geçerse 1,5). *PoA = 8/3*, *PoS = 1*.

Örnek 9.29

Problem. `poa_pos([3, 2, 2, 1], 2)` ne döndürür?

Çözüm. Toplam 8; {3, 1} | {2, 2} = 4 | 4 optimumdur. {3, 2} | {2, 1} (yükler 5 ve 3) de bir dengedir: 3'lük iş geçerse 6, 2'lik iş geçerse *3 + 2 = 5* (eşit, kazanmaz); M2'dekiler geçerse 7 ya da 6. Dengelerin en büyük yükleri 4 ve 5: fonksiyon `(1.25, 1.0)` döndürür, yani *PoA = 5/4*, *PoS = 1*.

Pürüzsüzlükle ilgili iki alıştırma, *(1, ½)*-pürüzsüz oyunun sınırı (2) ve *(y, z) = (2, 3)* için anahtar eşitsizliğin denetimi (8 ≤ 29/3), Örnek 9.8 ve Örnek 9.9'da çözülmüştür.

9.8 Özet

- **Anarşinin Bedeli** en kötü Nash dengesinin maliyetini sosyal optimumun maliyetine böler: $PoA = \max_{s \in NE} C(s) / C(OPT)$. **İstikrarın Bedeli** aynı oranı en iyi denge için yazar. Her zaman $1 \le PoS \le PoA$; tek denge varsa ikisi eşittir. Fayda problemlerinde oran ters çevrilir.
- PoA, hangi dengeye varılacağını bilmeden verilen **kötümser bir garanti**dir. Yaklaşım oranının oyun kuramındaki karşılığıdır: yaklaşım oranında hesaplama gücü, PoA'da koordinasyon eksiktir.
- **Pigou ağında** (üst yol *x*, alt yol 1) denge herkesi üste yollar ve maliyeti 1'dir; optimum trafiği yarıya böler ve maliyeti 3/4'tür: *PoA = 4/3*.

- Dengede yolların **gecikmeleri**, optimumda **marjinal maliyetleri** ($\ell + f \cdot \ell'$) eşitlenir. Aradaki $f \cdot \ell'$ terimi, bir sürücünün ötekilere yüklediği dışsallıktır.
- Gecikme x^p gibi dikleşince PoA sınırsız büyür ($p = 16$'da ≈ 4,73). **Doğrusal gecikmede** her ağda ve her talepte $PoA \le 4/3$'tür (Roughgarden–Tardos); ispat denge koşulu, kenar eşitsizliği ve toplamadan oluşur; sınır Pigou'da sıkıdır.
- **(λ, μ)-pürüzsüz** oyunlarda $PoA \le \lambda/(1-\mu)$; bölünebilir doğrusal trafik $(1, ¼)$ ile 4/3, atomik doğrusal tıkanıklık oyunları $(5/3, 1/3)$ ile 5/2 verir. Sınır karışık dengelere de taşınır (sağlam PoA).
- **Braess ağında** (4.000 araç) denge 320.000, aynı ağın optimumu 258.750 araç·dk'dır: $PoA \approx 1{,}237$. 80/65 Braess kaybıdır, PoA değildir.
- **Yük dengeleme** oyununda saf denge her zaman vardır. {2, 2, 1, 1} örneğinde $PoA = 4/3$, $PoS = 1$; m özdeş makinede sıkı değer $2 - 2/(m+1)$'dir. Farklı hızlı makinelerde ve karışık dengelerde oran m ile yavaşça büyür. PoA'nın değeri ölçülen sosyal maliyete bağlıdır.
- **Adil maliyet paylaşımında** PoA n'ye kadar çıkabilir, ama Rosenthal potansiyeli $\Phi = \Sigma\, c_e\, H(n_e)$ sayesinde $PoS \le H_n \approx \ln n$'dir; H_n örneği sınırın sıkı olduğunu gösterir.
- **Marjinal maliyet ücreti** $\tau_e = f_e \cdot \ell_e'(f_e)$ optimumu denge yapar (Pigou'da $\tau = x$ ya da sabit 1/2; iki yollu ağda Yol 1'e 7,5 dk). Ücret bir transferdir; fazla ücret yeniden kayıp yaratır. Stackelberg yönlendirme, kapasite artırımı, bilgi ve mekanizma tasarımı diğer araçlardır.

Örnek	Denge	Optimum	Oran
İki yol, 3.000 araç	75.000	71.250	PoA ≈ 1,053
Pigou (x ve 1)	1	3/4	PoA = 4/3
Doğrusal gecikme, her ağ	—	—	PoA ≤ 4/3 (sıkı)
Pigou, $\ell = x^p$	1	→ 0	PoA → ∞
Atomik tıkanıklık, doğrusal	—	—	PoA ≤ 5/2 (sıkı)
Braess, 4.000 araç	320.000	258.750	PoA ≈ 1,237
Yük dengeleme {2, 2, 1, 1}	4 (en kötü)	3	PoA = 4/3, PoS = 1
Adil paylaşım, H_n örneği	25/12	1,1	PoS ≈ 1,894 (≤ H_n)
Pigou + ücret τ = x	3/4	3/4	1

Tablo 9.14: Bölümün örnekleri bir bakışta.

9.9 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

Anarşinin Bedeli *Price of Anarchy (PoA)* · İstikrarın Bedeli *Price of Stability (PoS)* · koordinasyon oranı *coordination ratio* · en kötü denge *worst-case equilibrium* · sosyal optimum *social optimum* · sosyal maliyet *social cost* · verimlilik kaybı *efficiency loss* · yaklaşım oranı *approximation ratio* · sıkı sınır *tight bound* · Pigou ağı *Pigou network* · gecikme fonksiyonu *latency function*

bölünebilir (atomik olmayan) *nonatomic* · atomik *atomic* · Wardrop dengesi *Wardrop equilibrium* · sistem optimumu *system optimum* · bencil yönlendirme *selfish routing* · marjinal maliyet *marginal cost* · dışsallık *externality* · tıkanıklık oyunu *congestion game* · pürüzsüzlük, (λ, μ)-pürüzsüz oyun *smoothness* · sağlam PoA *robust Price of Anarchy* · Braess paradoksu *Braess's paradox* · yük dengeleme *load balancing* · en büyük yük *makespan* · ilişkili (farklı hızlı) makineler *related machines* · en iyi tepki dinamiği *best-response dynamics* · adil maliyet paylaşımı *fair cost sharing* · ağ tasarımı *network design* · harmonik sayı H_n *harmonic number* · potansiyel oyun *potential game* · Rosenthal potansiyeli *Rosenthal potential* · tıkanıklık ücreti *congestion toll* · marjinal maliyet ücreti *marginal-cost toll* · tıkanıklık fiyatlandırması *congestion pricing* · Pigou vergisi *Pigouvian tax* · Stackelberg yönlendirme *Stackelberg routing* · kapasite artırımı *capacity augmentation*

9.10 Kendini Dene

1. 1 birim trafik iki paralel yol arasında bölünür: Yol 1'in gecikmesi $\ell_1(x) = x$, Yol 2'ninki $\ell_2(x) = x + 1/2$. Wardrop dengesi, sosyal optimum ve PoA nedir?

   Cevap

   Denge: gecikmeler eşit, $x_1 = (1 - x_1) + 1/2 \Rightarrow x_1 = 3/4$, $x_2 = 1/4$; herkes 3/4 bekler, $C = 3/4$. Optimum: marjinal maliyetler eşit, $2x_1 = 2x_2 + 1/2$ ve $x_1 + x_2 = 1 \Rightarrow x_1 = 5/8$, $x_2 = 3/8$. $C = (5/8)^2 + (3/8)(3/8 + 1/2) = 25/64 + 21/64 = 23/32$. $PoA = (3/4)/(23/32) = 24/23 \approx 1{,}043$, 4/3'ün çok altında.

2. Pigou ağında (üst yol x, alt yol 1; burada x üst yoldaki trafik miktarıdır) toplam trafik $r = 1/2$ ve $r = 2$ olsun. Her iki durumda PoA nedir?

   Cevap

   $r = 1/2$: üst yolun gecikmesi en çok 1/2'dir, herkes üsttedir; $C = 1/2 \cdot 1/2 = 1/4$. Optimum $C(x) = x^2 + (1/2 - x)$, türev $2x - 1$ aralıkta hep ≤ 0 olduğundan en küçük değer $x = 1/2$'de: yine 1/4. $PoA = 1$; az trafikte dışsallık önemsizdir. $r = 2$: denge üst yolda 1 birim (gecikme 1), altta 1 birim; $C = 1 + 1 = 2$. Optimum $C(x) = x^2 + 2 - x$, $x = 1/2$, $C = 7/4$. $PoA = 8/7 \approx 1{,}143$.

3. Kenar eşitsizliği $\ell(f) \cdot f^{OPT} \le \ell(f^{OPT}) \cdot f^{OPT} + \tfrac{1}{4} \cdot \ell(f) \cdot f$ için $\ell(x) = x$ ve $f = 1$ alınsın. $f^{OPT} = 1/4$ için iki

yan nedir? Hangi f^{OPT} değerinde eşitlik olur?

Cevap

Sol $1 \cdot 1/4 = 1/4$; sağ $1/16 + 1/4 = 5/16$. $1/4 \le 5/16$ ✓; fark $(1/4 - 1/2)^2 = 1/16$. Fark her zaman $(f^{OPT} - f/2)^2$'dir; eşitlik yalnız $f^{OPT} = f/2 = 1/2$'de olur. Pigou ağı tam bu noktayı kullanır.

4. Bir oyunun $(2, 1/3)$-pürüzsüz olduğu biliniyor. PoA için hangi sınır çıkar? Aynı oyun $(1, 1)$-pürüzsüz olsaydı bu bilgi işe yarar mıydı?

Cevap

$PoA \le 2/(1 - 1/3) = 2/(2/3) = 3$. $(1, 1)$ tanımda izin verilen bir çift değildir ($\mu < 1$ gerekir). İspatın 4. adımında $(1 - \mu) \cdot C(s) \le \lambda \cdot C(OPT)$ eşitsizliği $0 \le C(OPT)$'ye döner; bu her zaman doğrudur ve hiçbir sınır vermez.

5. İşler {5, 4, 3, 2}, 2 özdeş makine, sosyal maliyet en büyük yük. Optimum nedir? {5, 4} | {3, 2} bir Nash dengesi midir? PoA ve PoS nedir?

Cevap

Toplam 14, en büyük yük en az 7; {5, 2} | {4, 3} = 7 | 7: OPT = 7. {5, 4} | {3, 2}'de yükler 9 ve 5. 5'lik iş geçerse $5 + 5 = 10$, 4'lük iş geçerse $5 + 4 = 9$ (eşit, kazanmaz) bekler; M2'dekiler geçerse 12 ya da 11. Denge ✓, en büyük yük 9. 16 atamanın taranması dengelerin en büyük yüklerinin 7, 8 ve 9 olduğunu gösterir: $PoA = 9/7 \approx 1{,}286 \le 4/3$, $PoS = 1$ (7 | 7 de denge).

6. 3 oyuncu s'den t'ye bağlantı kiralar: A 1 TL, B 2 TL; bedeli kullananlar eşit böler. Saf dengeler hangileridir? PoA ve PoS kaçtır?

Cevap

Hepsi A: kişi başı 1/3; B'ye geçen 2 öder, denge. Hepsi B: kişi başı 2/3; A'ya geçen 1 öder ($1 > 2/3$), denge. 1 A, 2 B: B'deki 1 öder, A'ya geçerse 1/2; kaçar. 2 A, 1 B: B'deki 2 öder, A'ya geçerse 1/3; kaçar. Dengeler “hepsi A” (toplam 1 = OPT) ve “hepsi B” (toplam 2): $PoA = 2$, $PoS = 1$.

7. H_n örneği iki oyuncuyla kurulur: kendi yolları 1 ve 1/2, ortak yol 1,2 (ε = 0,2). Optimum, tek denge

ve PoS nedir? $\Phi(OPT)$ ile $\Phi(denge)$ nasıl ilişkilidir?

Cevap

Optimum: ikisi ortakta, toplam 1,2 (diğer seçenekler 1,5, 1,7 ve 2,2). Optimumdan: oyuncu 2 ortakta 0,6 öder, kendi yolu 0,5 < 0,6, ayrılır; oyuncu 1 ortak yolun bedelini (1,2) tek başına öder, kendi yolu 1 < 1,2, ayrılır. Denge: herkes kendi yolunda, $C = 1{,}5$. $PoS = 1{,}5/1{,}2 = 1{,}25 \le H_2 = 1{,}5$. Potansiyel: $\Phi(OPT) = 1{,}2 \cdot H(2) = 1{,}8$, $\Phi(denge) = 1 + 1/2 = 1{,}5$; dinamik boyunca potansiyel 1,8'den 1,5'e düşmüştür.

8. 2.000 araç iki yol arasında bölünür: $\ell_1 = 10 + x_1/100$, $\ell_2 = 15 + x_2/100$ (dk). Denge ve optimum toplam gecikmesi nedir? Yol 1'e hangi sabit ücret optimumu denge yapar?

Cevap

Denge: $10 + x_1/100 = 15 + (2.000 - x_1)/100 \Rightarrow x_1 = 1.250$, herkes 22,5 dk; toplam 45.000 araç·dk. Optimum: $10 + x_1/50 = 15 + x_2/50 \Rightarrow x_1 - x_2 = 250$, $x_1 = 1.125$, $x_2 = 875$; gecikmeler 21,25 ve 23,75; toplam $1.125 \cdot 21{,}25 + 875 \cdot 23{,}75 = 44.687{,}5$. PoA ≈ 1,007. Ücret: $21{,}25 + \tau = 23{,}75 \Rightarrow \tau = 2{,}5$ dk; marjinal ücretlerin farkıyla aynıdır ($11{,}25 - 8{,}75 = 2{,}5$).

9.11 İleri okuma

- T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory* (2016), Ders 11–15: Pigou ve Braess örneklerinden pürüzsüzlüğe ve ağ tasarımında H_n sınırına kadar bu bölümün bütün konularını kısa ve açık derslerle anlatır.
- N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory* (2007), Bölüm 17–20: yük dengeleme, bencil yönlendirme, ağ oluşturma ve İstikrarın Bedeli için kapsamlı başvuru bölümleridir.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 8: trafikte denge ve Braess paradoksunu matematik bilgisi az olan okur için örneklerle anlatır.
- A. R. Karlin ve Y. Peres, *Game Theory, Alive* (2017), Bölüm 8: Anarşinin Bedeli'ni potansiyel oyunlar ve tıkanıklık oyunlarıyla birlikte öğrenci düzeyinde ispatlarla verir.

9.12 Kaynaklar

1. E. Anshelevich vd., "The Price of Stability for Network Design with Fair Cost Allocation", *IEEE FOCS*, 2004.
2. B. Awerbuch, Y. Azar, A. Epstein, "The Price of Routing Unsplittable Flow", *ACM STOC*, 2005.
3. D. Braess, "Über ein Paradoxon aus der Verkehrsplanung", *Unternehmensforschung* 12, 258–268, 1968.
4. G. Christodoulou, E. Koutsoupias, "The Price of Anarchy of Finite Congestion Games", *ACM STOC*, 2005.
5. A. Czumaj, B. Vöcking, "Tight Bounds for Worst-Case Equilibria", *ACM-SIAM SODA*, 2002.

6. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge, 2010 (Bölüm 8: trafik ve Braess).
7. G. Finn, E. Horowitz, "A Linear Time Approximation Algorithm for Multiprocessor Scheduling", *BIT* 19, 1979.
8. R. L. Graham, "Bounds for Certain Multiprocessing Anomalies", *Bell System Technical Journal* 45(9), 1966.
9. A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017 (Bölüm 8: Anarşinin Bedeli).
10. E. Koutsoupias, M. Mavronicolas, P. Spirakis, "Approximate Equilibria and Ball Fusion", *Theory of Computing Systems* 36(6), 2003.
11. E. Koutsoupias, C. Papadimitriou, "Worst-case Equilibria", *STACS*, 1999.
12. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 17–20).
13. C. Papadimitriou, "Algorithms, Games, and the Internet", *ACM STOC*, 2001.
14. A. C. Pigou, *The Economics of Welfare*, Macmillan, 1920.
15. T. Roughgarden, "The Price of Anarchy Is Independent of the Network Topology", *Journal of Computer and System Sciences*, 2003.
16. T. Roughgarden, "Stackelberg Scheduling Strategies", *SIAM Journal on Computing* 33(2), 2004.
17. T. Roughgarden, "Intrinsic Robustness of the Price of Anarchy", *ACM STOC*, 2009.
18. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press, 2016 (Ders 11–15).
19. T. Roughgarden, É. Tardos, "How Bad Is Selfish Routing?", *Journal of the ACM*, 2002.
20. J. G. Wardrop, "Some Theoretical Aspects of Road Traffic Research", *Proceedings of the Institution of Civil Engineers* 1(3), 325–362, 1952.

KISIM IV

İşbirliği ve Toplum

Bölüm 10–12

KISIM IV · BÖLÜM 10

Çoklu Ajan Sistemleri

Bir depoda yüzlerce robot sipariş toplar, bir arama-kurtarma ekibinde drone'lar bölgeleri paylaşır, bir akıllı şebekede evler enerji alıp satar. Bu sistemlerin her birinde kendi bilgisi ve kendi hedefi olan birden fazla karar verici vardır. Bu karar vericilere ajan, sistemin bütününe çoklu ajan sistemi denir. Ajanlar birbirinin kararından etkilendiği için aralarındaki etkileşim bir oyundur. Bu not, ajan kavramından başlayarak işbirliği ve koordinasyon sorunlarını, sözleşme ağıyla görev tahsisini, koalisyon oyunlarını (Shapley değeri, çekirdek, güç endeksleri) ve kamu malı oyunundaki beleşçiliği adım adım anlatır. Her konuda sayısal bir örnek ara adımlarıyla çözülür.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum10`

ÖĞRENME ÇIKTILARI

Ajan ve MAS
Ajanın temel özelliklerini ve çoklu ajan sistemlerinde işbirliği, rekabet ve koordinasyon sorunlarını açıklamak.

Görev tahsisi
Sözleşme ağı protokolünü (duyuru, teklif, ödül) küçük bir örnek üzerinde çalıştırmak ve optimumla karşılaştırmak.

Koalisyon oyunları
Karakteristik fonksiyonu yazmak; Shapley değerini tüm sıralamalardaki marjinal katkılardan hesaplamak; çekirdeği sınamak; ağırlıklı oylamada güç endeksini bulmak.

Kamu malı ve beleşçilik
Kamu malı oyununda katkı vermemenin neden baskın strateji olduğunu sayısal olarak göstermek ve çoklu ajan sistemlerindeki karşılıklarını tanımak.

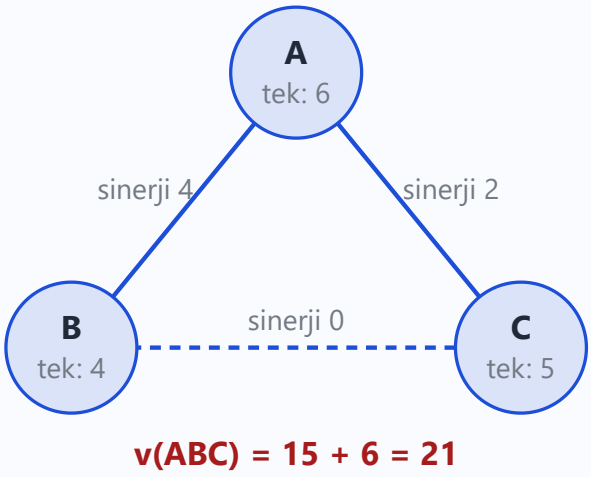
10.1 Ajanlar ve çoklu ajan sistemleri

Önceki konularda oyuncular çoğu zaman insanlardı: açık artırmada teklif veren alıcılar, oy veren seçmenler, pazar yerinde alışveriş yapan kullanıcılar. Bu bölümde oyuncuların önemli bir kısmı yazılımdır: robotlar, drone'lar, otonom araçlar, alım satım programları. Bu yazılımlara **ajan** denir. Ajanlar

insan gibi düşünmez, ama insan gibi kendi hedefini izler ve başka ajanların kararlarından etkilenir. Bu yüzden oyun kuramının araçları ajanlara da doğrudan uygulanır.

10.1.1 Bir oyunla başlangıç: Shapley koalisyon oyunu

A, B ve C adlı üç oyuncu birlikte çalışarak bir değer üretecektir. Her oyuncunun tek başına üretebileceği bir değer vardır. İki oyuncu birlikte çalışınca, tek başına değerlerinin toplamına ek olarak bir **sinerji bonusu** doğar. Oyunun kuralı şöyledir: bir koalisyonun (birlikte çalışan grubun) değeri, üyelerin tek başına değerleri ile koalisyonun içindeki her ikilinin sinerji bonusunun toplamıdır. Sonunda üç oyuncunun tamamından oluşan koalisyonun değeri, *v(N)*, oyunculara paylaştırılır.



Şekil 10.1: Shapley Koalisyon Oyunu'ndan örnek bir tur. Düğümlerde tek başına değerler (A 6, B 4, C 5), kenarlarda ikili sinerji bonusları (A–B 4, A–C 2, B–C 0) yazılıdır. Büyük koalisyonun değeri *v(ABC) = 15 + 6 = 21*'dir.

Şekil 10.1'de oyunun örnek bir turu görülür. Tek başına değerlerin toplamı *6 + 4 + 5 = 15*, sinerjilerin toplamı *4 + 2 + 0 = 6*'dır. Üçü birlikte çalışınca *v(ABC) = 15 + 6 = 21* birim üretilir. Oyunun sorduğu iki soru şunlardır:

- 21 birim eşit bölünürse herkes 7 alır. Bu paylaşımda A haksızlığa uğrar mı?
- Tek başına en değerli oyuncu her zaman en büyük payı mı almalıdır?

Eşit bölüşüm ilk bakışta adil görünür. Ancak A hem tek başına en değerli oyuncudur (6) hem de iki sinerjinin (4 ve 2) ortağıdır. B ise yalnız bir sinerjiye (4) katkı verir, C de yalnız bir sinerjiye (2). Bir paylaşımın "adil" sayılması için her oyuncunun katkısını ölçen bir kural gerekir. Bu kural Shapley değeridir ve 10.4.5'te tanımlanır. Oyunun cevabı 10.4.8'de hesaplanır: A'nın Shapley payı 9'dur, yani eşit bölüşüm A'ya katkısının 2 birim altında bir pay verir. İkinci sorunun cevabı da genel olarak "hayır"dır. Paylaşımı belirleyen şey tek başına değer değil, oyuncunun bütün olası gruplara kattığı değerin ortalamasıdır.

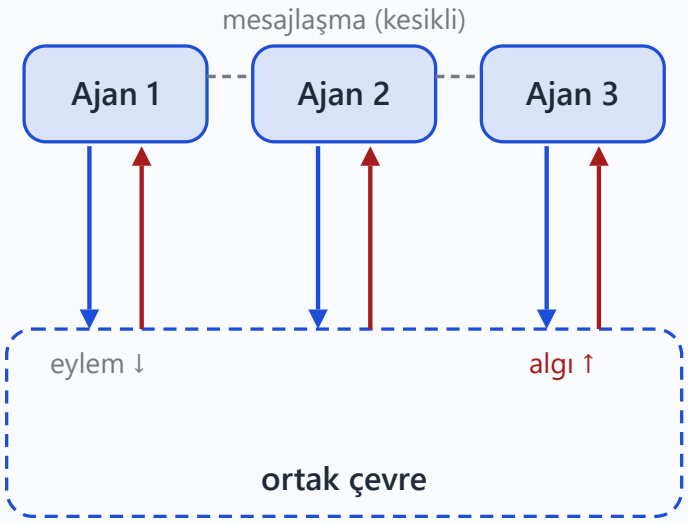
Bu oyun, bölümün iki büyük temasını birleştirir. Birincisi, ajanların birlikte çalışarak tek başlarına üretemeyecekleri bir değer üretmesidir (işbirliği). İkincisi, bu değerin ajanlar arasında nasıl bölüşüleceğidir (paylaşım). Paylaşım kötü yapılırsa bazı ajanlar gruptan ayrılmak ister ve işbirliği dağılır. Bölümün geri kalanı bu iki temanın üzerine kuruludur.

10.1.2 Çoklu ajan sistemi nedir?

Tanım 10.1

Çoklu ajan sistemi (*multi-agent system*, MAS), aynı çevrede algılayan, karar veren ve eylem yapan birden fazla otonom ajandan oluşan sistemdir. Her ajan çevreyi kendi algılayıcılarıyla gözler, kendi kararını verir ve çevreyi kendi eylemleriyle değiştirir. Ajanların ortak bir hedefi olabilir ya da her ajan kendi çıkarı için ötekilerle rekabet edebilir.

Tanımın üç parçası önemlidir. Birincisi "birden fazla"dır: tek bir ajanın sorunu bir karar sorunudur, birden fazla ajanın sorunu bir oyundur. İkincisi "otonom"dur: ajanlar kararlarını bir merkezden emir alarak değil, kendileri verir. Üçüncüsü "aynı çevre"dir: bir ajanın eylemi çevreyi değiştirir ve bu değişiklik diğer ajanların algısını ve kazancını etkiler.



Şekil 10.2: Çoklu ajan sistemi. Üç ajan ortak bir çevreye eylem uygular (mavi oklar, aşağı) ve çevreden algı alır (kırmızı oklar, yukarı). Ajanlar arasındaki kesikli çizgi mesajlaşmayı gösterir.

Şekil 10.2 bu yapıyı özetler. Her ajanın çevreyle iki yönlü bir bağı vardır: algı ve eylem. Ajanlar ayrıca birbirleriyle mesajlaşabilir. Şekildeki hiçbir ajan çevrenin tamamını görmez. **Her ajanın bilgisi yereldir.** Bir depo robotu yalnız kendi çevresindeki rafları görür. Bir drone yalnız kendi uçtuğu bölgeyi tarar. Bir otonom araç yalnız yakınındaki araçları algılar. Bu yüzden çoklu ajan sistemlerinde iki tasarım sorunu her zaman ortaya çıkar:

- **Koordinasyon:** Ajanların kararları birbiriyle uyumlu olmalıdır. İki robot aynı rafa gitmemeli, iki drone aynı bölgeyi taramamalıdır.
- **İletişim:** Ajanlar eksik bilgilerini mesajlarla tamamlar. Ancak mesajlaşmanın bir maliyeti (zaman, enerji, bant genişliği) vardır ve mesajlar gecikebilir ya da kaybolabilir.

Çoklu ajan sistemlerinin tipik örnekleri robot takımları, otonom araç filoları, dağıtık sensör ağları ve trafik ya da ekonomi simülasyonlarıdır. Bu örneklerin hepsinde ajan sayısı fazladır, bilgi dağınıktır ve tek bir merkezin her şeyi hesaplaması ya imkânsız ya da çok pahalıdır.

Oyun kuramı bağlantısı. Ajanlar birbirinin kararından etkileniyorsa aralarındaki etkileşim bir oyundur. Bir oyunun dört öğesi çoklu ajan sisteminde doğrudan karşılık bulur: oyuncular ajanlardır, stratejiler ajanların eylem seçenekleridir, kazançlar ajanların hedeflerini ölçen fayda fonksiyonlarıdır, bilgi yapısı da kimin neyi gözlediğidir. Ajanları tasarlayan kişi ise iki rol üstlenir. Ajanların içini (karar kuralını) programlar, ama ajanların karşılaştığı kuralları da (kim neyi alır, kim ne öder) belirler. İkinci rol, kuralları koyan **mekanizma tasarımcısının** rolüdür. Mekanizma tasarımı, oyunun kurallarını oyuncular istenen sonucu kendi çıkarları için seçecek biçimde kurmaktır.

10.1.3 Ajan kavramı ve dört özelliği

"Ajan" kelimesi gündelik dilde "birinin adına iş yapan" anlamına gelir. Yapay zekâda ajan, bir çevrede kendi başına hareket eden bir yazılım ya da makinedir. Wooldridge ve Jennings (1995), bir sistemin ajan sayılması için gereken en temel özellikleri dört maddede toplar. Bu tanıma **zayıf ajan** tanımı denir; "zayıf" kelimesi, tanımın ajanın bilinç ya da niyet gibi insana özgü yönlerini içermediğini, yalnız gözlenebilir davranışını tarif ettiğini belirtir.

Tanım 10.2

Bir **ajan** (*agent*) şu dört özelliği taşıyan bir bilgisayar sistemidir:

- **Otonomi** (*autonomy*): insan müdahalesi olmadan kendi kararını verir ve kendi iç durumunu yönetir.
- **Reaktivite** (*reactivity*): çevredeki değişikliği algılar ve zamanında tepki verir.
- **Proaktivite** (*pro-activeness*): yalnızca tepki vermez; hedefe ulaşmak için inisiyatif alır ve plan yapar.
- **Sosyal yetenek** (*social ability*): diğer ajanlarla iletişim kurar, pazarlık eder ve işbirliği yapar.

Dört özellik en iyi tek bir ajan üzerinde görülür. Bir arama-kurtarma drone'u düşünelim. Tablo 10.1 her özelliği bu drone'un bir davranışıyla eşler.

Özellik	Anlamı	Drone örneği
Otonomi	Kararı kendisi verir	Rotasını kendisi seçer

Özellik	Anlamı	Drone örneği
Reaktivite	Değişikliğe zamanında tepki verir	Engel görünce frenler
Proaktivite	Hedef için inisiyatif alır	Şarjı bitmeden istasyona gider
Sosyal yetenek	Diğer ajanlarla etkileşir	Bir görev için teklif verir

Tablo 10.1: Zayıf ajan tanımının dört özelliği ve bir drone'daki karşılıkları.

Reaktivite ile proaktivite arasındaki fark ince ama önemlidir. Reaktif davranış bir olayla tetiklenir: engel belirir, drone frenler. Olay olmasaydı davranış da olmazdı. Proaktif davranış ise bir hedefle tetiklenir: drone şarjının yolun geri kalanına yetmeyeceğini hesaplar ve henüz hiçbir uyarı yokken istasyona yönelir. Proaktif ajan geleceği öngörür, reaktif ajan şimdiye cevap verir. İyi bir ajan ikisini dengeler. Yalnız reaktif bir ajan uzun vadeli hedefleri kaçırır. Yalnız proaktif bir ajan ise planına o kadar bağlı kalabilir ki beklenmedik bir engele zamanında tepki veremez.

Sosyal yetenek, çoklu ajan sistemini tek ajanlı sistemden ayıran özelliktir. Sosyal yeteneği olmayan ajanlar aynı çevrede bulunsalar da birbirlerini yalnız çevre üzerinden etkiler. Sosyal yetenekli ajanlar ise mesajlaşır, teklif verir, anlaşma yapar. Görev tahsisi (10.3) ve koalisyon oluşumu (10.4) bu yeteneğe dayanır.

Oyun kuramında ajan. Oyun kuramı ajanın yalnız bir yönünü ayrıntılı modeller: kararını. Oyun kuramında ajan **rasyonel bir oyuncudur**. Tercihleri bir fayda fonksiyonuyla ifade edilir ve ajan beklenen faydasını maksimize etmeye çalışır. Burada "beklenen" kelimesi, sonucun belirsiz olduğu durumlarda her olası sonucun faydasının olasılığıyla ağırlıklandırılıp toplanmasını anlatır. Örneğin bir drone, yarı yarıya olasılıkla 10 ya da 4 birim fayda getirecek bir rotayı, kesin 6 birim getiren bir rotaya tercih eder; çünkü ilk rotanın beklenen faydası $0{,}5 \cdot 10 + 0{,}5 \cdot 4 = 7$'dir.

10.1.4 Ajan nasıl karar verir? Mimariler

Dört özellik ajanın *ne* yaptığını anlatır. Ajanın kararını *nasıl* verdiğini ise **ajan mimarisi** belirler. Mimari, algıdan eyleme giden iç yoldur. Tablo 10.2 beş yaygın mimariyi karşılaştırır.

Mimari	Karar mekanizması	Güçlü yanı	Örnek
Reaktif	Durum → eylem kuralları; iç model yok	Hızlı, sağlam	Süpürge robotu, sürü davranışı
Planlayan (*deliberative*)	Çevre modeli + planlama / arama	Uzun vadeli hedefler	Keşif robotu, lojistik planlayıcı
BDI	İnanç (*belief*), arzu (*desire*), niyet (*intention*)	İnsan benzeri akıl yürütme	Hava trafik yönetimi, sanal asistan
Öğrenen	Deneyimden politika öğrenir	Bilinmeyen çevreye uyum	Oyun oynayan ajanlar
Hibrit	Reaktif alt katman + planlayıcı üst katman	Hız ve öngörü birlikte	Otonom araç

Tablo 10.2: Ajan mimarileri.

Reaktif mimari en basit olanıdır. Ajanın içinde bir kural tablosu vardır: "önün boşsa ilerle, duvara çarptıysan rastgele dön." Ajan çevrenin bir modelini tutmaz, geleceği hesaplamaz. Bu yüzden çok hızlıdır ve algılayıcı hatalarına karşı sağlamdır. Ev süpürge robotları bu mimariye yakındır. Sürü davranışı da reaktif kuralların ürünüdür: Reynolds (1987) her kuşun yalnız üç yerel kurala (komşulara çarpma, komşularla aynı yöne uç, komşuların merkezine yaklaş) uymasıyla gerçekçi kuş sürülerinin ortaya çıktığını göstermiştir.

Planlayan mimari çevrenin bir modelini tutar ve bu model üzerinde arama yaparak bir eylem dizisi (plan) üretir. Bir keşif robotu haritayı belleğinde tutar ve hedefe en kısa yolu hesaplar. Planlama uzun vadeli hedeflere ulaşmayı sağlar, ama yavaştır. Çevre plan bitmeden değişirse plan geçersiz kalabilir.

BDI mimarisi planlamayı insan akıl yürütmesine benzeyen üç kavramla düzenler. *İnançlar* ajanın çevre hakkında doğru kabul ettiği bilgilerdir ("3. pist boş"). *Arzular* ajanın ulaşmak istediği durumlardır ("uçak indirilsin", "yakıt tasarrufu olsun"). Arzular birbiriyle çelişebilir. *Niyetler* ise ajanın arzular arasından seçtiği ve ulaşmaya karar verdiği hedeflerdir ("uçağı 3. piste indir"). Niyet, ajanın her an fikir değiştirmesini önler: ajan yeni bir bilgi önemli bir değişiklik gösterene kadar niyetine bağlı kalır.

Öğrenen mimari kuralları önceden yazmaz; hangi durumda hangi eylemin iyi sonuç verdiğini deneyimden çıkarır. Öğrenilen kural bir **politikadır**: her duruma bir eylem (ya da eylemler üzerinde bir olasılık dağılımı) atayan fonksiyon. Bu mimari bilinmeyen çevrelere uyum sağlar. Oyun oynayan ajanlar çoğunlukla böyle çalışır.

Hibrit mimari katmanları birleştirir. Bir otonom aracın alt katmanı reaktiftir: önüne aniden bir yaya çıkarsa planlayıcıyı beklemeden fren yapar. Üst katmanı planlayıcıdır: rotayı ve şerit değiştirmeyi önceden hesaplar. Böylece hız ve öngörü birlikte sağlanır.

Bir ayrımı akılda tutmak gerekir: **mimari, ajanın içini anlatır; oyun kuramı ise ajanlar arasındaki etkileşimi inceler.** Bir oyun kuramı modeli ajanın içinde reaktif kurallar mı yoksa BDI mantığı mı olduğuyla ilgilenmez. Yalnız ajanın hangi stratejileri seçebildiğine ve her sonuçtan ne kazandığına bakar. Bu bölümün geri kalanında ajanlar bu soyut düzeyde, yani stratejileri ve kazançlarıyla ele alınır.

10.1.5 Gerçek hayat uygulamaları

Çoklu ajan sistemleri bugün birçok alanda kullanılır. Her uygulama, bölümün ilerleyen kesimlerindeki bir oyun kuramı sorusuna bağlanır.

Otonom araçlar
: Kavşakta geçiş sırası, şerit değiştirme ve konvoy sürüşü için koordinasyon gerekir. İki araç aynı anda aynı şeride geçmeye çalışırsa çarpışma riski doğar. Bu bir koordinasyon oyunudur (10.2.5).

Drone takımları
: Arama-kurtarmada bölgeleri paylaşmak gerekir: hangi drone hangi bölgeyi tarayacak? Bu bir görev tahsisi problemidir (10.3).

Depo robotları

Yüzlerce robot sipariş toplar. Rotalar çarpışmasız olmalı, işler robotlara dengeli dağıtılmalıdır.

Çevrim içi pazar yerleri
: Alıcı ve satıcı ajanlar fiyat teklifi verir ve otomatik pazarlık yapar. Bu, açık artırma ve mekanizma tasarımının alanıdır.

Akıllı şebeke
: Evler, enerji depoları ve üreticiler arasında enerji alışverişi yapılır ve talep yönetilir. Herkes enerjiyi pik saatte çekerse şebeke zorlanır; bu bir teşvik sorunudur (10.2.6).

Simülasyon
: Trafik, salgın ve ekonomi modellerinde bireyler ajan olarak canlandırılır. Bireylerin basit kuralları sistem düzeyinde beklenmedik örüntüler üretebilir (10.2.7).

Bu uygulamaların hepsi aynı soruyu sorar: **kendi hedefi olan ajanlar bir araya geldiğinde sistem düzeyinde iyi bir sonuç nasıl elde edilir?** "İyi sonuç" bağlama göre değişir. Depoda toplam teslim süresinin kısa olması, şebekede toplam enerji maliyetinin düşük olması, koalisyonda kazancın adil paylaşılması iyi sonuçtur. Oyun kuramı iki soruyu ayırır. Analiz sorusu, ajanların kendi çıkarlarıyla hareket ettiğinde nereye varacağıdır (denge). Tasarım sorusu ise kuralların, varılan noktanın iyi olacağı biçimde nasıl kurulacağıdır (mekanizma).

10.1.6 Ajan kavramında sık hatalar

Sık yapılan hata

Proaktiviteyi reaktiviteyle karıştırmak. "Drone şarjı bitmeden istasyona döner" davranışı reaktivite değil, proaktivitedir. Ortada drone'u tetikleyen bir dış olay yoktur; drone ileride doğacak bir sorunu önlemek için kendi hedefi doğrultusunda harekete geçer. Reaktiviteye örnek "engel görünce frenlemek"tir.

Konvansiyonu bir ajan özelliği sanmak. "Herkes sağdan gider" gibi bir kural (konvansiyon) tek bir ajanın özelliği değil, ajanlar arasında paylaşılan bir koordinasyon kuralıdır.

Ajanı yalnız fiziksel robot sanmak. Bir alım satım programı, bir fiyat teklif botu ya da bir sanal asistan da ajandır. Ajan olmak için gövde gerekmez; dört özellik yeterlidir.

Mimariyle oyun kuramını karıştırmak. Bir ajanın BDI mimarisiyle yazılması, onun rasyonel bir oyuncu olarak modellenmesini engellemez. Mimari ajanın içini, oyun modeli ajanlar arasını anlatır.

10.2 İşbirliği ve koordinasyon

Ajanların hedefleri ne kadar örtüşür? Bu soru, bir çoklu ajan sisteminde hangi sorunun çözülmesi gerektiğini belirler. Hedefler tamamen aynıysa sorun yalnız koordinasyondur: herkes aynı şeyi ister, ama kararları uyumlu hâle getirmek gerekir. Hedefler tamamen zıtsa sorun stratejidir: her ajan raki-

binin hamlesini tahmin etmeye ve kendisini tahmin edilemez kılmaya çalışır. Gerçek sistemlerin çoğu bu iki uç arasındadır.

10.2.1 İşbirliği, rekabet ve karma sistemler

Çoklu ajan sistemleri, ajanların hedefleri arasındaki ilişkiye göre üç türe ayrılır.

İşbirlikçi MAS
: Ajanların ortak bir hedefi vardır. Bilgi, görev ve kaynak paylaşırlar. Temel sorun **koordinasyondur**: kim neyi, ne zaman yapacak? Örnek: bir arama-kurtarma drone takımı. Her drone kayıp kişiyi bulmak ister; hangi drone'un bulduğu önemli değildir.

Rekabetçi MAS
: Ajanların amaçları farklıdır. Her ajan diğerlerinin davranışını hesaba katarak kendi kazancını artırmaya çalışır. Temel sorun **stratejidir**. Örnek: bir açık artırmadaki teklifçi ajanlar. Bir teklifçinin kazandığı nesneyi diğeri kaybeder.

Karma güdülü MAS
: Bazı hedefler ortak, bazı kaynaklar rekabet konusudur (*mixed-motive*). Örnek: aynı elektrik şebekesini paylaşan evler. Hepsi şebekenin çökmemesini ister (ortak hedef), ama ucuz saatlerdeki sınırlı enerji için yarışır (rekabet). Gerçek sistemlerin çoğu bu türdendir.

Hangi yapının daha verimli olduğu sorusunun tek bir cevabı yoktur. Tablo 10.3 iki ucun hangi koşullarda verimli olduğunu özetler.

Yapı	Verimli olduğu koşul
İşbirliği	Hedefler örtüşüyor, iletişim ucuz, tek ajanın yapamayacağı iş var
Rekabet	Özel bilgi var ve fiyat/teklif ile açığa çıkarılabilir (piyasa, açık artırma)

Tablo 10.3: İşbirliği ve rekabet hangi koşulda daha verimlidir?

İşbirliği, ağır bir yükü birlikte taşıyan drone'lar gibi tek ajanın yapamayacağı işlerde vazgeçilmezdir. Rekabetin üstünlüğü ise bilgiyle ilgilidir. Her ajanın maliyetini yalnız kendisi biliyorsa merkez bu bilgiyi doğrudan göremez. Bir açık artırma ya da ihale, ajanları bu bilgiyi teklifleriyle açığa vurmaya iter: en düşük maliyetli ajan en düşük teklifi verir ve işi alır. Rekabet böylece bilgiyi toplayan bir araç olur. Mekanizma tasarımı da tam olarak bu işi yapar: rekabeti iyi sonuca yönlendiren kuralları kurar. Hangi yapının seçileceği, sistemin amacına ve kaynak yapısına bağlıdır.

10.2.2 Çıkar yelpazesi: üç küçük oyun

Üç sistem türü, üç küçük iki oyunculu oyunla temsil edilebilir. Her oyun bir **kazanç matrisiyle** verilir: satırlar 1. oyuncunun, sütunlar 2. oyuncunun seçenekleridir; her hücrede (1. oyuncunun kazancı, 2. oyuncunun kazancı) yazılıdır. Bir hücrenin **Nash dengesi** olması, iki oyuncunun da o hücrede **en iyi tepkide** olması demektir. Yani karşı tarafın seçimi sabitken hiçbir oyuncu tek başına seçimini değiştirerek kazancını artıramaz.



Şekil 10.3: Çıkar yelpazesi. Solda saf koordinasyon (ortak çıkar, $u_1 = u_2$), ortada Mahkûmun İkilemi (karma çıkar), sağda Yazı-Tura Eşleştirme (saf rekabet, $u_1 + u_2 = 0$). Turuncu hücreler saf Nash dengeleridir.

Şekil 10.3 üç oyunu, tam işbirliğinden tam rekabete uzanan bir yelpazenin üzerine yerleştirir. Her oyun bir sorun türünü temsil eder.

Solda: saf koordinasyon. İki oyuncunun kazancı her hücrede aynıdır ($u_1 = u_2$). Çıkarlar tamamen ortaktır. İkisi de Sağ'ı seçerse ikisi de 2, ikisi de Sol'u seçerse ikisi de 1 alır; farklı seçerlerse ikisi de 0 alır. Çatışma yoktur, ama bir sorun vardır: oyuncular birbirinin hangi tarafı seçeceğini bilmezse uyumsuz seçimler yapabilir. Çözüm iletişim ya da ortak bir kuraldır (konvansiyon, ör. "herkes sağdan").

Ortada: Mahkûmun İkilemi. Çıkarlar kısmen çatışır. İki oyuncu da sessiz kalırsa ikisi de −1 alır. Biri ihanet eder, diğeri sessiz kalırsa ihanet eden 0, sessiz kalan −3 alır. İkisi de ihanet ederse ikisi de −2 alır. Burada bireysel teşvik ortak iyiye ters düşer ve çözüm için teşvik tasarımı gerekir (10.2.3).

Sağda: Yazı-Tura Eşleştirme. Çıkarlar tamamen zıttır: bir oyuncunun kazancı diğerinin kaybıdır ($u_1 + u_2 = 0$, sıfır toplamlı oyun). İki oyuncu aynı yüzü seçerse 1. oyuncu, farklı yüzü seçerse 2. oyuncu kazanır. Burada önemli olan öngörülemez olmaktır. Bunun yolu **karışık stratejidir** (*mixed strategy*): her seçeneği belirli bir olasılıkla, ör. bir yazı-tura atarak seçmek.

Örnek 10.1

Üç oyunun saf Nash dengeleri, her hücrede iki oyuncunun da en iyi tepkide olup olmadığına bakılarak bulunur.

1. **Saf koordinasyon, (Sağ, Sağ) = (2, 2).** 2. oyuncu Sağ'dayken 1. oyuncu Sol'a geçerse 0 alır; 2 > 0 olduğu için geçmez. Oyun simetrik olduğu için 2. oyuncu da geçmez. Bu hücre dengedir.
2. **Saf koordinasyon, (Sol, Sol) = (1, 1).** Sağ'a geçen oyuncu 0 alır; 1 > 0 olduğu için kimse geçmez. Bu hücre de dengedir, ama ikisi için de (Sağ, Sağ)'dan kötüdür.
3. **Saf koordinasyon, (Sağ, Sol) ve (Sol, Sağ).** İkisi de 0 alır; taraf değiştiren oyuncu karşısındakiyle eşleşir ve 1 ya da 2 alır. Bu hücreler denge değildir. Sonuç: iki saf denge vardır ve sorun hangisinin oynanacağıdır.

4. **Mahkûmun İkilemi.** Karşı taraf Sessiz'se İhanet 0, Sessiz −1 verir: İhanet daha iyidir. Karşı taraf İhanet'teyse İhanet −2, Sessiz −3 verir: yine İhanet daha iyidir. İhanet, karşı taraf ne yaparsa yapsın daha iyidir. Tek denge (İhanet, İhanet) = (−2, −2)'dir.
5. **Yazı-Tura Eşleştirme.** (Yazı, Yazı)'da 2. oyuncu −1 alır ve Tura'ya geçerek 1 alır. (Yazı, Tura)'da 1. oyuncu −1 alır ve Tura'ya geçerek 1 alır. Dört hücrenin her birinde kaybeden oyuncu seçimini değiştirmek ister. Saf denge yoktur.
6. **Yazı-Tura'da karışık denge.** 2. oyuncu Yazı'yı q olasılıkla oynasın. 1. oyuncunun Yazı'dan beklenen kazancı $q \cdot 1 + (1 - q) \cdot (-1) = 2q - 1$, Tura'dan $q \cdot (-1) + (1 - q) \cdot 1 = 1 - 2q$'dir. 1. oyuncunun iki seçenek arasında kayıtsız kalması için $2q - 1 = 1 - 2q$, yani $q = 1/2$ olmalıdır. Simetriden 1. oyuncu da $p = 1/2$ oynar. Denge (½, ½)'dir ve her oyuncunun beklenen kazancı 0'dır.

Örnek 10.1 yelpazenin üç noktasında üç ayrı sorunu ortaya koyar. Koordinasyon oyununda birden çok denge vardır ve sorun seçimdir. Mahkûmun İkilemi'nde denge tektir ama verimsizdir. Yazı-Tura'da saf denge yoktur ve rastgelelik gerekir. Bir çoklu ajan sistemini tasarlayan kişi, sistemin bu yelpazenin neresinde olduğunu anlamadan doğru aracı seçemez. Koordinasyon sorunu ortak bir kuralla çözülür. Teşvik sorununda ise ortak kural yetmez; kazançları değiştiren bir düzenleme gerekir.

10.2.3 Bireysel rasyonellik ve ortak kötü sonuç

Mahkûmun İkilemi çoklu ajan sistemlerinin en önemli uyarısını taşır. Oyunda İhanet **baskın stratejidir**: karşı taraf ne yaparsa yapsın İhanet daha yüksek kazanç verir. Bu yüzden iki rasyonel oyuncu da İhanet'i seçer ve denge (−2, −2) olur. Oysa (Sessiz, Sessiz) ikisine de (−1, −1) verirdi. Her oyuncu kendi açısından en iyisini yapar, ama sonuç ikisi için de daha kötüdür. **Bireysel rasyonellik ortak kötü sonuç doğurur.**

Bu durum **Pareto optimallik** kavramıyla ifade edilir. Bir sonuç, hiç kimsenin durumunu kötüleştirmeden en az bir kişinin durumunu iyileştiren başka bir sonuç yoksa Pareto optimaldir. (İhanet, İhanet) Pareto optimal değildir: (Sessiz, Sessiz)'e geçmek iki oyuncuyu da iyileştirir. Mahkûmun İkilemi'nin dersi, Nash dengesinin Pareto optimal olmak zorunda olmadığıdır.

Çoklu ajan sistemlerinde bu yapının birçok karşılığı vardır.

Bant genişliği
: Her ajan kendi veri akışını artırınca kendi aktarımı hızlanır gibi görünür, ama ağ tıkanır ve herkes için yavaşlar.

Ortak kaynak
: Paylaşılan işlemci, enerji ya da veri havuzu aşırı kullanılır. Her ajan kendi işini öne alır, ortak kaynak tükenir.

Bilgi paylaşımı
: Bilgi paylaşmak ortak modeli iyileştirir, ama bilgiyi saklamak ajana kısa vadede avantaj sağlar.

Bu yapının çok oyunculu sürümü **kamu malı oyunudur** ve 10.5'te sayılarla incelenir. Ortak bir kaynağın aşırı kullanılarak tükenmesi ise **ortak mallar trajedisi** (*tragedy of the commons*) adını alır. İkisinde de mantık aynıdır: bireyin kazancı ile grubun kazancı ters yönde hareket eder.

10.2.4 Karar alma: merkezî mi, dağıtık mı?

Ajanlar çevresel bilgiye göre plan yapar. Kararı kim verir: tek bir merkez mi, her ajan kendisi mi? Bu, bir çoklu ajan sistemi tasarlanırken verilen ilk büyük karardır. Tablo 10.4 iki seçeneği karşılaştırır.

	Merkezî kontrol	**Dağıtık (tam otonomi)**
Avantaj	Tüm bilgi tek noktada: optimum hesaplanabilir; kolay koordinasyon, düşük çatışma riski	Esnek, ölçeklenebilir, yerel olaylara hızlı tepki; bir parça bozulunca sistem çalışmaya devam eder
Dezavantaj	Tek nokta hatası ve darboğaz; ajan sayısı artınca hesaplama patlar; ajanlar özel bilgisini dürüst bildirmeyebilir	Yerel bilgiyle alınan kararlar uyumsuz olabilir; bencil ajanlar verimsiz dengeye gidebilir (Anarşinin Bedeli)
Örnek	Trafik ışığı yönetim merkezi; sensör ağında merkezî sunucu	Drone takımları, otonom araçlar, robotik keşif

Tablo 10.4: Merkezî kontrol ile dağıtık kontrolün karşılaştırması.

Merkezî kontrolün en büyük gücü, bütün bilgiyi elinde tutan bir merkezin optimumu hesaplayabilmesidir. Ancak üç zayıflığı vardır. Birincisi **tek nokta hatasıdır**: merkez bozulursa bütün sistem durur. İkincisi **ölçeklenme** sorunudur: ajan sayısı arttıkça merkeze gelen mesajlar ve merkezin çözmesi gereken problem büyür. Örneğin n ajanı n göreve atamanın bütün yollarını denemek n! atamayı tek tek hesaplamak demektir (10.3.5); bu sayı n = 10'da 3.628.800'e ulaşır. Üçüncüsü **dürüstlük** sorunudur: merkez, maliyet gibi özel bilgiyi ajanların bildirmesine muhtaçtır. Bencil bir ajan maliyetini yüksek bildirerek daha kolay bir iş almaya çalışabilir.

Dağıtık kontrol bu zayıflıkları giderir: bir ajan bozulsa da diğerleri çalışır, yeni ajan eklemek kolaydır ve her ajan yerel olaylara anında tepki verir. Ama kendi zayıflıkları vardır. Yerel bilgiyle alınan kararlar uyumsuz olabilir (10.2.5). Bencil ajanlar da Mahkûmun İkilemi'ndeki gibi verimsiz bir dengeye varabilir. Bencil davranışın sistem düzeyindeki maliyeti, en kötü dengenin maliyetinin optimum maliyete oranıyla ölçülür. Bu orana **Anarşinin Bedeli** (*Price of Anarchy*) denir.

Çoğu gerçek çoklu ajan sistemi **hibrittir**: ajanlar otonom karar verir, ama bir merkez kuralları, fiyatları ya da ihaleleri belirler. Merkez her ajanın ne yapacağına karar vermez; yalnız oyunun kurallarını koyar. 10.3.2'de anlatılan sözleşme ağı buna bir örnektir: bir yönetici görevleri ihaleye çıkarır, ama maliyetleri bilmez ve kimin işi yapacağını tekliflere bırakır.

10.2.5 Dağıtık koordinasyon ve konvansiyonlar

Dağıtık koordinasyonda ajanlar merkez olmadan, yerel bilgi ve mesajlaşma ile karar verir. Bunu sağlayan bir protokol üç şeyi birlikte ele almalıdır. **İletişim gecikmesi**: mesaj ulaşana kadar durum değişmiş olabilir. **Hata toleransı**: bir mesaj kaybolursa ya da bir ajan yanıt vermezse sistem kilitlenmemelidir. **Ölçeklenebilirlik**: ajan sayısı arttıkça mesaj sayısı kontrolden çıkmamalıdır.

Dağıtık sistemin en tipik sorunu ortak bilgi eksikliğidir. İki robot dar bir koridorda karşılaşsın. Her biri karşıdakine yol vermek için bir tarafa kaçacaktır. İkisi de kendi sağına kaçarsa yollar ayrılır ve geçerler. Biri sağına, öteki soluna kaçarsa aynı tarafa yönelirler ve yine burun buruna gelirler. Bu durum Tablo 10.5'te bir oyun olarak yazılmıştır.

R1 ╲ R2	Sağ	Sol
Sağ	**(1, 1)**	(−1, −1)
Sol	(−1, −1)	**(1, 1)**

Tablo 10.5: Koridor oyunu. İki saf Nash dengesi (Sağ, Sağ) ve (Sol, Sol)'dur.

(Sağ, Sağ)'da tek başına Sol'a geçen robot −1'e düşer; bu yüzden kimse sapmaz. (Sol, Sol) için de aynısı geçerlidir. Diğer iki hücrede ikisi de −1 alır ve taraf değiştiren robot 1'e çıkar; bu hücreler denge değildir. İki denge de eşit iyidir. Sorun hangisinin daha iyi olduğu değil, **hangisinin oynanacağıdır**.

Tanım 10.3

Konvansiyon ya da **sosyal yasa** (*convention / social law*), bir koordinasyon oyununda dengelerden birini önceden seçen ve tüm ajanların bildiği ortak kuraldır. “Herkes sağdan gider” kuralı koridor oyununda (Sağ, Sağ) dengesini seçer.

Konvansiyonun gücü, kendini zorlamasıdır. Herkes sağdan gidiyorsa sağdan gitmek her ajanın en iyi tepkisidir; kuralı çiğneyen ajan yalnız kendine zarar verir. Bu yüzden konvansiyonu denetleyecek bir polise gerek yoktur. Bu gücün kaynağı, konvansiyonun bir dengeyi seçmesidir; denge olmayan bir sonucu seçen kural kendini zorlamaz.

Örnek 10.2

Koridor oyununda üç durumun beklenen kazancı karşılaştırılır: konvansiyon yok (karışık strateji), konvansiyon var, ortak sinyal var.

1. **Konvansiyon yok.** Robotlar birbirlerinin hangi tarafı seçeceğini bilmez. Simetrik tek makul davranış, her tarafı ½ olasılıkla seçmektir. R2 Sağ'ı q olasılıkla seçerse R1'in Sağ'dan beklenen kazancı $q - (1 - q) = 2q - 1$, Sol'dan $-q + (1 - q) = 1 - 2q$'dir. Kayıtsızlık $q = 1/2$'yi verir. Bu bir karışık Nash dengesidir.
2. **Karışık dengede kazanç.** Robotlar ½ olasılıkla aynı tarafı seçer (1 kazanır) ve ½ olasılıkla farklı tarafı seçer (−1 kazanır). Beklenen kazanç $½ \cdot 1 + ½ \cdot (-1) = 0$'dır. Karşılaşmaların yarısı çarpışmayla biter.
3. **Konvansiyon var.** “Herkes sağdan” kuralıyla iki robot da Sağ'ı seçer ve her biri kesin olarak 1 alır. Kazanç 0'dan 1'e çıkar.

4. **Ortak sinyal var.** Bir trafik ışığı gibi ortak bir sinyal, yarı yarıya olasılıkla "ikiniz de sağ" ya da "ikiniz de sol" desin. Sinyal "sağ" dediğinde karşıdakinin de sağa gideceğini bilen robot için en iyi tepki sağdır; "sol" için de aynısı geçerlidir. Kimse sinyale uymamak istemez ve her robot yine 1 alır.

Örnek 10.2'nin son adımı önemli bir kavramı gösterir. Trafik ışığı herkese ortak bir sinyal verir ve iki dengeden hangisinin oynanacağını seçer. Sinyale uymanın her oyuncu için en iyi tepki olduğu bu tür dengelere **korelasyonlu denge** (*correlated equilibrium*) denir. Korelasyonlu denge, oyunculara dışarıdan bir "tavsiye" verildiğinde ortaya çıkar ve tavsiyeye uymak her oyuncunun kendi çıkarınadır. Bir kavşaktaki trafik ışığı bu dengenin gündelik bir örneğidir.

10.2.6 Oyun kuramı ve teşvikler

Oyun kuramı çoklu ajan sistemlerinde iki ayrı iş görür.

Analiz: Nash dengesi
: Ajanların stratejik kararları dengede nereye varır? Denge verimli mi? Bu, sistemin kuralları verildiğinde sonucu öngörmektir.

Tasarım: mekanizma
: Kurallar, ödemeler ve açık artırmalar ajanları dürüst bildirime ve işbirliğine nasıl yönlendirir? Bu, istenen sonucu verecek kuralları seçmektir.

Tasarımın gücü iki örnekte görülür.

Enerji paylaşımı. Bir akıllı şebekede evlerin bir kısmı güneş panelleriyle enerji üretir. Teşvik dengeli kurulursa, ör. şebekeye verilen fazla üretime adil bir ödeme yapılırsa, evler fazla enerjiyi şebekeye verir ve herkes yeterli enerji alır. Teşvik yanlış kurulursa, ör. her saatte aynı fiyat uygulanırsa, herkes enerjiyi kendisine en uygun saatte, yani akşam pik saatinde çeker ve şebeke zorlanır.

Dağıtık ağ kaynakları. Bant genişliği ve hesaplama gücü paylaşılan kaynaklardır. Kullanım ücreti ya da kota olmazsa her ajan kaynağı kendi işi için sonuna kadar kullanır ve aşırı kullanım ortaya çıkar. Ücret ya da kota, her ajanın kendi kullanımının başkalarına yüklediği maliyeti hissetmesini sağlar.

İki örnekten çıkan sonuç şudur. **Doğru teşvik** işbirliğini artırır ve çatışmayı azaltır. **Yanlış teşvik** bencil davranış, aşırı risk ve beleşçilik doğurur. Teşvik tasarımı, ajanların bencil olmamasını beklemek yerine, bencil davranışın sistem için de iyi olacağı kuralları kurmaktır.

10.2.7 İletişim, öğrenme ve ortaya çıkan davranış

Oyun kuramının temel modellerinde oyuncular oyunu bilir ve dengeyi bir anda oynar. Çoklu ajan sistemlerinde ise üç gerçek durum bu basit resmi karmaşıklaştırır.

İletişim ve bilgi. Ajanların bilgisi farklıdır. Bir robot bir rafın boş olduğunu bilir, diğeri bilmez. Kimin neyi bildiği, yani **bilgi yapısı**, dengeyi değiştirir. Oyuncuların birbirinin kazancını ya da tipini tam bilmediği oyunlara eksik bilgili oyunlar denir. Ajanların bilgi alışverişi yapabilmesi için ortak bir dil ve bir mesajlaşma protokolü gerekir. Sözleşme ağı (10.3.2) böyle bir protokoldür.

Öğrenme ve uyum. Ajanlar dengeyi önceden hesaplamak yerine, geri bildirimle stratejilerini adım adım güncelleyebilir. Üç yaygın yöntem vardır. *En iyi tepki dinamiğinde* her ajan, diğerlerinin son hamlesine en iyi tepkiyi verir. *Kurgusal oyunda* (*fictitious play*) her ajan, rakibinin geçmiş hamlelerinin ortalamasına en iyi tepkiyi verir. *Pekiştirmeli öğrenmede* ajan, iyi sonuç veren eylemleri daha sık seçer. Ancak bir sorun vardır: diğer ajanlar da öğrendiği için her ajanın çevresi sürekli değişir. Böyle bir çevreye **durağan olmayan** (*non-stationary*) çevre denir. Durağan olmayan bir çevrede öğrenmenin bir dengeye yakınsaması garanti değildir.

Örnek 10.3

Koridor oyununda (Tablo 10.5) iki robot (Sağ, Sol) hücresinden başlasın; ikisinin de kazancı −1'dir. İki farklı güncelleme kuralı karşılaştırılır.

1. **Eşzamanlı en iyi tepki, 1. tur.** R1, R2'nin son seçimi Sol'a en iyi tepkiyi verir: Sol. R2 de aynı anda R1'in son seçimi Sağ'a en iyi tepkiyi verir: Sağ. Yeni hücre (Sol, Sağ)'dır ve ikisi yine −1 alır.
2. **Eşzamanlı en iyi tepki, 2. tur.** R1, Sağ'a en iyi tepki olarak Sağ'ı; R2, Sol'a en iyi tepki olarak Sol'u seçer. Hücre yeniden (Sağ, Sol) olur.
3. **Sonuç.** İki robot sonsuza dek (Sağ, Sol) ile (Sol, Sağ) arasında gidip gelir. Her ikisi de karşı tarafa uyum sağlamaya çalışır, ama karşı taraf da aynı anda değiştiği için uyum hiç olmaz. Dinamik yakınsamaz.
4. **Sıralı en iyi tepki.** Bu kez yalnız R1 güncellesin. R1, R2'nin Sol'una en iyi tepki olarak Sol'u seçer. Hücre (Sol, Sol) olur. Sıra R2'ye geldiğinde R2 zaten en iyi tepkidedir ve değişmez. Dinamik tek adımda bir dengeye varmıştır.

Örnek 10.3 durağan olmayan çevrenin sonucunu somut gösterir. Her robot için çevre, öteki robotun davranışıdır; öteki robot da öğrendiği için bu çevre her turda değişir. Güncellemeleri sıraya koymak (ya da robotlara küçük bir rastgele bekleme eklemek) bu döngüyü kırar. Ağ protokollerinde çarpışan mesajların rastgele bir süre bekleyip yeniden gönderilmesi aynı fikre dayanır.

Ortaya çıkan davranış. Basit yerel kurallar sistem düzeyinde beklenmedik örüntüler doğurabilir. Buna **ortaya çıkan davranış** (*emergent behavior*) denir. En ünlü örnek Schelling'in (1971) ayrışma modelidir. Bir ızgaraya iki gruptan bireyler yerleştirilir. Her birey, komşuları arasında kendi grubundan olanların oranı küçük bir eşiğin altına düşerse (ör. üçte birin altına) boş bir yere taşınır. Hiçbir birey ayrışma istemez; herkes karışık bir mahallede yaşamaya razıdır. Yine de taşınmalar zincirleme ilerler ve sonunda belirgin biçimde ayrışmış mahalleler oluşur. Hafif bir komşu tercihi, kimsenin istemediği güçlü bir ayrışmayı üretir.

Bu örnek bir yöntem dersi taşır: **sistemin davranışı tek tek ajanlara bakarak okunamaz.** Bu yüzden çoklu ajan sistemlerinde denge analizi ile simülasyon birlikte kullanılır. Denge analizi sistemin varabileceği durağan noktaları verir; simülasyon da sistemin bu noktalara nasıl gittiğini ya da gidip gitmediğini gösterir. Trafik sıkışıklığı ve fiyat dalgalanmaları da ajanlar arası etkileşimden doğan bu tür örüntülerdir.

10.2.8 Koordinasyonda sık hatalar

Sık yapılan hata

Çok dengeli oyunda tek dengeyi aramak. Koridor oyununda iki saf denge vardır: (Sağ, Sağ) ve (Sol, Sol). Ayrıca ½–½ karışık bir denge de vardır. “Kaç saf denge var?” sorusunun cevabı 2'dir; karışık denge bu sayıya eklenmez.

Koordinasyon sorununu teşvik sorunu sanmak. Koridor oyununda çıkar çatışması yoktur; iki robot da çarpışmamak ister. Burada ödeme ya da ceza gerekmez, ortak bir kural yeter. Mahkûmun İkilemi'nde ise ortak bir kural (“ikimiz de sessiz kalalım”) kendini zorlamaz, çünkü kural bir denge seçmez.

Ortak iyi olan sonucu denge sanmak. (Sessiz, Sessiz) iki oyuncu için de (İhanet, İhanet)'ten iyidir, ama denge değildir: karşı taraf sessizken İhanet'e geçen oyuncu −1 yerine 0 alır.

Merkezî kontrolü her zaman üstün sanmak. Merkez bütün bilgiyi bilirse optimumu hesaplayabilir, ama bu bilgiyi ajanlardan alması gerekir; ajanlar bilgilerini dürüst bildirmeyebilir. Ayrıca merkez tek nokta hatasıdır.

10.3 Görev tahsisi

Bir çoklu ajan sisteminin en sık karşılaştığı somut problem, işleri ajanlara dağıtmaktır. Hangi drone hangi bölgeyi tarayacak, hangi robot hangi siparişi toplayacak, hangi sunucu hangi hesaplamayı yapacak? En eski dağıtık yöntemlerden biri olan sözleşme ağı protokolü, küçük bir örnekte merkezî optimumla karşılaştırılabilir.

10.3.1 Görev ve kaynak tahsisi

Tanım 10.4

Görev tahsisi (*task allocation*), yetenek ve maliyetleri dikkate alarak işleri uygun ajanlara dağıtma problemidir. Amaç çoğu zaman bir toplam ölçütü en iyilemektir, ör. toplam süreyi en aza indirmek. Dağıtılan şey iş değil de işlemci, bant genişliği ya da enerji gibi bir kaynaksa probleme **kaynak tahsisi** denir.

Problem kâğıt üzerinde basit görünür: her ajanın her işi ne kadar sürede yapacağı bir tabloya yazılır ve toplamı en küçük atama aranır. Gerçekte üç zorluk vardır.

- **Özel bilgi.** Maliyetleri çoğu zaman yalnız ajanın kendisi bilir. Bir robotun bir siparişi kaç dakikada toplayacağı, robotun konumuna, şarjına ve yüküne bağlıdır. Merkez bu tabloyu göremez.
- **Ölçek.** Ajan sayısı büyüktür. Yüzlerce robot ve binlerce sipariş olabilir.
- **Zaman.** Görevler zamanla gelir. Bütün görevler baştan bilinmez; yeni sipariş geldikçe karar vermek gerekir.

İyi bir yöntem, çözüm kalitesini iletişim ve hesaplama maliyetiyle dengeler. Mükemmel bir atama bulmak için saatlerce hesap yapmak ya da binlerce mesaj göndermek, biraz daha kötü bir atamayı anında bulmaktan pahalı olabilir. Tablo 10.6 dört yaygın yöntemi karşılaştırır.

Yöntem	Nasıl çalışır?	Artı / eksi
Merkezî optimizasyon	Tüm maliyetler toplanır, atama problemi çözülür (Macar algoritması)	Optimum / tüm bilgi gerekir
Sözleşme ağı	Görev duyurulur, ajanlar teklif verir, en iyi teklif kazanır	Basit, dağıtık / açgözlü olabilir
Açık artırma	Görevler ya da kaynaklar teklifle satılır; kombinatoryal açık artırmada paketler	Teşvik uyumlu tasarlanabilir
Pazar tabanlı	Fiyatlar arz–talebe göre ayarlanır, ajanlar fiyata tepki verir	Ölçeklenir / yakınsama yavaş olabilir

Tablo 10.6: Görev tahsisi yöntemleri.

Merkezî optimizasyon tüm maliyetleri bir merkezde toplar ve **atama problemini** çözer: n görevi n ajana, her ajana tam olarak bir görev düşecek ve toplam maliyet en küçük olacak biçimde atamak. Bu problemin klasik çözümü **Macar algoritmasıdır** (*Hungarian algorithm*). Macar algoritması yaklaşık n^3 adımda optimumu bulur; yani n = 10 için yaklaşık bin adım yeter. Yöntemin eksiği hesap değil, bilgidir: merkezin bütün maliyet tablosunu bilmesi gerekir.

Sözleşme ağı bilgiyi merkezde toplamaz; görevleri tek tek ihaleye çıkarır. **Açık artırma** yöntemleri bu fikri genelleştirir. **Kombinatoryal açık artırmada** (*combinatorial auction*) ajanlar tek tek görevlere değil, görev paketlerine teklif verir: “G1 ile G2'yi birlikte 5 dakikada yaparım.” Paket teklifleri, görevler arasındaki tamamlayıcılığı (iki yakın siparişi birlikte toplamak gibi) yakalar. **Pazar tabanlı** yöntemlerde her görevin ya da kaynağın bir fiyatı vardır. Fiyat, talep fazlaysa artar, azsa düşer; ajanlar da fiyata bakarak hangi görevi alacağına karar verir. Bu yöntem çok büyük sistemlere ölçeklenir, ama fiyatların dengeye oturması zaman alabilir.

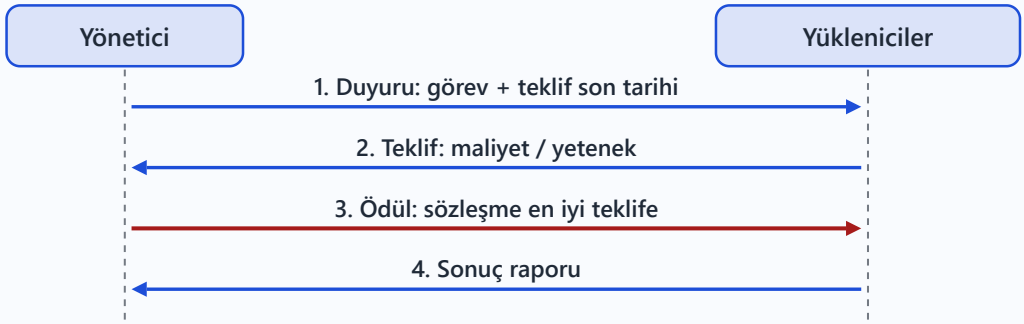
10.3.2 Sözleşme ağı protokolü

Sözleşme ağı protokolünü (*Contract Net Protocol*) Smith (1980) dağıtık problem çözme için önerdi. Protokol, inşaat sektöründeki ihale sürecini taklit eder: bir iş sahibi işi duyurur, müteahhitler teklif verir, iş en iyi teklife verilir. Ajan standartlarını belirleyen FIPA (*Foundation for Intelligent Physical Agents*) da protokolü 2002'de bir standart olarak yayımladı (FIPA, 2002).

Tanım 10.5

Sözleşme ağı protokolünde görevi olan ajan **yönetici** (*manager*), işi yapabilecek ajanlar **yüklenici** (*contractor*) rolündedir. Her görev için dört adım izlenir:

1. **Duyuru** (*task announcement*): yönetici görevi ve teklif son tarihini yüklenicilere yayınlar.
2. **Teklif** (*bid*): her uygun yüklenici maliyetini ya da yeteneğini bildiren bir teklif gönderir.
3. **Ödül** (*award*): yönetici teklifleri karşılaştırır ve sözleşmeyi en iyi teklife verir.
4. **Sonuç raporu**: yüklenici işi bitirince sonucu yöneticiye bildirir.



Şekil 10.4: Sözleşme ağı protokolünün mesaj sırası. Yönetici ile yükleniciler arasında dört mesaj gider gelir: duyuru, teklif, ödül (kırmızı) ve sonuç raporu.

Şekil 10.4 mesajların sırasını gösterir. Sıra değiştirilemez: duyurulmamış bir göreve teklif verilemez, teklifler toplanmadan ödül verilemez. Roller sabit değildir; **görev başına değişebilir**. Bir görevi yüklenici olarak alan robot, bu görevin bir alt parçasını başka robotlara ihaleye çıkararak o alt görev için yönetici olabilir. Böylece büyük bir iş, iç içe ihalelerle parçalara bölünür.

Protokolün en önemli özelliği bilgiyle ilgilidir. **Yönetici maliyetleri bilmez; teklifler bu özel bilgiyi açığa çıkarır.** Yönetici bütün maliyet tablosunu öğrenmek zorunda değildir; her görev için yalnız o görevin tekliflerini görür. Teklif verme, bir **ters açık artırmadır**: alışılmış açık artırmada en yüksek teklif kazanır, burada ise iş en düşük maliyeti bildirene gider. Bu yapı, teklifçilerin dürüst olup olmayacağı sorusunu doğurur; bu soru 10.3.6'da ele alınır.

10.3.3 Sözleşme ağı adım adım

Protokolü küçük bir örnekte çalıştıralım. Bir yönetici üç görevi (G1, G2, G3) sırayla ihaleye çıkaracaktır. Üç robot (R1, R2, R3) vardır. Her robot en çok bir görev alabilir. Her robotun her görev için tahmini süresi (dakika) Şekil 10.5'te verilmiştir. Bu süreler robotların maliyetidir ve robotlar bunları dürüstçe teklif olarak bildirir. Önemli bir ayrıntı: **tabloyu yalnız robotlar bilir, yönetici tabloyu görmez.** Her robot yalnız kendi sütununu bilir.



maliyet (dk) = teklif

	R1	R2	R3
G1	2	3	6
G2	3	7	8
G3	5	4	9

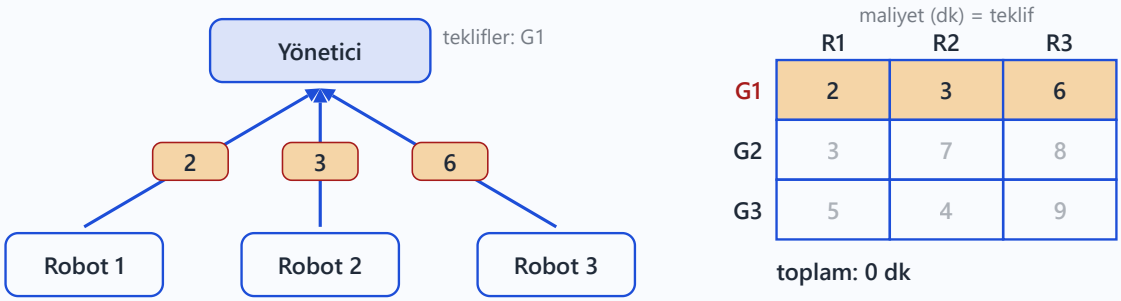
Şekil 10.5: Başlangıç durumu. Solda yönetici ve üç robot, sağda maliyet tablosu (dakika). Satırlar görevler, sütunlar robotlardır: ör. R1, G1'i 2 dakikada, G2'yi 3 dakikada, G3'ü 5 dakikada yapar. Tablo soluk çizilmiştir, çünkü yönetici onu görmez.

Tablonun kritik noktası şudur: R1 hem G1 için (2) hem G2 için (3) en ucuz robottur. R1 yalnız bir görev alabileceğine göre, hangi göreve gideceği toplam süreyi belirleyecektir.

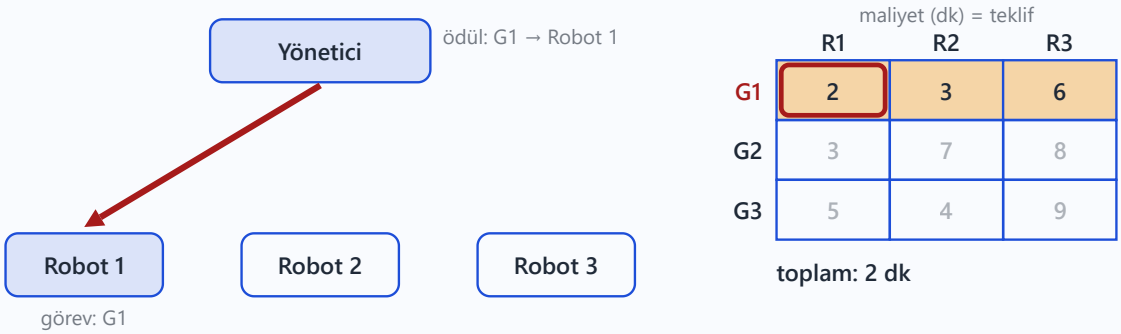
Örnek 10.4

1. **G1 duyurusu.** Yönetici G1'i boştaki bütün robotlara yayınlar: R1, R2, R3.
2. **G1 teklifleri.** Her robot G1 için kendi maliyetini bildirir: R1 → 2, R2 → 3, R3 → 6.
3. **G1 ödülü.** En düşük teklif R1'indir (2 dk). G1, R1'e verilir. R1 artık meşguldür. Toplam süre 2 dk.
4. **G2 duyurusu.** Yönetici G2'yi boştaki robotlara yayınlar: R2, R3. R1 meşgul olduğu için duyuruyu almaz.
5. **G2 teklifleri.** R2 → 7, R3 → 8. G2 için en ucuz robot R1 (3 dk) olduğu hâlde R1 teklif veremez.
6. **G2 ödülü.** En düşük teklif R2'nindir (7 dk). G2, R2'ye verilir. Toplam süre *2 + 7 = 9* dk.
7. **G3 duyurusu.** Boşta yalnız R3 kalmıştır.
8. **G3 teklifi.** R3 → 9. Tek teklif vardır.
9. **G3 ödülü.** G3, R3'e verilir; tek teklif kaldığı için seçim zorunludur. Toplam süre *9 + 9 = 18* dk.

Sözleşme ağı G1→R1, G2→R2, G3→R3 atamasını bulur ve toplam süre 18 dakikadır.



Şekil 10.6: G1 için teklifler. Üç robot da teklif verir (turuncu kutular 2, 3, 6); tabloda G1 satırı vurgulanmıştır.



Şekil 10.7: G1 ödülü. Kalın kırmızı ok sözleşmenin R1'e verildiğini, tablodaki kırmızı çerçeve kazanan teklifi gösterir. Toplam 2 dk.

Şekil 10.6 ve Şekil 10.7 ilk ihalenin teklif ve ödül adımlarını gösterir. Sonraki iki ihale aynı biçimde ilerler; tek fark, her ödülden sonra boştaki robot sayısının bir azalmasıdır. Tablo 10.7 üç ihaleyi bir arada özetler.

Duyuru	Boştaki robotlar	Teklifler	Ödül	Toplam
G1	R1, R2, R3	R1 2, R2 3, R3 6	R1 (2)	2
G2	R2, R3	R2 7, R3 8	R2 (7)	9
G3	R3	R3 9	R3 (9)	18

Tablo 10.7: Sözleşme ağının üç ihalesi.



Şekil 10.8: Sonuç. Kırmızı çerçeveler sözleşme ağının atamasını (toplam 18 dk), kesikli mavi çerçeveler merkezî optimumu gösterir: G1→R3, G2→R1, G3→R2, toplam 13 dk.

Şekil 10.8 sonucu optimumla karşılaştırır. Tüm atamaları deneyen bir merkez G1'i R3'e (6), G2'yi R1'e (3), G3'ü R2'ye (4) verir ve toplam *6 + 3 + 4 = 13* dakika elde eder. Sözleşme ağının 18 dakikası optimumdan 5 dakika fazladır. Oran olarak *18/13 ≈ 1,385*, yani sözleşme ağı yaklaşık **%38 daha pahalıdır**.

Fark nereden doğar? Sözleşme ağı her adımda o anki en iyi kararı verir: G1'i en ucuz robota, R1'e verir. Bu karar G1 için doğrudur, ama R1'i G2 için harcar. G2'nin en ucuz robotu R1 olduğu hâlde G2, 7 dakikalık R2'ye gider. Optimum ise G1'de biraz pahalı bir seçim yapar (R3, 6 dk), ama R1'i G2'ye saklar. Her adımda yerel olarak en iyi görünen seçimi yapan yöntemlere **açgözlü** (*greedy*) yöntem denir. Açgözlü yöntemler hızlıdır, ama geleceği hesaba katmadıkları için toplam optimumu kaçırabilir.

10.3.4 Kod izleme: sozlesme_agi()

Sözleşme ağının sıralı ihale mantığı birkaç satırlık bir Python fonksiyonuyla yazılabilir. Fonksiyon iki girdi alır. `gorevler`, ihaleye çıkış sırasıyla görevlerin listesidir: `["G1", "G2", "G3"]`. `maliyet` ise iç içe bir sözlüktür: `maliyet[r][g]`, r robotunun g görevi için maliyetidir. Örnekte `maliyet["R1"] = {"G1": 2, "G2": 3, "G3": 5}`'tir.

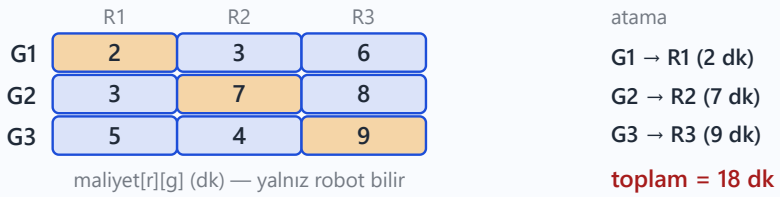
```
def sozlesme_agi(gorevler, maliyet):
  bos = set(maliyet)
  atama, toplam = {}, 0
  for g in gorevler:
    teklif = {r: maliyet[r][g] for r in bos}
    kazanan = min(teklif, key=teklif.get)
    atama[g] = kazanan
    toplam += teklif[kazanan]
    bos.remove(kazanan)
  return atama, toplam
```

Kod satır satır şöyle çalışır.

1. `bos = set(maliyet)` : Bir sözlükten küme oluşturulunca sözlüğün anahtarları alınır. `bos` başlangıçta bütün robotları içerir: {R1, R2, R3}.
2. `atama, toplam = {}, 0` : Atamalar boş bir sözlükte tutulur, toplam süre 0'dan başlar.
3. `for g in gorevler` : Görevler sırayla ihaleye çıkar. Her tur bir duyurudur.
4. `teklif = {r: maliyet[r][g] for r in bos}` : Yalnız boştaki robotlar teklif verir. Her robot g görevi için kendi maliyetini bildirir. Bu satır, protokolün teklif adımıdır.
5. `kazanan = min(teklif, key=teklif.get)` : `min`, sözlüğün anahtarları (robotlar) arasından `teklif.get` değeri en küçük olanı seçer. Bu satır ödül adımıdır.
6. `atama[g] = kazanan` ve `toplam += teklif[kazanan]` : Görev kazanana yazılır, kazanan teklif toplama eklenir.
7. `bos.remove(kazanan)` : Kazanan robot meşgul olur ve sonraki ihalelere katılmaz.

g	bos (tur başı)	teklif	kazanan	atama	toplam
—	{R1, R2, R3}	—	—	{}	0
G1	{R1, R2, R3}	{R1: 2, R2: 3, R3: 6}	R1	{G1: R1}	2
G2	{R2, R3}	{R2: 7, R3: 8}	R2	{G1: R1, G2: R2}	9
G3	{R3}	{R3: 9}	R3	{G1: R1, G2: R2, G3: R3}	18

Tablo 10.8: `sozlesme_agi(["G1", "G2", "G3"], maliyet)` izlemesi. Her satır döngünün bir turunun sonundaki durumdur.



Şekil 10.9: `sozlesme_agi()` izlemesinin son durumu. Turuncu hücreler kazanan teklifler, sağda atama listesi ve toplam 18 dk.

Tablo 10.8 ve Şekil 10.9 kodun örnekteki çalışmasını özetler. İzlemenin en öğretici yeri G2 turudur: bu turda `bos` kümesinde R1 yoktur. G1'i ucuza alan R1, G2 için de en ucuz adaydı (3), ama artık meşguldür. Fonksiyon her adımda yerel en iyi kararı verir, ama sonuç toplam optimum değildir. Kodun maliyeti de düşüktür: m görev ve n robot için her turda en çok n teklif karşılaştırılır; toplam iş yaklaşık $m \cdot n$ adımdır.

Kodun bir ayrıntısı daha önemlidir: `min` fonksiyonu eşitlik durumunda sözlükte ilk karşılaştığı robotu seçer. İki robot aynı teklifi verirse kazanan, kümenin iç sırasına bağlıdır. Gerçek bir protokolde eşitlik kuralı (ör. küçük numaralı robot kazanır) açıkça belirtilmelidir. Bu örnekte eşitlik çıkmaz.

10.3.5 Merkezî optimum: optimum()

Sözleşme ağının sonucunu değerlendirmek için optimumu bilmek gerekir. Tüm maliyetleri bilen bir merkez, robotların görevlere bütün atamalarını deneyebilir. Üç görev ve üç robot için atama, robotların bir sıralamasıdır: sıralamanın ilk robotu G1'i, ikincisi G2'yi, üçüncüsü G3'ü alır. Üç robotun *3! = 3 · 2 · 1 = 6* farklı sıralaması vardır.

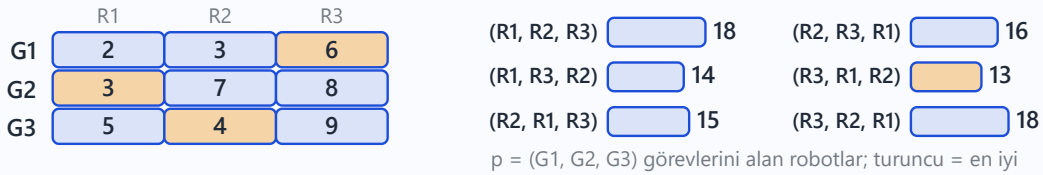
```
from itertools import permutations
def optimum(gorevler, maliyet):
  en_iyi = None
  for p in permutations(maliyet):
    t = sum(maliyet[r][g]
            for g, r in zip(gorevler, p))
    if en_iyi is None or t < en_iyi[0]:
      en_iyi = (t, p)
  return en_iyi
```

Kod satır satır şöyle çalışır.

1. `en_iyi = None`: Henüz hiçbir atama denenmemiştir.
2. `for p in permutations(maliyet)`: `permutations` (Python'un `itertools` modülünden) robotların bütün sıralamalarını üretir. `p = ("R3", "R1", "R2")`, G1'in R3'e, G2'nin R1'e, G3'ün R2'ye atanması demektir.
3. `zip(gorevler, p)`: Görevleri sıralamadaki robotlarla eşler: (G1, R3), (G2, R1), (G3, R2).
4. `t = sum(maliyet[r][g] for g, r in ...)`: Bu atamanın toplam süresini hesaplar.
5. `if en_iyi is None or t < en_iyi[0]`: İlk atama ya da şimdiye kadarkinden daha iyi bir atama bulunursa `en_iyi` güncellenir. Eşit süreli bir atama mevcut en iyiyi değiştirmez.

p	t	Karşılaştırma	en_iyi
(R1, R2, R3)	2 + 7 + 9 = 18	ilk atama	(18, R1 R2 R3)
(R1, R3, R2)	2 + 8 + 4 = 14	14 < 18 ⇒ yeni en iyi	(14, R1 R3 R2)
(R2, R1, R3)	3 + 3 + 9 = 15	15 ≥ 14 ⇒ değişmez	(14, R1 R3 R2)
(R2, R3, R1)	3 + 8 + 5 = 16	16 ≥ 14 ⇒ değişmez	(14, R1 R3 R2)
(R3, R1, R2)	6 + 3 + 4 = 13	13 < 14 ⇒ yeni en iyi	(13, R3 R1 R2)
(R3, R2, R1)	6 + 7 + 5 = 18	18 ≥ 13 ⇒ değişmez	(13, R3 R1 R2)

Tablo 10.9: `optimum()` izlemesi. p, sırasıyla G1, G2, G3'ü alan robotlardır.



Şekil 10.10: `optimum()` izlemesinin sonu. Sağda altı sıralamanın toplam süreleri; en iyisi (R3, R1, R2) ile 13 dk (turuncu). Soldaki turuncu hücreler optimum atamadır.

Tablo 10.9 ve Şekil 10.10 sonucu gösterir: optimum 13 dakikadır ve G1→R3, G2→R1, G3→R2 atamasıyla elde edilir. Sözleşme ağının bulduğu atama, listenin ilk satırıdır (18 dk). Tablodan bir şey daha görülür: altı atamanın ikisi 18 dakikadır, yani sözleşme ağı bu örnekte mümkün olan en kötü toplamı bulmuştur.

Bu yöntemin sorunu ölçektir. n robot için $n!$ sıralama vardır ve bu sayı çok hızlı büyür. Tablo 10.10 birkaç değeri gösterir.

n	n! (atama sayısı)	n^3
3	6	27
5	120	125
10	3.628.800	1.000
15	1.307.674.368.000	3.375

Tablo 10.10: Tüm atamaları denemenin maliyeti ($n!$) ile Macar algoritmasının yaklaşık adım sayısı (n^3).

n = 10'da üç buçuk milyondan fazla atama vardır; n = 15'te bu sayı bir trilyonu geçer. Tüm atamaları denemek küçük örnekler dışında pratik değildir. Macar algoritması aynı optimumu $O(n^3)$ zamanda bulur: n = 10'da yaklaşık bin, n = 15'te yaklaşık üç bin adım. Ancak hangi algoritma kullanılırsa kullanılsın, merkezî çözümün asıl engeli hesap değil bilgidir: merkezin bütün maliyetleri bilmesi gerekir.

10.3.6 Teklifler ve dürüstlük

Şimdiye kadar robotların maliyetlerini dürüstçe teklif ettiği varsayıldı. Robotlar bencil ajanlarsa bu varsayım geçerli midir? Cevap, kazanana ne ödendiğine bağlıdır. Sözleşme ağında teklif verme bir ters açık artırmadır: en düşük teklif kazanır ve kazanan, işi yaptığı için bir ödeme alır. Robotun kazancı, aldığı ödeme ile gerçek maliyeti arasındaki farktır.

Örnek 10.5

G1 ihalesini ele alalım. Gerçek maliyetler R1 2, R2 3, R3 6'dır. İki ödeme kuralı karşılaştırılır.

1. **Kazanana kendi teklifi ödenir (birinci fiyat).** R1 dürüstçe 2 teklif ederse kazanır, 2 alır ve maliyeti 2 olduğu için kazancı *2 − 2 = 0*'dır.
2. **Birinci fiyatta şişirme.** R1 teklifini 2,9'a çıkarsın. Diğer teklifler değişmezse R1 yine en düşük teklifi vermiştir (2,9 < 3), kazanır ve 2,9 alır. Kazancı *2,9 − 2 = 0,9*'dur. Bu kuralda yüklenici maliyetini şişirmek ister; dürüst teklif en iyisi değildir.
3. **Kazanana ikinci en düşük teklif ödenir (Vickrey).** R1 dürüstçe 2 teklif eder ve kazanır; ödeme ikinci en düşük teklif olan 3'tür. Kazancı *3 − 2 = 1*'dir.
4. **Vickrey'de şişirme işe yaramaz.** R1, 3'ün altında kalan herhangi bir teklifle (2,5 ya da 2,9) yine kazanır ve yine 3 alır: kazanç 1 değişmez. Teklif 3'ün üstüne çıkarsa R1 kaybeder ve kazancı 0'a düşer. Teklifi şişirmek ya kazancı değiştirmez ya da düşürür.
5. **Vickrey'de düşük teklif de işe yaramaz.** R2'nin maliyeti 3'tür. R2 işi almak için 1,5 teklif etsin. Kazanır, ama ödeme ikinci en düşük teklif olan R1'in 2'sidir. Kazancı *2 − 3 = −1*'dir: zarar eder. Dürüst teklifle (3) kaybeder ve kazancı 0 olur; bu daha iyidir.

Örnek 10.5 genel bir sonucun örneğidir. Kazanana kendi teklifi ödenirse yüklenici maliyetini şişirmek ister. İkinci en düşük teklif ödenirse (Vickrey kuralı) **dürüst teklif en iyisidir**: robotun teklifi yalnız kazanıp kazanmadığını belirler, ödemeyi belirlemez. Ödeme başkalarının teklifine bağlı olduğu için robotun yalan söyleyerek ödemeyi artırma olanağı yoktur. Robotun yapabileceği tek şey, kazanmanın kârlı olup olmadığına karar vermektir; dürüst teklif bu kararı her zaman doğru verir.

Bu sonuç tek bir görevin ihalesi için geçerlidir. Görevler sırayla ihaleye çıktığında bir robot, sonraki bir görevi almak için bir ihaleyi bilerek kaybetmek isteyebilir. Atamanın bütünü için dürüstlüğü, her robota ötekilere verdiği kaybı ödeten genel kural sağlar: **VCG mekanizması** (Vickrey–Clarke–Groves). VCG, Vickrey kuralının çok nesneli genellemesidir.

10.3.7 Güçlü yanlar, zayıf yanlar ve iyileştirmeler

Örnekteki sonuç sözleşme ağının hem gücünü hem zayıflığını gösterir.

Güçlü yanlar.

- **Dağıtık:** yönetici maliyet tablosunu bilmeden iyi bir atama yapar. Bilgi robotlarda kalır ve yalnız gerektiği kadar paylaşılır.
- **Az mesaj:** n robotlu bir sistemde görev başına mesaj sayısı *O(n)*'dir: bir duyuru (yayın), en çok n teklif ve bir ödül.
- **Esneklik:** yeni robot eklemek kolaydır; robot duyuruları dinlemeye başlar. Bozulan bir robotun görevi yeniden ihaleye çıkarılır.

Zayıf yanlar.

- **Açgözlü ve sıralı:** örnekte 18 dk, optimum 13 dk (%38 fazla). G1'i ucuza alan R1, G2 için en iyi aday olduğu hâlde meşgul kalmıştır.

- **Stratejik teklif:** robotlar maliyetlerini stratejik olarak şişirebilir (10.3.6).
- **Sıra bağımlılığı:** görevlerin ihaleye çıkış sırası sonucu değiştirir.

Sıra bağımlılığı en açık biçimde, aynı maliyet tablosuyla altı farklı ihale sırası denenerek görülür. Tablo 10.11 sonuçları verir.

İhale sırası	1. ihale	2. ihale	3. ihale	Toplam
G1, G2, G3	G1→R1 (2)	G2→R2 (7)	G3→R3 (9)	18
G1, G3, G2	G1→R1 (2)	G3→R2 (4)	G2→R3 (8)	14
G2, G1, G3	G2→R1 (3)	G1→R2 (3)	G3→R3 (9)	15
G2, G3, G1	G2→R1 (3)	G3→R2 (4)	G1→R3 (6)	13
G3, G1, G2	G3→R2 (4)	G1→R1 (2)	G2→R3 (8)	14
G3, G2, G1	G3→R2 (4)	G2→R1 (3)	G1→R3 (6)	13

Tablo 10.11: Aynı maliyet tablosunda altı ihale sırasının sonucu. Her hücrede görev, kazanan ve kazanan teklif yazılıdır.

Altı sıranın ikisi optimuma (13) ulaşır, biri en kötü sonucu (18) verir. G1'in en sona bırakıldığı sıralarda optimum bulunur: G1'in birkaç iyi adayı (R1 2, R2 3, R3 6) vardır, bu yüzden en son ihaleye çıkması fazla bir kayba yol açmaz. Buna karşılık G1 ilk sıradaysa onu R1 kapar; oysa R1, G2 için de en iyi adaydır. Kıt robotu ilk ihalede harcamak, sonraki ihaleleri pahalılaştırır.

Sözleşme ağının zayıflıkları üç iyileştirmeyle azaltılır.

Yeniden ihale (atamaları takasla düzeltme)
: İlk atamadan sonra robotlar ikili takas önerir. İki robot görevlerini değiştirince toplam süre azalıyorsa takas yapılır. Yerel düzeltmeler açgözlü kararların bir kısmını geri alır.

Kombinatoryal açık artırma
: Robotlar tek görevlere değil görev paketlerine teklif verir. Böylece bir robot, “G1'i alırsam G2'yi alamam” gibi görevler arasındaki bağı teklifine yansıtabilir.

İkinci fiyat ya da VCG ödemeleri
: Kazanana ikinci en iyi teklif (ya da genel olarak diğerlerine verdiği kayıp) ödenir. Dürüst teklif en iyi strateji olur (10.3.6).

Örnek 10.6

Sözleşme ağının 18 dakikalık ataması (G1→R1, G2→R2, G3→R3) yeniden ihaleyle iyileştirilir.

1. **R1 ile R2 görev takası.** Takastan önce R1 G1'i (2), R2 G2'yi (7) yapar: toplam 9. Takastan sonra R1 G2'yi (3), R2 G1'i (3) yapar: toplam 6. Takas 3 dakika kazandırır. Yeni atama G1→R2, G2→R1, G3→R3; toplam *3 + 3 + 9 = 15*.
2. **R2 ile R3 görev takası.** Takastan önce R2 G1'i (3), R3 G3'ü (9) yapar: toplam 12. Takastan sonra R2 G3'ü (4), R3 G1'i (6) yapar: toplam 10. Takas 2 dakika kazandırır. Yeni atama G1→R3, G2→R1, G3→R2; toplam *6 + 3 + 4 = 13*.

3. **Başka kârlı takas var mı?** 13, optimumdur (Tablo 10.9); hiçbir takas toplamı daha fazla düşüremez. İki takasla optimuma ulaşılmıştır.

Örnek 10.6 yeniden ihalenin gücünü gösterir, ama bir uyarı da gerekir: ikili takaslar her örnekte optimuma ulaşmayabilir. Hiçbir ikili takasın kazandırmadığı ama optimum olmayan atamalar vardır. Bu yüzden takas, açgözlü sonucu iyileştiren pratik bir araçtır; optimumun garantisi değildir.

10.3.8 Görev tahsisinde sık hatalar

Sık yapılan hata

Yöneticinin maliyetleri bildiğini varsaymak. Sözleşme ağında yönetici maliyet tablosunu görmez; maliyetleri yalnız tekliflerden öğrenir. Yönetici tabloyu bilseydi ihaleye gerek kalmaz, optimumu doğrudan hesaplardı.

Meşgul robotun teklif verdiğini sanmak. Her robot en çok bir görev alır. G2 ihalesinde R1'in maliyeti (3) en düşüktür, ama R1 meşgul olduğu için teklif vermez. G2'nin kazananı R2'dir (7).

Kaybın nedenini dürüstlükte aramak. Örnekte robotlar dürüst teklif verir ve sonuç yine optimum değildir. Kaybın nedeni stratejik teklif değil, açgözlü ve sıralı ihaledir. Maliyet şişirmek bu kaybı azaltmaz.

Mesaj sayısını abartmak. Görev başına mesaj sayısı $O(n)$'dir, $O(n^2)$ değil: bir duyuru, n teklif, bir ödül. Robotlar birbirleriyle değil, yalnız yöneticiyle mesajlaşır.

Yüzdeyi yanlış tabana göre hesaplamak. 18 dakika optimumdan $(18 - 13)/13 \approx \%38$ fazladır. $(18 - 13)/18 \approx \%28$ ise sözleşme ağının süresinin ne kadarının boşa gittiğidir; ikisi farklı sorulara cevap verir.

10.4 Koalisyon oyunları

Görev tahsisinde her robot işini tek başına yapıyordu. Birçok işte ise ajanlar birlikte çalışarak tek başlarına üretemeyecekleri bir değer üretir. Bu kesim, grupların oluşmasını ve grubun ürettiği değerin üyeler arasında paylaşılmasını inceler. Oyun kuramının bu dalına **işbirlikçi oyun kuramı** ya da **koalisyon oyunları** denir. Dalın üç temel aracı vardır: adalet ölçütü olarak Shapley değeri, kararlılık ölçütü olarak çekirdek ve oylamadaki gücü ölçen güç endeksleri.

10.4.1 Koalisyon oluşumu

Tek başına hedefe ulaşamayan ajanlar geçici ya da kalıcı **koalisyonlar** kurabilir. Kargo firmaları araç paylaşır: yarı boş giden iki kamyon yerine tek bir dolu kamyon gönderilir. Drone'lar tek başına kaldıramayacakları ağır bir yükü birlikte taşır. Hesaplama sunucuları bir işi parçalara bölüp birlikte çözer.

Her durumda grup, üyelerinin tek başına üretebileceğinden fazlasını üretir.

Tanım 10.6

Bir **koalisyon** (*coalition*), birlikte hareket etmeye karar veren ajanların oluşturduğu gruptur. Bütün ajanlardan oluşan koalisyona **büyük koalisyon** (*grand coalition*) denir. Ajanların hangi koalisyonlara ayrılacağı sürecine **koalisyon oluşumu** (*coalition formation*) denir.

Bu oyunlar önceki kesimlerdeki oyunlardan bir noktada ayrılır: **işbirlikçidir**. Mahkûmun İkilemi'nde oyuncular "ikimiz de sessiz kalalım" diye anlaşsa bile anlaşmanın bir yaptırımı yoktu; her oyuncu anlaşmayı bozmak isterdi. Koalisyon oyunlarında ise **bağlayıcı anlaşma** yapılabildiği varsayılır: bir sözleşme, bir yasa ya da bir yazılım protokolü, koalisyonun üyelerine verdiği sözü tutmalarını sağlar. Bu yüzden soru değişir. Soru artık "her oyuncu hangi stratejiyi seçecek?" değil, şudur: **kim kiminle birleşir ve kazanç nasıl paylaşılır?**

İki soru birbirine bağlıdır. Paylaşım adil değilse koalisyon dağılır: kendini haksızlığa uğramış hisseden ajanlar ayrılıp başka bir grup kurar. Koalisyonun kararlılığı paylaşım kuralına bağlıdır. Adalet ve kararlılık bu yüzden birlikte ele alınır.

Shapley değeri
: **Adalet** ölçütüdür: herkes ortalama marjinal katkısı kadar alır (10.4.5).

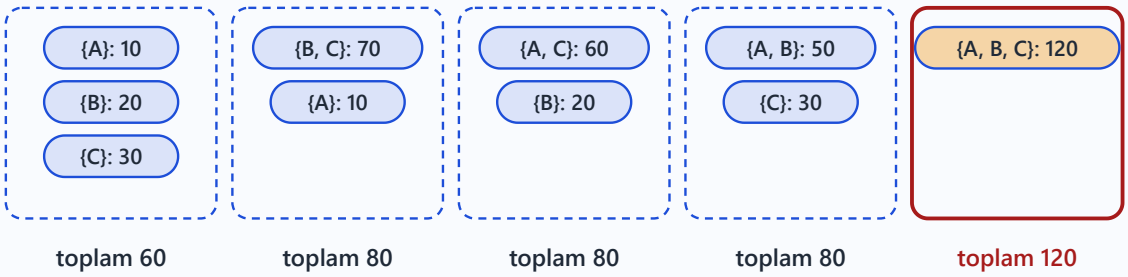
Çekirdek
: **Kararlılık** ölçütüdür: hiçbir alt grup ayrılarak daha fazlasını alamaz (10.4.9).

10.4.2 Koalisyon yapıları

Ajanlar tek bir büyük grupta birleşmek zorunda değildir. Üç lojistik firması A, B ve C düşünelim. Üçü ayrı çalışabilir, ikisi birleşip üçüncüsü ayrı kalabilir ya da üçü birden birleşebilir.

Tanım 10.7

Bir **koalisyon yapısı** (*coalition structure*), ajanların ayrık koalisyonlara bölünmesidir. Her ajan tam olarak bir koalisyondadır ve iki koalisyonun ortak üyesi yoktur. Bir yapının **toplam değeri**, içindeki koalisyonların değerlerinin toplamıdır.



Şekil 10.11: Üç firmanın beş koalisyon yapısı ve toplam değerleri. Herkes tek başına 60, bir ikili ve bir tekli 80, büyük koalisyon 120 değer üretir. En iyi yapı büyük koalisyondur (kırmızı çerçeve).

Şekil 10.11 üç ajanın bütün yapılarını gösterir. Değerler Tablo 10.12'de verilen karakteristik fonksiyondan gelir. Herkes tek başına çalışırsa toplam *10 + 20 + 30 = 60*'tır. {B, C} birleşir ve A ayrı kalırsa toplam *70 + 10 = 80*'dir. {A, C} ile {B}: *60 + 20 = 80*. {A, B} ile {C}: *50 + 30 = 80*. Üçü birlikte 120 üretir. Toplum açısından soru şudur: **toplam değeri en büyük yapı hangisidir?** Bu örnekte cevap büyük koalisyondur.

Yapı sayısı. n ajanı boş olmayan ayrık gruplara bölmenin yol sayısına **Bell sayısı** denir ve B_n ile gösterilir. Üç ajan için 5 yapı vardır. Bell sayıları şu ilişkiyle adım adım hesaplanabilir: (n + 1). ajanı içeren grubun, ilk n ajandan kaç tanesini içerdiğine göre sayım yapılır. Bu grup ilk n ajandan k tanesini içeriyorsa, bu k ajanı seçmenin $C(n, k)$ yolu vardır; geri kalan $n - k$ ajan da B_{n-k} yolla bölünür. Toplam:

$$B_{n+1} = \Sigma_{k=0}^{n} C(n, k) \cdot B_{n-k},\ B_0 = 1.$$

Bu ilişkiyle $B_1 = 1$, $B_2 = 1 + 1 = 2$, $B_3 = B_2 + 2B_1 + B_0 = 2 + 2 + 1 = 5$ bulunur. Dört ajan için $B_4 = C(3,0)\cdot B_3 + C(3,1)\cdot B_2 + C(3,2)\cdot B_1 + C(3,3)\cdot B_0 = 5 + 6 + 3 + 1 = 15$'tir. Sayı çok hızlı büyür: n = 8 için 4.140, n = 10 için 115.975 yapı vardır. Genel durumda en iyi yapıyı bulmak **NP-zordur** (Sandholm vd., 1999); yani ajan sayısı büyüdükçe bilinen hiçbir algoritma bunu her örnekte hızlı çözemez.

Örnekteki oyunda bu zorluk ortaya çıkmaz, çünkü oyun **süperadditiftir**: birleşmek hiçbir zaman zarar vermez. Bu durumda en iyi yapı her zaman büyük koalisyondur ve geriye tek bir soru kalır: büyük koalisyonun ürettiği 120 nasıl paylaşılacak?

10.4.3 Karakteristik fonksiyon

Tanım 10.8

Oyuncular kümesi $N = \{1, \ldots, n\}$ olsun. Bir **koalisyon oyunu**, her koalisyon $S \subseteq N$'ye bir $v(S)$ değeri atayan bir fonksiyonla verilir. $v(S)$, S'nin üyelerinin birlikte, diğerlerinden bağımsız olarak elde edebileceği toplam değerdir. Boş koalisyonun değeri 0'dır: $v(\emptyset) = 0$. v'ye **karakteristik fonksiyon** (*characteristic function*) denir.

Kazanç grup içinde serbestçe paylaştırılabiliyorsa, yani koalisyonun değeri para gibi bölünüp üyeler arasında istenen oranda dağıtılabiliyorsa, bu tür oyunlara **aktarılabilir faydalı oyun** (*transferable-utility game*, TU oyun) denir.

Karakteristik fonksiyon oyunun bütün bilgisini tek bir tabloda toplar. Oyuncuların hangi eylemleri seçtiği, nasıl çalıştıkları tabloda görünmez; yalnız her grubun sonunda ne kadar değer ürettiği görünür. Bu soyutlama işbirlikçi oyun kuramının temelidir: grup içi ayrıntılar bağlayıcı anlaşmalarla çözüldüğü için yalnız grupların gücü önemlidir.

Bölümün ana örneği üç lojistik firmasıdır. Firmalar birlikte 120 bin TL kâr eder, en iyi ikili 70, tek başına en fazla 30 kazanır. Karakteristik fonksiyonun bütün değerleri Tablo 10.12'de verilmiştir.

S	{A}	{B}	{C}	{A, B}	{A, C}	{B, C}	{A, B, C}
v(S)	10	20	30	50	60	70	120

Tablo 10.12: Üç lojistik firmasının karakteristik fonksiyonu, $v(S)$ (kâr, bin TL).

Tanım 10.9

Bir koalisyon oyunu, her ayrık S ve T koalisyonu için ($S \cap T = \emptyset$)

$$v(S \cup T) \geq v(S) + v(T)$$

sağlanıyorsa **süperadditiftir** (*superadditive*). İki ayrık grubun birleşmesi, ayrı ayrı ürettiklerinin toplamından az üretmez; yani birleşmek zarar vermez.

Örnek 10.7

Lojistik oyununun süperadditif olduğu, bütün ayrık koalisyon çiftleri denetlenerek gösterilir. Boş koalisyonu içeren çiftler kendiliğinden sağlanır ($v(\emptyset) = 0$). Geriye altı çift kalır.

1. **İki tekli birleşir.** $v(AB) = 50 \geq 10 + 20 = 30$ ✓ ; $v(AC) = 60 \geq 10 + 30 = 40$ ✓ ; $v(BC) = 70 \geq 20 + 30 = 50$ ✓.

2. **Bir ikili ile bir tekli birleşir.** $v(ABC) = 120 \geq v(BC) + v(A) = 70 + 10 = 80$ ✓; $120 \geq v(AC) + v(B) = 60 + 20 = 80$ ✓; $120 \geq v(AB) + v(C) = 50 + 30 = 80$ ✓.

Altı eşitsizliğin hepsi sağlanır; oyun süperadditiftir. Bu yüzden büyük koalisyon {A, B, C} oluşur ve paylaşılacak değer 120'dir.

Süperadditiflik her oyunda sağlanmaz. Bir koalisyon büyüdükçe koordinasyon maliyeti artabilir: çok büyük bir kargo ortaklığında araç planlaması zorlaşır. Böyle bir oyunda iki ayrı grup, birleşik gruptan fazla üretebilir ve en iyi yapı büyük koalisyon olmayabilir.

10.4.4 120 nasıl paylaşılmalı?

Büyük koalisyon 120 üretir. Bu değer nasıl paylaşılmalıdır? Her makul paylaşımın sağlaması gereken iki koşul vardır.

Tanım 10.10

Bir **paylaşım** (*imputation*) $x = (x_A, x_B, x_C)$, iki koşulu sağlayan bir ödeme vektörüdür:

- **Verimlilik:** toplamı büyük koalisyonun değerine eşittir, $x_A + x_B + x_C = v(N) = 120$. Hiçbir değer boşa gitmez.
- **Bireysel rasyonellik:** herkes en az tek başına değerini alır, $x_i \geq v(\{i\})$. Aksi hâlde o ajan koalisyondan ayrılıp tek başına çalışırdı.

Bu iki koşulu sağlayan sonsuz sayıda paylaşım vardır. Tablo 10.13 dört öneriyi karşılaştırır.

Öneri	x_A	x_B	x_C	Gerekçe / sorun
Eşit bölüşüm	40	40	40	Basit; ama C'nin daha fazla katkısını yok sayar
Tek başına değerle orantılı (10 : 20 : 30)	20	40	60	Yalnızca $v(\{i\})$'ye bakar; sinerjileri görmez
Güçlü C'nin önerisi	10	25	85	A ve B birlikte 50 alabilir; kabul etmezler
Shapley değeri	30	40	50	Ortalama marjinal katkı

Tablo 10.13: 120'nin dört paylaşım önerisi.

Dört önerinin hepsi bir paylaşımdır: toplamları 120'dir ve herkes en az tek başına değerini alır. Örneğin C'nin önerisinde A tam olarak 10 alır, yani tek başına değerinin altına düşmez. Yine de öneriler çok farklıdır.

Eşit bölüşüm en basit kuraldır, ama C'nin her gruba daha fazla değer kattığını yok sayar. **Orantılı paylaşım** 120'yi tek başına değerler oranında böler: toplam tek başına değer $10 + 20 + 30 = 60$ olduğu için her oyuncu tek başına değerinin $120/60 = 2$ katını alır. Bu kural sinerjileri görmez: A'nın B ile 20, C ile 20 birimlik ek değer yarattığı ($50 - 30$ ve $60 - 40$) hesaba katılmaz. **Güçlü C'nin önerisi** C'ye

85 verir. Ancak A ve B bu durumda birlikte $10 + 25 = 35$ alır; oysa ikisi ayrılıp kendi koalisyonlarını kursa $v(AB) = 50$ kazanırlar. Bu öneri kabul edilmez. **Shapley değeri** ise her oyuncunun bütün olası gruplara kattığı değerin ortalamasını verir.

Tablo iki ayrı ölçütü ortaya koyar: **paylaşım adil mi** (Shapley değeri) ve **kararlı mı** (çekirdek)? C'nin önerisi kararlı değildir, çünkü bir alt grup ayrılmak ister. Eşit bölüşüm bu oyunda kararlıdır, ama katkıları yansıtmaz.

10.4.5 Shapley değeri: tanım ve aksiyomlar

Shapley değerinin fikri basit bir hikâyeyle anlatılır. Oyuncular rastgele bir sırayla tek tek gelip koalisyona katılır. Her oyuncuya, geldiği anda koalisyona kattığı değer verilir. Örneğin sıra A → B → C ise A önce gelir ve boş koalisyona $v(A) - v(\emptyset) = 10$ katar. B gelince koalisyon {A}'dan {A, B}'ye büyür ve B'nin katkısı $v(AB) - v(A) = 50 - 10 = 40$'tır. C en son gelir ve $v(ABC) - v(AB) = 120 - 50 = 70$ katar. Bir oyuncunun bir koalisyona kattığı bu değere **marjinal katkı** (*marginal contribution*) denir.

Tek bir sıra adil değildir: son gelen oyuncu çoğu zaman en büyük sinerjiyi toplar. Bu yüzden Shapley değeri bütün sıraların ortalamasını alır.

Tanım 10.11

i oyuncusunun **Shapley değeri** (*Shapley value*) φ_i, n oyuncunun $n!$ geliş sırasının hepsi üzerinden i'nin ortalama marjinal katkısıdır:

$$\varphi_i = (1/n!) \cdot \Sigma_{\pi} \left[v(P_i^{\pi} \cup \{i\}) - v(P_i^{\pi}) \right]$$

Burada toplam bütün π sıraları üzerindendir ve P_i^{π}, π sırasında i'den önce gelen oyuncuların kümesidir.

Aynı değer, sıralar yerine koalisyonlar üzerinden de yazılabilir. i'den önce gelenlerin kümesi S olan sıra sayısı, S'nin üyelerinin kendi aralarında dizilişi ($|S|!$) ile i'den sonra gelenlerin dizilişinin ($(n - |S| - 1)!$) çarpımıdır. Bu yüzden

$$\varphi_i = \Sigma_{S \subseteq N \setminus \{i\}} \left[|S|! \cdot (n - |S| - 1)! \,/\, n! \right] \cdot \left[v(S \cup \{i\}) - v(S) \right]$$

yazılır. Bu formül $n!$ sıra yerine i'yi içermeyen 2^{n-1} koalisyon üzerinden toplam alır. Üç oyunculu oyunda ağırlıklar şöyledir: $|S| = 0$ için $0! \cdot 2!/3! = 2/6 = 1/3$, $|S| = 1$ için $1! \cdot 1!/3! = 1/6$, $|S| = 2$ için $2! \cdot 0!/3! = 2/6 = 1/3$. Ağırlıkların toplamı $1/3 + 2 \cdot 1/6 + 1/3 = 1$'dir.

Hesap maliyeti. Kesin hesap ya $n!$ sırayı ya da 2^{n-1} alt kümeyi gerektirir. Her ikisi de n büyüdükçe çok hızlı artar. Büyük n için Shapley değeri **örnekleme** ile yaklaşık hesaplanır: rastgele birkaç bin sıra seçilir ve bu sıralardaki marjinal katkıların ortalaması alınır. Örnek sayısı arttıkça ortalama gerçek değere yaklaşır.

Shapley değerinin önemi, yalnız sezgisel bir kural olmasından gelmez. Shapley (1953), bu değerin dört doğal koşulu sağlayan tek kural olduğunu göstermiştir. Bu koşullara **aksiyom** denir.

Teorem 10.1 (Shapley, 1953)

Her koalisyon oyununa bir ödeme vektörü atayan kurallar arasında aşağıdaki dört aksiyomu sağlayan tek kural Shapley değeridir.

- **Verimlilik:** $\Sigma_i \varphi_i = v(N)$. Büyük koalisyonun değeri tam olarak dağıtılır.
- **Simetri:** Aynı katkıyı yapan iki oyuncu aynı payı alır. Biçimsel olarak, i ve j'yi içermeyen her S için $v(S \cup \{i\}) = v(S \cup \{j\})$ ise $\varphi_i = \varphi_j$.
- **Kukla oyuncu:** Hiçbir koalisyona katkısı olmayan oyuncu, yani her S için $v(S \cup \{i\}) = v(S)$ olan oyuncu, $\varphi_i = 0$ alır.
- **Toplamsallık:** İki oyunun değerleri toplanırsa Shapley payları da toplanır: $\varphi_i(v + w) = \varphi_i(v) + \varphi_i(w)$.

Teoremin kanıtı bu notun kapsamı dışındadır. Ancak tanımın aksiyomları neden sağladığı kolayca görülür. **Verimlilik:** her sırada marjinal katkıların toplamı, boş koalisyondan büyük koalisyona kadar adım adım eklenen değerlerin toplamıdır ve teleskopik olarak $v(N) - v(\emptyset) = v(N)$'ye eşittir. Ortalama alınca da toplam $v(N)$ kalır. **Kukla oyuncu:** her sırada katkısı 0 olan oyuncunun ortalaması da 0'dır. **Simetri:** i ile j'nin yerini değiştiren bir eşleme, i'nin her sıradaki katkısını j'nin başka bir sıradaki katkısıyla eşler; ortalamalar eşit çıkar. **Toplamsallık:** marjinal katkı v'ye göre doğrusaldır; toplamın katkısı, katkıların toplamıdır.

Shapley değeri bugün oyun kuramının dışında da kullanılır. Makine öğrenmesinde bir modelin tek bir kararını girdi özelliklerinin katkılarına bölen **SHAP** (*SHapley Additive exPlanations*) yöntemi, özellikleri "oyuncu", modelin tahminini "koalisyon değeri" sayar ve her özelliğe Shapley değerini atar.

10.4.6 Shapley değeri adım adım

Lojistik oyununda Shapley değeri, altı geliş sırasının her birinde marjinal katkılar yazılarak hesaplanır.

Örnek 10.8

Her sırada ajanlar tek tek gelir; her birine, geldiği anda koalisyona kattığı değer yazılır. Değerler Tablo 10.12'den okunur.

1. **A → B → C.** A: $10 - 0 = 10$; B: $v(AB) - v(A) = 50 - 10 = 40$; C: $v(ABC) - v(AB) = 120 - 50 = 70$. Satır toplamı $10 + 40 + 70 = 120$.
2. **A → C → B.** A: $10 - 0 = 10$; C: $v(AC) - v(A) = 60 - 10 = 50$; B: $v(ABC) - v(AC) = 120 - 60 = 60$. Satır toplamı 120.

3. **B → A → C.** B: *20 − 0 = 20*; A: *v(AB) − v(B) = 50 − 20 = 30*; C: *120 − 50 = 70*. Satır toplamı 120.
4. **B → C → A.** B: *20*; C: *v(BC) − v(B) = 70 − 20 = 50*; A: *v(ABC) − v(BC) = 120 − 70 = 50*. Satır toplamı 120.
5. **C → A → B.** C: *30 − 0 = 30*; A: *v(AC) − v(C) = 60 − 30 = 30*; B: *120 − 60 = 60*. Satır toplamı 120.
6. **C → B → A.** C: *30*; B: *v(BC) − v(C) = 70 − 30 = 40*; A: *120 − 70 = 50*. Satır toplamı 120.
7. **Sütunları topla ve 6'ya böl.** A: *10 + 10 + 30 + 50 + 30 + 50 = 180*, $\varphi_A = 180/6 = 30$. B: *40 + 60 + 20 + 20 + 60 + 40 = 240*, $\varphi_B = 40$. C: *70 + 50 + 70 + 50 + 30 + 30 = 300*, $\varphi_C = 50$.

Shapley değeri *φ = (30, 40, 50)*'dir. Toplamı 120'dir; verimlilik aksiyomu sağlanır.

Sıra	A	B	C
A → B → C	10	40	70
A → C → B	10	60	50
B → A → C	30	20	70
B → C → A	50	20	50
C → A → B	30	60	30
C → B → A	50	40	30
Toplam	180	240	300
φ = toplam / 6	30	40	50

Tablo 10.14: Altı geliş sırasında marjinal katkılar. Her satırın toplamı *v(ABC) = 120*'dir.

Tablo 10.14 hesabı tek bakışta gösterir. Tablodaki iki düzen, hesabın doğruluğunu denetlemeye yarar. Birincisi, her satırın toplamı 120'dir; bir satır 120 tutmuyorsa o satırda bir çıkarma hatası vardır. İkincisi, bir oyuncunun ilk geldiği sıralarda katkısı her zaman tek başına değeridir: A'nın ilk iki satırı 10, B'nin üçüncü ve dördüncü satırları 20, C'nin son iki satırı 30'dur.

Koalisyon formülüyle denetim. Aynı sonuç 10.4.5'teki ağırlıklı formülle de bulunur. A için i'yi içermeyen dört koalisyon vardır: ∅ (ağırlık 1/3), {B} ve {C} (ağırlık 1/6), {B, C} (ağırlık 1/3).

$$\varphi_A = (1/3)\cdot 10 + (1/6)\cdot(50 - 20) + (1/6)\cdot(60 - 30) + (1/3)\cdot(120 - 70) = 10/3 + 5 + 5 + 50/3 = 30$$

B için: *(1/3)·20 + (1/6)·(50 − 10) + (1/6)·(70 − 30) + (1/3)·(120 − 60) = 20/3 + 20/3 + 20/3 + 20 = 40*. C için: *(1/3)·30 + (1/6)·(60 − 10) + (1/6)·(70 − 20) + (1/3)·(120 − 50) = 10 + 25/3 + 25/3 + 70/3 = 50*. İki yöntem aynı sonucu verir.

Sonucun yorumu. C en büyük payı (50) alır, çünkü hem tek başına en değerli oyuncudur (30) hem de her gruba en çok değeri katar. A en az payı (30) alır. Yine de A'nın payı tek başına değerinin (10) üç katıdır, çünkü A'nın gruplara kattığı sinerji büyüktür: A, {B, C}'ye 50 katar. Shapley değeri, her oyuncu-

ya tek başına değerini ve buna ek olarak yarattığı sinerjilerden ortalama payını verir.

10.4.7 Kod izleme: shapley()

Shapley değerinin tanımı doğrudan koda dönüştürülebilir. Fonksiyon iki girdi alır: oyuncu listesi `N = ["A", "B", "C"]` ve karakteristik fonksiyon `v`. `v`, anahtarları `frozenset` olan bir sözlüktür: `v[frozenset()] = 0`, `v[frozenset({"A"})] = 10`, …, `v[frozenset({"A", "B", "C"})] = 120`. `frozenset`, değiştirilemeyen bir kümedir; sıradan küme sözlük anahtarı olamaz, değiştirilemeyen küme olabilir.

```
from itertools import permutations
def shapley(N, v):
  phi = {i: 0 for i in N}
  siralar = list(permutations(N))
  for sira in siralar:
    S = frozenset()
    for i in sira:
      katki = v[S | {i}] - v[S]
      phi[i] += katki
      S = S | {i}
  return {i: phi[i] / len(siralar) for i in N}
```

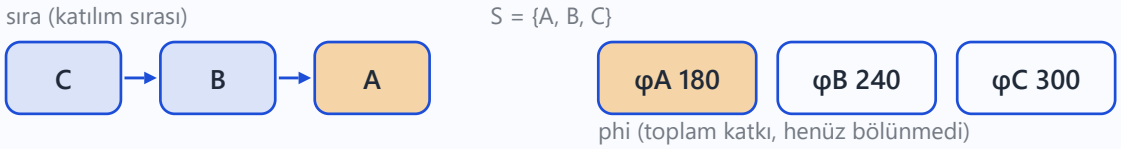
Kod satır satır şöyle çalışır.

1. `phi = {i: 0 for i in N}`: Her oyuncunun toplam katkısı 0'dan başlar.
2. `siralar = list(permutations(N))`: `permutations` bütün geliş sıralarını üretir: üç oyuncu için 3! = 6 sıra.
3. `for sira in siralar`: Her sıra ayrı ayrı işlenir. `S = frozenset()` ile koalisyon boş başlar.
4. `for i in sira`: Oyuncular sıradaki dizilişleriyle gelir.
5. `katki = v[S | {i}] - v[S]`: i'nin marjinal katkısı. `S | {i}`, S'ye i'nin eklenmiş hâlidir; `frozenset` ile sıradan kümenin birleşimi yine `frozenset` olduğu için sözlük anahtarı olarak kullanılabilir.
6. `phi[i] += katki` ve `S = S | {i}`: Katkı i'nin toplamına eklenir, i koalisyona katılır.
7. `return {i: phi[i] / len(siralar) for i in N}`: Toplamlar sıra sayısına (6) bölünür.

Sıra	i	S (önce)	katki	phi (A, B, C)
(A, B, C)	A	∅	10 − 0 = 10	(10, 0, 0)
	B	{A}	50 − 10 = 40	(10, 40, 0)
	C	{A, B}	120 − 50 = 70	(10, 40, 70)
(A, C, B)	A	∅	10 − 0 = 10	(20, 40, 70)
	C	{A}	60 − 10 = 50	(20, 40, 120)
	B	{A, C}	120 − 60 = 60	(20, 100, 120)

Sıra	i	S (önce)	katki	phi (A, B, C)
(B, A, C)	B	∅	20 − 0 = 20	(20, 120, 120)
	A	{B}	50 − 20 = 30	(50, 120, 120)
	C	{A, B}	120 − 50 = 70	(50, 120, 190)
(B, C, A)	B	∅	20 − 0 = 20	(50, 140, 190)
	C	{B}	70 − 20 = 50	(50, 140, 240)
	A	{B, C}	120 − 70 = 50	(100, 140, 240)
(C, A, B)	C	∅	30 − 0 = 30	(100, 140, 270)
	A	{C}	60 − 30 = 30	(130, 140, 270)
	B	{A, C}	120 − 60 = 60	(130, 200, 270)
(C, B, A)	C	∅	30 − 0 = 30	(130, 200, 300)
	B	{C}	70 − 30 = 40	(130, 240, 300)
	A	{B, C}	120 − 70 = 50	(180, 240, 300)

Tablo 10.15: `shapley(N, v)` izlemesi. Her satır iç döngünün bir adımıdır; son sütun `phi` 'nin o adımdan sonraki hâlidir.



Şekil 10.12: `shapley()` izlemesinde son sıranın son adımı. Sıra C → B → A, en son A katılır (turuncu); `phi` toplamları (180, 240, 300) henüz 6'ya bölünmemiştir.

Tablo 10.15 ve Şekil 10.12 izlemeyi gösterir. Döngüler bitince `phi = {A: 180, B: 240, C: 300}` olur. Son satır her toplamı 6'ya böler ve `{A: 30.0, B: 40.0, C: 50.0}` döndürür. İzleme Tablo 10.14'ün sütunlarını satır satır biriktirir; her sıranın sonunda `phi` 'nin toplamı 120 artar (ilk sıradan sonra 120, ikinciden sonra 240, …, altıncıdan sonra 720 = 6 · 120).

Kodun maliyeti $n! \cdot n$ adımdır: n! sıranın her birinde n oyuncu işlenir. n = 10 için bu yaklaşık 36 milyon adımdır ve bir bilgisayar için hâlâ yapılabilir; n = 20 için ise $20! \cdot 20 \approx 4{,}9 \cdot 10^{19}$ adımdır ve yapılamaz. Büyük oyunlarda `siralar` listesi yerine rastgele seçilmiş birkaç bin sıra kullanılır; geri kalan kod değişmez.

10.4.8 Oyuna dönüş: ikili sinerji modeli

Açılıştaki Shapley Koalisyon Oyunu'na (10.1.1) dönelim. Oyunda bir koalisyonun değeri, üyelerin tek başına değerleri ile içindeki her ikilinin sinerji bonusunun toplamıdır. Örnek turda tek başına değerler A 6, B 4, C 5; sinerjiler A–B 4, A–C 2, B–C 0'dır. Karakteristik fonksiyon şöyledir: $v(A) = 6$, $v(B) = 4$, $v(C) = 5$, $v(AB) = 6 + 4 + 4 = 14$, $v(AC) = 6 + 5 + 2 = 13$, $v(BC) = 4 + 5 + 0 = 9$, $v(ABC) = 15 + 6 = 21$.

Bu modelde altı sırayı tek tek hesaplamaya gerek yoktur; Shapley değeri kısa bir kurala iner.

Teorem 10.2 (İkili sinerji modelinde Shapley değeri)

Her oyuncunun tek başına değeri a_i, her ikilinin sinerjisi s_{ij} ve bir koalisyonun değeri $v(S) = \Sigma_{i \in S}\, a_i + \Sigma_{\{i,j\} \subseteq S}\, s_{ij}$ olsun. O zaman

$$\varphi_i = a_i + \tfrac{1}{2} \cdot \Sigma_{j \neq i}\, s_{ij}$$

Yani her oyuncu tek başına değerini ve ortağı olduğu her sinerjinin yarısını alır.

İspat. Kanıt, Shapley aksiyomlarıyla yapılır. Oyun, daha basit oyunların toplamı olarak yazılır. Her i için bir "tek başına oyun" tanımlanır: $u_i(S) = a_i$ (i ∈ S ise), aksi hâlde 0. Her {i, j} ikilisi için bir "sinerji oyunu" tanımlanır: $w_{ij}(S) = s_{ij}$ (i ve j ikisi de S'deyse), aksi hâlde 0. Tanımdan $v = \Sigma_i\, u_i + \Sigma_{\{i,j\}}\, w_{ij}$ olduğu görülür.

Tek başına oyun u_i'de i dışındaki her oyuncu kukladır: varlıkları hiçbir koalisyonun değerini değiştirmez. Kukla oyuncu aksiyomuyla onlar 0 alır; verimlilik aksiyomuyla i, $u_i(N) = a_i$'nin tamamını alır.

Sinerji oyunu w_{ij}'de i ve j dışındaki oyuncular kukladır ve 0 alır. i ile j simetriktir: i'yi ve j'yi içermeyen her S için $w_{ij}(S \cup \{i\}) = w_{ij}(S \cup \{j\}) = 0$'dır. Simetri aksiyomuyla eşit pay alırlar; verimlilikle bu paylar toplamı s_{ij} olduğu için her biri $s_{ij}/2$ alır.

Toplamsallık aksiyomuyla v'nin Shapley değeri, bu basit oyunların Shapley değerlerinin toplamıdır. i, kendi tek başına oyunundan a_i ve ortağı olduğu her sinerji oyunundan $s_{ij}/2$ alır. □

Oyuncu	Tek	Sinerji payı	φ	Eşit pay
A	6	4/2 + 2/2 = 3	9	7
B	4	4/2 + 0 = 2	6	7
C	5	2/2 + 0 = 1	6	7
Toplam	15	6	21	21

Tablo 10.16: Oyunun örnek turunda Shapley değeri ve eşit bölüşüm.

Örnek 10.9

Kuralın verdiği $\varphi = (9, 6, 6)$ sonucu, altı geliş sırasıyla denetlenir.

1. **A → B → C:** A 6, B $14 - 6 = 8$, C $21 - 14 = 7$.
2. **A → C → B:** A 6, C $13 - 6 = 7$, B $21 - 13 = 8$.
3. **B → A → C:** B 4, A $14 - 4 = 10$, C 7.
4. **B → C → A:** B 4, C $9 - 4 = 5$, A $21 - 9 = 12$.
5. **C → A → B:** C 5, A $13 - 5 = 8$, B 8.
6. **C → B → A:** C 5, B $9 - 5 = 4$, A 12.

7. **Ortalama.** A: *(6 + 6 + 10 + 12 + 8 + 12)/6 = 54/6 = 9*. B: *(8 + 8 + 4 + 4 + 8 + 4)/6 = 36/6 = 6*. C: *(7 + 7 + 7 + 5 + 5 + 5)/6 = 36/6 = 6*.

Altı katılım sırasının ortalaması da *(9, 6, 6)* verir.

Artık açılıştaki iki soru cevaplanabilir. **Eşit bölüşüm A'yı haksızlığa uğratır mı?** Katkıya göre ölçülürse evet: A'nın Shapley payı 9'dur, eşit bölüşüm ise 7 verir. A, katkısının 2 altında kalır; B ve C ise katkılarının 1'er üstünde alır. **Tek başına en değerli oyuncu her zaman en büyük payı mı almalıdır?** Bu turda A hem tek başına en değerli hem de en büyük payı alan oyuncudur, ama bu bir rastlantıdır. Pay, tek başına değere sinerjilerin yarısı eklenerek bulunur. Tek başına değeri düşük ama büyük sinerjilerin ortağı olan bir oyuncu, tek başına değeri yüksek ama kimseyle sinerjisi olmayan bir oyuncudan fazla alabilir. Örneğin sinerjiler A–B 10, A–C 0, B–C 0 olsaydı B'nin payı *4 + 5 = 9*, C'nin payı 5 olurdu; tek başına değeri daha yüksek olan C daha az alırdı.

Oyun, sonucu bir **eşit bölüşüm hatası** ile ölçer: eşit bölüşümün Shapley değerinden toplam sapması. Örnek turda bu hata *|7 − 9| + |7 − 6| + |7 − 6| = 2 + 1 + 1 = 4*'tür.

Gerçekte insanlar nasıl paylaşır? Deneyler kuramın tek başına yeterli olmadığını gösterir. Üç kişilik koalisyon oyunlarında başkaları için “adil” paylaşım öneren deneklerin seçimleri, Shapley değeri ile eşit bölüşümün bir karışımı olarak en iyi açıklanmıştır (de Clippel ve Rozen, 2022). İnsanlar katkıyı önemser, ama eşitliği de önemser. Örnekte yarı yarıya bir karışım *½ · (9, 6, 6) + ½ · (7, 7, 7) = (8; 6,5; 6,5)* verir. Katkıya göre paylaşım (Shapley) ile eşit paylaşım çoğu zaman ayrışır; hangisinin adil sayılacağı bağlama bağlıdır. Bir ortaklıkta yatırımcılar katkıya göre, bir ailede ise kardeşler eşit paylaşımı adil bulabilir.

10.4.9 Çekirdek

Shapley değeri bir adalet ölçütüdür: her oyuncuya katkısının ortalamasını verir. Ancak adil bir paylaşım kararlı olmayabilir. Bir alt grup büyük koalisyondan ayrılıp kendi değerini paylaşınca daha fazla alıyorsa, paylaşım ne kadar adil görünürse görünsün o grup ayrılır. Kararlılığı ölçen kavram çekirdektir.

Tanım 10.12

Bir S koalisyonu, üyelerinin x paylaşımında aldığı toplam *v(S)*'den küçükse, yani $\Sigma_{i \in S}\, x_i < v(S)$ ise, x'i **engeller** (*blocks*): S'nin üyeleri ayrılıp *v(S)*'yi kendi aralarında paylaşarak hepsi daha iyi duruma geçebilir. x paylaşımı **çekirdektedir** (*core*) ancak ve ancak

$$\Sigma_{i \in N}\, x_i = v(N) \text{ ve her } S \subseteq N \text{ için } \Sigma_{i \in S}\, x_i \geq v(S)$$

sağlanıyorsa; yani hiçbir koalisyon x'i engellemiyorsa.

"Engelleme" fikri şöyle somutlaşır. S'nin üyeleri x'te toplam 35 alıyor, ama ayrılırlarsa birlikte 50 kazanabiliyorsa, 50'yi öyle paylaşabilirler ki her biri x'te aldığından fazlasını alır (ör. fazladan 15'i eşit bölerek). Bu yüzden S'nin x'i kabul etmesi için bir neden yoktur. Çekirdek, hiçbir grubun böyle bir itirazının olmadığı paylaşımların kümesidir.

Tanımdaki eşitsizlikler tek oyunculu S'ler için bireysel rasyonelliği, büyük koalisyon için verimliliği içerir. Bu yüzden çekirdekteki her ödeme vektörü bir paylaşımdır. Lojistik oyununda çekirdeği tanımlayan kısıtlar şunlardır:

Tekli kısıtlar

$$x_A \geq 10, x_B \geq 20, x_C \geq 30$$

İkili kısıtlar

$$x_A + x_B \geq 50, x_A + x_C \geq 60, x_B + x_C \geq 70$$

Verimlilik

$$x_A + x_B + x_C = 120$$

Çekirdek bir **doğrusal eşitsizlikler sistemidir** (Gillies, 1959). Doğrusal eşitsizliklerin çözüm kümesi bir çokgen (genel olarak çokyüzlü) bölgedir. Bu yüzden çekirdek boş olabilir, tek bir noktadan oluşabilir ya da bir bölge olabilir. Boş çekirdek ve bölge biçimli çekirdek bu kesimde örneklerle görülür.

İkili kısıtların kısa yolu. Verimlilik eşitliği kullanılarak ikili kısıtlar tek oyunculu üst sınırlara çevrilir. $x_A + x_B \geq 50$ ve $x_A + x_B = 120 - x_C$ olduğundan $120 - x_C \geq 50$, yani $x_C \leq 70$ bulunur. Aynı yolla $x_B \leq 120 - 60 = 60$ ve $x_A \leq 120 - 70 = 50$ çıkar. Genel olarak üç oyunculu bir oyunda her oyuncu en çok $v(N) - v(N \setminus \{i\})$ alabilir. Bu değer, i'nin büyük koalisyona son katılan olarak kattığı değerdir; i bundan fazlasını alırsa geri kalan ikisi ayrılıp kendi değerlerini alır.

Shapley ile çekirdek. Shapley değeri adalet, çekirdek kararlılık ölçütüdür; ikisi her zaman örtüşmez. Shapley değeri çekirdeğin dışında kalabilir, çekirdek boşken de Shapley değeri her zaman vardır. Ancak bir oyun sınıfında ikisi uyumludur.

Tanım 10.13

Bir koalisyon oyununda her oyuncunun marjinal katkısı, katıldığı koalisyon büyüdükçe azalmıyorsa oyun **dışbükeydir** (*convex game*): her i ve $S \subseteq T \subseteq N \setminus \{i\}$ için $v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T)$. Dışbükey bir oyunda bir tür "kartopu etkisi" vardır: grup büyüdükçe yeni gelen üyenin kattığı değer artar ya da sabit kalır.

Teorem 10.3 (Shapley, 1971)

Dışbükey bir oyunda çekirdek boş değildir ve Shapley değeri her zaman çekirdektedir.

Teoremin kanıtı uzundur, ama lojistik oyununun dışbükey olduğu kolayca denetlenir. Tablo 10.17 her oyuncunun dört koalisyona kattığı değeri listeler. Her satırda değerler soldan sağa azalmaz: koalisyon büyüdükçe katkı küçülmez. Örneğimiz dışbükey bir oyundur; bu yüzden Shapley paylaşımı *(30, 40, 50)* çekirdektedir. Bu, 10.4.11'deki kısıt denetimiyle de doğrulanır.

Oyuncu	∅'ye	tek bir oyuncuya	tek bir oyuncuya	ikiliye
A	10	{B}: 30	{C}: 30	{B, C}: 50
B	20	{A}: 40	{C}: 40	{A, C}: 60
C	30	{A}: 50	{B}: 50	{A, B}: 70

Tablo 10.17: Lojistik oyununda marjinal katkılar. Koalisyon büyüdükçe katkı azalmaz; oyun dışbükeydir.

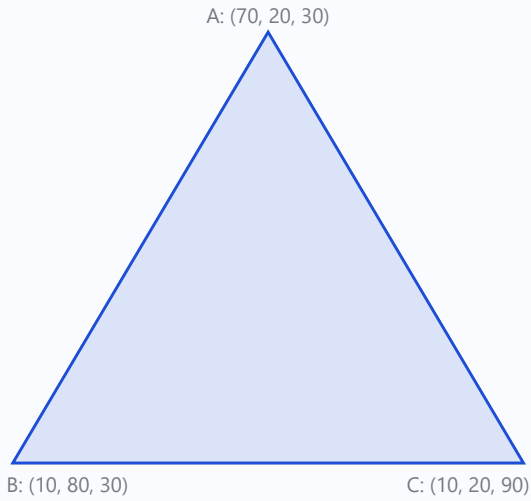
Dışbükeylik yeterli bir koşuldur, gerekli değildir: dışbükey olmayan bir oyunda da Shapley değeri çekirdekte olabilir. Çekirdeğin boş olup olmadığını kesin olarak belirleyen koşul **Bondareva–Shapley teoremidir** (Bondareva, 1963; Shapley, 1967): çekirdek ancak ve ancak oyun **dengeli** (*balanced*) ise boş değildir. Genel tanım teknik olsa da üç oyunculu oyunlarda koşul somut bir hâl alır. Üç oyunculu bir oyunda çekirdek ancak ve ancak şu iki koşul sağlanırsa boş değildir:

- Her koalisyon yapısının toplam değeri $v(N)$'yi aşmaz: $v(N) \geq v(A) + v(B) + v(C)$ ve her i için $v(N) \geq v(\{i\}) + v(N \setminus \{i\})$.
- İkililerin değerlerinin yarısının toplamı $v(N)$'yi aşmaz: $v(N) \geq \tfrac{1}{2} \cdot [v(AB) + v(AC) + v(BC)]$.

İkinci koşulun neden gerekli olduğu kolayca görülür: çekirdekteki bir x için üç ikili kısıt toplanırsa $2(x_A + x_B + x_C) \geq v(AB) + v(AC) + v(BC)$ bulunur; sol taraf $2v(N)$'dir. Lojistik oyununda $\tfrac{1}{2} \cdot (50 + 60 + 70) = 90 \leq 120$ ve her yapının değeri en çok 80'dir; iki koşul da sağlanır.

10.4.10 Çekirdeğin geometrisi

Üç oyunculu bir oyunda paylaşımlar bir üçgen üzerinde çizilebilir. Verimlilik eşitliği $x_A + x_B + x_C = 120$ sayesinde üç sayıdan ikisi seçilince üçüncüsü belirlenir; bu yüzden paylaşımlar iki boyutlu bir bölge oluşturur.

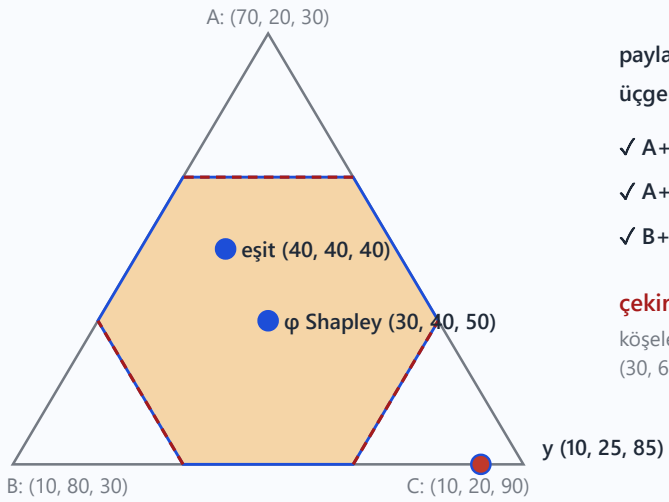


Şekil 10.13: Bireysel rasyonel ve verimli bütün paylaşımlar bir üçgen oluşturur. Her köşede bir oyuncu, tek başına değerlerin üstünde kalan 60'ın tamamını alır: A köşesi (70, 20, 30), B köşesi (10, 80, 30), C köşesi (10, 20, 90).

Şekil 10.13 başlangıç durumunu gösterir. Her oyuncu önce tek başına değerini alır: $10 + 20 + 30 = 60$. Geriye $120 - 60 = 60$ birimlik bir fazla kalır. Bu fazlanın üç oyuncu arasında her bölünüşü üçgenin bir noktasıdır. Köşelerde fazlanın tamamı tek bir oyuncuya gider. Örneğin C köşesinde C, kendi 30'una ek olarak 60'ın hepsini alır ve $(10, 20, 90)$ paylaşımı elde edilir.

Çekirdek, bu üçgenden ikili kısıtların kestiği parçalar çıkarılarak bulunur.

1. **{A, B} kısıtı.** $x_A + x_B \geq 50 \Leftrightarrow x_C \leq 70$. C'ye 70'ten fazlası verilirse A ve B birlikte 50'den az alır; ayrılıp kendi 50'lerini alırlar. Bu kısıt, C köşesinin çevresindeki küçük üçgeni keser.
2. **{A, C} kısıtı.** $x_A + x_C \geq 60 \Leftrightarrow x_B \leq 60$. B köşesi kesilir.
3. **{B, C} kısıtı.** $x_B + x_C \geq 70 \Leftrightarrow x_A \leq 50$. A köşesi de kesilir.



Şekil 10.14: Üç köşe kesildikten sonra kalan turuncu altıgen çekirdektir. Shapley değeri $\varphi = (30, 40, 50)$ ve eşit

bölüşüm *(40, 40, 40)* çekirdektedir (mavi noktalar); *y = (10, 25, 85)* dışarıdadır (kırmızı nokta).

Şekil 10.14'te kalan bölge altı köşeli bir çokgendir. Köşeleri, her oyuncunun payının alt sınırında (tek başına değer) ya da üst sınırında ($v(N) - v(N \setminus \{i\})$) olduğu noktalardır: $x_A \in [10, 50]$, $x_B \in [20, 60]$, $x_C \in [30, 70]$. Bir oyuncu alt sınırında, bir başkası üst sınırındaysa üçüncünün payı verimlilikten belirlenir. Örneğin A 10, B 60 alırsa C $120 - 70 = 50$ alır. Altı köşe şunlardır: *(10, 40, 70)*, *(10, 60, 50)*, *(30, 20, 70)*, *(30, 60, 30)*, *(50, 20, 50)*, *(50, 40, 30)*.

Şekildeki üç nokta çekirdeğin üç farklı durumunu gösterir. Shapley değeri altıgenin içindedir: üç sınırın hiçbirine dayanmaz. Eşit bölüşüm de içeridedir: $40 \le 50$, $40 \le 60$, $40 \le 70$. C'nin önerisi y ise C köşesinin kesilen parçasındadır: $x_C = 85 > 70$. Geometri, "hangi koalisyon engeller?" sorusunu da cevaplar: y'nin dışarıda kaldığı kenar {A, B} kısıtının kenarıdır; y'yi engelleyen koalisyon {A, B}'dir.

10.4.11 Çekirdek kontrolü adım adım

Bir paylaşımın çekirdekte olup olmadığı, yedi kısıtın her biri tek tek denetlenerek bulunur. Bir paylaşım çekirdekteyse hiçbir koalisyon ayrılıp kendi değerini ($v(S)$) alarak daha iyi duruma geçemez.

Örnek 10.10

Shapley paylaşımı *x = (30, 40, 50)* ile C'nin önerisi *y = (10, 25, 85)* yedi kısıtla sınanır.

1. **{A}: v(A) = 10.** x için $30 \ge 10$ ✓; y için $10 \ge 10$ ✓ (eşitlik yeterlidir).
2. **{B}: v(B) = 20.** x için $40 \ge 20$ ✓; y için $25 \ge 20$ ✓.
3. **{C}: v(C) = 30.** x için $50 \ge 30$ ✓; y için $85 \ge 30$ ✓.
4. **{A, B}: v(AB) = 50.** x için $30 + 40 = 70 \ge 50$ ✓; y için $10 + 25 = 35 \ge 50$ ✗. {A, B} birlikte ayrılırsa 50 alır; y'yi engeller.
5. **{A, C}: v(AC) = 60.** x için $30 + 50 = 80 \ge 60$ ✓; y için $10 + 85 = 95 \ge 60$ ✓.
6. **{B, C}: v(BC) = 70.** x için $40 + 50 = 90 \ge 70$ ✓; y için $25 + 85 = 110 \ge 70$ ✓.
7. **{A, B, C}: toplam = 120.** x için $30 + 40 + 50 = 120$ ✓; y için $10 + 25 + 85 = 120$ ✓.

Sonuç: Shapley paylaşımı x çekirdektedir. y ise çekirdek dışındadır, çünkü onu {A, B} engeller. Oyun dışbükey olduğundan Shapley değerinin çekirdekte çıkması Teorem 10.3'ün beklentisiyle de uyumludur.

Koalisyon S	Kısıt	x = (30, 40, 50)	y = (10, 25, 85)
{A}	Σ ≥ 10	30 ✓	10 ✓
{B}	Σ ≥ 20	40 ✓	25 ✓
{C}	Σ ≥ 30	50 ✓	85 ✓
{A, B}	Σ ≥ 50	70 ✓	35 ✗

Koalisyon S	Kısıt	x = (30, 40, 50)	y = (10, 25, 85)
{A, C}	Σ ≥ 60	80 ✓	95 ✓
{B, C}	Σ ≥ 70	90 ✓	110 ✓
{A, B, C}	toplam = 120	120 ✓	120 ✓

Tablo 10.18: Yedi çekirdek kısıtı ve iki paylaşımın denetimi.

Tablo 10.18 iki paylaşımın denetimini özetler. Tek bir X paylaşımı çekirdek dışına atmaya yeter. Bir paylaşımı çekirdekten çıkarmak için engelleyen bir koalisyon bulmak yeterlidir; çekirdekte olduğunu göstermek için ise bütün kısıtların denetlenmesi gerekir.

Kısa yolla denetim. 10.4.9'daki üst sınırlar ($x_A \le 50, x_B \le 60, x_C \le 70$) ve alt sınırlar (tek başına değerler) bilinirse, toplamı 120 olan bir paylaşım altı sayı karşılaştırılarak denetlenir. Örneğin orantılı paylaşım *(20, 40, 60)* için: $10 \le 20 \le 50$, $20 \le 40 \le 60$, $30 \le 60 \le 70$. Hepsi sağlanır; orantılı paylaşım da çekirdektedir. Lojistik oyununda çekirdek geniştir ve Tablo 10.13'teki dört önerinin üçü (eşit, orantılı, Shapley) çekirdektedir. Çekirdek bu oyunda tek bir paylaşım seçmez; yalnız kabul edilemez paylaşımları eler.

10.4.12 Çekirdeği boş bir oyun: çoğunluk oyunu

Lojistik oyununda çekirdek geniş bir altıgendi. Her oyunda böyle bir kararlı paylaşım bulunmaz.

Örnek 10.11

Üç ajan 1 birimlik bir ödülü paylaşacaktır. Herhangi iki ajan anlaşırsa ödülü alır. Karakteristik fonksiyon şöyledir: $v(S) = 1$ ($|S| \ge 2$ ise), aksi hâlde 0. Bu oyuna **çoğunluk oyunu** denir.

1. **Kısıtları yaz.** Tekli kısıtlar $x_i \ge 0$'dır. İkili kısıtlar: $x_A + x_B \ge 1$, $x_A + x_C \ge 1$, $x_B + x_C \ge 1$. Verimlilik: $x_A + x_B + x_C = 1$.
2. **İkili kısıtları topla.** Her oyuncu iki kısıtta geçer. Toplam $2(x_A + x_B + x_C) \ge 3$, yani $x_A + x_B + x_C \ge 3/2$'dir.
3. **Çelişki.** Verimlilik toplamın tam olarak 1 olmasını ister. Ama $1 < 3/2$'dir. Hiçbir paylaşım üç ikili kısıtı aynı anda sağlayamaz: **çekirdek boştur**.
4. **Shapley değeri.** Üç oyuncu simetriktir; simetri aksiyomuyla eşit alırlar, verimlilikle toplam 1'dir. $\varphi = (1/3, 1/3, 1/3)$. Bu paylaşım adildir, ama her çift yalnız $2/3$ alır ve ayrılırsa 1 alabilir: kararlı değildir.

Çekirdeğin boş olmasının ne anlama geldiği bir pazarlık dizisiyle görülür. Diyelim ki A ve B anlaşıp ödülü *(½, ½, 0)* paylaştı. Dışarıda kalan C, A'ya *(0,6; 0; 0,4)* önerir: A 0,5 yerine 0,6, C 0 yerine 0,4 alır; ikisi de kazanır. Bu kez dışarıda kalan B, C'ye *(0; 0,3; 0,7)* önerir: C 0,4 yerine 0,7, B 0 yerine 0,3 alır.

Şimdi de A dışarıdadır ve B'ye daha iyi bir teklif yapabilir. Her paylaşımda en az bir çift toplam en çok *2/3* alır (çünkü üç çiftin toplamı 2'dir) ve bu çift üçüncüyü dışlayıp 1'i paylaşmak ister. Pazarlık hiç bitmez.

Ders: kararlı paylaşım her oyunda yoktur. Bu durumda kuralı gevşetmek gerekir. "Hiçbir grup ayrılarak hiçbir şey kazanamasın" yerine "hiçbir grup ayrılarak çok fazla kazanamasın" denir. Bu fikir **ε-çekirdek** kavramıyla ifade edilir: hiçbir koalisyon ayrılarak ε'dan fazla kazanamaz, yani her S için $\Sigma_{i \in S} x_i \geq v(S) - \varepsilon$. Çoğunluk oyununda ikili kısıtlar $x_i + x_j \geq 1 - \varepsilon$ olur. Üçü toplanınca $2 \geq 3(1 - \varepsilon)$, yani $\varepsilon \geq 1/3$ çıkar. $\varepsilon = 1/3$ için $(1/3, 1/3, 1/3)$ paylaşımı her ikili kısıtı tam eşitlikle sağlar: $2/3 = 1 - 1/3$. ε'u en küçük yapan paylaşımlar kümesine **en küçük çekirdek** denir. Bu fikri adım adım sürdürerek her oyunda tek bir paylaşım seçen çözüme **çekirdekçik** (*nucleolus*) denir; çoğunluk oyununda çekirdekçik de $(1/3, 1/3, 1/3)$'tür.

10.4.13 Ağırlıklı oylama ve güç endeksleri

Koalisyon oyunlarının önemli bir uygulaması oylamadır. Bir mecliste partilerin sandalye sayıları, bir şirkette ortakların hisse oranları, bir ağda düğümlerin oy hakları farklı olabilir. Ağırlığı fazla olanın gücü de fazla mıdır? Cevap "her zaman değil"dir.

Tanım 10.14

Bir **ağırlıklı oylama oyunu** $[q; w_1, \ldots, w_n]$ ile gösterilir. w_i, i'nin oy ağırlığı; q, kararın geçmesi için gereken **kotadır**. Bir koalisyonun ağırlıkları toplamı kotaya ulaşırsa koalisyon kazanır ve $v(S) = 1$; ulaşmazsa kaybeder ve $v(S) = 0$.

Bir geliş sırasında, kendisinden öncekilerle birlikte toplamı kotanın altından kotaya ya da üstüne çıkaran oyuncuya **kilit oyuncu** (*pivotal player*) denir. Kazanan bir koalisyondan ayrılınca koalisyonu kaybettiren üyeye **kritik üye** (*critical / swing member*) denir.

Bir oyuncunun oylamadaki gücü, kararları ne sıklıkla belirlediğiyle ölçülür. Bunu ölçen iki klasik endeks vardır.

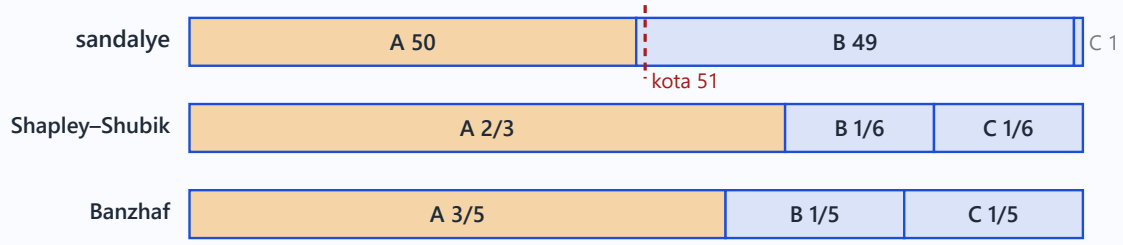
Shapley–Shubik endeksi (Shapley ve Shubik, 1954), ağırlıklı oylama oyununun Shapley değeridir. Oylama oyununda marjinal katkı yalnız 0 ya da 1 olabilir: oyuncu kotayı geçiren kişiyse katkısı 1, değilse 0'dır. Bu yüzden i'nin Shapley değeri, i'nin kilit olduğu sıraların oranıdır.

Banzhaf endeksi ise sıralara değil koalisyonlara bakar. Her kazanan koalisyonda hangi üyelerin kritik olduğu sayılır. Bir oyuncunun Banzhaf endeksi, kritik üyelik sayısının bütün oyuncuların kritik üyelik sayıları toplamına oranıdır.

Örnek 10.12

Bir mecliste 100 sandalye vardır: A partisinin 50, B partisinin 49, C partisinin 1 sandalyesi vardır. Karar için salt çoğunluk, yani 51 oy gerekir. Oyun $[51; 50, 49, 1]$'dir.

1. **Kazanan koalisyonlar.** Hiçbir parti tek başına 51'e ulaşamaz. İkililer: $A + B = 99$ ✓, $A + C = 51$ ✓, $B + C = 50$ ✗. Üçü birlikte 100 ✓. Kazanan koalisyonlar {A, B}, {A, C}, {A, B, C}'dir.
2. **Shapley–Shubik: altı sıra.** A → B → C: A ile toplam 50, B ile 99; kilit B. A → C → B: A ile 50, C ile 51; kilit C. B → A → C: B ile 49, A ile 99; kilit A. B → C → A: B ile 49, C ile 50, A ile 100; kilit A. C → A → B: C ile 1, A ile 51; kilit A. C → B → A: C ile 1, B ile 50, A ile 100; kilit A.
3. **Shapley–Shubik endeksi.** A dört sırada, B bir sırada, C bir sırada kilittir: *(4/6, 1/6, 1/6) = (2/3, 1/6, 1/6)*.
4. **Banzhaf: kritik üyeler.** {A, B}'de A ayrılırsa 49, B ayrılırsa 50 kalır; ikisi de kritiktir. {A, C}'de A ayrılırsa 1, C ayrılırsa 50 kalır; ikisi de kritiktir. {A, B, C}'de A ayrılırsa 50 kalır (kaybeder); B ayrılırsa 51 kalır (yine kazanır); C ayrılırsa 99 kalır (yine kazanır). Yalnız A kritiktir.
5. **Banzhaf endeksi.** Kritik üyelik sayıları A 3, B 1, C 1; toplam 5. Endeks *(3/5, 1/5, 1/5)*'tir.



Şekil 10.15: Meclis *[51; 50, 49, 1]*'de sandalye payları ile güç endeksleri. B'nin 49 sandalyesi ile C'nin 1 sandalyesi aynı gücü getirir: Shapley–Shubik endeksinde 1/6, Banzhaf endeksinde 1/5.

Şekil 10.15 sonucun çarpıcılığını gösterir. B ile C arasında 48 sandalye fark vardır, ama iki parti eşit güçtedir. Neden? Kazanmanın tek yolu A'yı içeren bir koalisyondur ve A'ya kotayı geçmek için tek bir sandalye yeter. B'nin fazladan 48 sandalyesi hiçbir koalisyonun sonucunu değiştirmez. Güç, ağırlıkla değil, oyuncunun sonucu değiştirebildiği durumların sayısıyla ölçülür. **Ağırlık ≠ güç.**

İki endeks farklı sayılar verir, çünkü farklı şeyleri sayar. Shapley–Shubik endeksi koalisyonların hangi sırayla kurulduğunu önemser ve büyük koalisyonun oluşum sürecini modeller. Banzhaf endeksi sıraya bakmaz; her kazanan koalisyonu eşit sayar. İkisi de A'yı en güçlü, B ile C'yi eşit bulur. Bu, sıralama açısından iki endeksin bu oyunda anlaştığını gösterir.

Tarihsel bir örnek: 1958 AET Bakanlar Konseyi. Avrupa Ekonomik Topluluğu'nun kuruluşunda Bakanlar Konseyi'nde oylar şöyle dağıtılmıştı: Fransa, Almanya ve İtalya 4'er; Belçika ve Hollanda 2'şer; Lüksemburg 1 oy. Nitelikli çoğunluk için 12 oy gerekiyordu: oyun *[12; 4, 4, 4, 2, 2, 1]*'dir. Lüksemburg'un oyu hiçbir koalisyonun sonucunu değiştiremiyordu. Nedeni bir çift-tek sayı gözlemidir: Lüksemburg dışındaki bütün ağırlıklar çifttir, bu yüzden Lüksemburg'suz her koalisyonun toplamı çifttir. Lüksemburg'un 1 oyu bir koalisyonu kotanın altından kotaya ancak toplam tam 11 iken taşı-

yabilirdi; ama 11 tek sayıdır ve hiçbir koalisyonun toplamı 11 olamaz. Bu yüzden Lüksemburg hiçbir sırada kilit değildir ve gücü 0'dır. Diğer ülkelerin Shapley–Shubik endeksi büyük üçlü için 7/30, Belçika ve Hollanda için 3/20'dir.

10.4.14 Kod izleme: guc_endeksi()

Shapley–Shubik endeksi, Shapley değerinin kodundan daha basit bir kodla hesaplanır: koalisyon değerine bakmak yerine ağırlıklar toplanır ve kotayı geçen oyuncu kaydedilir. Fonksiyon üç girdi alır: oyuncu listesi `N`, ağırlık sözlüğü `w = {"A": 50, "B": 49, "C": 1}` ve kota `q = 51`.

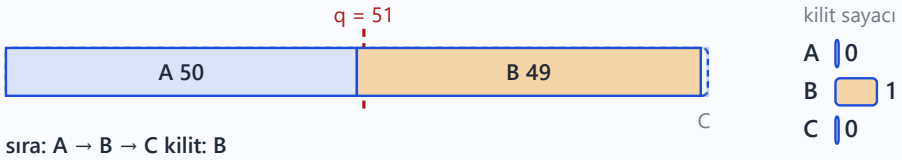
```
from itertools import permutations
def guc_endeksi(N, w, q):
  kilit = {i: 0 for i in N}
  siralar = list(permutations(N))
  for sira in siralar:
    toplam = 0
    for i in sira:
      toplam += w[i]
      if toplam >= q:
        kilit[i] += 1
        break
  return {i: kilit[i] / len(siralar) for i in N}
```

Kod satır satır şöyle çalışır.

1. `kilit = {i: 0 for i in N}`: Her oyuncunun kilit olduğu sıra sayısı 0'dan başlar.
2. `siralar = list(permutations(N))`: Bütün geliş sıraları (3! = 6) üretilir.
3. `toplam = 0`: Her sıra boş koalisyonla, yani 0 oyla başlar.
4. `toplam += w[i]`: Gelen oyuncunun ağırlığı toplama eklenir.
5. `if toplam >= q`: Toplam kotaya ilk kez ulaştıysa gelen oyuncu kilittir; `kilit[i]` bir artırılır.
6. `break`: Bu sırada kilit bulundu; sıranın geri kalanı önemli değildir. İç döngü kesilir ve bir sonraki sıraya geçilir. Her sırada tam olarak bir kilit oyuncu vardır.
7. `return {i: kilit[i] / len(siralar) for i in N}`: Kilit sayıları sıra sayısına bölünür.

Sıra	Toplamın gelişimi	Kilit	kilit (A, B, C)
A → B → C	50 < 51 → 99 ≥ 51	B	(0, 1, 0)
A → C → B	50 < 51 → 51 ≥ 51	C	(0, 1, 1)
B → A → C	49 < 51 → 99 ≥ 51	A	(1, 1, 1)
B → C → A	49 → 50 < 51 → 100 ≥ 51	A	(2, 1, 1)
C → A → B	1 < 51 → 51 ≥ 51	A	(3, 1, 1)
C → B → A	1 → 50 < 51 → 100 ≥ 51	A	(4, 1, 1)

Tablo 10.19: `guc_endeksi(N, w, 51)` izlemesi.



Şekil 10.16: İlk sıra A → B → C. A'nın 50 oyu kotanın (kesikli kırmızı çizgi, 51) altında kalır; B gelince toplam 99 olur ve kota geçilir. Kilit B'dir (turuncu); sayaç B = 1.



Şekil 10.17: İzlemenin sonu: kilit sayaçları A 4, B 1, C 1.

Tablo 10.19, Şekil 10.16 ve Şekil 10.17 izlemeyi gösterir. Sonuçta `kilit = {A: 4, B: 1, C: 1}` olur ve fonksiyon *(2/3, 1/6, 1/6)* döndürür. B'nin 49 sandalyesi C'nin 1 sandalyesinden fazla güç getirmez. İzlemenin `break` satırı önemlidir: B → C → A sırasında B ve C gelince toplam 50 olur; A gelince 100'e çıkar ve A kilittir. `break` olmasaydı kod, kota geçildikten sonra gelen oyunculara da kilit sayardı ve sonuç yanlış olurdu.

10.4.15 Koalisyon oyunlarında sık hatalar

Sık yapılan hata

Marjinal katkıyı koalisyon değeriyle karıştırmak. A → B → C sırasında B'nin katkısı *v(AB) = 50* değil, *v(AB) − v(A) = 40*'tır. Satır toplamının *v(N)*'yi vermesi bu hatayı yakalar: 10 + 50 + 70 = 130 ≠ 120.

Shapley değerini tek bir sıradan hesaplamak. Tek bir sıra son gelene büyük pay verir. A → B → C sırası C'ye 70 verir; Shapley değeri ise 50'dir. Bütün sıraların ortalaması alınmalıdır.

Çekirdekte yalnız tekli kısıtları denetlemek. *y = (10, 25, 85)* tekli kısıtların hepsini sağlar, ama {A, B} onu engeller. Bireysel rasyonellik çekirdeğin yalnız bir parçasıdır.

Adil olanı kararlı sanmak. Shapley değeri her zaman vardır, ama çekirdekte olmak zorunda değildir. Çoğunluk oyununda *(1/3, 1/3, 1/3)* adildir, ama çekirdek boştur ve her çift ayrılmak ister.

Ağırlığı güç sanmak. *[51; 50, 49, 1]*'de ağırlık payları *(0,50; 0,49; 0,01)*'dir, Shapley–Shubik endeksi ise *(2/3, 1/6, 1/6)*'dır. Lüksemburg örneğinde 1 oy hiç güç getirmez.

İki güç endeksini karıştırmak. *(3/5, 1/5, 1/5)* Banzhaf endeksidir (kritik üyelikler), *(2/3, 1/6, 1/6)* Shapley–Shubik endeksidir (kilit sıralar). Soru hangi endeksi istiyorsa o hesaplanmalıdır.

10.5 Kamu malı oyunu

Koalisyon oyunlarında bağlayıcı anlaşma yapılabildiği varsayılıyordu. Birçok çoklu ajan sisteminde ise böyle bir anlaşma yoktur: her ajan ortak bir kaynağa ne kadar katkı vereceğine kendi başına karar verir ve kimse onu katkı vermeye zorlayamaz. Kamu malı oyunu, Mahkûmun İkilemi'nin çok oyunculu sürümüdür. Oyun, ortak iyinin neden kendiliğinden ortaya çıkmadığını sayılarla gösterir.

10.5.1 Kamu malı nedir?

Tanım 10.15

Kamu malı (*public good*), iki özelliği olan bir maldır:

- **Dışlanamazlık:** kimse maldan yararlanmaktan dışlanamaz. Katkı vermeyen de yararlanır.
- **Rakip olmama:** bir kişinin kullanımı diğerininkini azaltmaz.

Açık kaynak yazılım iyi bir örnektir. Kodu yazan da yazmayan da yazılımı indirip kullanabilir (dışlanamazlık). Bir kişinin yazılımı kullanması başkasının kullanımını azaltmaz (rakip olmama). Temiz hava ve ortak bir veri kümesi de kamu malıdır. Bir ekmek ise kamu malı değildir: parasını ödemeyen alamaz, bir kişinin yediği ekmeği başkası yiyemez.

Kamu mallarının sorunu, dışlanamazlıktan doğar. Mal herkese açık olduğu için her kişi şunu düşünebilir: “Ben katkı vermesem de başkaları verir, ben de yararlanırım.” Herkes böyle düşünürse kimse katkı vermez ve mal hiç üretilmez.

10.5.2 Oyunun kuralları

Tanım 10.16

Kamu malı oyununda (*public goods game*) n oyuncunun her biri e jetonla başlar. Oyuncu i, ortak havuza 0 ile e arasında k_i jeton katar. Havuzdaki jetonlar r çarpanıyla büyür ve n oyuncuya eşit dağıtılır. Oyuncu i'nin kazancı

$$u_i = e - k_i + (r/n) \cdot \Sigma_j k_j$$

olur. Burada $\Sigma_j k_j$, havuza giren toplam jetondur. r/n, bir jetonun katkı yapana geri dönen kısmıdır ve **kişi başı marjinal getiri** (*marginal per-capita return*, MPCR) adını alır.

Kazanç formülü iki parçadan oluşur. $e - k_i$, oyuncunun cebinde tuttuğu jetonlardır. $(r/n) \cdot \Sigma k_j$, havuzdan aldığı paydır. Bu pay, oyuncunun kendi katkısına değil, herkesin toplam katkısına bağlıdır. Katkı vermeyen oyuncu da havuzdan aynı payı alır; dışlanamazlık formülün içindedir.

Bölümün örneği şu sayıları kullanır: $n = 4$ oyuncu, her biri $e = 10$ jeton, çarpan $r = 1{,}6$. Kişi başı marjinal getiri $r/n = 1{,}6/4 = 0{,}4$'tür. Bir jetonluk katkının iki ayrı sonucu vardır.

Bireysel sonuç
: 1 jeton katkı veren oyuncunun cebinden 1 jeton çıkar, havuzdan kendisine 0,4 döner. Net etki $-1 + 0{,}4 = -0{,}6$'dır.

Sosyal sonuç
: 1 jeton havuzda 1,6'ya dönüşür ve dört oyuncuya 0,4'er dağıtılır. Grubun toplam kazancı $-1 + 1{,}6 = +0{,}6$ artar.

Bireysel ve sosyal sonucun işaretleri terstir. Katkı oyuncuya zarar verir, ama gruba fayda sağlar. İkilemin özü budur.

Örnek 10.13

Dört senaryoda kazançlar hesaplanır ($n = 4$, $e = 10$, $r = 1{,}6$).

1. **Herkes 10 verir.** Havuz $40 \cdot 1{,}6 = 64$, kişi başı pay $64/4 = 16$. Herkes $10 - 10 + 16 = 16$ alır. Grubun toplamı 64.
2. **Kimse vermez.** Havuz 0. Herkes cebindeki 10'u tutar: kişi başı 10, toplam 40.
3. **O1 beleşçi, diğerleri 10 verir.** Havuza 30 jeton girer: $30 \cdot 1{,}6 = 48$, kişi başı 12. O1 $10 - 0 + 12 = 22$, diğerleri $10 - 10 + 12 = 12$ alır. Toplam $22 + 3 \cdot 12 = 58$.
4. **Yalnız O1 10 verir.** Havuza 10 jeton girer: $10 \cdot 1{,}6 = 16$, kişi başı 4. O1 $0 + 4 = 4$, diğerleri $10 + 4 = 14$ alır. Toplam $4 + 3 \cdot 14 = 46$.

Senaryo	Σk	Havuz × 1,6	O1	Diğerleri (her biri)	Toplam
Herkes 10	40	64	16	16	64
Kimse	0	0	10	10	40
O1 beleşçi	30	48	22	12	58
Yalnız O1	10	16	4	14	46

Tablo 10.20: Dört senaryoda kazançlar ($n = 4$, $e = 10$, $r = 1{,}6$).

Tablo 10.20 ikilemi açıkça gösterir. O1 için en iyi durum, başkaları verirken kendisinin vermemesidir (22). En kötü durum, başkaları vermezken kendisinin vermesidir (4). Grup için ise en iyi durum herkesin vermesidir (64), en kötü durum kimsenin vermemesidir (40). Toplam sütunu bir kural daha gösterir: grubun toplamı $n \cdot e + (r - 1) \cdot \Sigma k = 40 + 0{,}6 \cdot \Sigma k$'dır. Havuza giren her jeton grubun toplamını 0,6 artırır.

10.5.3 Neden beleşçilik?

O1'in kazancı, diğer üç oyuncunun toplam katkısı K ile yazılırsa

$$u_1 = 10 - k_1 + 0{,}4 \cdot (k_1 + K) = 10 + 0{,}4K - 0{,}6\, k_1$$

olur. O1 bir jeton daha verirse cebinden 1 jeton çıkar, havuzdan 0,4 geri gelir: net −0,6. Bu değer K'ye bağlı değildir. **Diğerleri ne yaparsa yapsın, her jetonluk katkı O1'e 0,6 kaybettirir.**

Teorem 10.4 (Kamu malı oyununda baskın strateji)

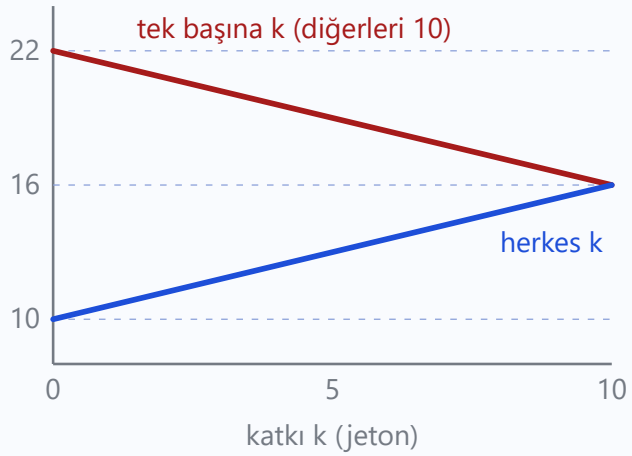
$r/n < 1$ ise kamu malı oyununda $k_i = 0$ her oyuncu için baskın stratejidir ve oyunun tek Nash dengesi herkesin 0 vermesidir. $r > 1$ ise bu denge Pareto optimal değildir: herkesin e vermesi her oyuncuya daha fazla kazanç sağlar.

İspat. Oyuncu i'nin kazancı, diğerlerinin toplam katkısı K ile $u_i = e + (r/n)K - (1 - r/n)\, k_i$ biçiminde yazılır. $r/n < 1$ olduğu için k_i'nin katsayısı $-(1 - r/n)$ negatiftir. Bu yüzden K ne olursa olsun, k_i'yi azaltmak kazancı artırır ve en büyük kazanç $k_i = 0$'da elde edilir. Bu, 0 vermenin baskın strateji olduğunu gösterir.

Her oyuncunun baskın stratejisi 0 olduğu için herkesin 0 verdiği profil bir Nash dengesidir. Başka denge yoktur: bir profilde pozitif katkı veren bir oyuncu varsa, o oyuncu katkısını 0'a indirerek kazancını kesin olarak artırır.

Dengede her oyuncu e alır. Herkes e verirse havuz $n \cdot e \cdot r$ olur ve her oyuncu $r \cdot e$ alır. $r > 1$ olduğu için $r \cdot e > e$'dir: herkes daha iyi durumdadır. Denge Pareto optimal değildir. □

Örnekte tek Nash dengesi herkesin 0 vermesidir ve kişi başı kazanç 10'dur. Oysa herkes 10 verse kişi başı 16 olurdu. Denge Pareto optimal değildir. Katkı vermeden ortak maldan yararlanan oyuncuya **beleşçi** (*free rider*), bu davranışa **beleşçilik** (*free riding*) denir.



Şekil 10.18: Katkı *k*'nin kazanca etkisi. Kırmızı doğru: diğerleri tam katkı (10) verirken O1'in kazancı, O1 k verdikçe 22'den 16'ya düşer. Mavi doğru: herkes aynı k'yi verirse kişi başı kazanç 10'dan 16'ya çıkar.

Şekil 10.18 ikilemi iki doğruyla çizer. Kırmızı doğru O1'in bireysel bakışıdır: diğerleri 10 verirken $u_1 = 10 + 0{,}4 \cdot 30 - 0{,}6k = 22 - 0{,}6k$. O1'in her jetonu kazancını 0,6 düşürür; en iyisi hiç vermemektir. Mavi doğru grubun bakışıdır: herkes k verirse $u = 10 - k + 0{,}4 \cdot 4k = 10 + 0{,}6k$. Herkesin her jetonu kişi başı kazancı 0,6 artırır; en iyisi herkesin tam vermesidir. İki doğrunun eğimleri zıt işaretlidir: bireysel teşvik (−0,6) ile sosyal refah (+0,6) ters yöndedir. Bu, Mahkûmun İkilemi'nin n kişilik hâlidir.

10.5.4 İkilem ne zaman vardır?

İkilemin varlığı iki koşula bağlıdır. Bir jetonluk katkı, katkıyı verene r/n, gruba toplam r olarak döner.

- **Bireysel koşul:** $r/n < 1$, yani $r < n$. Katkı bireysel olarak zararlıdır; vermemek baskındır.
- **Sosyal koşul:** $r > 1$, yani $r/n > 1/n$. Her jeton grubun toplamını $r - 1 > 0$ artırır; katkı sosyal olarak faydalıdır.

İki koşul birlikte, yani $1 < r < n$ (eşdeğer olarak $1/n < r/n < 1$), ikilemi oluşturur. Koşulun dışındaki iki durumda ikilem yoktur.

Örnek 10.14

$n = 4$, $e = 10$ sabit tutulur ve r değiştirilir.

1. **r = 0,8 (r < 1).** Bir jetonluk katkı, katkı verene $0{,}2$ döndürür (net −0,8) ve grubun toplamını $0{,}8 - 1 = -0{,}2$ değiştirir. Katkı hem bireye hem gruba zararlıdır. Herkesin 0 vermesi hem denge hem de grup için en iyisidir; ikilem yoktur.

2. **r = 1,6 (1 < r < 4).** Ana örnek: bireysel −0,6, sosyal +0,6. İkilem vardır.
3. **r = 4 (r = n).** Bir jetonluk katkı, katkı verene tam 1 döndürür: net 0. Oyuncu katkı verse de vermese de kazancı değişmez. Tam katkı zayıf baskındır: hiçbir zaman daha kötü değildir. Sosyal olarak her jeton grubu 3 artırır.
4. **r = 6 (r > n).** Bir jeton kendine *6/4 = 1,5* döndürür: net +0,5. Tam katkı kesin baskındır. Bireysel ve sosyal teşvik aynı yöndedir; ikilem kalkar.

Örnek 10.14 tasarım açısından önemli bir sonuç verir. İkilemi ortadan kaldırmanın bir yolu, katkının bireye dönen kısmını 1'in üstüne çıkarmaktır. Bunun için çarpanı grup büyüklüğünden büyük yapmak çoğu zaman gerçekçi değildir. Ama katkıya bağlı bir ödül (ör. katkı verene öncelik) aynı etkiyi yaratır: katkı verenin kişisel getirisi artar.

10.5.5 Çoklu ajan sistemlerindeki karşılıklar

Kamu malı oyunu soyut bir laboratuvar oyunu değildir; çoklu ajan sistemlerinde doğrudan karşılıkları vardır.

Dosya paylaşım ağları
: Eşler arası (*peer-to-peer*, P2P) bir ağda her eş dosya indirir ve başkalarına dosya yükler. Yükleme bant genişliği harcar, ama ağın bütününe fayda sağlar. Yükleme yapmadan yalnızca indiren eş beleşçidir. Herkes böyle davranırsa ağda indirilecek dosya kalmaz.

Federe öğrenme
: Birçok istemci (telefon, hastane, şirket) ortak bir makine öğrenmesi modelini birlikte eğitir (*federated learning*). Her istemci kendi verisiyle hesapladığı güncellemeyi gönderir. Güncelleme göndermek işlemci, enerji ve veri gizliliği açısından maliyetlidir. Güncelleme göndermeden ortak modeli kullanan istemci beleşçidir.

Ortak hesaplama ve veri havuzları
: Bir topluluk ortak bir veri kümesi ya da açık kaynak bir kütüphane oluşturur. Katkı vermeden kullanan çoktur; katkı veren az sayıda kişi yükün tamamını taşır.

Bu sistemlerin tasarımcıları beleşçiliği azaltmak için oyunun kazançlarını değiştirir. Katkıyı ödüllendirmek (yükleme yapan eşe daha hızlı indirme hakkı vermek), katkı vermeyeni dışlamak (malı kısmen dışlanabilir yapmak) ya da etkileşimi tekrarlı kılıp itibar kaydı tutmak yaygın yollardır. Hepsinin ortak fikri, katkının bireysel getirisini artırarak bireysel teşviki sosyal refahla aynı yöne çevirmektir. Bu, 10.2.6'daki teşvik tasarımı ilkesinin kamu malına uygulanmasıdır.

10.5.6 Kamu malında sık hatalar

Sık yapılan hata

Havuzu çarpmayı ya da bölmeyi unutmak. Herkes 10 verdiğinde havuz $40 \cdot 1{,}6 = 64$'tür ve dörde bölünür: 16. Havuzun tamamını (64) bir oyuncunun kazancı sanmak ya da çarpanı unutup $40/4 = 10$ yazmak sık yapılan hatalardır.

Beleşçinin havuzdan pay almadığını sanmak. Kamu malı dışlanamaz. Beleşçi O1 de havuzdan diğerleri kadar (12) alır; üstelik cebindeki 10'u da tutar ve 22'ye ulaşır.

"Katkı sosyal olarak faydalıysa birey de katkı verir" sanmak. Sosyal getiri (+0,6) ile bireysel getiri (−0,6) farklıdır. Rasyonel birey yalnız kendi getirisine bakar; ikilemin kaynağı budur.

İkilem koşulunu yanlış yazmak. İkilem $1 < r < n$ iken vardır. $r < 1$ ise katkı gruba da zararlıdır; $r \geq n$ ise tam katkı (zayıf) baskındır. Her iki durumda da bireysel ve sosyal teşvik aynı yöndedir.

10.6 Özet

- **Ajan**, otonom, reaktif, proaktif ve sosyal yetenekli bir sistemdir (Wooldridge ve Jennings, 1995). **Çoklu ajan sistemi**, aynı çevrede algılayan ve eylem yapan birden fazla ajandan oluşur; her ajanın bilgisi yereldir. Oyun kuramında ajan, beklenen faydasını maksimize eden rasyonel bir oyuncudur.
- **Mimari** (reaktif, planlayan, BDI, öğrenen, hibrit) ajanın içini; oyun kuramı ajanlar arasındaki etkileşimi anlatır.
- Sistemler **işbirlikçi** (sorun koordinasyon), **rekabetçi** (sorun strateji) ya da **karma güdülü** olabilir. Üç küçük oyun bu yelpazeyi temsil eder: koordinasyon oyununda iki denge, Mahkûmun İkilemi'nde verimsiz tek denge, Yazı-Tura'da yalnız karışık denge (½, ½).
- Çoğu gerçek sistem **hibrit kontrol** kullanır: ajanlar otonom karar verir, merkez kuralları, fiyatları ya da ihaleleri belirler. **Konvansiyon**, koordinasyon oyununda bir dengeyi önceden seçer; trafik ışığı gibi ortak bir sinyal **korelasyonlu denge** üretir.
- Öğrenen ajanların çevresi **durağan değildir**; eşzamanlı en iyi tepki koridor oyununda sonsuz döngüye girebilir. Basit yerel kurallar **ortaya çıkan davranış** üretir (Schelling'in ayrışma modeli).
- **Sözleşme ağı**: duyuru → teklif → ödül. Dağıtık ve basittir, görev başına $O(n)$ mesaj gerektirir, ama açgözlü ve sıralıdır: örnekte 18 dk, optimum 13 dk (%38 fazla). Görev sırası sonucu değiştirir; yeniden ihale (takas) örnekte iki adımda optimuma ulaşır.
- **Tüm atamaları denemek** $n!$ iş gerektirir (n = 10'da 3.628.800); Macar algoritması $O(n^3)$'tür. Kazanana kendi teklifi ödenirse yüklenici maliyetini şişirir; ikinci en düşük teklif ödenirse dürüst teklif en iyisidir.
- **Karakteristik fonksiyon** $v(S)$ her koalisyonun değerini verir. Süperadditif oyunda büyük koalisyon oluşur. Koalisyon yapılarının sayısı Bell sayısıdır (3 ajan 5, 4 ajan 15, 10 ajan 115.975).

- **Shapley değeri** = ortalama marjinal katkı; dört aksiyomu (verimlilik, simetri, kukla oyuncu, toplamsallık) sağlayan tek kuraldır. Lojistik örneğinde $\varphi = (30, 40, 50)$. İkili sinerji modelinde φ_i = tek başına değer + sinerjilerin yarısı: oyunun örnek turunda $(9, 6, 6)$, eşit bölüşüm hatası 4.
- **Çekirdek**: hiçbir koalisyonun ayrılmak istemediği paylaşımlar. Lojistik örneğinde altı köşeli bir çokgendir ve Shapley değerini içerir (oyun dışbükeydir). $y = (10, 25, 85)$'i {A, B} engeller. Çoğunluk oyununda çekirdek boştur; en küçük ε-çekirdek $\varepsilon = 1/3$ ile $(1/3, 1/3, 1/3)$'tür.
- **Ağırlıklı oylamada güç ≠ ağırlık**: $[51; 50, 49, 1]$'de Shapley–Shubik $(2/3, 1/6, 1/6)$, Banzhaf $(3/5, 1/5, 1/5)$. 1958 AET Konseyi'nde Lüksemburg'un gücü 0'dı.
- **Kamu malı oyununda** $r/n < 1$ ise katkı vermemek baskındır: denge kişi başı 10, herkes tam katkı verirse 16. İkilem $1 < r < n$ iken vardır. **Beleşçi** (yalnızca indiren eş, güncelleme göndermeyen istemci) bireysel olarak kazanır, sistem kaybeder.

Kavram	Açıklama	Örnek
Koordinasyon oyunu	Birden çok eşit iyi denge; sorun hangisinin oynanacağı	Koridor oyunu; “herkes sağdan”
Sözleşme ağı	Duyuru, teklif, ödül; açgözlü ve sıralı	18 dk; optimum 13 dk
Shapley değeri	Bütün sıralarda ortalama marjinal katkı	(30, 40, 50); oyunda (9, 6, 6)
Çekirdek	Hiçbir koalisyonun engellemediği paylaşımlar	altıgen; (10, 25, 85) dışarıda
Boş çekirdek	Kararlı paylaşım yok	çoğunluk oyunu
Güç endeksi	Kilit sıraların ya da kritik üyeliklerin oranı	[51; 50, 49, 1]: (2/3, 1/6, 1/6)
Kamu malı oyunu	$r/n < 1$ ise vermemek baskın	denge 10, optimum 16

Tablo 10.21: Bölümün kavramları bir bakışta.

10.7 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

ajan *agent* · çoklu ajan sistemi *multi-agent system (MAS)* · otonomi *autonomy* · reaktivite *reactivity* · proaktivite *pro-activeness* · sosyal yetenek *social ability* · planlayan ajan *deliberative agent* · inanç–arzu–niyet (BDI) *belief–desire–intention (BDI)* · karma güdü *mixed-motive* · koordinasyon oyunu *coordination game* · konvansiyon / sosyal yasa *convention / social law* · korelasyonlu denge *correlated equilibrium (CE)* · durağan olmayan *non-stationary* · kurgusal oyun *fictitious play* · ortaya çıkan davranış *emergent behavior* · görev tahsisi *task allocation* · kaynak tahsisi *resource allocation* · sözleşme ağı protokolü *Contract Net Protocol* · yönetici / yüklenici *manager / contractor (contract net)* · duyuru / teklif / ödül *task announcement / bid / award* · açgözlü *greedy (algorithm)* · Macar algoritması *Hungarian algorithm* · yeniden ihale *re-auction (re-tendering)* · kombinatoryal açık artırma *combinatorial auction* · koalisyon *coalition* · büyük koalisyon *grand coalition*

koalisyon oluşumu *coalition formation* · koalisyon yapısı *coalition structure* · Bell sayısı *Bell number* · karakteristik fonksiyon *characteristic function* · aktarılabilir faydalı (TU) oyun *transferable-utility (TU) game* · süperadditif *superadditive* · paylaşım *imputation* · marjinal katkı *marginal contribution* · Shapley değeri *Shapley value* · verimlilik aksiyomu *efficiency axiom* · simetri aksiyomu *symmetry axiom* · kukla oyuncu *dummy (null) player* · toplamsallık *additivity* · çekirdek *core* · engellemek *to block (a coalition blocks)* · dışbükey oyun *convex game* · dengeli oyun *balanced game* · Bondareva–Shapley teoremi *Bondareva–Shapley theorem* · çoğunluk oyunu *majority game* · ε-çekirdek *ε-core* · çekirdekçik *nucleolus* · ağırlıklı oylama oyunu *weighted voting game* · kota *quota* · kilit oyuncu *pivotal player* · kritik üye *critical (swing) member* · güç endeksi *power index* · Shapley–Shubik endeksi *Shapley–Shubik index* · Banzhaf endeksi *Banzhaf index* · kamu malı *public good* · kamu malı oyunu *public goods game* · kişi başı marjinal getiri (MPCR) *marginal per-capita return (MPCR)* · beleşçi *free rider* · beleşçilik *free riding* · ortak mallar trajedisi *tragedy of the commons*

10.8 Kendini Dene

1. Saf koordinasyon oyununda (Sağ, Sağ) her iki oyuncuya 2, (Sol, Sol) her ikisine 1, farklı seçimler 0 verir. Saf Nash dengeleri nelerdir? Karışık dengede her oyuncu Sağ'ı hangi olasılıkla seçer, beklenen kazanç nedir ve bu kazanç saf dengelerin kazancıyla nasıl karşılaştırılır?

   Cevap

   Saf dengeler (Sağ, Sağ) = (2, 2) ve (Sol, Sol) = (1, 1)'dir; ikisinde de taraf değiştiren oyuncu 0'a düşer. Karışık denge: 2. oyuncu Sağ'ı q ile oynasın. 1. oyuncu için Sağ $2q$, Sol $1 - q$ verir; kayıtsızlık $2q = 1 - q \Rightarrow q = 1/3$. Simetriden $p = 1/3$. Beklenen kazanç $2 \cdot (1/3)(1/3) + 1 \cdot (2/3)(2/3) = 2/9 + 4/9 = 2/3$'tür. Bu, iki saf dengenin de (2 ve 1) altındadır: oyuncular sık sık uyumsuz seçim yapar. Bir konvansiyon (“herkes sağdan”) kazancı 2'ye çıkarır.

2. Bir yönetici G1, G2, G3 görevlerini bu sırayla ihaleye çıkarıyor. Maliyetler (dk): G1 için R1 4, R2 5, R3 6; G2 için R1 2, R2 3, R3 6; G3 için R1 7, R2 4, R3 9. Her robot en çok bir görev alır. Sözleşme ağının ataması ve toplamı nedir? Merkezî optimum nedir ve sözleşme ağı ondan yüzde kaç fazla-

dır?

Cevap

G1: teklifler 4, 5, 6 → R1 (4). G2: R2 3, R3 6 → R2 (3). G3: yalnız R3 kalır → 9. Toplam *4 + 3 + 9 = 16* dk. Altı atama: (R1, R2, R3) 16; (R1, R3, R2) *4 + 6 + 4 = 14*; (R2, R1, R3) *5 + 2 + 9 = 16*; (R2, R3, R1) *5 + 6 + 7 = 18*; (R3, R1, R2) *6 + 2 + 4 = 12*; (R3, R2, R1) *6 + 3 + 7 = 16*. Optimum 12 dk: G1→R3, G2→R1, G3→R2. Sözleşme ağı *(16 − 12)/12 ≈ %33* fazladır. Kayıp yine açgözlülükten doğar: G1'i alan R1, G2 için en ucuz robottu (2).

3. Bir görev için robotların gerçek maliyetleri R1 5, R2 7, R3 9'dur ve en düşük teklif kazanır. (a) Kazanana kendi teklifi ödenirse dürüst teklifle R1'in kazancı nedir; R1 teklifini 6,5'e çıkarırsa ne olur? (b) Kazanana ikinci en düşük teklif ödenirse dürüst R1 ne kazanır? (c) Bu kuralda R2 işi almak için 4 teklif ederse ne olur?

Cevap

(a) Dürüst teklifle R1 kazanır, 5 alır, kazancı *5 − 5 = 0*. 6,5 teklif ederse (diğerleri dürüstse) yine en düşük tekliftir; kazanır ve *6,5 − 5 = 1,5* kazanır. Bu kuralda şişirmek kârlıdır. (b) R1 kazanır ve ikinci en düşük teklif olan 7'yi alır: kazanç *7 − 5 = 2*. 7'nin altındaki her teklif aynı sonucu verir. (c) R2 kazanır, ama ödeme ikinci en düşük teklif olan R1'in 5'idir; maliyeti 7 olduğu için kazancı *5 − 7 = −2*'dir. Dürüst teklifle kaybeder ve 0 alır; dürüstlük daha iyidir.

4. Üç ajanlı bir oyunda *v(A) = 0*, *v(B) = 0*, *v(C) = 10*, *v(AB) = 30*, *v(AC) = 40*, *v(BC) = 40*, *v(ABC) = 60*'tır. Altı sıra üzerinden hesaplanan Shapley değeri nedir? Oyun süperadditif midir, dışbükey midir?

Cevap

Marjinal katkılar (A, B, C): A→B→C (0, 30, 30); A→C→B (0, 20, 40); B→A→C (30, 0, 30); B→C→A (20, 0, 40); C→A→B (30, 20, 10); C→B→A (20, 30, 10). Toplamlar A 100, B 100, C 160; *φ = (100/6, 100/6, 160/6) = (50/3, 50/3, 80/3) ≈ (16,67; 16,67; 26,67)*, toplam 60 ✓. Süperadditiftir: *30 ≥ 0*, *40 ≥ 10*, *40 ≥ 10* ve her ikili artı tekli en çok *40 + 0 = 40* ya da *30 + 10 = 40*'tır, 60'ı aşmaz. Dışbükey değildir: A'nın {B}'ye katkısı 30, daha büyük {B, C}'ye katkısı *60 − 40 = 20*'dir; katkı koalisyon büyüyünce azalmıştır.

5. 4. sorudaki oyunda Shapley değeri çekirdekte midir? *(20, 20, 20)* ve *(30, 10, 20)* paylaşımları çekir-

dekte midir? Dışarıda kalan paylaşımı hangi koalisyon engeller?

Cevap

Üst sınırlar $x_i \le v(N) - v(N \setminus \{i\})$: $x_A \le 60 - 40 = 20$, $x_B \le 20$, $x_C \le 60 - 30 = 30$; alt sınırlar 0, 0, 10. Shapley $(16{,}67;\ 16{,}67;\ 26{,}67)$ bütün sınırların içindedir: çekirdektedir. Oyun dışbükey olmadığı hâlde Shapley değeri çekirdektedir; dışbükeylik yeterli koşuldur, gerekli değildir. $(20, 20, 20)$: sınırların içinde (A ve B tam üst sınırda), çekirdektedir. $(30, 10, 20)$: $x_A = 30 > 20$; {B, C} yalnız $10 + 20 = 30 < 40$ alır ve ayrılıp 40'ı paylaşır. {B, C} engeller.

6. Ağırlıklı oylama oyunu $[7; 4, 3, 2, 1]$'dir (A 4, B 3, C 2, D 1). Kazanan koalisyonlar, Banzhaf endeksi ve Shapley–Shubik endeksi nedir? C ile D'nin gücü nasıl karşılaştırılır?

Cevap

Toplam ağırlık 10'dur. Kazanan koalisyonlar: AB (7), ABC (9), ABD (8), ACD (7), ABCD (10). A'sız en iyi koalisyon BCD = 6 < 7, yani A her kazanan koalisyonda vardır. Kritik üyelikler: AB'de A, B; ABC'de A, B (C ayrılınca 7 kalır); ABD'de A, B; ACD'de A, C, D; ABCD'de A. Sayılar A 5, B 3, C 1, D 1, toplam 10: Banzhaf $(1/2, 3/10, 1/10, 1/10)$. Shapley–Shubik (24 sıra): kilit sayıları A 14, B 6, C 2, D 2 $\Rightarrow$ $(7/12, 1/4, 1/12, 1/12)$. Örneğin C, ancak önünde tam olarak A ve D varsa (toplam 5) kilittir; bu $2! \cdot 1! = 2$ sırada olur. C'nin ağırlığı D'nin iki katıdır, ama iki endekste de güçleri eşittir.

7. Kamu malı oyununda $n = 3$, $e = 20$, $r = 1{,}5$'tir. Herkes 20 verirse, kimse vermezse ve O1 beleşçi olup diğerleri 20 verirse kazançlar nedir? Kişi başı marjinal getiri, bir jetonun bireysel ve sosyal net etkisi nedir? İkilem var mı?

Cevap

Herkes 20: havuz $60 \cdot 1{,}5 = 90$, kişi başı 30; herkes 30 alır. Kimse: herkes 20. O1 beleşçi: havuz $40 \cdot 1{,}5 = 60$, kişi başı 20; O1 $20 + 20 = 40$, diğerleri $0 + 20 = 20$. MPCR $r/n = 0{,}5$: bir jetonluk katkı bireye $-1 + 0{,}5 = -0{,}5$, gruba $-1 + 1{,}5 = +0{,}5$ net etki yapar. $1 < 1{,}5 < 3$ olduğu için ikilem vardır: vermemek baskın, denge kişi başı 20; herkes verirse 30.

8. Dört ajan 1 birimlik ödülü paylaşacaktır; en az üç ajan anlaşırsa ödülü alır: $v(S) = 1$ ($|S| \ge 3$), aksi hâlde 0. Çekirdek boş mudur? Shapley değeri nedir?

Cevap

Dört üçlü kısıt vardır: her üçlünün payları toplamı en az 1. Her ajan dört üçlünün üçündedir; kısıtlar toplanınca $3(x_A + x_B + x_C + x_D) \ge 4$, yani toplam ≥ 4/3 çıkar. Ama toplam 1 olmalıdır: çekirdek boştur. Dört ajan simetriktir; simetri ve verimlilikten $\varphi = (1/4, 1/4, 1/4, 1/4)$. Her üçlü bu paylaşımda yalnız 3/4 alır ve dördüncüyü dışlayarak 1'i paylaşmak ister.

10.9 İleri okuma

- M. Wooldridge, *An Introduction to MultiAgent Systems* (2. baskı, 2009): ajan kavramını, ajan mimarilerini (reaktif, BDI, hibrit), sözleşme ağı gibi iletişim protokollerini ve koalisyon oluşumunu az matematikle, bilgisayar mühendisliği bakışıyla anlatan temel ders kitabıdır.
- G. Chalkiadakis, E. Elkind ve M. Wooldridge, *Computational Aspects of Cooperative Game Theory* (2011): karakteristik fonksiyon, Shapley değeri, çekirdek, çekirdekçik ve ağırlıklı oylama oyunlarını hesaplama karmaşıklığı açısından inceleyen kısa ve düzenli bir kitaptır.
- A. R. Karlin ve Y. Peres, *Game Theory, Alive* (2017), koalisyonlar ve Shapley değeri bölümü: Shapley aksiyomlarını ve teklik kanıtını çok sayıda örnekle, adım adım verir.
- Y. Shoham ve K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* (2009), Bölüm 12: işbirlikçi oyun kuramının çözüm kavramlarını (çekirdek, Shapley değeri, çekirdekçik) ve Bondareva–Shapley teoremini biçimsel olarak anlatır; kitabın PDF'i yazarların sitesinde ücretsiz yayımlanmıştır.

10.10 Kaynaklar

1. O. N. Bondareva, “Some applications of linear programming methods to the theory of cooperative games”, *Problemy Kibernetiki* 10, 1963.
2. G. Chalkiadakis, E. Elkind, M. Wooldridge, *Computational Aspects of Cooperative Game Theory*, Morgan & Claypool, 2011.
3. G. de Clippel, K. Rozen, “Fairness through the lens of cooperative game theory: An experimental approach”, *American Economic Journal: Microeconomics* 14(3), 2022.
4. FIPA, *Contract Net Interaction Protocol Specification* (SC00029H), 2002.
5. D. B. Gillies, “Solutions to general non-zero-sum games”, *Contributions to the Theory of Games IV*, Princeton, 1959.
6. A. R. Karlin, Y. Peres, *Game Theory, Alive*, American Mathematical Society, 2017.
7. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 15: maliyet paylaşımı, çekirdek ve Shapley değeri).
8. C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model”, *SIGGRAPH '87*, 1987.
9. T. Sandholm, K. Larson, M. Andersson, O. Shehory, F. Tohmé, “Coalition structure generation with worst case guarantees”, *Artificial Intelligence* 111, 1999.
10. T. C. Schelling, “Dynamic models of segregation”, *Journal of Mathematical Sociology* 1(2), 1971.
11. L. S. Shapley, “A value for n-person games”, *Contributions to the Theory of Games II*, Princeton, 1953.
12. L. S. Shapley, “On balanced sets and cores”, *Naval Research Logistics Quarterly* 14(4), 1967.
13. L. S. Shapley, “Cores of convex games”, *International Journal of Game Theory* 1, 1971.
14. L. S. Shapley, M. Shubik, “A method for evaluating the distribution of power in a committee system”, *American Political Science Review* 48(3), 1954.

15. Y. Shoham, K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2009 (Bölüm 12).

16. R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver", *IEEE Transactions on Computers* C-29(12), 1980.

17. M. Wooldridge, *An Introduction to MultiAgent Systems*, 2. baskı, Wiley, 2009.

18. M. Wooldridge, N. R. Jennings, "Intelligent agents: Theory and practice", *Knowledge Engineering Review* 10(2), 1995.

KISIM IV · BÖLÜM 11

Algoritmik Adalet ve Etik

Kredi başvurusunu onaylayan, özgeçmişleri eleyen ya da bir sanığın risk puanını hesaplayan algoritmalar, insanların hayatını doğrudan etkileyen kararlar verir. Bu kararlar "doğru" olabilir ama yine de bazı gruplara sistematik olarak daha çok hata yükleyebilir. Bu not adaleti ölçmenin yollarını tek bir sayısal örnek üzerinden kurar ve bu ölçütlerin neden aynı anda sağlanamadığını gösterir. Kaynakların adil paylaşımı (pasta kesme ve bölünemez malların bölüşümü), etik ilkeler, denetim ve insanların algoritmaya stratejik tepkisi de notun konularıdır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum11`

ÖĞRENME ÇIKTILARI

Ölçütleri hesaplamak
İki grubun karışıklık matrisinden demografik eşitlik, eşitlenmiş olasılıklar, tahmin eşitliği ve kalibrasyonu hesaplamak.

Uyumsuzluğu göstermek
Taban oranları farklıyken ölçütlerin neden birlikte sağlanamadığını sayısal olarak göstermek.

Adil bölüşüm
Kes-ve-seç yöntemini uygulamak; orantılılık ve kıskançsızlığı ayırt etmek.

Etik ve teşvikler
Şeffaflık, hesap verebilirlik, denetim ve stratejik davranışın tasarıma etkisini tartışmak.

11.1 Giriş

Mekanizma tasarımı ve oyun kuramı çoğunlukla verimlilikle, yani bir mekanizmanın toplam refahı büyütüp büyütmediğiyle ilgilenir. Bu notun sorusu farklıdır: bir kararın ya da paylaşımın sonuçları kişiler ve gruplar arasında *nasıl dağılıyor*? Bu soru iki ayrı alanda karşımıza çıkar. Birincisi, makine öğrenmesi modellerinin verdiği evet/hayır kararlarıdır. İkincisi, bir kaynağın (zaman, arazi, miras, ders kontenjanı) kişiler arasında bölüşülmesidir. İki alanda da "adil" sözcüğünün tek bir anlamı yoktur. Bu yüzden önce farklı anlamları ayrı ayrı tanımlamak, sonra aralarındaki ilişkiyi incelemek gerekir.

11.1.1 Bir oyunla başlangıç: Adil Bölüşüm Oyunu

Bölümün etkileşimli oyunu olan Adil Bölüşüm Oyunu basit bir kurala dayanır. İki oyuncu, bölünemeyen malları sırayla ve birer birer seçer. Her malın kartında iki oyuncunun o mala verdiği değer yazar. Oyunun sabit örnek turunda dört mal kullanılır; diğer seviyelerde değerler her turda rastgele çekilir.



Şekil 11.1: Adil Bölüşüm Oyunu'nun örnek turu. Her kartta birinci oyuncunun (O1) ve ikinci oyuncunun (O2) o mala verdiği değer yazar: Pasta 5/1, Kitap 4/9, Kulaklık 3/8, Top 1/1. Seçim sırası O1, O2, O1, O2'dir; O2 (bilgisayar) her seferinde kendi gözünde en değerli kalan malı alır.

Şekil 11.1'de görülen değerler iki oyuncunun malları çok farklı değerlendirdiğini gösterir. O1 en çok Pasta'yı (5), O2 en çok Kitap'ı (9) sever. Oyun iki soru sorar:

- Rakip hep kendi en sevdiği malı alıyorsa, O1'in de hep en değerli malı alması en iyisi midir?
- Herkesin kendi payını ötekininkinden değerli bulduğu bir paylaşım her zaman var mıdır?

İlk soru bir oyun kuramı sorusudur: O1'in stratejisi, rakibin davranışına karşı en iyi tepki olmalıdır. İkinci soru bir adalet sorusudur: "kimsenin başkasının payını kıskanmadığı" bir paylaşım her zaman mümkün müdür? İki sorunun cevabı da "hayır"dır. Gerekçeler 11.4.8 ve 11.4.9'da hesaplanır. Oyun daha baştan iki şeyi gösterir: adalet tek bir sayıyla anlatılamaz ve kurala uyan oyuncular bile stratejik davranabilir.

11.1.2 Algoritmik kararlar neden önemli?

Algoritmalar artık yüksek etkili kararları ya doğrudan verir ya da verdikleri önerilerle yönlendirir. Kredi onayı, işe alım, sağlık teşhisi, sosyal medya önerileri ve ceza adaleti bu alanların başlıcalarıdır. Bu kararlarda yapılan bir hata, bir kişiye evine, işine ya da özgürlüğüne mal olabilir.

Adalet ve etik ihlallerinin iki tür zararı vardır. Birincisi doğrudan zarardır: haksız yere reddedilen başvuran kredi alamaz, haksız yere "yüksek riskli" sayılan sanık daha uzun tutuklu kalabilir. İkincisi dolaylı zarardır: bu tür olaylar duyuldukça kurumlara ve teknolojiye duyulan toplumsal güven aşınır. Güven kaybolunca yararlı uygulamalar bile kabul görmez.

Oyun kuramı açısından bir karar kuralı bir **mekanizmadır**. Kural, girdileri (başvuranın verileri) çıktılara (onay ya da ret) dönüştürür. Bu mekanizmanın çevresinde çıkarları çatışan paydaşlar vardır: başvuran kredi almak ister, kurum geri ödenmeyen krediden kaçınmak ister, toplum ise fırsatların gruplar arasında hakkaniyetle dağılmasını ister. Mekanizma tasarımında olduğu gibi burada da hangi amacın gözetileceği bir tasarım kararıdır.

Üç örnek, sorunun gerçek hayatta nasıl ortaya çıktığını gösterir:

- **Amazon işe alım aracı (2018).** Şirketin geçmiş özgeçmişlerle eğittiği eleme modeli, "kadın" sözcüğü geçen başvuruları (örneğin "kadın satranç kulübü kaptanı") cezalandırıyordu. Model, geçmişte çoğunlukla erkeklerin işe alındığı veriden bu örüntüyü öğrenmişti. Araç kullanımdan kaldırıldı (Dastin, 2018).
- **COMPAS (2016).** ABD'de sanıklar için hesaplanan bir yeniden suç işleme risk puanıdır. Gazeteciler, puanın siyah ve beyaz sanıklarda farklı hata oranları ürettiğini gösterdi; şirket ise puanın iki grupta aynı riski anlattığını savundu (Angwin vd., 2016). Bu tartışma, taban oranları farklıyken adalet ölçütlerinin çatıştığını gösteren en bilinen örnektir. 11.3.5'te sayılarıyla ele alınır.
- **Kredi skorlama.** Burada iki soru öne çıkar. Aynı puan herkes için aynı riski mi gösteriyor? Kimler hiç kredi alamıyor? İlk soru kurumun, ikincisi başvuranın bakış açısını yansıtır. Bölümün sayısal örneği de bir kredi modelidir.

11.1.3 Önyargı ve ayrımcılık nereden gelir?

Bir modelin adaletsiz sonuç üretmesi için onu yazan kişinin kötü niyetli olması gerekmez. **Önyargı** (*bias*), modelin geliştirildiği ve kullanıldığı sürecin pek çok noktasından içeri sızabilir. Dört kaynak öne çıkar.

Tarihsel veri
: Geçmişteki eşitsizlikler veri etiketlerine yansır. Model bu etiketleri "doğru cevap" sayar ve onları yeniden üretmeyi öğrenir. Amazon örneğindeki cinsiyet önyargısı buna örnektir: "geçmişte işe alındı" etiketi, liyakati değil geçmişin işe alım alışkanlıklarını kaydetmiştir.

Temsil eksikliği
: Bir grup veride az temsil ediliyorsa model o grupta daha çok hata yapar. Model, eğitimdeki toplam hatayı küçültmeye çalışır; az sayıdaki örneğin hatası bu toplama az katkı yapar. Ticari yüz tanıma sistemlerinde en yüksek hata oranı koyu tenli kadınlarda ölçülmüştür (Buolamwini ve Gebru, 2018; "Gender Shades" çalışması).

Ölçüm ve vekil değişken
: Asıl ölçülmek istenen hedef yerine ölçmesi kolay bir **vekil değişken** (*proxy variable*) kullanılabilir. Örneğin "sağlık ihtiyacı" yerine "sağlık harcaması" ölçülürse, sağlık hizmetine erişimi az

olan grup daha az harcadığı için daha sağlıklı görünür. Model bu grubun ihtiyacını sistematik olarak küçümser.

Tasarım ve geri besleme

Yanlış seçilmiş bir optimizasyon hedefi, korunan özelliğin vekilleri (örneğin posta kodu) ve modelin kararlarının gelecekte toplanacak veriyi değiştirmesi bu gruptadır. Son durum bir **geri besleme döngüsüdür**; 11.5.6'daki kod izlemesi böyle bir döngüyü adım adım gösterir.

Bu kaynaklardan çıkan önemli bir ders vardır. **Korunan özellik** (*protected attribute*), yani cinsiyet ya da etnik köken gibi ayrımcılık yapılması yasak olan bir özellik, modelden çıkarılsa bile sorun çözülmez. Diğer değişkenler onu dolaylı olarak taşır. Bir kentte mahalleler etnik kökene göre ayrışmışsa posta kodu, etnik kökenin iyi bir vekilidir. Model posta kodunu kullanarak aynı grup farkını yeniden üretebilir. Bu nedenle adaleti güvenceye almanın yolu, girdiden bir sütun silmek değil, *sonuçları grup bazında ölçmektir*.

Sık yapılan hata

- **"Model cinsiyeti görmüyorsa adildir."** Bu yaklaşıma "körlükle adalet" denir ve vekil değişkenler yüzünden çoğu zaman işe yaramaz. Üstelik korunan özellik silinince, grup bazlı sonuçları ölçmek de zorlaşır.
- **"Veri gerçeği yansıtır, model yalnız onu öğrenir."** Veri, geçmişteki kararların ve ölçüm biçimlerinin kaydıdır. Etiket bir vekil ise ya da geçmiş kararlarla toplandıysa, "gerçek" sandığımız şey zaten çarpıktır.
- **"Önyargı yalnız eğitim verisinden gelir."** Hedefin seçimi, eşiğin konması ve modelin kullanıldığı bağlam da önyargı kaynağıdır.

11.2 Adalet ölçütleri

Bir modelin "adil" olup olmadığını tartışmak için önce neyin ölçüleceği üzerinde anlaşmak gerekir. Dört temel ölçütün hepsi, ikili (evet/hayır) karar veren bir modelin sonuçlarını özetleyen karışıklık matrisinden türetilir. Bütün hesaplarda aynı iki grup ve aynı model kullanılır. Böylece ölçütler arasındaki farklar sayıların değişmesinden değil, sorulan sorunun değişmesinden kaynaklanır.

11.2.1 Karışıklık matrisi

Bir kredi modeli düşünelim. Her başvuran için iki şey vardır. Birincisi gerçek sonuçtur: kişi krediyi gerçekte geri öder mi? Bu sonuç Y ile gösterilir; $Y = 1$ "öder", $Y = 0$ "ödemez" demektir. İkincisi modelin kararıdır: model krediyi onaylar mı? Bu karar $\hat{Y}$ ("Y şapka") ile gösterilir; $\hat{Y} = 1$ "onay", $\hat{Y} = 0$ "ret" demektir. Şapka işareti, bunun gerçek değer değil bir tahmin olduğunu hatırlatır.

İki ikili değişkenin dört birleşimi vardır. Her başvuran bu dört kutudan birine düşer. Kutulardaki kişi sayılarını gösteren tabloya karışıklık matrisi denir.

Tanım 11.1

Karışıklık matrisi (*confusion matrix*), gerçek sonuç Y ile modelin kararı $\hat{Y}$'nın dört birleşimindeki kişi sayılarını gösteren 2 × 2 tablodur:

- **TP** (doğru pozitif, *true positive*): öder ve onaylandı ($Y = 1$, $\hat{Y} = 1$);
- **FN** (yanlış negatif, *false negative*): öder ama reddedildi ($Y = 1$, $\hat{Y} = 0$);
- **FP** (yanlış pozitif, *false positive*): ödemez ama onaylandı ($Y = 0$, $\hat{Y} = 1$);
- **TN** (doğru negatif, *true negative*): ödemez ve reddedildi ($Y = 0$, $\hat{Y} = 0$).

Toplam başvuran sayısı $n = TP + FP + FN + TN$'dir.

Adlandırmanın bir mantığı vardır. İkinci harf (P ya da N) modelin ne dediğini söyler: P "pozitif", yani onay; N "negatif", yani ret. İlk harf (T ya da F) bu sözün doğru mu yanlış mı olduğunu söyler. FN, "yanlış negatif"tir: model "hayır" dedi ve yanıldı, çünkü kişi aslında ödeyecekti.

Matristen beş oran türetilir. Her oran, matrisin belli bir satırına ya da sütununa bakar ve farklı bir soruya cevap verir.

Oran	Tanım	Soru	Matrisin neresi?
Seçim oranı	$(TP + FP) / n$	Kaçta kaçı onay alıyor?	onay sütunu, bütün grup
TPR (doğru pozitif oranı)	$TP / (TP + FN)$	Ödeyecek olanların kaçı onaylandı?	"öder" satırı
FPR (yanlış pozitif oranı)	$FP / (FP + TN)$	Ödemeyecek olanların kaçı onaylandı?	"ödemez" satırı
PPV (kesinlik)	$TP / (TP + FP)$	Onaylananların kaçı ödüyor?	onay sütunu
Taban oranı p	$(TP + FN) / n$	Grupta ödeyenlerin payı	"öder" satırı, bütün grup

Tablo 11.1: Karışıklık matrisinden türetilen oranlar (kredi örneği).

Bu oranlar koşullu olasılık olarak da yazılabilir. Koşullu olasılık $P(A \mid B)$, "B olduğu bilindiğinde A'nın olasılığı" demektir ve yalnız B'yi sağlayanlar arasında A'nın payıdır. TPR, ödeyenler arasında onay alanların payıdır: $TPR = P(\hat{Y} = 1 \mid Y = 1)$. FPR, ödemeyenler arasında onay alanların payıdır: $FPR = P(\hat{Y} = 1 \mid Y = 0)$. PPV ise yönü tersine çevirir; onay alanlar arasında ödeyenlerin payıdır: $PPV = P(Y = 1 \mid \hat{Y} = 1)$. TPR ile PPV'yi karıştırmamak için "çizginin sağındaki koşul, hangi kalabalığın içinde saydığımızı söyler" kuralı işe yarar. TPR'de kalabalık ödeyenlerdir, PPV'de onay alanlardır. Bu iki oranın paydaları farklıdır ve genelde farklı sayılar verir.

TPR'ye tıp ve istatistikte **duyarlılık** (*sensitivity*) de denir. PPV'nin öteki adı **kesinlik**tir (*precision*). Taban oranı (*base rate*) modelden bağımsızdır: modelin ne karar verdiğine bakmaz, yalnız gerçek sonucun gruptaki sıklığını ölçer. Model değişse de taban oranı değişmez. Bölümün ana sonuçlarından biri, taban oranları farklı olduğunda ölçütlerin birlikte sağlanamamasıdır (11.3).

Her grup adaleti ölçütü, bu oranlardan birinin gruplar arasında eşit olmasını ister. Hangi oranın eşitleneceği teknik bir ayrıntı değil, bir *değer seçimidir*: hangi hatanın kimin için daha önemli olduğuna verilen cevaptır.

11.2.2 Örnek: iki grup, tek model

Bölüm boyunca kullanılacak örnek şudur. A ve B adlı iki grubun her birinde 100 başvuran vardır. Model her başvurana iki puandan birini verir: 0,8 (“düşük risk”) ya da 0,2 (“yüksek risk”). Model 0,5 eşiğiyle karar verir: puanı 0,5'ten büyük olanlar onaylanır, ötekiler reddedilir. Aşağıdaki tablo, her puan grubunda kaç kişinin gerçekte ödediğini gösterir.

grup	puan	kişi	öder	ödemez	karar (eşik 0,5)
A	0,80	50	40	10	onay
A	0,20	50	10	40	ret
B	0,80	20	16	4	onay
B	0,20	80	16	64	ret

Tablo 11.2: Örnek veri. Her grupta 100 başvuran; model iki puan veriyor.

Tablodaki ilk gözlem, iki grubun gerçekte farklı olduğudur. A grubunda ödeyenler 40 + 10 = 50 kişidir; taban oranı $p_A = 50/100 = 0{,}50$'dir. B grubunda ödeyenler 16 + 16 = 32 kişidir; taban oranı $p_B = 0{,}32$'dir. İkinci gözlem, modelin B grubuna daha seyrek yüksek puan verdiğidir: A'da 50 kişi 0,8 alırken B'de yalnız 20 kişi alır.

Örnek 11.1

Tablo 11.2'den iki grubun karışıklık matrisleri kurulur.

1. **Onaylananlar.** Puanı 0,8 olanlar onaylanır. A'da bunların 40'ı öder (TP = 40), 10'u ödemez (FP = 10). B'de 16'sı öder (TP = 16), 4'ü ödemez (FP = 4).
2. **Reddedilenler.** Puanı 0,2 olanlar reddedilir. A'da bunların 10'u aslında ödeyecek kişilerdir (FN = 10), 40'ı doğru biçimde reddedilmiştir (TN = 40). B'de 16 FN ve 64 TN vardır.
3. **Denetim.** Her matrisin dört hücresinin toplamı 100 olmalıdır: A'da 40 + 10 + 10 + 40 = 100, B'de 16 + 4 + 16 + 64 = 100 ✓.
4. **Doğruluk.** Modelin doğru karar verdiği kişiler TP ve TN kutularındadır. **Doğruluk** (*accuracy*) $(TP + TN)/n$'dir. A'da $(40 + 40)/100 = 0{,}80$, B'de $(16 + 64)/100 = 0{,}80$'dir.

Model iki grupta da %80 doğrudur. İlk bakışta bu, modelin iki gruba eşit davrandığı izlenimini verir. Oysa bu izlenim eksiktir; ölçütler eksikliği sayılarla ortaya koyar.



Şekil 11.2: İki grubun karışıklık matrisleri. Satırlar gerçek sonucu (öder $Y = 1$, ödemez $Y = 0$), sütunlar modelin kararını (onay $\hat{Y} = 1$, ret $\hat{Y} = 0$) gösterir. A: TP = 40, FN = 10, FP = 10, TN = 40. B: TP = 16, FN = 16, FP = 4, TN = 64.

Şekil 11.2 iki matrisi yan yana gösterir. Matrisin *satırları* gerçeğe göre, *sütunları* karara göre ayrılır. Bu düzen, sonraki ölçütlerin hangi hücrelere baktığını görmeyi kolaylaştırır: satır üzerinden hesaplanan oranlar (TPR, FPR) "gerçek sonuç bilindiğinde model ne yaptı?" sorusunu, sütun üzerinden hesaplananlar (PPV) "karar bilindiğinde gerçek ne çıktı?" sorusunu sorar.

11.2.3 Demografik eşitlik

En sade adalet fikri, iki grubun eşit oranda onay almasıdır. Bu fikir modelin ne kadar isabetli olduğuna bakmaz; yalnız sonuçların dağılımına bakar.

Tanım 11.2

Demografik eşitlik (*demographic parity*), iki grubun seçim oranlarının eşit olmasıdır:

$$P(\hat{Y} = 1 \mid A) = P(\hat{Y} = 1 \mid B).$$

Burada $P(\hat{Y} = 1 \mid A) = (TP_A + FP_A) / n_A$, A grubunun **seçim oranıdır** (*selection rate*).

Örnek 11.2

1. **Onay sütunu.** Seçim oranı yalnız onay sütununa bakar. A'da onaylananlar 40 + 10 = 50, B'de 16 + 4 = 20 kişidir.
2. **Seçim oranları.** A: 50/100 = 0,50. B: 20/100 = 0,20. Fark 0,50 − 0,20 = 0,30'dur. Demografik eşitlik sağlanmaz.
3. **Beşte dört kuralı.** Küçük seçim oranının büyüğüne oranı 0,20 / 0,50 = 0,40'tır. Bu sayı 0,8'den küçük olduğu için aşağıda tanımlanan kural da ihlal edilir.
4. **Gruplar farklı.** Gerçekte ödeyenler A'da 50, B'de 32 kişidir. B'de seçim oranını 0,50'ye çıkarmak, onaylanan sayısını 20'den 50'ye çıkarmak demektir. B'de ödeyen yalnız 32 kişi olduğu için bu yeni onayların en az 50 − 32 = 18'i ödemeyecek kişilere gider. Tersine, A'yı 0,20'ye indirmek, ödeyecek 50 kişinin en az 30'unu reddetmek demektir.



Şekil 11.3: Demografik eşitliğin baktığı hücreler: onay sütunu (TP + FP). A'da 50, B'de 20 kişi onay alır; seçim oranları 0,50 ve 0,20.

ABD istihdam hukukunda bu ölçütün pratik bir sürümü kullanılır. İşe alımda bir grubun seçim oranı, seçim oranı en yüksek grubunkinin beşte dördünden (%80) azsa, bu durum dolaylı ayrımcılık etkisi (*disparate impact*) için ilk kanıt sayılır.

Tanım 11.3

Beşte dört kuralı (*four-fifths rule*), küçük seçim oranının büyük seçim oranına oranının en az 0,8 olmasını ister:

$$\min(s_A, s_B) / \max(s_A, s_B) \geq 0{,}8.$$

Kural bir *orana* bakar, bir *farka* değil. Bu ayrım önemlidir. Seçim oranları 0,50 ve 0,40 ise fark 0,10, oran 0,8'dir; kural sınırda sağlanır. Seçim oranları 0,10 ve 0,02 ise fark yalnız 0,08'dir ama oran 0,2'dir; kural açıkça ihlal edilir. Küçük oranlarda fark yanıltıcı biçimde küçük görünür.

Demografik eşitliğin çekici yanı, gerçek sonuç Y'yi bilmeyi gerektirmemesidir. Etiketlerin kendisi tarihsel önyargı taşıyorsa (11.1.3), etiketlere dayanmayan bir ölçüt avantajlıdır. Zayıf yanı ise yukarıdaki son adımda görülür: taban oranları gerçekten farklıysa bu ölçütü zorlamak ya ödemeyecek kişilere kredi vermek ya da ödeyecek kişileri reddetmek demektir. İkisi de bir tür hatadır ve bu hataların bedeli de bir gruba yüklenir.

11.2.4 Eşitlenmiş olasılıklar ve fırsat eşitliği

İkinci fikir, gerçek sonucu hesaba katar. Soru şudur: gerçek durumu aynı olan iki kişinin onay alma şansı, gruplarına göre değişiyor mu? Ödeyecek iki kişiden biri A'da, öteki B'de ise ikisinin onay şansı eşit olmalıdır. Aynı şey ödemeyecek kişiler için de geçerli olmalıdır.

Tanım 11.4

Eşitlenmiş olasılıklar (*equalized odds*; Hardt vd., 2016), gerçek sonuç koşulu altında hata oranlarının gruplar arasında eşit olmasıdır:

$$TPR_A = TPR_B \text{ ve } FPR_A = FPR_B.$$

Yalnız TPR eşitliği istenirse ölçütün adı **fırsat eşitliğidir** (*equal opportunity*).

Fırsat eşitliğinin adı içeriğini anlatır: "hak eden", yani krediyi ödeyecek olan herkesin onay alma fırsatı grubundan bağımsız olmalıdır. Eşitlenmiş olasılıklar buna ikinci bir koşul ekler: hak etmeyenlerin yanlışlıkla onay alma şansı da gruptan bağımsız olmalıdır.

Örnek 11.3

1. **TPR (öder satırı).** A: $40/(40 + 10) = 40/50 = 0{,}80$. B: $16/(16 + 16) = 16/32 = 0{,}50$. B'de krediyi ödeyecek olanların yarısı reddediliyor; A'da bu oran beşte birdir.
2. **FPR (ödemez satırı).** A: $10/(10 + 40) = 10/50 = 0{,}20$. B: $4/(4 + 64) = 4/68 \approx 0{,}059$.

3. **Karşılaştırma.** TPR farkı 0,80 − 0,50 = 0,30, FPR farkı 0,20 − 0,059 = 0,141'dir. İki oran da farklı olduğundan eşitlenmiş olasılıklar sağlanmaz. Yalnız TPR'ye bakan fırsat eşitliği de sağlanmaz.

Sonuç ilginç bir tablo çizer: B grubu iki yönden de A'dan farklı muamele görür. Ödeyecek B üyeleri daha sık reddedilir (düşük TPR); ödemeyecek B üyeleri ise daha seyrek onay alır (düşük FPR). Model B'ye karşı genel olarak “daha temkinli”dir.



Şekil 11.4: Eşitlenmiş olasılıkların baktığı hücreler. Üst sıra: TPR için "öder" satırı (A'da $40/50 = 0{,}80$, B'de $16/32 = 0{,}50$). Alt sıra: FPR için "ödemez" satırı (A'da $10/50 = 0{,}20$, B'de $4/68 \approx 0{,}059$).

Şekil 11.4 ölçütün satır bazlı olduğunu gösterir. Bu ölçütün bakış açısı başvuranınkidir. Ödeyebilecek bir başvuran şunu sorar: "Ödeyebilecek biri olarak onay alma şansım grubuma bağlı mı?" A'daki ödeyecek bir başvuranın şansı %80, B'deki ödeyecek bir başvuranınki %50'dir. Bu fark, başvuranın kendi davranışıyla değil, yalnız grubuyla açıklanır.

TPR'nin tümleyeni de sık kullanılır. **Yanlış negatif oranı** (FNR, *false negative rate*), ödeyecek olanlar arasında reddedilenlerin payıdır: $FNR = FN/(TP + FN) = 1 - TPR$. Örnekte A'da FNR = 0,20, B'de 0,50'dir. TPR eşitliği ile FNR eşitliği aynı koşuldur; COMPAS tartışmasında (11.3.5) FNR biçimi kullanılır.

11.2.5 Tahmin eşitliği

Üçüncü fikir bakış açısını kurumun tarafına taşır. Kurum, modelin "onay" kararını bir bilgi olarak kullanır. Bu bilginin anlamı iki grupta aynı olmalıdır: onay alan bir A üyesiyle onay alan bir B üyesi, kredi yi aynı olasılıkla ödemelidir.

Tanım 11.5

Tahmin eşitliği (*predictive parity*), onay alanların gerçekten ödeme oranının, yani PPV'nin gruplar arasında eşit olmasıdır:

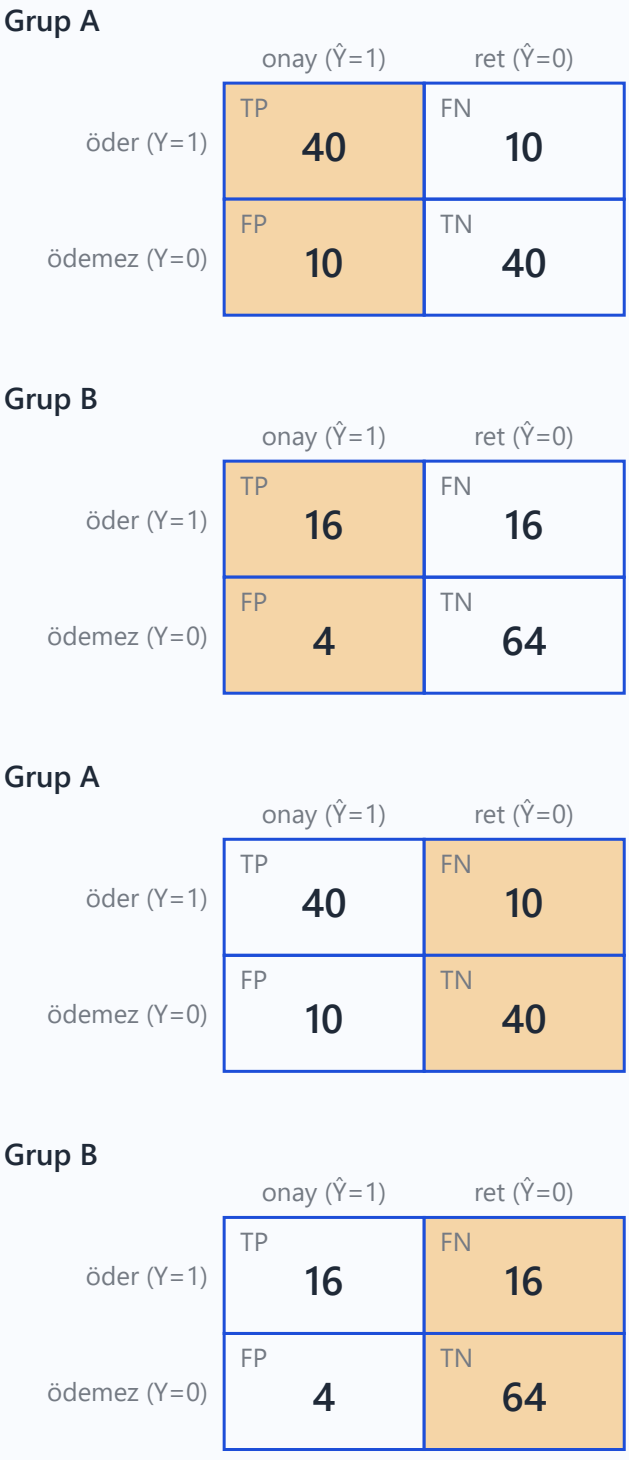
$$PPV_A = PPV_B, \text{ burada } PPV = TP/(TP + FP) = P(Y = 1 \mid \hat{Y} = 1).$$

Ret tarafının karşılığı **negatif kestirim değeridir** (NPV, *negative predictive value*): reddedilenlerin gerçekten ödemeyen payı, $NPV = TN/(TN + FN)$.

Örnek 11.4

1. **PPV (onay sütunu).** A: $40/(40 + 10) = 0{,}80$. B: $16/(16 + 4) = 16/20 = 0{,}80$. İki grupta da onaylananların %80'i öder; tahmin eşitliği sağlanır.
2. **NPV (ret sütunu).** A: $40/(40 + 10) = 40/50 = 0{,}80$. B: $64/(64 + 16) = 64/80 = 0{,}80$. Reddedilenlerin de iki grupta %80'i gerçekten ödemez.

"Onay" ve "ret" etiketleri iki grupta aynı anlamı taşır. Kurum için model iki grupta aynı ölçüde güvenilirdir.



Şekil 11.5: Tahmin eşitliğinin baktığı hücreler. Üst sıra: PPV için onay sütunu (A'da 40/50, B'de 16/20; ikisi de 0,80). Alt sıra: NPV için ret sütunu (A'da 40/50, B'de 64/80; ikisi de 0,80).

Aynı model, aynı sayılar: eşitlenmiş olasılıklara göre B grubu dezavantajlıdır, tahmin eşitliğine göre model iki gruba aynı davranır. Çelişki yoktur; iki ölçüt matrisin farklı yönlerine bakar. Şekil 11.4 satırları, Şekil 11.5 sütunları karşılaştırır. Hangi yönün önemli olduğu, kimin bakış açısının öncelikli sayıldığına bağlıdır.

11.2.6 Kalibrasyon

Şimdiye kadarki ölçütler modelin onay/ret *kararına* baktı. Oysa model önce bir *puan* üretir; karar bu puanın eşikle karşılaştırılmasıyla verilir. Puanın kendisi de bir iddiada bulunur: "bu kişinin ödeme olasılığı 0,8'dir." Kalibrasyon bu iddianın doğru olup olmadığını sorar.

Tanım 11.6

Kalibrasyon (grup içi kalibrasyon, *calibration within groups*): puanı s olan kişilerin gerçekten ödeme oranı, her grupta s'dir:

> $P(Y = 1 \mid \text{puan} = s, \text{grup} = g) = s$, her s ve her g için.

Kalibrasyon, puanın grup ne olursa olsun aynı riski anlatmasıdır. Bir hava durumu uygulaması düşünelim: "%30 yağmur" dediği günlerin gerçekten yaklaşık %30'unda yağmur yağıyorsa uygulama kalibredir. Kalibrasyonun grup içi olması, bu denetimin her grup için ayrı yapılması demektir. Bir model toplamda kalibre olup bir grupta puanları sistematik olarak şişirebilir.

Örnek 11.5

Tablo 11.2'deki dört satırın her biri için gerçekleşme oranı hesaplanır ve puanla karşılaştırılır.

1. **A grubu, puan 0,8:** 50 kişiden 40'ı öder ⇒ 40/50 = 0,80. Puanla aynı.
2. **A grubu, puan 0,2:** 50 kişiden 10'u öder ⇒ 10/50 = 0,20. Puanla aynı.
3. **B grubu, puan 0,8:** 20 kişiden 16'sı öder ⇒ 16/20 = 0,80. Puanla aynı.
4. **B grubu, puan 0,2:** 80 kişiden 16'sı öder ⇒ 16/80 = 0,20. Puanla aynı.

Dört satırın dördünde de puan gerçekleşme oranına eşittir. Model iki grupta da kalibredir: puan dürüst bir olasılıktır.

grup	puan s	kişi	öder	gerçekleşme oranı	s ile aynı mı?
A	0,80	50	40	40/50 = 0,80	evet
A	0,20	50	10	10/50 = 0,20	evet
B	0,80	20	16	16/20 = 0,80	evet
B	0,20	80	16	16/80 = 0,20	evet

Tablo 11.3: Kalibrasyon denetimi. Gerçekleşme oranı = öder / kişi.

Kalibrasyonla tahmin eşitliği yakından ilişkilidir ama aynı şey değildir. Tahmin eşitliği yalnız "onay" sütununa, yani eşiğin üstündeki herkese birlikte bakar. Kalibrasyon her puan düzeyine ayrı ayrı bakar. Bu örnekte eşiğin üstünde tek bir puan (0,8) olduğu için kalibrasyon PPV eşitliğini doğrudan verir:

onaylananların hepsinin puanı 0,8'dir ve kalibre bir modelde bunların %80'i öder. Daha çok puan düzeyi olan bir modelde iki gruptaki onaylananların puan karışımı farklı olabilir; o zaman kalibre bir model bile farklı PPV'ler verebilir.

Kalibre bir modelin ilginç bir özelliği vardır: bir gruptaki ortalama puan, o grubun taban oranına eşittir. A'da ortalama puan *(50 · 0,8 + 50 · 0,2)/100 = 0,50 =* p_A, B'de *(20 · 0,8 + 80 · 0,2)/100 = 0,32 =* p_B'dir. Bu yüzden taban oranları farklıysa kalibre bir model iki gruba ortalamada farklı puanlar vermek *zorundadır*. Bu gözlem 11.3.3'teki Kleinberg teoreminin özüdür.

11.2.7 Kod izleme: olcutler()

Dört grup ölçütünün hepsi karışıklık matrisinin dört sayısından hesaplanır. Aşağıdaki kısa Python fonksiyonu bu hesabı yapar ve iki grubun sonuçlarını yan yana yazdırır.

```
def olcutler(TP, FP, FN, TN):
  n = TP + FP + FN + TN
  secim = (TP + FP) / n   # demografik eşitlik
  tpr = TP / (TP + FN)    # eşitlenmiş olasılıklar
  fpr = FP / (FP + TN)
  ppv = TP / (TP + FP)    # tahmin eşitliği
  return secim, tpr, fpr, ppv

A = olcutler(40, 10, 10, 40)
B = olcutler(16, 4, 16, 64)
ADLAR = ["seçim", "TPR", "FPR", "PPV"]
for i, ad in enumerate(ADLAR):
  print(ad, round(A[i], 3), round(B[i], 3))
```

Fonksiyon dört argüman alır: TP, FP, FN ve TN. İlk satırda toplam kişi sayısı `n` hesaplanır. Sonraki dört satır, Tablo 11.1'deki tanımları birebir uygular. Yorumlar her oranın hangi ölçütte kullanıldığını söyler: `secim` demografik eşitlikte, `tpr` ve `fpr` eşitlenmiş olasılıklarda, `ppv` tahmin eşitliğinde karşılaştırılır. Fonksiyon dört sayıyı bir Python demeti (*tuple*) olarak döndürür. Demet, virgülle ayrılmış değerlerin değiştirilemeyen sıralı bir listesidir; `A[0]` ilk değeri, `A[3]` son değeri verir.

Programın alt kısmı fonksiyonu iki grubun matrisiyle çağırır. `for i, ad in enumerate(ADLAR)` satırı, listedeki her ad için hem sırasını (`i`) hem de adın kendisini (`ad`) verir. Böylece `A[i]` ve `B[i]` aynı oranın iki gruptaki değerleridir.

Satır	A grubu	B grubu
`n`	40 + 10 + 10 + 40 = 100	16 + 4 + 16 + 64 = 100
`secim`	(40 + 10)/100 = 0,5	(16 + 4)/100 = 0,2
`tpr`	40/(40 + 10) = 0,8	16/(16 + 16) = 0,5
`fpr`	10/(10 + 40) = 0,2	4/(4 + 64) ≈ 0,0588
`ppv`	40/(40 + 10) = 0,8	16/(16 + 4) = 0,8

Satır	A grubu	B grubu
dönüş	(0,5; 0,8; 0,2; 0,8)	(0,2; 0,5; 0,0588…; 0,8)

Tablo 11.4: `olcutler()` izlemesi. A = olcutler(40, 10, 10, 40), B = olcutler(16, 4, 16, 64).

Döngü dört satır yazdırır: `seçim 0.5 0.2`, `TPR 0.8 0.5`, `FPR 0.2 0.059`, `PPV 0.8 0.8`. Çıktı satır satır okunduğunda ölçütlerin durumu hemen görülür. `i = 0`'da seçim oranları farklıdır (0,5 ≠ 0,2): demografik eşitlik yok. `i = 1`'de TPR farklıdır: fırsat eşitliği yok. `i = 2`'de FPR de farklıdır: eşitlenmiş olasılıklar yok. `i = 3`'te PPV eşittir: tahmin eşitliği var.

Örnek 11.6

Üçüncü bir C grubunun matrisi TP = 30, FP = 20, FN = 10, TN = 40 olsun. Fonksiyon `olcutler(30, 20, 10, 40)` ile çağrılır.

1. $n = 30 + 20 + 10 + 40 = 100$.
2. Seçim oranı $(30 + 20)/100 = 0{,}5$.
3. TPR $= 30/(30 + 10) = 30/40 = 0{,}75$.
4. FPR $= 20/(20 + 40) = 20/60 \approx 0{,}333$.
5. PPV $= 30/(30 + 20) = 30/50 = 0{,}6$.

C'yi A ile karşılaştıralım: A'nın dört değeri (0,5; 0,8; 0,2; 0,8)'dir. Yalnız seçim oranları eşittir (0,5 = 0,5); demografik eşitlik sağlanır. TPR (0,75 ≠ 0,8), FPR (0,333 ≠ 0,2) ve PPV (0,6 ≠ 0,8) farklıdır. Demek ki iki grubun aynı oranda onay alması, onayların aynı isabetle dağıtıldığı anlamına gelmez.

11.2.8 Dört ölçüt, tek model: sonuç tablosu

Hesapların hepsi bir tabloda toplanabilir. Tablo, aynı modelin bir ölçüte göre adil, ötekine göre adaletsiz sayılabileceğini açıkça gösterir.

Ölçüt	Eşit olması gereken	A	B	Sağlanıyor mu?
Demografik eşitlik	$P(\hat{Y} = 1)$	0,50	0,20	hayır
Fırsat eşitliği	TPR = $P(\hat{Y} = 1 \mid Y = 1)$	0,80	0,50	hayır
Eşitlenmiş olasılıklar	TPR ve FPR = $P(\hat{Y} = 1 \mid Y = 0)$	0,80; 0,20	0,50; 0,059	hayır
Tahmin eşitliği	PPV = $P(Y = 1 \mid \hat{Y} = 1)$	0,80	0,80	evet
Kalibrasyon	$P(Y = 1 \mid \text{puan} = s) = s$	0,8→0,8; 0,2→0,2	0,8→0,8; 0,2→0,2	evet

Tablo 11.5: Dört ölçüt, tek model. Örnek modelin bütün ölçütlere göre durumu.

Tablodaki ölçütler iki bakış açısına ayrılır:

- **Başvuranın bakışı:** "Ödeyebilecek biri olarak onay alma şansım grubuma bağlı mı?" Bu soru gerçek sonucu sabit tutar ve kararın dağılımına bakar ⇒ TPR ve FPR (fırsat eşitliği, eşitlenmiş olasılıklar).
- **Kurumun bakışı:** "Onay verdiğim kişinin ödeme olasılığı grubuna bağlı mı?" Bu soru kararı ya da puanı sabit tutar ve gerçek sonucun dağılımına bakar ⇒ PPV ve kalibrasyon.

Demografik eşitlik bu iki bakışın dışında kalır; gerçek sonucu hiç kullanmaz ve yalnız fırsatların sayısal dağılımıyla ilgilenir. Örnek model kurumun ölçütlerini sağlar, başvuranınkileri sağlamaz. Bu bir rastlantı değildir: taban oranları farklı olduğu sürece kaçınılmazdır (11.3).

11.2.9 Grup adaletinin ötesi: bireysel adalet

Yukarıdaki ölçütlerin hepsi **grup adaleti** ölçütleridir: grupların ortalama oranlarını karşılaştırırlar. Ortalamalar eşitlenirken bireylere keyfî davranılabilir. Örneğin B grubundan rastgele seçilen 30 kişiyi ek olarak onaylamak, B'nin seçim oranını 0,50'ye çıkarır ve demografik eşitliği sağlar. Ama hangi 30 kişinin onay aldığı tamamen şansa bağlıdır; birbirine çok benzeyen iki B üyesinden biri onay alır, öteki almaz. Dwork ve arkadaşları (2012) bu soruna farklı bir ilkeyle yaklaşır: benzer kişilere benzer davranılmalıdır.

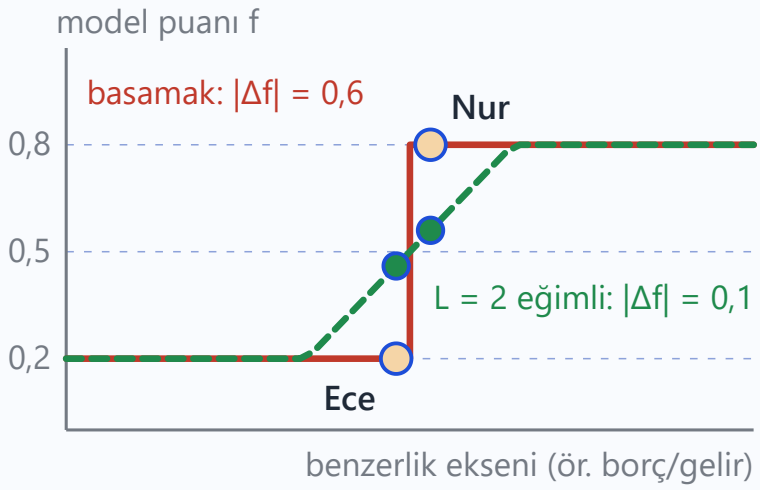
Tanım 11.7

Bireysel adalet (*individual fairness*), model puanının **Lipschitz koşulunu** (*Lipschitz condition*) sağlamasıdır: her iki kişi x ve y için

$$|f(x) - f(y)| \le L \cdot d(x, y).$$

Burada $f(x)$, x kişisinin model puanıdır; $d(x, y)$ iki kişinin birbirinden ne kadar farklı olduğunu ölçen uzaklıktır; L izin verilen en büyük eğimdir.

Koşul şunu söyler: iki kişinin puan farkı, aralarındaki farkın L katını aşmamalıdır. $d(x, y)$ küçükse, yani kişiler benzerse, puanları da birbirine yakın olmak zorundadır. "Lipschitz" adı matematikteki bir süreklilik koşulundan gelir: grafiği hiçbir yerde L'den dik olmayan fonksiyonlar bu koşulu sağlar. Ani sıçrama yapan bir fonksiyon, sıçrama noktasının iki yanındaki çok yakın iki noktaya çok farklı değerler verdiği için koşulu ihlal eder.



Şekil 11.6: Bireysel adalet. Yatay eksen bir benzerlik ölçüsü (örneğin borç/gelir oranı), dikey eksen model puanıdır. Kırmızı basamak model, eşiğin iki yanındaki Ece ve Nur'a 0,2 ve 0,8 verir: $|\Delta f| = 0{,}6$. Yeşil kesikli model $L = 2$ eğimiyle yükselir ve aynı iki kişiye en çok 0,1 fark verir.

Örnek 11.7

Şekil 11.6'daki Ece ile Nur arasındaki uzaklık $d = 0{,}05$, izin verilen eğim $L = 2$'dir.

1. **İzin verilen en büyük fark:** $L \cdot d = 2 \cdot 0{,}05 = 0{,}1$.
2. **Basamak model:** Ece'ye 0,2, Nur'a 0,8 verir. Fark $|0{,}8 - 0{,}2| = 0{,}6$'dır. $0{,}6 > 0{,}1$ olduğu için koşul ihlal edilir.
3. **Yumuşak model:** Puanı eğimi 2'yi aşmayacak biçimde yavaş yavaş artırır. İki kişinin puan farkı en çok $2 \cdot 0{,}05 = 0{,}1$ olur; koşul sağlanır.
4. **Bir aralık sorusu:** $L = 2$, $d(x, y) = 0{,}1$ ve $f(x) = 0{,}8$ olsun. Koşul $|f(y) - 0{,}8| \le 2 \cdot 0{,}1 = 0{,}2$ verir; yani $0{,}6 \le f(y) \le 1{,}0$. Üst sınır zaten puanın alabileceği en büyük değer olduğu için izin verilen aralık [0,6; 1]'dir.

Bireysel adalet çekici bir ilkedir ama bir zorluğu vardır: uzaklık ölçüsü d'yi kim belirler? "Kimler benzer?" sorusu da bir değer yargısıdır. Borç/gelir oranı aynı olan iki kişi benzer midir, yoksa birinin düzensiz geliri onu farklı mı kılar? Bireysel adalet sorunu ortadan kaldırmaz; onu d'nin seçimine taşır. Ayrıca koşul, puanın kendisini değil yalnız puanlar arasındaki farkı sınırlar. Herkese 0 veren bir model de koşulu sağlar. Bu yüzden bireysel adalet, isabetli bir modelle birlikte düşünülmesi gereken bir kısıttır.

11.2.10 Ölçüt hesaplarında sık hatalar

Sık yapılan hata

- **TPR ile PPV'yi karıştırmak:** İkisinin de payı TP'dir ama paydaları farklıdır. TPR'de payda ödeyenlerdir (TP + FN), PPV'de onay alanlardır (TP + FP). B grubunda TPR = 16/32 = 0,5, PPV = 16/20 = 0,8'dir.
- **FPR'nin paydasına FN yazmak:** FPR'nin paydası ödemeyenlerin toplamıdır: FP + TN. B'de 4/(4 + 64) = 4/68'dir, 4/20 değildir.
- **Beşte dört kuralında fark almak:** Kural oran ister: küçük ÷ büyük. Ters oran (büyük ÷ küçük) de yanlıştır; 0,50/0,20 = 2,5 sonucu kuralı "sağlıyor" gibi gösterir.
- **Doğruluğu adalet sanmak:** Örnekte iki grubun doğruluğu da %80'dir ama TPR'ler 0,8 ve 0,5'tir. Eşit doğruluk, hataların aynı türden olduğunu ya da aynı kişilere düştüğünü göstermez.
- **Kalibrasyonu toplamda denetlemek:** Kalibrasyon her grup içinde ayrı denetlenir. İki grubun verisi birleştirilerek yapılan denetim, gruplardan birindeki sistematik sapmayı gizleyebilir.
- **Bireysel adalette *L*'yi unutmak:** İzin verilen fark d değil $L \cdot d$'dir.

11.3 Ölçütlerin uyumsuzluğu

Önceki kısımdaki model tahmin eşitliğini ve kalibrasyonu sağladı ama eşitlenmiş olasılıkları sağlamadı. Akla gelen ilk düşünce, daha iyi bir modelin hepsini birden sağlayabileceğidir. Oysa bu düşünce genelde yanlıştır. Ölçütler aynı dört sayıdan türetildiği için birbirine bağlıdır. Taban oranları farklıysa, mükemmel tahmin dışında hiçbir model hepsini birden sağlayamaz. Bu tür sonuçlara **uyumsuzluk teoremleri** (*impossibility theorems*) denir.

11.3.1 Chouldechova özdeşliği

Chouldechova (2017) ölçütler arasındaki bağı tek bir eşitlikle yazmıştır. Eşitliği türetmek için bir gruptaki karışıklık matrisini taban oranı p, TPR ve FPR cinsinden ifade etmek yeterlidir.

Teorem 11.1 (Chouldechova özdeşliği)

Taban oranı p olan bir grupta, $0 < p < 1$ ve $PPV > 0$ iken

$$FPR = \frac{p}{1-p} \cdot \frac{1-PPV}{PPV} \cdot TPR.$$

İspat. **1. Matris hücrelerini oranlarla yazmak.** Grupta n kişi olsun. Taban oranı tanımı gereği ödeyenler $p \cdot n$, ödemeyenler $(1 - p) \cdot n$ kişidir. TPR ödeyenlerin onay alan payı olduğundan $TP = p \cdot n \cdot TPR$'dir. FPR ödemeyenlerin onay alan payı olduğundan $FP = (1 - p) \cdot n \cdot FPR$'dir.

2. PPV'yi yerine koymak. $PPV = TP / (TP + FP)$'dir. İki ifadeyi yerine koyup pay ve paydadaki ortak n'yi sadeleştirince $PPV = p \cdot TPR / (p \cdot TPR + (1 - p) \cdot FPR)$ bulunur.

3. Paydayla çarpmak. İki taraf paydayla çarpılır: $PPV \cdot [p \cdot TPR + (1 - p) \cdot FPR] = p \cdot TPR$. Sol taraf açılır ve $PPV \cdot p \cdot TPR$ terimi sağa geçirilir: $(1 - p) \cdot FPR \cdot PPV = p \cdot TPR - p \cdot TPR \cdot PPV = p \cdot TPR \cdot (1 - PPV)$.

4. FPR'yi yalnız bırakmak. İki taraf $(1 - p) \cdot PPV$'ye bölünür; bu sayı $p < 1$ ve $PPV > 0$ olduğu için sıfır değildir. Sonuç $FPR = p/(1 - p) \cdot (1 - PPV)/PPV \cdot TPR$'dir. □

Özdeşlik, üç oranı (FPR, TPR, PPV) ve taban oranını birbirine bağlar. Bunlardan herhangi üçü bilinirse dördüncüsü hesaplanabilir. Örnek modelin sayıları da bu bağı izler.

Örnek 11.8

1. **A grubu:** $p = 0{,}5$, $PPV = 0{,}8$, $TPR = 0{,}8$. Çarpanlar: $p/(1 - p) = 0{,}5/0{,}5 = 1$; $(1 - PPV)/PPV = 0{,}2/0{,}8 = 0{,}25$. Sonuç: $FPR = 1 \cdot 0{,}25 \cdot 0{,}8 = 0{,}20$ ✓ (tabloyla aynı).
2. **B grubu:** $p = 0{,}32$, $PPV = 0{,}8$, $TPR = 0{,}5$. Çarpanlar: $0{,}32/0{,}68 = 8/17 \approx 0{,}471$; $0{,}25$. Sonuç: $FPR = (8/17) \cdot 0{,}25 \cdot 0{,}5 = 1/17 = 4/68 \approx 0{,}059$ ✓.
3. **Yorum:** İki grupta PPV eşittir (0,8), bu yüzden ortadaki çarpan $(1 - PPV)/PPV$ iki grupta da 0,25'tir. Ama $p/(1 - p)$ çarpanı A'da 1, B'de 0,471'dir.

Asıl soru şudur: PPV eşitken TPR ve FPR de eşit olabilir mi? İki grupta PPV ve TPR eşit olsun. O zaman özdeşliğin sağ tarafında yalnız $p/(1 - p)$ çarpanı gruptan gruba değişir. p farklıysa bu çarpan da farklıdır (çünkü $p/(1 - p)$, p büyüdükçe kesin olarak büyür). Dolayısıyla FPR'ler farklı çıkar. TPR'ler de FPR'ler de eşit olacaksa, ancak $TPR = 0$ ya da $PPV = 1$ gibi uç durumlarda sağ taraflar eşitlenebilir. $PPV = 1$, onay alan herkesin ödediği, yani hiç yanlış pozitif olmayan durumdur; $TPR = 0$ ise kimseye kredi verilmeyen durumdur.

Teorem 11.2 (Tahmin eşitliği ile eşitlenmiş olasılıkların uyumsuzluğu)

İki grubun taban oranları farklıysa ($p_A \neq p_B$), tahmin eşitliği ve eşitlenmiş olasılıklar, mükemmel tahmin (FPR = 0 ve PPV = 1) ya da kimseye onay vermeme gibi uç durumlar dışında birlikte sağlanamaz.

İspat. Tahmin eşitliği ve eşitlenmiş olasılıklar birlikte sağlansın: $PPV_A = PPV_B = P$, $TPR_A = TPR_B = T$, $FPR_A = FPR_B$. Özdeşlik iki grup için yazılır ve FPR'ler eşitlenir: $p_A/(1 - p_A) \cdot (1 - P)/P \cdot T = p_B/(1 - p_B) \cdot (1 - P)/P \cdot T$. $(1 - P)/P \cdot T$ sıfır değilse iki taraf ona bölünür ve $p_A/(1 - p_A) = p_B/(1 - p_B)$ kalır. Bu fonksi-

yon kesin artan olduğu için $p_A = p_B$ çıkar; bu, varsayımla çelişir. Demek ki $(1 - P)/P \cdot T = 0$ olmalıdır, yani ya $P = 1$ ya da $T = 0$'dır. İki durumda da özdeşlik FPR = 0 verir. □

Teoremin söylediği şey bir model hatası değil, bir aritmetik gerçektir. Ne kadar veri toplanırsa toplansın ve ne kadar gelişmiş bir algoritma kullanılırsa kullanılsın, taban oranları farklı ve tahmin kusurlu olduğu sürece bir ölçüt seçmek, ötekinden vazgeçmek demektir.

Örnek 11.9

Bir grupta $p = 0{,}2$, $PPV = 0{,}5$, $TPR = 0{,}6$ verilmiş olsun. FPR kaçtır?

1. $p/(1 - p) = 0{,}2/0{,}8 = 0{,}25$.
2. $(1 - PPV)/PPV = 0{,}5/0{,}5 = 1$.
3. $FPR = 0{,}25 \cdot 1 \cdot 0{,}6 = 0{,}15$.

Denetim için 100 kişilik bir grup kurulabilir: 20 kişi öder, 80 kişi ödemez. TP = 20 · 0,6 = 12, FP = 80 · 0,15 = 12. PPV = 12/24 = 0,5 ✓.

11.3.2 Eşitlenmiş olasılıkları zorlarsak ne olur?

Uyumsuzluğun ne demek olduğunu somut görmek için ters yönden gidelim. B grubu için A ile aynı hata oranlarını veren, varsayımsal ve daha iyi ayırt eden bir model olsun. Bu modelde $TPR_B = 0{,}8$ ve $FPR_B = 0{,}2$'dir. Bugünkü model, yalnız 0,8 ve 0,2 puanı verdiği için, eşiği değiştirerek ya da rastgele onay vererek bu iki hata oranına birlikte ulaşamaz. Yeni modelin bu oranlara nasıl ulaştığını bilmemize gerek yoktur; karışıklık matrisi yalnız hata oranlarından hesaplanabilir.

Örnek 11.10

1. **Hedef:** B'de TPR = 0,8 ve FPR = 0,2 (A ile aynı). B'de 32 kişi öder, 68 kişi ödemez.
2. **Matrisin satırları:** Hata oranları sabitlenince karışıklık matrisinin üst ve alt satırları belirlenir. TP = 32 × 0,8 = 25,6; FP = 68 × 0,2 = 13,6. Bunlar beklenen sayılardır; bu yüzden kesirli çıkabilirler.
3. **Seçim oranı:** Onaylanan = 25,6 + 13,6 = 39,2 kişi ⇒ seçim oranı ≈ 0,39. B'de onay oranı 0,20'den 0,39'a çıkar.
4. **PPV:** $PPV_B = 25{,}6/39{,}2 \approx 0{,}65$. A'da PPV 0,80'di. Onaylananların ödeme oranı B'de düşer.

Eşitlenmiş olasılıklar sağlandı, ama tahmin eşitliği bozuldu: B'deki "onay" artık daha düşük bir ödeme olasılığı anlatıyor. Model yalnız onay/ret verdiği için bozulan ölçüt PPV'dir.

Sonuç özdeşlikle de bulunabilir. B için $0{,}2 = (8/17) \cdot (1 - PPV)/PPV \cdot 0{,}8$ yazılırsa $(1 - PPV)/PPV = 0{,}2 \cdot 17/(8 \cdot 0{,}8) = 0{,}53125$ ve $PPV = 1/1{,}53125 \approx 0{,}653$ çıkar. İki yol aynı sonucu verir.

Hangi ölçütün korunması gerektiği sorusuna matematik cevap vermez. Cevap bağlama bağlıdır: hangi zararın daha ağır olduğu (ödeyebilecek birinin reddedilmesi mi, ödeyemeyecek birinin borçlandırılması mı) ve hukukun ne istediği. Ödeyemeyecek birine kredi vermek ona da zarar verebilir; geri ödenmeyen kredi kişinin kredi geçmişini bozar. Bu yüzden "daha çok onay her zaman daha adildir" sonucu da çıkmaz.

11.3.3 Kalibrasyon ve sınıf dengesi: Kleinberg teoremi

Chouldechova özdeşliği onay/ret kararları içindir. Kleinberg, Mullainathan ve Raghavan (2017) aynı sorunu puan düzeyinde incelemiştir. Puan düzeyinde eşitlenmiş olasılıkların karşılığı **sınıf dengesidir** (*balance*).

Tanım 11.8

Pozitif sınıf dengesi: gerçekte ödeyenlerin ($Y = 1$) ortalama puanı gruplar arasında eşittir.
Negatif sınıf dengesi: gerçekte ödemeyenlerin ($Y = 0$) ortalama puanı gruplar arasında eşittir.

Pozitif sınıf dengesinin anlamı, TPR eşitliğinin anlamına benzer: gerçek durumu aynı olan kişiler (ödeyenler) ortalamada aynı puanı almalıdır. Fark, karar yerine puana bakılmasıdır.

Örnek 11.11

1. **A, ödeyenler ($Y = 1$):** 50 ödeyenden 40'ı 0,8, 10'u 0,2 almıştır. Ortalama $(40 \cdot 0{,}8 + 10 \cdot 0{,}2)/50 = (32 + 2)/50 = 0{,}68$.
2. **B, ödeyenler:** 32 ödeyenden 16'sı 0,8, 16'sı 0,2 almıştır. Ortalama $(16 \cdot 0{,}8 + 16 \cdot 0{,}2)/32 = (12{,}8 + 3{,}2)/32 = 0{,}50$. Kalibre modelimizde B'nin ödeyen üyeleri ortalamada daha düşük puan alıyor: pozitif sınıf dengesi yok.
3. **A, ödemeyenler ($Y = 0$):** 50 ödemeyenden 10'u 0,8, 40'ı 0,2 almıştır. Ortalama $(10 \cdot 0{,}8 + 40 \cdot 0{,}2)/50 = (8 + 8)/50 = 0{,}32$.
4. **B, ödemeyenler:** 68 ödemeyenden 4'ü 0,8, 64'ü 0,2 almıştır. Ortalama $(4 \cdot 0{,}8 + 64 \cdot 0{,}2)/68 = (3{,}2 + 12{,}8)/68 = 16/68 \approx 0{,}235$. Negatif sınıf dengesi de yok.

Kalibrasyon sağlandığı hâlde iki denge de bozuktur.

Teorem 11.3 (Kleinberg, Mullainathan, Raghavan, 2017)

Bir risk puanı için grup içi kalibrasyon, pozitif sınıf dengesi ve negatif sınıf dengesi ancak iki durumda birlikte sağlanabilir: tahmin mükemmelse (her kişi 0 ya da 1 puan alır ve puanı gerçek sonucuna eşittir) ya da grupların taban oranları eşitse.

Teoremin tam ispatı uzundur, ama ana fikri 11.2.6'daki gözlemle görülebilir. Kalibre bir modelde bir grubun ortalama puanı taban oranına eşittir. Bu ortalama, ödeyenlerin ortalamasıyla ödemeyenlerin ortalamasının ağırlıklı ortalamasıdır: $p = p \cdot$ *(ödeyenlerin ortalaması)* $+ (1 - p) \cdot$ *(ödemeyenlerin ortalaması)*. İki sınıf dengesi sağlanırsa parantez içindeki iki ortalama iki grupta aynı olur; bu iki sayıya x ve y diyelim. O zaman her grup için $p = p \cdot x + (1 - p) \cdot y$, yani $p \cdot (1 - x + y) = y$ sağlanmalıdır. $1 - x + y$ sıfır değilse bu eşitlik p'yi tek bir değere bağlar ve iki grubun taban oranı eşit çıkar. Taban oranları farklıysa geriye $1 - x + y = 0$ kalır; bu ise $y = 0$ ve $x = 1$ ile, yani mükemmel tahminle mümkündür (ortalama puanlar 0 ile 1 arasında olduğundan $x - y = 1$ ancak böyle olur). Örneğimizde taban oranları farklı (0,5 ve 0,32) ve tahmin mükemmel değildir; ölçütlerin çatışması kaçınılmazdır.

11.3.4 Bu model adil mi? Cevap ölçüte bağlı

Hesaplar bir araya konunca “Bu model adil mi?” sorusunun tek bir cevabı olmadığı ortaya çıkar. İki tablo bütün durumu özetler.

grup	onay: öder / ödemez	ret: öder / ödemez	taban oranı
A	40 / 10	10 / 40	0,50
B	16 / 4	16 / 64	0,32

Tablo 11.6: Örnek modelin iki grupta kararları (her grupta 100 başvuran, aynı model, aynı eşik 0,5).

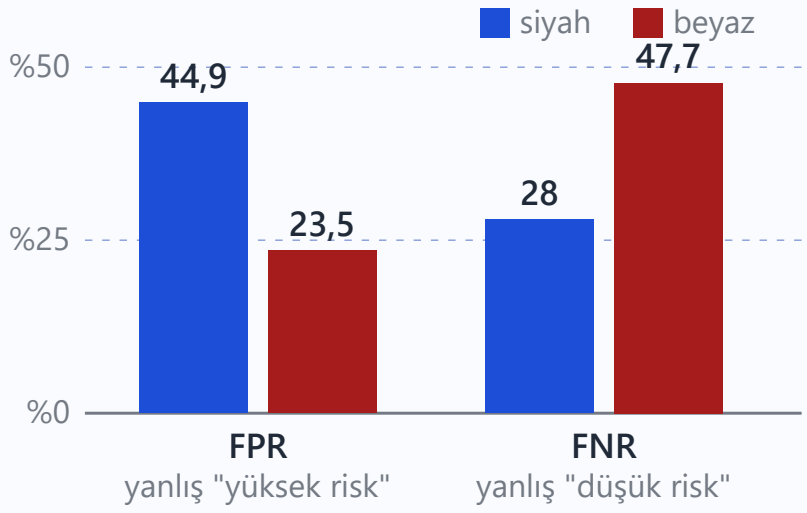
Ölçüt	A	B	Sağlanıyor mu?
Demografik eşitlik (onay oranı)	0,50	0,20	hayır
Eşitlenmiş olasılıklar (TPR; FPR)	0,80; 0,20	0,50; 0,059	hayır
Tahmin eşitliği (PPV)	0,80	0,80	evet

Tablo 11.7: Üç ölçüte göre sonuç.

Model kurumun ölçütüne göre adildir, başvuranın ölçütlerine göre adil değildir. Taban oranları farklı olduğu için (0,50 ve 0,32) bu ölçütler, tahmin mükemmel olmadıkça, aynı anda sağlanamaz (Chouldechova, 2017; Kleinberg vd., 2017). Hangi ölçütün seçileceği bir değer kararıdır. Bu kararın teknik bir ayrıntı gibi geliştiricilere bırakılması da bir seçimdir. Kararı kimin verdiği ve gerekçesinin yazılı olup olmadığı, 11.5.4'te anlatılan denetimin konusudur.

11.3.5 Gerçek bir örnek: COMPAS tartışması

COMPAS, ABD'de bazı mahkemelerde kullanılan bir risk değerlendirme aracıdır. Sanığın iki yıl içinde yeniden suç işleme riskini puanlar. 2016'da ProPublica adlı haber kuruluşu, Florida'daki Broward County'den elde ettiği verilerle aracı inceledi (Angwin vd., 2016).



Şekil 11.7: ProPublica'nın (2016) Broward County verisiyle hesapladığı hata oranları (yüzde). FPR: yeniden suç işlemeyen sanıklardan "yüksek risk" denenlerin payı (siyah %44,9, beyaz %23,5). FNR: yeniden suç işleyenlerden "düşük risk" denenlerin payı (siyah %28,0, beyaz %47,7).

İki taraf iki farklı ölçüte baktı:

- **ProPublica:** Yeniden suç işlemeyen siyah sanıkların %44,9'u "yüksek riskli" etiketlenmişti; beyazlarda bu oran %23,5'ti. Yani FPR eşit değildi. Yanlış negatif oranı (FNR) için tablo tersine dönüyordu: yeniden suç işleyen beyaz sanıkların %47,7'sine "düşük risk" denmişti, siyahlarda bu oran %28,0'dı. ProPublica bu bulgularla aracın siyah sanıklara karşı önyargılı olduğunu ileri sürdü.
- **Northpointe (aracı geliştiren şirket):** Aynı risk puanı iki grupta benzer yeniden suç oranına karşılık geliyordu. Yani araç kalibreydi ve tahmin eşitliğini sağlıyordu. Şirket bu gerekçeyle aracın adil olduğunu savundu.

İki taraf da kendi hesabında haklıydı. İki grubun yeniden suç kayıtlarındaki taban oranları farklı olduğu için iki ölçüt birlikte sağlanamazdı (Teorem 11.1'deki Chouldechova özdeşliği). Tartışma aslında hangi adalet tanımının seçileceğiyle ilgilidir. Chouldechova'nın 2017 makalesi tam olarak bu tartışmaya cevap olarak yazılmıştır. Ders şudur: tek bir doğruluk puanı yerine grup bazlı hata oranları raporlanmalıdır. Ayrıca bir taraf "model adil" dediğinde ilk sorulacak soru "hangi ölçüte göre?" olmalıdır.

Örneğin yapısı bölümün sayısal örneğine çok benzer. COMPAS'ta taban oranı yüksek olan grubun (kayıtlarda yeniden suç oranı daha yüksek görünen grubun) FPR'si de daha yüksektir. Bu da özdeşliğin öngördüğü yöndür: PPV eşitken p büyüdükçe FPR büyür. Bölümün örneğinde taban oranı yüksek olan A grubunun FPR'si (0,20) B'ninkinden (0,059) büyüktür.

11.3.6 Adalet ve doğruluk ödünleşimi

Model doğruluğu tek başına hataların kimler üzerinde yoğunlaştığını göstermez. Örneğimizde doğruluk iki grupta da %80'dir, ama TPR 0,8'e 0,5'tir. Doğruluk iki tür hatayı (FP ve FN) toplar ve aralarındaki farkı siler.

Bir adalet kısıtı eklemek toplam hatayı artırabilir. Bu bir **ödünleşimdir** (*trade-off*): bir şeyi kazanmak için ötekinden vazgeçmek. Ama asıl seçim "ne kadar doğruluk feda edilir?" sorusundan daha incedir. Asıl seçim, hangi hataların kabul edilebilir olduğudur: yanlış ret mi, yanlış onay mı? Aynı toplam hata, farklı gruplara farklı biçimde dağıtılabilir.

Adalet kısıtı modelin geliştirilme sürecinin üç noktasında eklenebilir:

Ön işleme
: Model eğitilmeden önce veri değiştirilir. Örneğin az temsil edilen grubun örneklerine daha çok ağırlık verilir ya da o gruptan daha çok veri toplanır. Amaç temsil eksikliğini gidermektir.

Eğitim sırasında
: Eğitimde en aza indirilen hata ölçüsüne, yani **kayıp fonksiyonuna** (*loss function*), adalet kısıtı ya da ceza terimi eklenir. Model hem az hata yapmaya hem de grup oranlarını yakın tutmaya çalışır; bunun için bir miktar doğruluktan vazgeçer.

Son işleme
: Eğitilmiş modelin puanları değiştirilmeden kararı veren eşik gruba göre ayarlanır (Hardt vd., 2016). Bu yöntem basittir, ama kararda grup üyeliğini açıkça kullandığı için bazı hukuk sistemlerinde tartışmalı olabilir.

Hangi yöntemin ve hangi ölçütün seçileceğine kullanım bağlamı, hukuki gereklilikler, düzenlemeler ve etkilenen grupların görüşleri yön vermelidir. Örneğin düşük gelirli başvuranları adil değerlendirmek isteyen bir kredi skorlaması, bu grubun ödeyebilecek üyelerinin reddedilmesini (düşük TPR) en ağır hata sayabilir ve fırsat eşitliğini öne alabilir.

11.3.7 Kod izleme: rastgeleleştirme ile fırsat eşitliği

Son işlemenin en basit biçimi şudur: B grubunda yalnız 0,8 puanlıları onaylamak yerine, 0,2 puanlıların bir kısmını da rastgele onaylamak. Onay olasılığı q olsun. Aşağıdaki fonksiyon, q'nun her değeri için B grubunun TPR, FPR ve doğruluğunu hesaplar.

```
def b_grubu(q):   # 0,2 puanlıyı q ile onayla
  TP = 16 + 16 * q
  FP = 4 + 64 * q
  tpr = TP / 32
  fpr = FP / 68
  dogru = TP + (68 - FP)   # TP + TN
  return tpr, fpr, round(dogru / 100, 3)

for q in [0, 0.2, 0.4, 0.6]:
  tpr, fpr, d = b_grubu(q)
  print(q, round(tpr, 2), round(fpr, 2), d)
```

Fonksiyonun mantığı Tablo 11.2'den gelir. 0,8 puanlı 20 kişinin hepsi onaylanır; bunlardan 16'sı öder (TP), 4'ü ödemez (FP). 0,2 puanlı 80 kişiden her biri q olasılıkla onaylanır. Bu 80 kişinin 16'sı öder, 64'ü ödemez. Beklenen değer olarak ödeyenlerden $16 \cdot q$, ödemeyenlerden $64 \cdot q$ kişi ek onay alır.

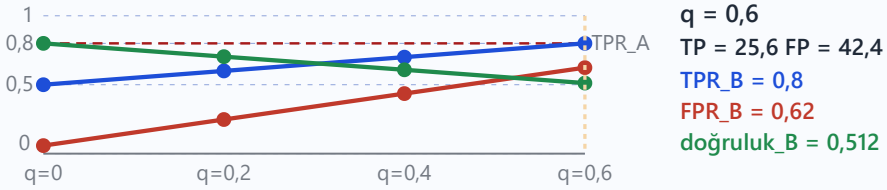
Böylece *TP = 16 + 16q* ve *FP = 4 + 64q* olur. TPR'nin paydası B'deki bütün ödeyenlerdir (32), FPR'ninki bütün ödemeyenlerdir (68). Doğru kararlar TP ile TN'nin toplamıdır; TN, ödemeyenlerden onay almayanlardır: *TN = 68 − FP*. Kodda `dogru = TP + (68 - FP)` satırı bunu yapar. Bu yüzden doğruluk *(16 + 16q + 68 − 4 − 64q)/100 = (80 − 48q)/100*'dür.

q	TP	FP	TPR	FPR	doğruluk	seçim oranı	PPV
0	16	4	0,5	0,06	0,800	0,20	0,80
0,2	19,2	16,8	0,6	0,25	0,704	0,36	0,53
0,4	22,4	29,6	0,7	0,44	0,608	0,52	0,43
0,6	25,6	42,4	0,8	0,62	0,512	0,68	0,38

Tablo 11.8: `b_grubu(q)` izlemesi. Son iki sütun, kodda hesaplanmayan ama aynı sayılardan çıkan seçim oranı ve PPV'dir.

Tablo satır satır şöyle okunur:

1. ***q = 0*:** Yalnız 0,8 puanlılar onaylanır (bugünkü model). TP = 16, FP = 4. TPR = 16/32 = 0,5; FPR = 4/68 ≈ 0,06; doğruluk (16 + 64)/100 = 0,80.
2. ***q = 0,2*:** 0,2 puanlı 80 kişinin %20'si onaylanır. TP = 16 + 16 · 0,2 = 19,2; FP = 4 + 64 · 0,2 = 16,8. TPR 0,6'ya çıkar ama doğruluk 0,704'e düşer. Neden bu kadar düşer? 0,2 puanlıların yalnız %20'si öder; yeni onaylanan her 5 kişiden 4'ü ödemeyecek olanlardandır.
3. ***q = 0,4*:** TP = 22,4; FP = 29,6. TPR 0,7; FPR 0,44'e fırlar; doğruluk 0,608.
4. ***q = 0,6*:** TP = 25,6; FP = 42,4. $TPR_B = 25{,}6/32 = 0{,}8 = TPR_A$: fırsat eşitliği sağlandı. Bedeli: doğruluk 0,512'ye, PPV 0,38'e düştü; FPR 0,62'ye çıktı.



Şekil 11.8: B grubunda 0,2 puanlıları *q* olasılıkla onaylamanın etkisi. Mavi: TPR_B (0,5'ten 0,8'e çıkar ve *q = 0,6*'da A'nın TPR'sine, kesikli çizgiye, ulaşır). Kırmızı: FPR_B (0,06'dan 0,62'ye). Yeşil: B'nin doğruluğu (0,8'den 0,512'ye).

Şekil 11.8 bedeli açıkça gösterir. Fırsat eşitliği için 0,2 puanlıların %60'ı rastgele onaylanmıştır. Ödemeyecek 64 kişiden yaklaşık *64 · 0,6 ≈ 38*'i kredi alır. B'nin FPR'si (0,62), A'dakinin (0,2) üç katından fazladır; bu yüzden eşitlenmiş olasılıklar yine sağlanmaz. Tahmin eşitliği de bozulmuştur: PPV 0,80'den 0,38'e düşmüştür. Bir ölçütü rastgeleleştirmeyle sağlamak mümkündür ama bu, modelin ayırt etme gücünü artırmaz; yalnız hataları yeniden dağıtır.

Örnek 11.12

A'nın TPR'si 0,8 yerine 0,7 olsaydı, B'de fırsat eşitliği için q kaç olmalıydı ve B'nin doğruluğu ne olurdu?

1. $TPR_B = (16 + 16q)/32 = 0{,}5 + 0{,}5q$.
2. $0{,}5 + 0{,}5q = 0{,}7 \Rightarrow 0{,}5q = 0{,}2 \Rightarrow q = 0{,}4$.
3. Doğruluk $(80 - 48 \cdot 0{,}4)/100 = (80 - 19{,}2)/100 = 0{,}608$. Tablo 11.8'in $q = 0{,}4$ satırıyla aynıdır.

Sık yapılan hata

- **"Daha iyi model bütün ölçütleri sağlar."** Taban oranları farklıyken bu ancak mükemmel tahminle olur. Modeli iyileştirmek çatışmanın boyutunu küçültebilir ama ortadan kaldırmaz.
- **Özdeşlikte $p/(1-p)$ yerine p yazmak:** Çarpan bir olasılık değil bir orandır ("ödeyenlerin ödemeyenlere oranı"). $p = 0{,}5$'te 1'dir, 0,5 değildir.
- **Rastgeleleştirmenin bedelini yalnız doğrulukta aramak:** q arttıkça doğruluk düşer, ama asıl önemli değişiklik FPR'nin ve PPV'nin değişmesidir. Bir ölçüt sağlanırken ötekiler daha da uzaklaşabilir.
- **Sınıf dengesini kalibrasyonla karıştırmak:** Kalibrasyon "puanı bilinen kişilerin gerçek sonucunu", sınıf dengesi "gerçek sonucu bilinen kişilerin puanını" sorar. Koşulun yönü terstir.

11.4 Adil bölüşüm

Adaletin ikinci yüzü, ortak bir kaynağın kişiler arasında paylaştırılmasıdır. Miras bölüşümü, ortak bir makinenin işlem süresinin kullanıcılara dağıtılması, bir arazinin parsellenmesi ya da derslerin öğrencilere atanması bu türden sorunlardır. Burada ölçülen şey bir modelin hata oranları değil, her kişinin kendi gözünde ne aldığıdır. Kaynak bölünebilir (pasta) ya da bölünemeyen mallardan oluşabilir; iki durumda adalet kavramları ve onları sağlayan algoritmalar farklıdır.

11.4.1 Pasta kesme modeli

Bölünebilir kaynak geleneksel olarak bir pasta ile anlatılır. Pasta, sayı doğrusundaki $[0, 1]$ aralığıdır. Bu aralık zamanı (bir günün saatleri), araziyi (bir kıyı şeridi) ya da işlem süresini temsil edebilir. Pasta n kişi arasında paylaştırılacaktır.

Kişiler pastanın farklı kısımlarına farklı değer verir. Biri çikolatalı kısmı, öteki vanilyalı kısmı sever; biri sabah saatlerini, öteki akşam saatlerini ister. Bu farklılık bir değer fonksiyonuyla anlatılır.

Tanım 11.9

Pasta kesme (*cake cutting*) probleminde her kişi i'nin bir **değer fonksiyonu** v_i vardır. $v_i(X)$, X parçasının i. kişinin gözündeki değeridir. Bütün pasta her kişi için 1 değerindedir: $v_i([0, 1]) = 1$. Değerler toplanabilir: ayrık iki parçanın birleşiminin değeri, değerlerinin toplamıdır. X_i, i. kişinin aldığı parçadır.

Bir algoritma kişilerin değer fonksiyonlarını tümüyle bilmez; bunu bilmek, sonsuz sayıda bilgi demektir. Bunun yerine kişilere soru sorar. Robertson ve Webb'in önerdiği standart modelde iki tür soru vardır:

- **Değer sorgusu:** "$[a, b]$ parçasına ne değer verirsin?"
- **Kesim sorgusu:** "a'dan başlayan ve senin gözünde değeri α olan parça nerede biter?"

Bu modele **Robertson–Webb modeli** (*Robertson–Webb query model*) denir. Bir algoritmanın verimliliği, kaç soru sorduğuyla ölçülür. Bu ölçü, bilgisayar biliminde bir algoritmanın kaç karşılaştırma yaptığını saymaya benzer.

Paylaşımın "adil" olduğunu söylemenin iki temel yolu vardır.

Tanım 11.10

Bir paylaşım $(X_1, \ldots, X_n)$:

- **orantılıdır** (*proportional*), eğer herkes kendi gözünde en az $1/n$ alıyorsa: her i için $v_i(X_i) \geq 1/n$;
- **kıskançsızdır** (*envy-free*), eğer kimse başkasının parçasını kendininkine tercih etmiyorsa: her i ve j için $v_i(X_i) \geq v_i(X_j)$.

İki tanımda da değerlendirme her kişinin *kendi* değer fonksiyonuyla yapılır. Yani adaleti dışarıdan bir hakem değil, kişilerin kendi tercihleri tanımlar. Ali ile Ayşe aynı pastadan farklı büyüklükte parçalar alabilir ve paylaşım yine de ikisinin gözünde adil olabilir.

Orantılılık bir "alt sınır" koşuludur: kişi yalnız kendi payına bakar. Kıskançsızlık bir "karşılaştırma" koşuludur: kişi kendi payını herkesinkiyle karşılaştırır. Pasta tamamen paylaşıldığında kıskançsızlık orantılılığı gerektirir. Bunun nedeni basittir: kıskançsız bir kişi kendi parçasını n parçanın her birinden değerli bulur; n parçanın toplam değeri onun gözünde 1 olduğundan, en büyük parçanın değeri en az $1/n$'dir. Ters yön ise genelde doğru değildir (11.4.5).

Oyun kuramı açısından kişiler stratejik olabilir: algoritmanın sorularına yalan cevap vererek daha büyük parça almaya çalışabilirler. İyi bir protokol, dürüst davranan herkese bir garanti verir; başkaları ne yaparsa yapsın. Bu garanti, Nash dengesi gibi herkesin rasyonel olduğunu varsayan bir kavram-

dan daha güçlüdür. Kişi, başkalarının davranışı hakkında hiçbir varsayım yapmadan payını güvenceye alabilir.

11.4.2 Kes-ve-seç adım adım

İki kişi için bilinen en eski adil bölüşüm yöntemi, çocukların pasta paylaşırken kullandığı kuraldır: biri keser, öteki seçer.

Tanım 11.11

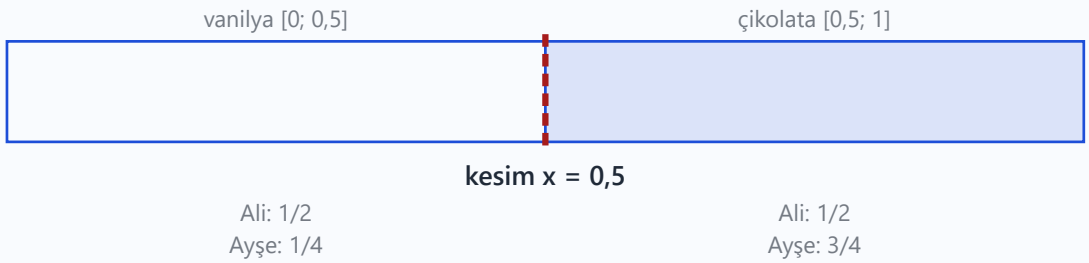
Kes-ve-seç (*cut-and-choose*) yöntemi iki adımdan oluşur:

1. **Kesen**, pastayı kendi gözünde değeri eşit iki parçaya böler.
2. **Seçen**, iki parçadan kendi gözünde değeri büyük olanı alır; kalan parça kesene kalır.

Örnek pasta iki yarıdan oluşur: sol yarı *[0; 0,5]* vanilyalı, sağ yarı *[0,5; 1]* çikolatalıdır. Ali pastanın her yerine eşit değer verir. Ayşe çikolatayı vanilyadan 3 kat daha çok sever: $v_{Ayşe}(vanilya) = 1/4$, $v_{Ayşe}(çikolata) = 3/4$. Her yarının içinde değer düzgün dağılır; yani bir yarının yarısı, o yarının değerinin yarısı kadar değerlidir.

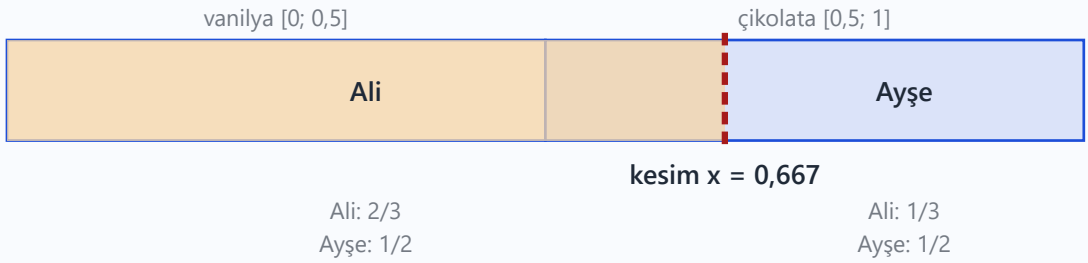
Örnek 11.13

1. **Ali keser.** Kendi gözünde iki eşit parça yapar. Ali için değer düzgün olduğundan $x = 1/2$'den keser. Her parça Ali için 1/2'dir.
2. **Ayşe değerlendirir.** Sol parça tam vanilya olduğu için onun gözünde 1/4, sağ parça tam çikolata olduğu için 3/4'tür.
3. **Ayşe seçer.** Değeri büyük olan sağ parçayı (çikolata) alır. Sol parça Ali'ye kalır.
4. **Sonuç.** Orantılılık: Ali 1/2 ≥ 1/2, Ayşe 3/4 ≥ 1/2 ✓. Kıskançsızlık: Ali kendi parçasına 1/2, Ayşe'ninkine 1/2 verir (1/2 = 1/2, kıskanmaz); Ayşe kendi parçasına 3/4, Ali'ninkine 1/4 verir (3/4 > 1/4) ✓.



Şekil 11.9: Kes-ve-seç, Ali keser. Kesim $x = 0,5$'tedir. Sol parça Ali için 1/2, Ayşe için 1/4; sağ parça Ali için 1/2, Ayşe için 3/4. Ayşe sağı seçer.

Roller değişirse sonuç da değişir. Bu kez Ayşe kesip Ali seçsin. Ayşe kendi gözünde iki eşit parça yapmalıdır. Sol yarının tamamı (vanilya) onun için yalnız 1/4 olduğundan kesim çikolatanın içinde olmalıdır. Kesim noktası $x > 1/2$ ise sol parçanın Ayşe için değeri $1/4 + (3/4) \cdot (x - 1/2)/(1/2) = 1/4 + 1{,}5 \cdot (x - 1/2)$'dir. Bunu 1/2'ye eşitlemek $1{,}5 \cdot (x - 1/2) = 1/4$, yani $x - 1/2 = 1/6$ ve $x = 2/3$ verir.



Şekil 11.10: Roller değişince: Ayşe $x = 2/3 \approx 0{,}667$'den keser (kendi gözünde 1/2–1/2). Ali sol parçayı seçer ve kendi gözünde 2/3 alır; Ayşe 1/2 alır.

Şekil 11.10'da Ali'nin seçimi kolaydır: sol parça onun için 2/3, sağ parça 1/3'tür. Ali 2/3 alır, Ayşe kendi gözünde tam 1/2 alır. İki duruma birlikte bakınca bir örüntü görülür: Ali seçerken 2/3, keserken 1/2 alır; Ayşe seçerken 3/4, keserken 1/2 alır. Seçen, kesenden avantajlıdır. Kesen yalnız garantisini (1/2) alır; seçen ise iki kişinin değerlendirmeleri arasındaki farkın kazancını toplar.

11.4.3 Kod izleme: kesim sorgusu ikiye bölme ile

Ayşe'nin kesim noktasını (2/3) yukarıda formülle bulduk. Bir algoritma ise Ayşe'nin değer fonksiyonunun formülünü bilmez; yalnız Robertson–Webb sorguları sorabilir. Kesim sorgusunun cevabı, yalnız değer sorguları kullanılarak **ikiye bölme** (*bisection*) yöntemiyle yaklaşık olarak bulunabilir. Aşağıdaki kod bu fikri uygular.

```
def v_ayse(x):          # [0, x] parçasının değeri
  if x <= 0.5: return x / 2
  return 0.25 + 1.5 * (x - 0.5)
def kes_noktasi(v, hedef, adim):
  lo, hi = 0.0, 1.0
  for _ in range(adim):
    orta = (lo + hi) / 2
    if v(orta) < hedef: lo = orta
    else: hi = orta
  return (lo + hi) / 2
print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

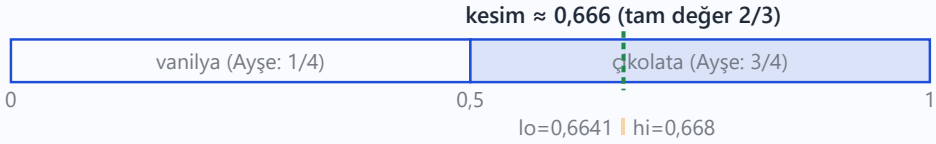
`v_ayse(x)`, *[0, x]* parçasının Ayşe için değerini verir. $x \le 0{,}5$ iken parça tamamen vanilyadır ve değeri $x/2$'dir (bütün vanilya, yani 0,5 uzunluk, 1/4 değerindedir). $x > 0{,}5$ iken bütün vanilya (0,25) ile çikolatanın bir kısmı alınır; çikolatanın her birim uzunluğu $(3/4)/(1/2) = 1{,}5$ değerindedir.

`kes_noktasi(v, hedef, adim)` fonksiyonu aradığı noktanın *[lo, hi]* aralığında olduğunu bilir. Başta *lo = 0*, *hi = 1*'dir. Her adımda aralığın ortasını sorar. Ortadaki değer hedeften küçükse kesim noktası daha sağdadır ve `lo` ortaya taşınır; değilse kesim noktası ortada ya da solundadır ve `hi` ortaya taşınır. Her adım aralığı yarıya indirir. Bu yöntemin çalışması için değer fonksiyonunun azalmayan olması yeterlidir: *[0, x]* parçası büyüdükçe değeri azalmaz.

sorgu	orta	v(orta)	karşılaştırma	yeni [lo, hi]	genişlik
1	0,5	0,25	< 0,5 ⇒ lo = orta	[0,5; 1]	1/2
2	0,75	0,625	≥ 0,5 ⇒ hi = orta	[0,5; 0,75]	1/4
3	0,625	0,4375	< 0,5 ⇒ lo = orta	[0,625; 0,75]	1/8
4	0,6875	0,5312	≥ 0,5 ⇒ hi = orta	[0,625; 0,6875]	1/16
5	0,6562	0,4844	< 0,5 ⇒ lo = orta	[0,6562; 0,6875]	1/32
6	0,6719	0,5078	≥ 0,5 ⇒ hi = orta	[0,6562; 0,6719]	1/64
7	0,6641	0,4961	< 0,5 ⇒ lo = orta	[0,6641; 0,6719]	1/128
8	0,668	0,502	≥ 0,5 ⇒ hi = orta	[0,6641; 0,668]	1/256

Tablo 11.9: `kes_noktasi(v_ayse, 0.5, 8)` izlemesi. Her satır bir değer sorgusudur.

Sekiz sorgudan sonra aralığın genişliği *$1/2^8 = 1/256$*'dır. Fonksiyon aralığın ortasını, *≈ 0,666*'yı döndürür. Gerçek kesim 2/3 ≈ 0,6667'dir; hata yaklaşık 0,00065'tir. Ayşe kesen olsaydı buradan keserdi.



Şekil 11.11: Sekiz sorgudan sonra arama aralığı *[0,6641; 0,668]*'e daralır; orta nokta ≈ 0,666, gerçek kesim 2/3'tür.

Hatanın nasıl küçüldüğü genel olarak yazılabilir. *k* sorgudan sonra aralığın genişliği *2^{-k}*'dir. Gerçek nokta bu aralığın içindedir ve fonksiyon aralığın ortasını döndürür; bu yüzden hata en çok aralığın yarısı, yani *$2^{-(k+1)}$*'dir. Hatanın her zaman 0,001'in altında kalması için *$2^{-(k+1)} < 0{,}001$* gerekir. *$2^{10} = 1024 > 1000$* olduğundan *$k + 1 = 10$*, yani *$k = 9$* sorgu yeterlidir; *$k = 8$*'de sınır *$2^{-9} \approx 0{,}00195$*'tir ve güvence vermez. Bir basamak daha doğruluk için yaklaşık üç ya da dört sorgu daha gerekir.

Örnek 11.14

`kes_noktasi(v_ayse, 0.5, 3)` çağrıldığında döngü sonunda *[lo, hi]* nedir?

1. Orta 0,5: *v = 0,25 < 0,5 ⇒ lo = 0,5*.
2. Orta 0,75: *v = 0,625 ≥ 0,5 ⇒ hi = 0,75*.
3. Orta 0,625: *v = 0,4375 < 0,5 ⇒ lo = 0,625*.

Aralık *[0,625; 0,75]*, döndürülen değer 0,6875'tir. Tablo 11.9'un ilk üç satırıyla aynıdır.

11.4.4 Kes-ve-seç neden adil?

Kes-ve-seçin adil olmasının nedeni, iki rolün garantilerine ayrı ayrı bakınca anlaşılır.

Kesenin garantisi. Kesen hangi parçanın kendisine kalacağını bilmez; seçen hangisini isterse onu alır. Kesen en kötü durumu düşünürse elinde iki parçanın küçüğü kalır. Bu yüzden kesenin garantisi *min(v(sol), v(sağ))*'dır. Kesim noktası sağa kaydıkça sol parçanın değeri artar, sağınki azalır; iki değerin küçüğü, ikisi eşit olduğunda en büyüktür. Bu nokta tam kesenin gözündeki yarıdır ve garanti 1/2'dir. Dürüst kesmek, yani iki eşit parça yapmak, en kötü durumdaki kazancı en büyük yapan, yani **maksimin** stratejidir.



Şekil 11.12: Kesenin garantisi *min(v(sol), v(sağ))*, kesim noktası *x*'in fonksiyonu olarak. Ali için (mavi) tepe *x = 1/2*'de, Ayşe için (kırmızı) *x = 2/3*'te; iki tepe de 0,5 yüksekliğindedir.

Şekil 11.12 bu "çatı" biçimini gösterir. Ali'nin eğrisi düzgündür, çünkü değeri düzgün dağılmıştır. Ayşe'nin eğrisinin sol kolu 1/2'ye kadar yavaş (vanilya az değerli), sonra hızlı yükselir. İki eğrinin tepesi de 0,5'tir: kimin değer fonksiyonu ne olursa olsun, dürüst kesen 1/2 garanti eder.

Seçenin garantisi. Seçen iki parçadan kendi gözünde büyük olanı alır. İki parçanın değerleri toplamı 1 olduğundan büyük olan en az 1/2'dir. Ayrıca seçtiği parça, kalan parçadan daha az değerli değildir. Bu iki özellik kesim nasıl yapılırsa yapılsın geçerlidir; kesen dürüst davranmasa bile seçen hem orantılı hem kıskançsız bir pay alır.

Bilginin değeri. Kesen, seçenin tercihini bilirse kesimi ona göre kaydırıp 1/2'den fazlasını alabilir; ama garantisi yine 1/2'dir. Aşağıdaki örnek bunu sayılarla gösterir.

Örnek 11.15

Ali kesiyor ve Ayşe'nin çikolatayı 3 kat sevdiğini biliyor. Ayşe iki parçayı eşit değerli bulursa çikolatalı (sağ) parçayı seçer. Ali nereden keser ve kendi gözünde ne kazanır?

1. Ayşe, sol parçanın kendi gözündeki değeri en çok 1/2 olduğu sürece sağı seçer.
2. Kesim $x \geq 1/2$ ise sol parçanın Ayşe için değeri $1/4 + 1{,}5\,(x - 1/2)$'dir. Koşul $1/4 + 1{,}5\,(x - 1/2) \leq 1/2$, yani $x \leq 2/3$'tür.
3. Ali sol parçayı almak istediği için onu olabildiğince büyütür: $x = 2/3$'ten keser.
4. Ali kendi gözünde $2/3 > 1/2$ alır; Ayşe yine 1/2 alır.

Bilgi, kesene avantaj sağlar. Ama Ali'nin bilgisi yanlışsa (örneğin Ayşe aslında vanilyayı seviyorsa) bu kaydırılmış kesim Ali'yi 1/2'nin altına düşürebilir. Dürüst kesimin değeri, bilgi gerektirmeyen bir garanti vermesidir.

Kes-ve-seçin iki kişide hem orantılı hem kıskançsız olması bir rastlantı değildir; iki kişide bu iki kavram aynıdır.

Teorem 11.4 (İki kişide orantılılık ve kıskançsızlık)

Pasta iki kişi arasında tamamen paylaşılmışsa, bir kişi için orantılılık ile kıskançsızlık eşdeğerdir: $v(X_1) \geq 1/2 \Leftrightarrow v(X_1) \geq v(X_2)$.

İspat. Pasta tamamen paylaşıldığı için $v(X_1) + v(X_2) = 1$'dir. ($\Rightarrow$) $v(X_1) \geq 1/2$ olsun. O zaman $v(X_2) = 1 - v(X_1) \leq 1/2 \leq v(X_1)$'dir; kişi ötekini kıskanmaz. ($\Leftarrow$) $v(X_1) \geq v(X_2) = 1 - v(X_1)$ olsun. İki tarafa $v(X_1)$ eklenince $2\,v(X_1) \geq 1$, yani $v(X_1) \geq 1/2$ bulunur. □

Üç ve daha fazla kişide bu eşdeğerlik bozulur: kıskançsızlık hâlâ orantılılığı gerektirir, ama orantılı bir paylaşım kıskançlık içerebilir. Kıskançsızlık daha güçlü bir koşul hâline gelir.

11.4.5 Orantılı ama kıskançlı: üç kişilik örnek

Üç kişi (Ali, Ayşe, Cem) pastanın daha önceden kesilmiş üç parçasını (X, Y, Z) paylaşacaktır. Aşağıdaki tablo her kişinin her parçaya verdiği değeri gösterir; her satırın toplamı 1'dir.

kişi	X parçası	Y parçası	Z parçası	aldığı	≥ 1/3?
Ali	**0,40**	0,30	0,30	X: 0,40	evet
Ayşe	0,20	**0,35**	0,45	Y: 0,35	evet
Cem	0,30	0,30	**0,40**	Z: 0,40	evet

Tablo 11.10: Üç kişinin üç parçaya verdiği değerler ve paylaşım Ali → X, Ayşe → Y, Cem → Z.

Örnek 11.16

1. **Orantılılık.** *1/3 ≈ 0,333*'tür. Ali 0,40, Ayşe 0,35, Cem 0,40 alır; üçü de 0,333'ten büyüktür. Paylaşım orantılıdır.
2. **Ali'nin kıskançlığı.** Ali için Y ve Z 0,30'dur, kendi X'i 0,40'tır. Ali kimseyi kıskanmaz.
3. **Cem'in kıskançlığı.** Cem için X ve Y 0,30'dur, kendi Z'si 0,40'tır. Cem kimseyi kıskanmaz.
4. **Ayşe'nin kıskançlığı.** Ayşe için Cem'in parçası Z 0,45'tir, kendi Y'si 0,35'tir. *0,45 > 0,35*: Ayşe Cem'i kıskanır.

Paylaşım orantılıdır ama kıskançsız değildir. Ayşe 1/3'ün üstünde aldığı için "haksızlığa uğradım" diyemez, ama Cem'le yer değiştirmeyi ister.

Bu örnekte basit bir düzeltme vardır: Ayşe ile Cem parçalarını takas etse Ayşe 0,45 alır, Cem 0,30 alır. Ama bu kez Cem 1/3'ün altına düşer ve paylaşım orantılı olmaktan çıkar. Kıskançsız bir paylaşım bulmak, orantılı bir paylaşım bulmaktan genelde çok daha zordur. Bu fark, bilinen algoritmaların karmaşıklığına da yansır:

Son azalan
: *n* kişi için orantılı bir paylaşım bulur (11.4.6).

Selfridge–Conway
: 3 kişi için kıskançsız bir paylaşım bulur ve en fazla 5 kesim yapar.

Aziz–Mackenzie (2016)
: *n* kişi için kıskançsız bir paylaşım bulan ve sonlu sayıda soruyla biten ilk algoritmadır. Ama soru sayısının üst sınırı devasadır; *n*'nin üst üste üslerle büyüyen bir fonksiyonudur. Algoritma kuramsal açıdan önemlidir, pratikte kullanılmaz.

11.4.6 Son azalan

Kes-ve-seçi ikiden fazla kişiye genişletmenin bir yolu, ilk parçayı herkesin denetiminden geçirmektir. Bu fikir **son azalan** (*last diminisher*) yöntemine yol açar.

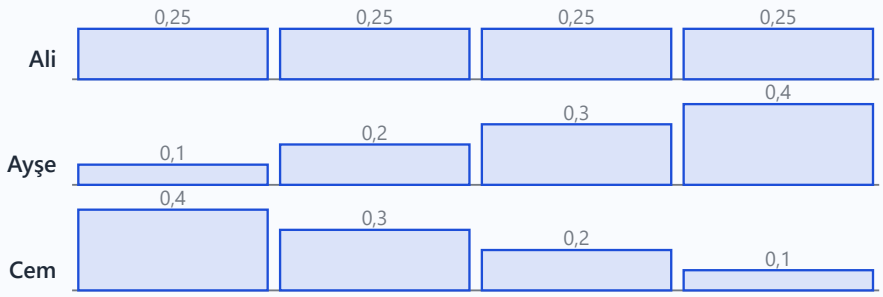
Tanım 11.12

Son azalan yöntemi *n* kişi için şöyle işler:

1. Birinci kişi pastanın solundan, kendi gözünde *1/n* değerinde bir parça keser.
2. Parça sırayla öteki kişilere gösterilir. Parçayı kendi gözünde *1/n*'den büyük bulan kişi onu kendi gözünde tam *1/n* olacak şekilde kırpar (azaltır). Parçayı *1/n* ya da daha küçük bulan kişi dokunmaz ("geçer").
3. Parçayı son kırpan kişi alır ve oyundan çıkar. Hiç kimse kırpmadıysa parça ilk kesene kalır.
4. Kalan pasta, kalan *n − 1* kişi arasında aynı yöntemle paylaştırılır. İki kişi kaldığında kes-ve-seç uygulanır.

Yöntemin neden orantılı olduğunu görmek için iki tür kişiye bakmak yeterlidir. Parçayı alan kişi onu son kırpan kişidir; kendi gözünde tam *1/n* alır. Parçayı almayanlar için ise parça, onların gözünde en çok *1/n*'dir: ya ilk baştan öyleydi ya da kendileri kırparak öyle yaptılar (kendilerinden sonra başkası kırptıysa parça daha da küçüldü). Bu yüzden kalan pasta onların gözünde en az $1 - 1/n = (n-1)/n$ değerindedir. Kalan $n - 1$ kişi bu pastayı orantılı paylaşırsa her biri en az $((n-1)/n) \cdot (1/(n-1)) = 1/n$ alır.

Örnekte üç kişi vardır ve pasta dört eşit çeyreğe ayrılmıştır. Her çeyreğin içinde değer düzgün dağılır. Ali'nin değeri tekdüzedir; Ayşe sağı, Cem solu sever.



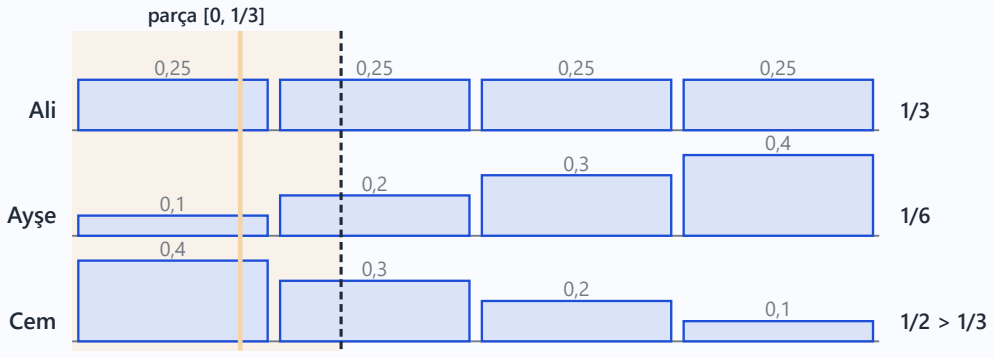
Şekil 11.13: Son azalan örneği. Çubuklar her kişinin çeyrek başına değeridir (her satırın toplamı 1): Ali 0,25–0,25–0,25–0,25; Ayşe 0,1–0,2–0,3–0,4; Cem 0,4–0,3–0,2–0,1.

Hesaplarda kullanılacak bir araç şudur: bir çeyreğin uzunluğu 1/4 olduğundan, değeri *w* olan bir çeyrekte birim uzunluğun değeri *4w*'dir. Örneğin Cem'in birinci çeyreğinde birim uzunluk $4 \cdot 0{,}4 = 1{,}6$ değerindedir; $[0;\ 0{,}1]$ parçası onun için $0{,}1 \cdot 1{,}6 = 0{,}16$ eder.

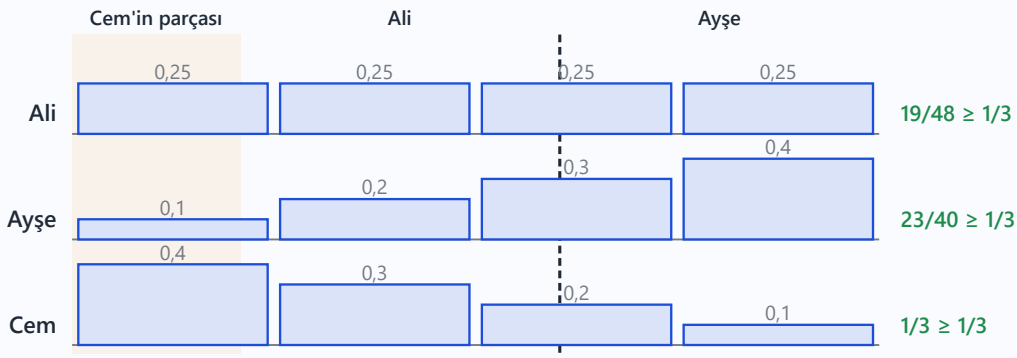
Örnek 11.17

1. **Ali keser.** Kendi gözünde 1/3 değerinde bir parça: *[0, 1/3]*. Ali'nin değeri tekdüze olduğu için uzunluk ile değer aynıdır.
2. **Ayşe bakar.** Parça birinci çeyreğin tamamını (0,1) ve ikinci çeyrekten *1/3 − 1/4 = 1/12* uzunluğu içerir. İkinci çeyrekte birim uzunluk *4 · 0,2 = 0,8* değerindedir. Değer *0,1 + (1/12) · 0,8 = 0,1 + 0,0667 = 1/6*'dır. *1/6 < 1/3*; Ayşe dokunmaz.
3. **Cem bakar.** Değer *0,4 + (1/12) · 1,2 = 0,4 + 0,1 = 1/2*'dir. *1/2 > 1/3*; Cem fazlasını kırpar. Birinci çeyrek tek başına 0,4 > 1/3 olduğu için kesim birinci çeyreğin içindedir: *1,6 · x = 1/3 ⇒ x = 1/(3 · 1,6) = 5/24 ≈ 0,208*. Parça artık *[0, 5/24]*'tür ve Cem'in gözünde tam 1/3'tür.
4. **Cem alır.** Parçayı son kırpan Cem'dir; parçayı alır ve çıkar. Kırpmayan Ali için parça *5/24 ≤ 1/3*, Ayşe için *(5/24) · 0,4 = 1/12 ≤ 1/3*'tür. Kalan pasta ikisinin gözünde en az 2/3'tür.
5. **Kalan *[5/24, 1]* iki kişide kes-ve-seç.** Kalan pasta Ali için *1 − 5/24 = 19/24* değerindedir. Ali onu kendi gözünde iki eşit parçaya böler: yarısı *19/48*'dir, kesim *5/24 + 19/48 = 10/48 + 19/48 = 29/48 ≈ 0,604*'tedir.
6. **Ayşe seçer.** Sağ parça *[29/48, 1]* Ayşe için: üçüncü çeyrekten *3/4 − 29/48 = 7/48* uzunluk (birim değer 1,2) ve dördüncü çeyreğin tamamı (0,4). Değer *(7/48) · 1,2 + 0,4 = 0,175 + 0,4 = 0,575 = 23/40*. Sol parça *[5/24, 29/48]* onun için *41/120 ≈ 0,342*'dir. Ayşe sağı seçer.

Sonuç: Cem *[0, 5/24]* ve kendi gözünde 1/3; Ali *[5/24, 29/48]* ve *19/48 ≈ 0,396*; Ayşe *[29/48, 1]* ve *23/40 = 0,575* alır. Herkes kendi gözünde en az 1/3 aldı; paylaşım orantılıdır.



Şekil 11.14: Ali'nin kestiği *[0, 1/3]* parçası (taralı alan). Parça Ali için 1/3, Ayşe için 1/6, Cem için 1/2'dir. Cem kesimi 5/24'e çeker (turuncu çizgi).



Şekil 11.15: Son azalanın sonucu. Cem *[0, 5/24]*, Ali *[5/24, 29/48]*, Ayşe *[29/48, 1]* alır. Sağdaki sayılar her kişinin kendi parçasına verdiği değerdir: 19/48, 23/40 ve 1/3; hepsi ≥ 1/3.

Şekil 11.15 paylaşımın orantılı olduğunu gösterir. Ama paylaşım kıskançsız değildir. Cem, Ali'nin parçası *[5/24, 29/48]*'e şu değeri verir: birinci çeyrekten *1/24* uzunluk (birim 1,6) ⇒ 1/15; ikinci çeyreğin tamamı ⇒ 0,3; üçüncü çeyrekten *5/48* uzunluk (birim 0,8) ⇒ 1/12. Toplam *1/15 + 3/10 + 1/12 = 9/20 = 0,45*'tir. Cem'in kendi parçası 1/3 ≈ 0,333'tür: Cem Ali'yi kıskanır. Son azalan yalnız orantılılığı güvenceye alır.

Yöntemin soru sayısı da hesaplanabilir. İlk turda n kişinin her birine parça gösterilir; ikinci turda $n - 1$ kişiye ve böyle sürer. Toplam yaklaşık $n + (n - 1) + \ldots + 2 \approx n^2/2$ sorgudur; yani $O(n^2)$.

Örnek 11.18

Cem'in değerleri Ayşe'ninki gibi (0,1; 0,2; 0,3; 0,4) olsaydı ilk parçayı kim alırdı?

Ali yine *[0, 1/3]*'ü keser. Ayşe için bu parça 1/6'dır (yukarıdaki hesap). Cem'in değerleri aynı olduğundan onun için de 1/6'dır. İkisi de *1/6 < 1/3* bulur ve dokunmaz. Kimse kırpmadığı için parçayı ilk kesen, yani Ali alır.

11.4.7 Hareketli bıçak ve Even–Paz

Son azalan yönteminin sürekli bir sürümü de vardır. Dubins ve Spanier (1961) bunu bir bıçak imgesiyle anlatır.

Tanım 11.13

Hareketli bıçak (*moving knife*, Dubins–Spanier) yönteminde bir bıçak pastanın solundan sağa yavaşça kayar. Bıçağın solundaki parça birinin gözünde *1/n* değerine ulaşınca o kişi "dur!" der, parçayı alır ve çıkar. Bıçak kalanlarla sürer; sonuncu kişi kalanı alır.

Yöntemi son azalan örneğindeki değerlerle (çeyrek başına Ali 0,25; Ayşe 0,1–0,4; Cem 0,4–0,1) çalıştıralım. Her kişi için "sol parça ne zaman 1/3 olur?" sorusunun cevabı hesaplanır; en küçük cevabı veren kişi önce "dur" der.

kişi	**1. tur: *v([0, x]) = 1/3***	**2. tur: *v([5/24, x]) = 1/3***	**aldığı**
Cem	*x = 5/24 ≈ 0,21* (dur!)	—	*[0, 5/24]*: 1/3
Ali	*x = 1/3 ≈ 0,33*	*x = 13/24 ≈ 0,54* (dur!)	*[5/24, 13/24]*: 1/3
Ayşe	*x = 19/36 ≈ 0,53*	*x = 43/72 ≈ 0,60*	*[13/24, 1]*: 13/20

Tablo 11.11: Hareketli bıçak. Her hücre, o kişinin "dur" diyeceği bıçak konumudur.

Örnek 11.19

1. **1. tur, Cem:** Birinci çeyrekte birim değer 1,6; $1{,}6\,x = 1/3 \Rightarrow x = 5/24$.
2. **1. tur, Ali:** Tekdüze değer; $x = 1/3$.
3. **1. tur, Ayşe:** İlk iki çeyrek ona 0,1 + 0,2 = 0,3 verir; bu 1/3'ten azdır. Üçüncü çeyrekte birim değer 1,2: $0{,}3 + 1{,}2\,(x - 1/2) = 1/3 \Rightarrow x - 1/2 = 1/36 \Rightarrow x = 19/36$.
4. En küçük konum Cem'indir (5/24). Bıçak oraya gelince Cem durdurur ve *[0, 5/24]*'ü alır. İlk parça son azalandakiyle aynıdır: kesimi en önce Cem durdurur.
5. **2. tur, Ali:** $5/24 + 1/3 = 5/24 + 8/24 = 13/24$.
6. **2. tur, Ayşe:** *[5/24, 1/4]* ona $(1/24) \cdot 0{,}4 = 1/60$, ikinci çeyrek 0,2 verir; 1/2'ye kadar toplam $13/60 \approx 0{,}217$'dir. Eksik $1/3 - 13/60 = 7/60$, üçüncü çeyrekte $(7/60)/1{,}2 = 7/72$ uzunlukla tamamlanır: $x = 1/2 + 7/72 = 43/72$.
7. Ali'nin konumu (13/24 ≈ 0,542) Ayşe'ninkinden (43/72 ≈ 0,597) küçüktür. Ali durdurur ve *[5/24, 13/24]*'ü alır. Ayşe kalanı alır: *[13/24, 1]*, değeri $(3/4 - 13/24) \cdot 1{,}2 + 0{,}4 = (5/24) \cdot 1{,}2 + 0{,}4 = 0{,}25 + 0{,}4 = 13/20$.

Herkes en az 1/3 alır; paylaşım orantılıdır. Kıskançsız değildir: Cem, Ali'nin parçası *[5/24, 13/24]*'e $1/15 + 0{,}3 + 1/30 = 2/5 > 1/3$ değer verir.

Hareketli bıçak, son azalanın "sürekli zamanda" çalışan hâlidir; bir algoritma olarak çalıştırmak için her kişinin "dur" noktası bir kesim sorgusuyla bulunur. Her turda kalan herkese bir kesim sorgusu sorulur; toplam sorgu sayısı yine $O(n^2)$'dir. Even ve Paz (1984), pastayı her adımda ikiye bölerek orantılı paylaşımı çok daha az soruyla, $O(n \log n)$ sorguyla bulur. Fikir şudur: herkes pastayı kendi gözünde ortadan işaretler; işaretlerin ortancasından kesilir; kişilerin yarısı sol parçayı, yarısı sağ parçayı aralarında aynı yöntemle paylaşır. Grup her adımda yarıya indiği için yaklaşık $\log_2 n$ düzey vardır ve her düzeyde n sorgu sorulur.

11.4.8 Bölünemez mallar: sırayla seçim ve EF1

Pasta istenen yerden kesilebilir; ama bir araba, bir ev ya da bir ders kontenjanı bölünemez. **Bölünemez mallarda** (*indivisible goods*) her mal tümüyle bir kişiye verilir. Bir kişinin aldığı mallar kümesine **demet** (*bundle*) denir. Değerlerin toplanabilir olduğu varsayılır: bir demetin değeri, içindeki malların değerlerinin toplamıdır.

Bölünemez mallarda kıskançsızlık her zaman mümkün değildir. En basit örnek, tek bir arabanın iki kardeş arasında paylaşılmasıdır. Arabayı alan kardeş ötekini kıskanmaz, ama öteki hiçbir şey almadığı için arabayı alanı kıskanır. Hangi paylaşım seçilirse seçilsin biri kıskanır. Bu yüzden zayıflatılmış bir adalet kavramı gerekir.

Tanım 11.14

Bir paylaşımda kıskançlığın olduğu her durumda kıskanılan demetten uygun bir mal çıkarılınca kıskançlık ortadan kalkıyorsa, paylaşım **bir mala kadar kıskançsızdır** (EF1, *envy-free up to one good*). Yani i, j'yi kıskanıyorsa bazı $g \in X_j$ için $v_i(X_i) \geq v_i(X_j \setminus \{g\})$ sağlanır.

Sırayla seçim (*round-robin*): kişiler sabit bir sırayla dönerek her turda kalan mallardan kendi gözlerinde en değerlisini alır.

EF1, "kıskançlık varsa bile bu kıskançlık tek bir malın şansından kaynaklanır" demektir. Tek araba örneğinde arabayı alan kardeşin demetinden araba çıkarılırsa boş demet kalır; öteki kardeşin kıskançlığı kalkar. Bu paylaşım EF1'dir.

Örnekte iki kişi ve dört mal vardır. Ali başlar.

kişi	m1	m2	m3	m4
Ali	7	5	3	1
Ayşe	8	6	1	1

Tablo 11.12: Bölünemez dört mal ve iki kişinin değerleri.

Örnek 11.20

1. **Ali** m1'i (7) alır. Ali: kendi 7. Ayşe: kendi 0, Ali'nin demeti gözünde 8.
2. **Ayşe** kalanlardan m2'yi (6) alır. Ayşe: kendi 6, Ali'nin demeti 8. Ali: kendi 7, Ayşe'nin demeti 5.
3. **Ali** m3'ü (3) alır. Ali: kendi 10, Ayşe'nin demeti 5. Ayşe: kendi 6, Ali'nin demeti 8 + 1 = 9.
4. **Ayşe** m4'ü (1) alır. Ayşe: kendi 7, Ali'nin demeti 9. Ali: kendi 10, Ayşe'nin demeti 5 + 1 = 6.

Sonuç: Ali {m1, m3} = 10, Ayşe {m2, m4} = 7. Ali kıskanmaz (10 ≥ 6). Ayşe ise Ali'nin demetine 9 değer verir ve 7 alır: kıskanır! Ama Ali'nin demetinden m1 çıkarılırsa Ayşe'nin gözünde 1 kalır ve $1 \leq 7$: paylaşım EF1'dir.

Sırayla seçimin her zaman EF1 bir paylaşım ürettiği gösterilebilir. İki kişide gerekçe kısadır. Ayşe'nin her seçimi, Ali'nin bir sonraki seçiminden önce gelir ve o anda kalan en değerli malı alır. Bu yüzden Ayşe'nin k. seçimi, onun gözünde Ali'nin $(k + 1)$. seçiminden az değerli değildir. Ali'nin ilk seçimi (m1) çıkarılınca Ali'nin geri kalan malları Ayşe'nin malları ile birebir eşleşir ve her eşte Ayşe'ninki ağır basar. Örnekte Ayşe'nin m2'si (6), Ali'nin m3'ünden (Ayşe için 1) değerlidir. Başlayan kişi (Ali) ise hiç kıskanmaz, çünkü her turda ötekinden önce seçer.

Örnek 11.21

Sırayla seçimde Ayşe başlasaydı (aynı değerlerle) kim neyi alırdı? Bir kişi iki malı eşit değerli bulursa numarası küçük olanı alır.

1. Ayşe m1'i (8) alır.
2. Ali kalanlardan m2'yi (5) alır.
3. Ayşe için m3 ve m4 eşittir (1 ve 1); numarası küçük olan m3'ü alır.
4. Ali m4'ü (1) alır.

Ali {m2, m4} = 6, Ayşe {m1, m3} = 9. Ali, Ayşe'nin demetine 7 + 3 = 10 değer verir: kıskanır. Ama m1 çıkarılınca 3 kalır ve $3 \leq 6$: yine EF1'dir. Bu kez başlayan Ayşe kıskanmaz.

11.4.9 Oyuna dönüş: açgözlü seçim en iyi tepki değil

Açılıştaki Adil Bölüşüm Oyunu'na (11.1.1) dönelim. Malların değerleri (O1 / O2): Pasta 5/1, Kitap 4/9, Kulaklık 3/8, Top 1/1. O1 başlar ve O2 her turda kendi gözünde en değerli kalan malı alır. O1'in iki planını karşılaştıralım.

O1'in planı	Seçim sırası	O1	O2
hep en değerlisi	Pasta, Kitap, Kulaklık, Top	8	10
önce Kitap	Kitap, Kulaklık, Pasta, Top	9	9

Tablo 11.13: O1'in iki planı ve sonuçları.

Örnek 11.22

1. **Plan 1 (açgözlü).** O1 kendi en değerlisi Pasta'yı (5) alır. O2 Kitap'ı (9) alır. O1 kalanlardan Kulaklık'ı (3), O2 Top'u (1) alır. O1: 5 + 3 = 8. O2: 9 + 1 = 10.
2. **Plan 2 (önce Kitap).** O1 Kitap'ı (4) alır. O2 kalanlardan kendi en değerlisi Kulaklık'ı (8) alır. O1 Pasta'yı (5), O2 Top'u (1) alır. O1: 4 + 5 = 9. O2: 8 + 1 = 9.

3. **Neden işe yarar?** O2 Pasta'ya yalnız 1 değer verir; Pasta beklese de kaçmaz. O2'nin gerçekten istediği Kitap'tır. O1 Kitap'ı önce alırsa O2'nin en sevdiği malı elinden alır, Pasta'yı ise bir sonraki turda yine alabilir.

Dört malın bütün seçim sıraları denendiğinde O1'in bu O2'ye karşı alabileceği en büyük değer 9'dur; Plan 2 en iyi tepkidir. Sırayla seçimde her turda en sevilen malı almak her zaman en iyi strateji değildir.

Adalet açısından ne olduğuna da bakalım. İkinci planda O2, O1'in demeti {Kitap, Pasta}'ya $9 + 1 = 10$ değer verir; kendi demeti {Kulaklık, Top} 9'dur. O2 kıskanır. Kitap çıkarılırsa $1 \leq 9$: paylaşım yine EF1'dir. Kuramın güvencesi şudur: iki taraf da hep favorisini alırsa sırayla seçim EF1 üretir. Bir taraf stratejik davranınca bu güvence artık kuramdan gelmez; bu örnekte EF1 yine sağlanmıştır ama her durumda sağlanacağının garantisi yoktur.

Bu gözlem gerçek hayatta da görülmüştür. Harvard İşletme Okulu derslerini öğrencilere sırayla seçime (*draft*) dayalı bir yöntemle dağıtıyordu. Budish ve Cantillon (2012) bu sistemin verisini inceledi. Öğrenciler tam bu türden stratejik seçimler yapıyordu: popüler olmayan ama kendileri için değerli dersleri bekletiyor, başkalarının da istediği dersleri öne alıyorlardı. Bu davranış refah kaybına yol açıyordu. Adil bir protokol bile stratejik oyunculara açıktır: EF1 güvencesi, herkes sırası gelince favorisini alırsa geçerlidir.

11.4.10 Üç sosyal refah ölçütü

Birden çok paylaşım arasında seçim yapmak için onları bir sayıyla sıralamak gerekir. Bu sayıya sosyal refah ölçütü denir. Üç ölçüt yaygındır.

Tanım 11.15

- **Faydacı refah** (*utilitarian welfare*): sosyal refah $SW = \Sigma\, v_i(X_i)$, yani değerlerin toplamı, en büyük yapılır. Bu ölçüt verimliliği ölçer.
- **Eşitlikçi refah** (*egalitarian welfare*): en küçük $v_i(X_i)$ en büyük yapılır (maksimin). En kötü durumdaki kişinin payı korunur.
- **Nash sosyal refahı** (*Nash social welfare*): değerlerin çarpımı $\Pi\, v_i(X_i)$ en büyük yapılır. İki kişide $v_{Ali} \times v_{Ayşe}$'dir.

Nash sosyal refahı iki ucun arasında durur. Toplamda bir kişinin büyük kazancı ötekinin küçük kaybını kolayca telafi eder; çarpımda ise herhangi bir değerin sıfıra yaklaşması bütün çarpımı sıfıra çeker. Bu yüzden çarpım dengeli paylaşımları ödüllendirir. Nash sosyal refahının bir özelliği daha vardır: ölçek değişikliğinden etkilenmez. Ali'nin bütün değerleri 10 katına çıkarsa her paylaşımın çarpımı 10 katına çıkar; sıralama değişmez ve en iyi paylaşım aynı kalır. Toplamda ise Ali'nin değerleri yeterince büyütülürse Ali'nin değer verdiği bütün mallar ona gider. Kişilerin değer ölçeklerini karşılaştırmak çoğu zaman anlamsız olduğu için bu özellik önemlidir.

Tablo 11.12'deki değerlerle 16 olası paylaşımdan üçü karşılaştırılabilir.

Paylaşım	Ali	Ayşe	toplam	en küçük	çarpım	kıskançsız?
Sırayla seçim: Ali {m1, m3}, Ayşe {m2, m4}	10	7	17	7	70	hayır, EF1
Faydacı: Ali {m3, m4}, Ayşe {m1, m2}	4	14	18	4	56	hayır, EF1 bile değil
Nash / eşitlikçi (tek değil): Ali {m2, m3, m4}, Ayşe {m1}	9	8	17	8	72	evet

Tablo 11.14: Üç paylaşımın refah ölçütlerine göre karşılaştırılması.

Örnek 11.23

1. **Faydacı paylaşım.** Toplamı en büyük yapmak için her mal ona daha çok değer verene gider: m1 (8 > 7) ve m2 (6 > 5) Ayşe'ye, m3 (3 > 1) Ali'ye. m4'te eşitlik vardır (1 = 1); burada Ali'ye verilir. Toplam 4 + 14 = 18.
2. **Faydacı paylaşımın adaletsizliği.** Ali, Ayşe'nin demeti {m1, m2}'ye *7 + 5 = 12* verir, kendisi 4 alır. m1 çıkarılsa bile *5 > 4*; m2 çıkarılsa *7 > 4*. Hangi mal çıkarılırsa çıkarılsın kıskançlık sürer: EF1 bile değildir.
3. **Nash paylaşımı.** Ali {m2, m3, m4} = 5 + 3 + 1 = 9, Ayşe {m1} = 8; çarpım 72. Ali, Ayşe'nin demetine 7 verir (< 9); Ayşe, Ali'nin demetine 6 + 1 + 1 = 8 verir (= 8). Kimse kıskanmaz.
4. **Tek değil.** Ali {m2, m3}, Ayşe {m1, m4} paylaşımı da (8, 9) verir ve çarpımı yine 72'dir. İki paylaşım da en küçük değeri 8 yapar; ikisi de hem Nash hem eşitlikçi optimumdur.

Örnekte Nash sosyal refahını en büyük yapan paylaşımın aynı zamanda adil olması bir rastlantı değildir. Caragiannis ve arkadaşları (2016), böyle bir paylaşımın her zaman en az EF1 olduğunu göstermiştir.

Teorem 11.5 (Caragiannis vd., 2016)

Değerler toplanabilir olduğunda, Nash sosyal refahını maksimize eden paylaşım hem Pareto optimal hem de EF1'dir.

Bu sonuç makalenin adına da yansımıştır: “en büyük Nash refahının akıl almaz adaleti”. Teorem, verimlilik ile adaletin birbirini dışlamak zorunda olmadığını gösterir. Tek bir sayıyı (çarpımı) en büyük yapmak, hem kimsenin durumunu bozmadan iyileştirilemeyen hem de en çok bir mallık kıskançlık içeren bir paylaşım verir.

11.4.11 Adaletin bedeli ve Pareto verimliliği

Adalet kısıtının verimlilikten ne kadar götürdüğünü ölçmek için iki kavram gerekir.

Tanım 11.16

- Bir paylaşım **Pareto verimlidir** (Pareto optimal, *Pareto efficient*), eğer kimseyi kötüleştirmeden en az bir kişiyi iyileştiren başka bir paylaşım yoksa.
- **Adaletin bedeli** (*price of fairness*), bütün paylaşımlar arasındaki en büyük sosyal refahın, adil paylaşımlar arasındaki en büyük sosyal refaha oranıdır. Oran 1'e yakınsa adalet ucuzdur.

Tablo 11.12'nin değerleriyle 16 paylaşımın hepsi taranabilir (dört malın her biri iki kişiden birine gider: $2^4 = 16$). Taramanın sonuçları şunlardır. 16 paylaşımın 11'i Pareto optimaldir; sırayla seçimin (10, 7) paylaşımı da bunlardan biridir. Örneğin Ali'ye 10'dan, Ayşe'ye 7'den az vermeyen ve birini artıran başka bir paylaşım yoktur. Toplamı en büyük yapan ve kıskançsız olan paylaşımlar aşağıdaki tablodadır.

Paylaşım	Ali	Ayşe	toplam	kıskançsız?
Ali {m3, m4}, Ayşe {m1, m2}	4	14	18	hayır
Ali {m3}, Ayşe {m1, m2, m4}	3	15	18	hayır
Ali {m2, m3, m4}, Ayşe {m1}	9	8	17	evet
Ali {m2, m3}, Ayşe {m1, m4}	8	9	17	evet

Tablo 11.15: En büyük toplamlı paylaşımlar ve kıskançsız paylaşımlar.

Örnek 11.24

1. **En büyük sosyal refah:** 18. İki paylaşım bunu verir; ikisi de m4'ün kime gittiği dışında aynıdır (m4'te iki kişinin değeri eşittir).
2. **Kıskançsız paylaşımlar:** 16 paylaşımdan yalnız ikisi kıskançsızdır. m4'teki eşitlik yüzünden ikisi de Nash ve eşitlikçi optimumdur; toplamları 17'dir.
3. **Adaletin bedeli:** $18/17 \approx 1{,}059$.
4. **Yorum:** Kıskançsızlık için sosyal refahın $1 - 17/18 = 1/18 \approx \%5{,}6$'sı feda edilir.

Adaletin bedeli, Anarşinin Bedeli gibi bir orandır. Anarşinin Bedeli'nde kayıp, oyuncuların bencil davranmasından gelir: en iyi sonucun, bencil oyuncuların ulaştığı en kötü dengeye oranıdır. Adaletin bedelinde kayıp bencillikten değil, adalet kısıtından gelir. İki oranın da paydasında "kısıt altında ulaşılabilen" refah, payında ise kısıtsız en iyi refah vardır.

Bu örnekte bedel küçüktür. Ama bedel her zaman küçük değildir. Bölünebilir pastada n kişide orantılılığın bedeli kabaca $\sqrt{n}$ ile orantılı büyür (Caragiannis vd., 2012). Kişi sayısı arttıkça, herkese en az $1/n$ verme zorunluluğu, pastayı en çok değer verene bırakmaya göre giderek daha pahalı hâle gelir.

Sık yapılan hata

- **Değerleri ortak bir ölçekle değerlendirmek:** Orantılılık ve kıskançsızlık her kişinin kendi değer fonksiyonuyla denetlenir. Ayşe'nin Ali'yi kıskanıp kıskanmadığı Ayşe'nin değerleriyle hesaplanır, Ali'ninkilerle değil.
- **"Orantılıysa kıskançsızdır":** Bu yalnız iki kişide doğrudur. Üç kişilik örnekte (Tablo 11.10) paylaşım orantılıdır ama Ayşe Cem'i kıskanır.
- **EF1'de herhangi bir malı çıkarmak:** EF1, kıskanılan kişinin demetinden *uygun bir* malın çıkarılmasını ister; genellikle kıskanan kişinin gözünde en değerli olan mal çıkarılır. Rastgele bir mal çıkarıp kıskançlık sürüyorsa paylaşımın EF1 olmadığı sonucu çıkmaz.
- **Son azalanda parçayı ilk kesene vermek:** Parçayı son kırpan kişi alır. Yalnız hiç kimse kırpmadıysa parça ilk kesene kalır.
- **Kesimi uzunlukla bulmak:** Kesen "kendi gözünde" eşit parçalar yapar. Ayşe'nin yarısı 1/2 uzunlukta değil, 2/3 noktasındadır.

11.5 Etik ve yönetişim

Ölçütler ve algoritmalar adaletin ölçülebilen yüzünü anlatır. Ama bir modelin hangi ölçüte göre değerlendirileceğine, kimin itiraz edebileceğine ve hata olursa kimin sorumlu olacağına matematik karar vermez. Bu sorular etik ilkelere ve hukuki çerçeveye, insanların algoritmaya stratejik tepkisine, denetime ve kararların kendi verisini nasıl değiştirdiğine uzanır.

11.5.1 Etik ilkeler ve standartlar

Algoritmik karar sistemleri için önerilen ilkeler farklı kurumlarda farklı adlarla geçse de dört ilke hemen hepsinde bulunur.

Şeffaflık ve açıklanabilirlik
: **Şeffaflık** (*transparency*), karar sürecinin, yani hangi verinin hangi amaçla kullanıldığının anlaşılır olmasıdır. **Açıklanabilirlik** (*explainability*), tek bir kararın gerekçesinin ilgili kişiye anlatılabilmesidir. Kredisi reddedilen kişi, kararın nedenini (örneğin "borç/gelir oranı yüksek") öğrenebilmelidir.

Hesap verebilirlik
: **Hesap verebilirlik** (*accountability*), hatalı bir kararın sorumlusunun belli olmasıdır. "Algoritma öyle karar verdi" bir gerekçe sayılmaz. Kararın etkilediği kişi itiraz edebilmeli ve bir insanın kararı yeniden incelemesini isteyebilmelidir.

Adalet ve eşitlik
: Kararların sosyal etkileri ve grup bazlı sonuçları hesaba katılmalıdır. Bu bölümde hesaplanan ölçütler bu ilkenin ölçülebilir karşılığıdır.

Mahremiyet

Mahremiyet (*privacy*), kişisel verinin en az gerekli ölçüde ve yalnız açıklanan amaç için toplanmasıdır. Adalet ölçmek için grup bilgisine ihtiyaç duyulması, mahremiyetle gerilim yaratabilir: korunan özelliği toplamayan bir kurum, kendi modelinin bu özelliğe göre adil olup olmadığını da ölçemez.

Bu ilkeler hukuki düzenlemelere de yansımıştır.

Düzenleme	İçerik
GDPR (AB, 2018), madde 22	Yalnızca otomatik işlemeye dayanan ve kişiyi önemli ölçüde etkileyen kararlara itiraz ve insan müdahalesi hakkı.
AB Yapay Zekâ Yasası (2024)	Risk tabanlı yaklaşım: kredi skorlama ve işe alım gibi uygulamalar "yüksek risk" sınıfındadır ve ek yükümlülüklere (risk yönetimi, veri kalitesi, insan gözetimi) tabidir.
KVKK (Türkiye), OECD YZ ilkeleri	Kişisel veri işleme kuralları (6698 sayılı Kanun); güvenilir yapay zekâ için ortak ilkeler.

Tablo 11.16: Algoritmik kararlarla ilgili başlıca düzenlemeler ve ilkeler.

Düzenlemelerin ortak noktası, yüksek etkili kararlarda insanın devreden tümüyle çıkarılmamasıdır. AB Yapay Zekâ Yasası'nın risk tabanlı yaklaşımı ayrıca bir orantılılık ilkesi taşır: bir film öneri sistemiyle bir kredi skorlama sistemine aynı yükümlülükler getirilmez; yükümlülük, olası zararın büyüklüğüyle artar.

11.5.2 Stratejik tasarım ve teşvikler

Önceki kısımlarda kişiler, modelin karşısında edilgen bir veri noktası gibi ele alındı. Oysa algoritma bir kural koyduğunda insanlar ona tepki verir. Sınıflandırılan kişiler de oyunculardır. Bu bakış açısına **stratejik sınıflandırma** (*strategic classification*) denir. Oyun kuramının dili burada doğrudan kullanılabilir: kurum bir kural seçer (lider), kişiler bu kurala en iyi tepkiyi verir (izleyiciler).

Bu tepkinin genel bir sonucu **Goodhart yasası** (*Goodhart's law*) ile özetlenir: bir ölçü hedef hâline gelince iyi bir ölçü olmaktan çıkar. Tıklama sayısı başarı ölçüsü olarak kullanılmaya başlanınca içerik üreticileri tıklama tuzağı başlıklar yazar; tıklama sayısı artar ama ölçtüğü şey, yani içeriğin ilgi çekiciliği, artmaz. Ölçü ile ölçülmek istenen nitelik arasındaki bağ, ölçünün hedeflenmesiyle kopar.

Kullanıcılara doğru teşvikler vermek ve kötü niyetli manipülasyonu (sahte yorumlar, bot hesaplar) önlemek tasarımın bir parçasıdır. Üç durum öne çıkar:

Oyunlanabilir özellik
: Kredi puanında "kişinin kredi kartı sayısı" gibi kolay değiştirilen bir özellik, gerçek riski değiştirmeden puanı değiştirir. Kişinin ödeme gücü aynı kalırken puanı yükselir. Bu tür davranışa **oyunlama** (*gaming*) denir.

Öneri sistemleri
: Etkileşimi (izlenme süresi, tıklama) ödüllendiren bir hedef, kullanıcının uzun vadeli yararıyla çatışabilir. Daha iyi bir hedef, kullanıcının uzun vadeli yararını ödüllendirendir; ama bu yararı ölçmek çok daha zordur.

Maliyet eşitsizliği
: Oyunlamanın maliyeti gruplar arasında farklıysa stratejik davranış yeni bir adaletsizlik doğurur. Kaynağı bol olan grup kurala daha kolay uyum sağlar. 11.5.3'teki örnek bunu adım adım gösterir.

Mekanizma tasarımının temel sorusu burada da geçerlidir: kurala uyan dürüst davranış, bireyin çıkarına olmalıdır. Stratejiye dayanıklı bir mekanizmada dürüst davranmak en iyi stratejidir; iyi tasarlanmış bir sınıflandırıcıda da puanı yükseltmenin en ucuz yolu, gerçekten daha nitelikli olmak olmalıdır.

11.5.3 Stratejik sınıflandırma adım adım

Hu, Immorlica ve Vaughan (2019) stratejik davranışın gruplar arası etkisini basit bir modelle incelemiştir. Aşağıdaki örnek bu modelin sayısal bir sürümüdür.

Kurum, puanı eşiğe eşit ya da eşikten büyük olanları kabul eder. Başta eşik 0,6'dır. Gerçek nitelik de bu puanla ölçülür: gerçek puanı 0,6 ve üstü olan kişiler gerçekten niteliklidir. Kişiler puanlarını bir maliyetle (örneğin kart sayısını artırarak) yükseltebilir. A grubunun bütçesi 0,15'tir: A'daki herkes puanını en çok 0,15 artırabilir. B grubunun bütçesi yalnız 0,03'tür. Her grupta 5 niteliksiz ve 5 nitelikli kişi vardır.

grup	bütçe	niteliksiz (puan < 0,6)	nitelikli (puan ≥ 0,6)
A	0,15	0,30; 0,42; 0,50; 0,52; 0,58	0,62; 0,66; 0,71; 0,80; 0,90
B	0,03	0,25; 0,35; 0,48; 0,56; 0,58	0,63; 0,68; 0,72; 0,78; 0,88

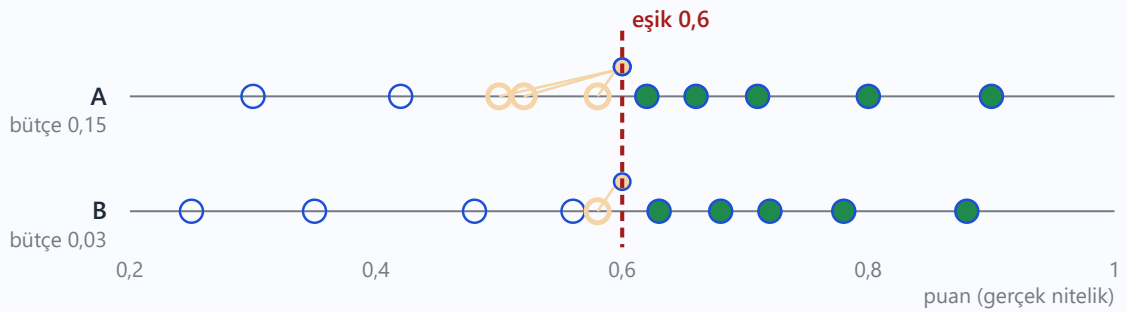
Tablo 11.17: Stratejik sınıflandırma örneğinde gerçek puanlar.

Kişinin kararı basittir. Gerçek puanı eşiğin altındaysa ve eşiğe olan uzaklığı bütçesinden küçük ya da ona eşitse, puanını tam eşiğe taşır ve kabul edilir. Uzaklık bütçeden büyükse yükseltme bir işe yaramaz; kişi maliyete katlanmaz.

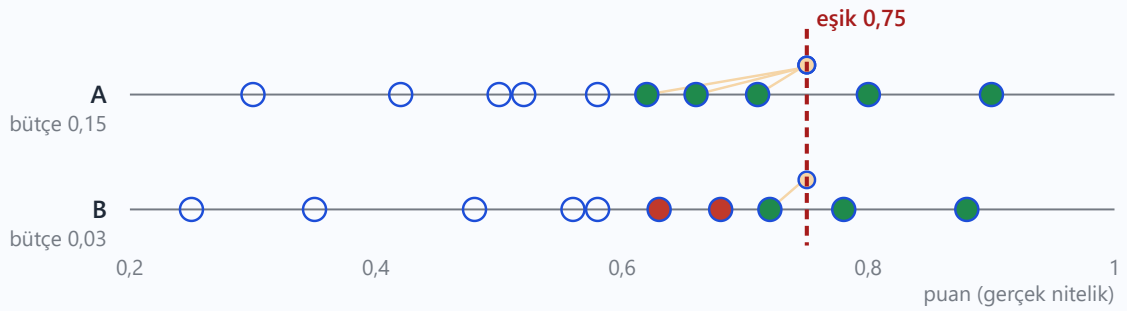
Örnek 11.25

1. **Eşik 0,6, oyunlama.** A'da gerçek puanı en az $0{,}6 - 0{,}15 = 0{,}45$ olan niteliksizler eşiğe ulaşabilir: 0,50, 0,52 ve 0,58. A'dan 3 niteliksiz kişi geçer. B'de en az $0{,}6 - 0{,}03 = 0{,}57$ gerekir: yalnız 0,58. B'den 1 niteliksiz kişi geçer.
2. **Kurum tepki verir.** Kurum oyunlamayı fark eder ve eşiği A'nın bütçesini hesaba katarak $0{,}6 + 0{,}15 = 0{,}75$'e yükseltir. Niteliksizlerin ulaşabileceği en yüksek puan A'da $0{,}58 + 0{,}15 = 0{,}73$, B'de $0{,}58 + 0{,}03 = 0{,}61$'dir. İkisi de 0,75'in altındadır; artık hiçbir niteliksiz kişi geçemez.
3. **A'nın nitelikli üyeleri.** Eşiği geçmek için gerçek puanı en az $0{,}75 - 0{,}15 = 0{,}60$ olmalıdır. A'nın beş nitelikli üyesinin hepsi bunu sağlar (en düşüğü 0,62). A'da 5/5 geçer.
4. **B'nin nitelikli üyeleri.** Gerçek puanı en az $0{,}75 - 0{,}03 = 0{,}72$ olmalıdır. 0,72, 0,78 ve 0,88 geçer; 0,63 ve 0,68 geçemez. B'de yalnız 3/5 geçer.

Stratejik davranışa karşı alınan önlem yeni bir eşitsizlik doğurdu. Önlem niteliksizleri dışarıda tutmayı başardı, ama bedelini bütçesi küçük B grubunun nitelikli üyeleri ödedi. Kurumun kuralı iki gruba aynı görünür (aynı eşik), ama iki grubun bu kurala uyum sağlama gücü farklıdır.



Şekil 11.16: Eşik 0,6'da oyunlama. Dolu daireler nitelikli kişilerdir (puan ≥ 0,6). A'dan üç, B'den bir niteliksiz kişi puanını eşiğe taşıyarak geçer (turuncu halkalar ve oklar).



Şekil 11.17: Eşik 0,75'e yükseltildikten sonra. Hiçbir niteliksiz kişi geçemez. A'nın nitelikli beş üyesinin hepsi bütçesiyle eşiğe ulaşır; B'nin nitelikli 0,63 ve 0,68 puanlı üyeleri (kırmızı) ulaşamaz.

Şekil 11.16 ve Şekil 11.17 iki aşamayı karşılaştırır. Eşiği yükseltmek, oyunlamanın bedelini herkese yüklemenin kaba bir yoludur. Daha incelikli çözümler de vardır: oyunlanması zor özelliklere ağırlık vermek, ya da oyunlama için gereken kaynağı (örneğin danışmanlık, eğitim) düşük bütçeli gruba da erişilebilir kılmak. Örnekte bir aralık hesabı da yapılabilir: niteliksizlerin hiçbirinin geçmemesi için eşik 0,73'ten büyük olmalıdır; B'nin bütün niteliklilerinin geçmesi için eşik en çok *0,63 + 0,03 = 0,66* olmalıdır. İki koşul çeliştiği için tek bir eşikle iki amaca birlikte ulaşılamaz.

11.5.4 Algoritmik denetim

Algoritmik denetim (*algorithmic audit*), bir karar sisteminin amacına uygun, doğru ve adil çalışıp çalışmadığının sistematik olarak incelenmesidir. Denetim, sistemin yalnız kodunu değil bütün yaşam döngüsünü kapsar.

Ne denetlenir?

Veri kaynağı, hedef değişken, grup bazlı performans ve insan gözetimi birlikte incelenir. Hedef değişkenin bir vekil olup olmadığı (11.1.3'teki sağlık harcaması örneği) çoğu zaman en önemli sorudur.

Belgeleme

Model kartları (*model cards*) ve veri kümesi kartları, bir modelin ya da veri kümesinin varsayımlarını, kullanım amacını ve bilinen sınırlamalarını görünür kılan kısa belgelerdir. Bir model kartı örneğin şunu yazar: "Bu model 2015–2020 kredi başvurularıyla eğitildi; 25 yaş altı başvuranlarda doğruluğu düşüktür."

Dağıtım sonrası

Model kullanılmaya başlandıktan sonra da izlenir. **Veri kayması** (*data drift*), yani modelin karşılaştığı verinin eğitim verisinden zamanla uzaklaşması, itirazlar ve beklenmeyen zararlar düzenli olarak takip edilir.

Katılım

Etkilenen grupların görüşü alınır. İtiraz hakkı ve açık bir sorumluluk zinciri tanımlanır.

Bölümün sayısal kısmı, bir denetimin en somut adımını verir: bir **denetim listesi**. Listede taban oranları, seçim oranı, TPR, FPR, PPV ve kalibrasyon her grup için ayrı ayrı raporlanır. Uyumsuzluk teoremleri yüzünden bu ölçütlerin hepsi birden sağlanamayacağı için, hangi ölçütün neden önceliklendirildiği yazılı olarak gerekçelendirilir. Tablo 11.5 böyle bir raporun en küçük hâlidir.

11.5.5 Algoritmalar tarafsız olabilir mi?

Algoritmaların "matematik oldukları için tarafsız" olduğu sıkça söylenir. Bölümün bulguları bu iddiayı dört yönden sınırlar:

- Algoritmalar, hedef, veri, ölçüt ve eşik seçimlerini yapan kurumların değerlerinden bağımsız değildir. "Neyi tahmin edeceğiz?" sorusunun cevabı bile bir değer seçimidir.
- Tarihsel eşitsizlikler eğitim verisine ve hedef değişkenlere yansıyabilir.
- Farklı adalet ölçütleri aynı karar için çelişen sonuçlar üretir (uyumsuzluk teoremleri). Bir ölçüt seçmek, ötekinden vazgeçmektir.
- Model, dağıtımdan sonra davranışı değiştirerek yeni geri besleme döngüleri yaratabilir (11.5.6).

Temel zorluklar da açıktır. Adalet kavramı bağlama göre değişir: kredi kararında önemli olan hata türü, tıbbi tarama kararında farklı olabilir. Doğruluk, mahremiyet ve grup adaleti arasında ödünleşimler vardır. Adalet anlayışları kültürden kültüre değişir. Bu zorluklara verilen makul cevap, mutlak tarafsızlık iddiasından vazgeçmektir. Bunun yerine kararın etkileri ölçülür, yapılan tercihler belgelenir ve etkilenen kişiler için bir itiraz mekanizması kurulur.

11.5.6 Kod izleme: kendini doğrulayan veri

Geri besleme döngüsünün en bilinen örneği tahmine dayalı polisliktir (*predictive policing*). Devriye, geçmiş kayıtlarda olayın çok olduğu bölgeye gönderilir. Ama olay yalnız devriyenin bulunduğu yerde kaydedilir. Bu durumda kayıtlar, olayların gerçek dağılımını değil devriyenin nereye gittiğini yansıt-

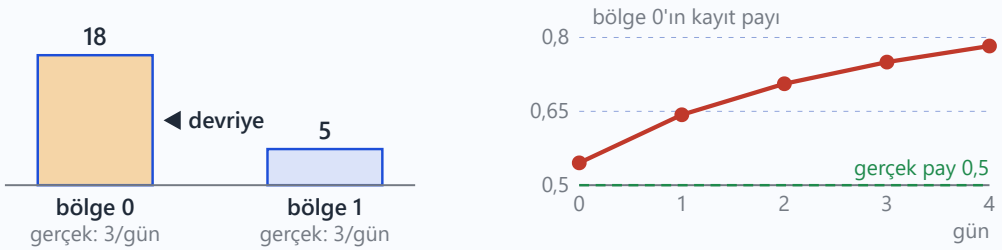
maya başlar (Ensign vd., 2018). Aşağıdaki kod bu döngünün en sade modelidir.

```
kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
gercek = [3, 3]    # gerçek günlük olay: eşit!
for gun in range(1, 5):
  hedef = 0 if kayit[0] >= kayit[1] else 1
  kayit[hedef] += gercek[hedef]  # yalnız orada
  pay = kayit[0] / sum(kayit)
  print(gun, hedef, kayit, round(pay, 2))
```

Kodda iki bölge vardır. `kayit` listesi her bölgenin geçmiş olay kaydını tutar: bölge 0'da 6, bölge 1'de 5 kayıt vardır. `gercek` listesi her bölgede her gün gerçekte olan olay sayısıdır: iki bölgede de 3'tür. Yani iki bölge gerçekte eşittir; kayıtlardaki bir fark belki tesadüftür. Her gün devriye kaydı fazla olan bölgeye gider (eşitlikte bölge 0'a). Yalnız devriyenin gittiği bölgede o günün olayları kayda geçer. `pay`, bölge 0'ın toplam kayıt içindeki payıdır.

gün	hedef	kayit	pay	görünmeyen olay (bölge 1)
1	0 (6 ≥ 5)	[9, 5]	9/14 ≈ 0,64	3
2	0 (9 ≥ 5)	[12, 5]	12/17 ≈ 0,71	3
3	0 (12 ≥ 5)	[15, 5]	15/20 = 0,75	3
4	0 (15 ≥ 5)	[18, 5]	18/23 ≈ 0,78	3

Tablo 11.18: Geri besleme kodunun izlemesi. Başlangıç: kayit = [6, 5], pay = 6/11 ≈ 0,55.



Şekil 11.18: Dört gün sonra kayıtlar [18, 5]'tir. Sağdaki grafik bölge 0'ın kayıt payını gösterir: 0,55'ten 0,78'e çıkar. Gerçek pay (yeşil kesikli çizgi) 0,5'te sabittir.

Her gün devriye bölge 0'a gider, çünkü bölge 0'ın kaydı hep fazladır. Bölge 0'ın kaydı her gün 3 artar; bölge 1'deki 3 olay ise hiç görülmez ve kaydı 5'te donar. Şekil 11.18 sonucu gösterir: pay 0,55'ten 0,78'e çıkar. Veri, modelin kararıyla toplandığı için modeli "doğrular". Bir gözlemci kayıtlara bakıp "bölge 0'da suç bölge 1'in üç katından fazla" sonucuna varabilir; oysa gerçekte iki bölge eşittir. Başlangıçtaki tek kayıtlık fark (6 ile 5), belki yalnız tesadüftü, ama kendini büyüterek kalıcı bir "gerçek" hâline geldi.

Örnek 11.26

Aynı kod `kayit = [5, 6]` ile çalıştırılırsa 4 gün sonra bölge 0'ın payı kaçtır?

1. Gün 1: bölge 1'in kaydı fazladır (6 > 5); devriye bölge 1'e gider: [5, 9].
2. Gün 2: [5, 12]. Gün 3: [5, 15]. Gün 4: [5, 18].
3. Pay *5/23 ≈ 0,22*'dir.

Devriye hep bölge 1'e gider. Başlangıçtaki tek kayıtlık fark bu kez ters yönde büyür. Sonucun yönünü gerçek suç oranları değil, başlangıçtaki rastlantısal fark belirler.

Döngüyü kırmanın yolları vardır. Devriye zaman zaman rastgele bölgelere gönderilebilir; böylece her bölgeden bir miktar yansız veri toplanır. Ya da kayıtlar, o bölgeye ne kadar devriye gönderildiğine bölünerek düzeltilebilir. Bu iki fikir, öğrenme kuramındaki "keşif ve sömürü" ödünleşimine karşılık gelir: yalnız bilinen en iyi seçeneği sömüren bir sistem, öteki seçenekler hakkında hiç bilgi toplamaz.

11.5.7 Gerçek hayat uygulamaları

Bölümün kavramları üç alanda özellikle sık karşımıza çıkar:

Kredi ve bankacılık
: Skorlama modelleri. Asıl seçim kalibrasyon ile fırsat eşitliği arasındadır: kurum puanın her grupta aynı riski anlatmasını, başvuranlar ise ödeyebilecek olanların aynı şansı almasını ister.

İşe alım ve insan kaynakları
: Özgeçmiş eleme. Tarihsel veri önyargısı (Amazon örneği) ve beşte dört kuralı en sık konulardır.

Sağlık
: Teşhis ve tedavi önerisi. Vekil hedef sorunu (harcama ≠ ihtiyaç) burada ağır sonuçlar doğurabilir.

Bu alanlarda uygulanan temel analiz, bir eşiğin değiştirilmesinin ölçütleri nasıl etkilediğini incelemektir. Bölümün örnek modeli yalnız iki puan verdiği için yalnız üç anlamlı karar kuralı vardır: herkesi reddetmek (eşik 0,8'in üstünde), yalnız 0,8 puanlıları onaylamak (eşik 0,2 ile 0,8 arasında) ve herkesi onaylamak (eşik 0,2'nin altında).

Örnek 11.27

1. **Herkesi onayla.** İki grupta da seçim oranı 1, TPR 1, FPR 1'dir. Demografik eşitlik ve eşitlenmiş olasılıklar sağlanır. Ama PPV, onaylananların (yani herkesin) ödeme oranıdır: A'da 0,50, B'de 0,32. Tahmin eşitliği bozulur. Doğruluk A'da 0,50, B'de 0,32'ye düşer.
2. **Yalnız 0,8 puanlıları onayla** (bugünkü model). Tahmin eşitliği ve kalibrasyon sağlanır; demografik eşitlik ve eşitlenmiş olasılıklar sağlanmaz (Tablo 11.5). Doğruluk iki grupta da 0,80'dir.

3. **Herkesi reddet.** Seçim oranı, TPR ve FPR iki grupta da 0'dır; demografik eşitlik ve eşitlenmiş olasılıklar sağlanır. PPV tanımsızdır (onaylanan yok). Doğruluk A'da 0,50, B'de 0,68'dir. Kurum hiç kredi vermez.

Eşitlenmiş olasılıkları sağlayan iki kural da (herkesi onayla, herkesi reddet) modeli tümüyle işe yaramaz hâle getirir. Herkesi reddetmek, 11.3.1'deki uyumsuzluk teoreminin *TPR = 0* uç durumudur; herkesi onaylamak ise eşitlenmiş olasılıkları sağlarken tahmin eşitliğini bozar. Üç kural bir şeyi somut olarak gösterir: modelin ayırt etme gücü değişmedikçe yalnız eşikle oynayarak bütün ölçütler birlikte sağlanamaz.

Sık yapılan hata

- **Kuralı herkese aynı uygulamayı adalet saymak:** Stratejik sınıflandırma örneğinde eşik iki gruba aynıdır, ama gruplar kurala farklı maliyetle uyum sağlar. Biçimsel eşitlik, sonuç eşitliği getirmez.
- **Kayıtları gerçek sanmak:** Geri besleme döngüsünde kayıtlar devriyenin nereye gittiğini ölçer. Kararın etkilediği bir süreçle toplanan veri, o kararı değerlendirmek için kullanılırken dikkat ister.
- **Denetimi tek seferlik saymak:** Veri kayması ve stratejik tepki yüzünden bugün adil olan bir model yarın olmayabilir. Denetim, dağıtımdan sonra da sürer.
- **"İnsan incelemesi" ile hesap verebilirliği eşitlemek:** Kararı yalnız onaylayan, gerekçesini sorgulamayan bir insan hesap verebilirlik sağlamaz. İtiraz edenin kararın gerekçesine ulaşabilmesi gerekir.

11.6 Özet

- **Önyargı** tarihsel veriden, temsil eksikliğinden, vekil değişkenlerden ve tasarım ile geri besleme döngülerinden gelir. Korunan özelliği modelden çıkarmak yetmez; vekiller onu taşır. Bu yüzden sonuçlar grup bazında ölçülür.
- **Karışıklık matrisi** (TP, FP, FN, TN) bütün grup ölçütlerinin kaynağıdır: seçim oranı $(TP + FP)/n$, TPR $TP/(TP + FN)$, FPR $FP/(FP + TN)$, PPV $TP/(TP + FP)$, taban oranı $(TP + FN)/n$.
- **Ölçütler:** demografik eşitlik (seçim oranı; beşte dört kuralı oranı ≥ 0,8 ister), eşitlenmiş olasılıklar (TPR ve FPR; yalnız TPR ise fırsat eşitliği), tahmin eşitliği (PPV), kalibrasyon (puanı s olanların gerçekleşme oranı her grupta s). Eşitlenmiş olasılıklar başvuranın, son ikisi kurumun bakışıdır; demografik eşitlik gerçek sonucu hiç kullanmaz.
- **Örneğimiz:** kalibre ve PPV'si eşit (0,8) bir model; buna rağmen seçim oranı 0,5/0,2, TPR 0,8/0,5, FPR 0,2/0,059. İki grupta da doğruluk %80'dir; doğruluk hataların kime düştüğünü göstermez.
- **Bireysel adalet** benzer kişilere benzer puan ister: $|f(x) - f(y)| \le L \cdot d(x, y)$. Zorluğu, uzaklık d'nin seçiminin de bir değer yargısı olmasıdır.

- **Uyumsuzluk:** $FPR = p/(1-p) \cdot (1-PPV)/PPV \cdot TPR$ (Chouldechova). Taban oranları farklıysa tahmin eşitliği ile eşitlenmiş olasılıklar, mükemmel tahmin dışında, birlikte sağlanamaz. Kleinberg vd.: kalibrasyon ve iki sınıf dengesi de ancak mükemmel tahminde ya da eşit taban oranında birlikte sağlanır. COMPAS tartışmasında iki taraf da kendi ölçütüne göre haklıydı.
- **Ödünleşim:** eşitlenmiş olasılıkları zorlamak B'de PPV'yi 0,8'den 0,65'e düşürür; rastgeleleştirmeyle fırsat eşitliği ($q = 0{,}6$) doğruluğu 0,8'den 0,512'ye indirir. Adalet kısıtı ön işleme, eğitim ya da son işlemeyle eklenir; hangi ölçütün seçileceği bir değer kararıdır.
- **Adil bölüşüm:** orantılılık $v_i(X_i) \geq 1/n$, kıskançsızlık $v_i(X_i) \geq v_i(X_j)$. Kes-ve-seç iki kişide orantılı ve kıskançsızdır; dürüst kesmek maksimin stratejisidir ve seçen kesenden avantajlıdır. İki kişide orantılılık ⇔ kıskançsızlık; üç ve daha fazla kişide kıskançsızlık daha güçlüdür.
- **Algoritmalar:** son azalan ve hareketli bıçak n kişide orantılıdır ($O(n^2)$ sorgu), Even–Paz daha az soruyla; Selfridge–Conway 3 kişide, Aziz–Mackenzie n kişide kıskançsızdır. Kesim sorgusu ikiye bölmeyle bulunur; k sorgudan sonra hata en çok $2^{-(k+1)}$'dir.
- **Bölünemez mallar:** kıskançsızlık her zaman mümkün değildir; sırayla seçim EF1 üretir ama açgözlü seçim en iyi tepki olmayabilir. Nash sosyal refahını maksimize eden paylaşım Pareto optimal ve EF1'dir. Örnekte adaletin bedeli $18/17 \approx 1{,}059$'dur.
- **Etik:** şeffaflık, hesap verebilirlik, adalet, mahremiyet; GDPR madde 22, AB Yapay Zekâ Yasası, KVKK. Denetimde her grup için taban oranı, seçim oranı, TPR, FPR, PPV ve kalibrasyon raporlanır ve ölçüt seçimi gerekçelendirilir.
- **Teşvikler:** algoritmaya tepki veren insanlar Goodhart yasasını işletir. Stratejik sınıflandırmada oyunlama maliyeti gruplar arasında farklıysa yeni bir eşitsizlik doğar; geri besleme döngüsünde veri, kararın kendisini doğrular.

Kavram	Ne söyler?	Nasıl hesaplanır?
Demografik eşitlik	Gruplar eşit oranda onay alır	Seçim oranları karşılaştırılır
Eşitlenmiş olasılıklar	Aynı gerçek durumdakilerin şansı eşittir	TPR ve FPR karşılaştırılır
Tahmin eşitliği	Onay iki grupta aynı anlamı taşır	PPV karşılaştırılır
Kalibrasyon	Puan her grupta aynı riski anlatır	Her puan düzeyinde gerçekleşme oranı
Bireysel adalet	Benzer kişilere benzer puan	Lipschitz koşulu
Chouldechova özdeşliği	Taban oranı farklıysa ölçütler çatışır	$FPR = p/(1-p) \cdot (1-PPV)/PPV \cdot TPR$
Orantılılık	Herkes kendi gözünde ≥ 1/n alır	Kes-ve-seç, son azalan, hareketli bıçak
Kıskançsızlık	Kimse başkasının payını tercih etmez	Kes-ve-seç (2), Selfridge–Conway (3)
EF1	Kıskançlık en çok bir mallıktır	Sırayla seçim, en büyük Nash refahı
Adaletin bedeli	Adalet kısıtının refah kaybı	En büyük SW ÷ adil paylaşımlarda en büyük SW

Tablo 11.19: Bölümün kavramları bir bakışta.

11.7 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

algoritmik adalet *algorithmic fairness* · önyargı *bias* · korunan özellik *protected attribute* · vekil değişken *proxy variable* · karışıklık matrisi *confusion matrix* · doğruluk *accuracy* · seçim oranı *selection rate* · doğru pozitif oranı *true positive rate (TPR)* · yanlış pozitif oranı *false positive rate (FPR)* · yanlış negatif oranı *false negative rate (FNR)* · kesinlik (PPV) *positive predictive value, precision* · negatif kestirim değeri (NPV) *negative predictive value* · taban oranı *base rate* · grup adaleti *group fairness* · demografik eşitlik *demographic parity* · beşte dört kuralı *four-fifths rule* · eşitlenmiş olasılıklar *equalized odds* · fırsat eşitliği *equal opportunity* · tahmin eşitliği *predictive parity* · kalibrasyon *calibration within groups* · bireysel adalet *individual fairness* · Lipschitz koşulu *Lipschitz condition* · uyumsuzluk teoremleri *impossibility theorems* · Chouldechova özdeşliği *Chouldechova's identity* · sınıf dengesi *balance for the positive/negative class* · kayıp fonksiyonu *loss function* · ön işleme / son işleme *pre-processing / post-processing* · adil bölüşüm *fair division* · pasta kesme *cake cutting* · değer fonksiyonu *valuation function* · Robertson–Webb modeli *Robertson–Webb query model* · kesim sorgusu *cut query* · ikiye bölme *bisection* · orantılılık *proportionality* · kıskançsızlık *envy-freeness* · kes-ve-seç *cut-and-choose* · son azalan *last diminisher* · hareketli bıçak *moving knife* · bölünemez mallar *indivisible goods* · demet *bundle* · sırayla seçim *round-robin* · bir mala kadar kıskançsız (EF1) *envy-free up to one good* · faydacı refah *utilitarian welfare* · eşitlikçi refah *egalitarian welfare* · Nash sosyal refahı *Nash social welfare* · Pareto verimliliği *Pareto efficiency* · adaletin bedeli *price of fairness* · şeffaflık *transparency* · açıklanabilirlik *explainability* · hesap verebilirlik *accountability* · mahremiyet *privacy* · stratejik sınıflandırma *strategic classification* · oyunlama *gaming* · Goodhart yasası *Goodhart's law* · algoritmik denetim *algorithmic audit* · model kartı *model card* · veri kayması *data drift* · geri besleme döngüsü *feedback loop*

11.8 Kendini Dene

1. Bir D grubunun karışıklık matrisi TP = 24, FP = 6, FN = 16, TN = 54'tür. Seçim oranı, TPR, FPR, PPV ve doğruluk nedir? Bölümdeki A grubuyla (seçim 0,5; TPR 0,8; FPR 0,2; PPV 0,8) karşılaştırıldığın-

da hangi ölçüt sağlanır? Beşte dört kuralı ne der?

Cevap

n = 100. Seçim *(24 + 6)/100 = 0,30*; TPR *24/40 = 0,6*; FPR *6/60 = 0,1*; PPV *24/30 = 0,8*; doğruluk *(24 + 54)/100 = 0,78*. Yalnız tahmin eşitliği sağlanır (0,8 = 0,8). Beşte dört kuralı: *0,30/0,50 = 0,6 < 0,8* ⇒ ihlal. Özdeşlikle denetim: taban oranı *p = 40/100 = 0,4*; *(0,4/0,6) · (0,2/0,8) · 0,6 = 0,1* ✓.

2. Bir grupta taban oranı *p = 0,3*, *PPV = 0,6*, *TPR = 0,7*'dir. FPR kaçtır? Taban oranı 0,5 olan ikinci bir grupta PPV ve TPR aynı tutulursa FPR ne olur? Bu iki grupta eşitlenmiş olasılıklar sağlanabilir mi?

Cevap

Birinci grup: *(0,3/0,7) · (0,4/0,6) · 0,7 = 0,3 · (2/3) = 0,2*. İkinci grup: *(0,5/0,5) · (2/3) · 0,7 = 7/15 ≈ 0,467*. TPR'ler eşit ama FPR'ler farklıdır (0,2 ≠ 0,467); eşitlenmiş olasılıklar sağlanmaz. PPV eşitken taban oranı farkı, *p/(1 − p)* çarpanı üzerinden FPR'ye taşınır.

3. Bir model iki puan veriyor: 0,7 ve 0,1. Bir grupta puanı 0,7 olan 30 kişiden 21'i, puanı 0,1 olan 70 kişiden 7'si krediyi öder. Model bu grupta kalibre midir? Taban oranı nedir? Ödeyenlerin ve ödemeyenlerin ortalama puanı kaçtır?

Cevap

21/30 = 0,7 ve *7/70 = 0,1*: iki puan düzeyinde de gerçekleşme oranı puana eşittir, model kalibredir. Taban oranı *(21 + 7)/100 = 0,28*; bu, ortalama puana *(30 · 0,7 + 70 · 0,1)/100 = 0,28* eşittir. Ödeyenlerin ortalama puanı *(21 · 0,7 + 7 · 0,1)/28 = 15,4/28 = 0,55*. Ödemeyenler 9 (0,7 puanlı) ve 63 (0,1 puanlı) kişidir; ortalama *(9 · 0,7 + 63 · 0,1)/72 = 12,6/72 = 0,175*.

4. Pasta [0, 1]; sol yarı vanilya, sağ yarı çikolata. Ali her yere eşit değer verir. Ayşe için vanilyanın tamamı 2/5, çikolatanın tamamı 3/5 değerindedir (her yarıda düzgün). (a) Ali keser, Ayşe seçerse herkes kendi gözünde ne alır? (b) Ayşe keserse nereden keser ve kim ne alır?

Cevap

(a) Ali *x = 1/2*'den keser. Ayşe için sol 2/5, sağ 3/5'tir; sağı seçer. Ali 1/2, Ayşe 3/5 alır. (b) Vanilyanın tamamı Ayşe için 2/5 < 1/2 olduğundan kesim çikolatadadır. Çikolatada birim uzunluk *(3/5)/(1/2) = 6/5* değerindedir. *2/5 + (6/5)(x − 1/2) = 1/2* ⇒ *x − 1/2 = 1/12* ⇒ *x = 7/12*. Ali sol parçayı seçer ve *7/12 ≈ 0,583* alır; Ayşe 1/2 alır. Seçen yine avantajlıdır.

5. Bireysel adalet koşulunda *L = 3*, *d(x, y) = 0,1* ve *f(x) = 0,25*'tir. Puanlar [0, 1] aralığındaysa *f(y)* hangi

aralıkta olabilir?

Cevap

$|f(y) - 0{,}25| \le 3 \cdot 0{,}1 = 0{,}3 \Rightarrow -0{,}05 \le f(y) \le 0{,}55$. Puan 0'ın altına inemediği için aralık $[0; 0{,}55]$'tir.

6. İki kişi dört bölünemez malı paylaşıyor. Değerler: Ali (m1 8, m2 4, m3 2, m4 1), Ayşe (m1 5, m2 5, m3 4, m4 1). Ali'nin başladığı sırayla seçimde kim neyi alır; paylaşım kıskançsız mı, EF1 mi? Nash sosyal refahını en büyük yapan paylaşım hangisidir?

Cevap

Ali m1 (8), Ayşe m2 (5), Ali m3 (2), Ayşe m4 (1). Ali {m1, m3} = 10, Ayşe {m2, m4} = 6. Ayşe, Ali'nin demetine $5 + 4 = 9 > 6$ değer verir: kıskanır. m1 çıkarılınca $4 \le 6$: EF1. Çarpım 60'tır. 16 paylaşım taranınca en büyük çarpım Ali {m1, m4} = 9, Ayşe {m2, m3} = 9 paylaşımındadır: 81. Bu paylaşım kıskançsızdır (Ali, Ayşe'nin demetine 6, Ayşe Ali'ninkine 6 verir) ve toplamı (18) da en büyüktür; bu örnekte adaletin bedeli 1'dir.

7. Geri besleme kodu `kayit = [7, 5]` ve `gercek = [2, 4]` ile çalıştırılıyor; yani bölge 1'de gerçekte daha çok olay oluyor. 4 gün sonra kayıtlar ve bölge 0'ın kayıt payı nedir? Gerçek pay kaçtır?

Cevap

Her gün bölge 0'ın kaydı fazladır ve devriye oraya gider: [9, 5], [11, 5], [13, 5], [15, 5]. Paylar 0,64; 0,69; 0,72; 0,75. Gerçek pay $2/(2 + 4) \approx 0{,}33$'tür. Kayıtlar gerçeği tersine çevirir: olayların çoğu bölge 1'dedir ama kayıtlara göre bölge 0 üç kat “daha tehlikeli” görünür.

11.9 İleri okuma

- S. Barocas, M. Hardt ve A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities* (2023): grup ölçütlerini, uyumsuzluk sonuçlarını ve nedensellik temelli yaklaşımları ayrıntılı ve örnekli anlatan ders kitabıdır; çevrim içi sürümü ücretsizdir.
- A. Chouldechova, “Fair prediction with disparate impact” (2017) ve J. Kleinberg, S. Mullainathan, M. Raghavan, “Inherent trade-offs in the fair determination of risk scores” (2017): bu bölümdeki iki uyumsuzluk sonucunun özgün makaleleridir; ikisinin de giriş bölümleri az matematikle okunabilir.
- A. D. Procaccia, “Cake cutting algorithms”, *Handbook of Computational Social Choice*, Bölüm 13 (2016): Robertson–Webb modeli, son azalan, Even–Paz ve kıskançsız protokollerin karmaşıklığını özetler.
- S. J. Brams ve A. D. Taylor, *Fair Division: From Cake-Cutting to Dispute Resolution* (1996): adil bölüşüm yöntemlerini gerçek anlaşmazlık örnekleriyle, teknik olmayan bir dille anlatır.

11.10 Kaynaklar

1. J. Angwin, J. Larson, S. Mattu, L. Kirchner, “Machine Bias”, *ProPublica*, 2016.
2. H. Aziz, S. Mackenzie, “A discrete and bounded envy-free cake cutting protocol for any number of agents”, *IEEE FOCS*, 2016.
3. S. Barocas, M. Hardt, A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*, MIT Press, 2023.
4. S. J. Brams, A. D. Taylor, *Fair Division: From Cake-Cutting to Dispute Resolution*, Cambridge University Press, 1996.
5. E. Budish, E. Cantillon, “The multi-unit assignment problem: theory and evidence from course allocation at Harvard”, *American Economic Review* 102(5), 2012.
6. J. Buolamwini, T. Gebru, “Gender Shades: Intersectional accuracy disparities in commercial gender classification”, *Proc. FAT**, 2018.
7. I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, “The efficiency of fair division”, *Theory of Computing Systems* 50(4), 2012.
8. I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, J. Wang, “The unreasonable fairness of maximum Nash welfare”, *ACM EC*, 2016.
9. A. Chouldechova, “Fair prediction with disparate impact: a study of bias in recidivism prediction instruments”, *Big Data* 5(2), 2017.
10. J. Dastin, “Amazon scraps secret AI recruiting tool that showed bias against women”, *Reuters*, 10 Ekim 2018.
11. L. E. Dubins, E. H. Spanier, “How to cut a cake fairly”, *American Mathematical Monthly* 68(1), 1961.
12. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, “Fairness through awareness”, *ITCS*, 2012.
13. D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, S. Venkatasubramanian, “Runaway feedback loops in predictive policing”, *Proc. FAT**, 2018.
14. S. Even, A. Paz, “A note on cake cutting”, *Discrete Applied Mathematics* 7(3), 1984.
15. M. Hardt, E. Price, N. Srebro, “Equality of opportunity in supervised learning”, *NeurIPS*, 2016.
16. L. Hu, N. Immorlica, J. W. Vaughan, “The disparate effects of strategic manipulation”, *Proc. FAT**, 2019.
17. A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017 (adil bölüşüm bölümü).
18. J. Kleinberg, S. Mullainathan, M. Raghavan, “Inherent trade-offs in the fair determination of risk scores”, *ITCS*, 2017.
19. A. D. Procaccia, “Cake cutting algorithms”, F. Brandt vd. (ed.), *Handbook of Computational Social Choice* içinde, Bölüm 13, Cambridge University Press, 2016.
20. Avrupa Parlamentosu ve Konseyi, Genel Veri Koruma Tüzüğü (GDPR), Tüzük (AB) 2016/679, md. 22 (uygulama 2018).
21. Avrupa Parlamentosu ve Konseyi, Yapay Zekâ Yasası, Tüzük (AB) 2024/1689.
22. 6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK), *Resmî Gazete*, 7 Nisan 2016.
23. OECD, *Recommendation of the Council on Artificial Intelligence* (OECD YZ ilkeleri), 2019.

KISIM IV · BÖLÜM 12

Davranışsal ve Kolektif Karar

Klasik oyun kuramındaki oyuncular kusursuz hesap yapan, yalnızca kendi kazancını önemseyen varlıklardır. Laboratuvarda ve gündelik hayatta gözlenen insanlar ise bu tablodan sistematik biçimde ayrılır: kayıpları kazançlardan daha ağır hisseder, küçük hatalar yapar, adalete ve niyete önem verir. Gruplar hâlinde karar verirken de ortak kaynakları tüketir, kamu malına az katkı yapar, birbirini taklit ederek bilgiyi kaybeder ya da tam tersine bağımsız tahminleri birleştirerek tek tek üyelerinden daha isabetli olur. Bu not, bu sapmaları ölçen modelleri (beklenti kuramı, kuantal tepki, eşitsizlikten kaçınma) ve grup kararlarını açıklayan modelleri (ortak mallar, kamu malı ve ceza, eşikli projeler, bilgi şelaleleri, jüri teoremi) sayısal örneklerle adım adım anlatır.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum12`

ÖĞRENME ÇIKTILARI

Davranışsal karar
Beklenti kuramıyla bir bahsin değerini hesaplamak; kayıptan kaçınma, bağış etkisi ve çerçevelemeyi açıklamak; kuantal tepkinin Nash öngörüsünü nasıl değiştirdiğini göstermek.

Sosyal tercihler
Diktatör, ültimatom ve güven oyunlarındaki sapmaları eşitsizlikten kaçınma ve karşılıklılıkla açıklamak; Fehr–Schmidt kabul eşiğini hesaplamak.

Ortak kaynaklar ve kamu malları
Ortak mallar dengesini optimumla karşılaştırmak; Ostrom ilkelerini, ceza ve ikinci derece beleşçiliği, eşikli projelerin dengelerini yorumlamak.

Toplu bilgi
Kalabalığın bilgeliğini çeşitlilik teoremiyle, bilgi şelalesini Bayes hesabıyla, Condorcet jüri teoremini binom olasılığıyla hesaplamak.

12.1 Davranışsal karar

Oyun kuramının klasik modelinde her oyuncu beklenen faydasını maksimize eder, hesap hatası yapmaz ve yalnızca kendi kazancını önemser. Bu varsayımlar modelleri sade tutar ve çoğu zaman şaşırtıcı derecede iyi öngörüler verir. Ama insanlarla yapılan deneyler, bazı durumlarda öngörülerin açıkça tutmadığını gösterir. Davranışsal oyun kuramı bu sapmaları "insanlar irrasyoneldir" diye geçiştirmez; sapmaların hangi yönde ve ne büyüklükte olduğunu ölçer ve modele katar. Bu kısım önce bireysel kararları ele alır: risk altında seçim, referans noktası, çerçeveleme ve hatalı ama akıllı seçim.

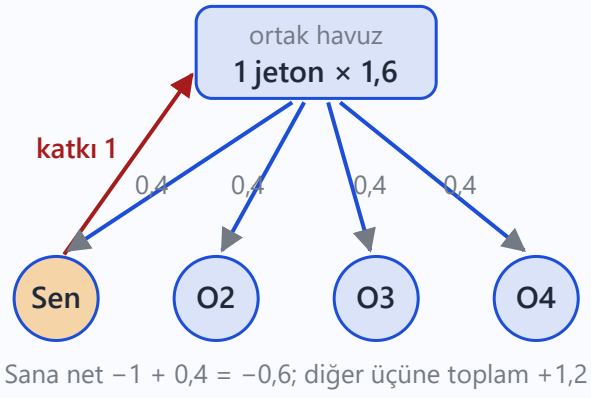
12.1.1 Bir oyunla başlangıç: Kamu Malı Oyunu

Bölümün bütün konularını bağlayan oyun, **kamu malı oyunu**dur (*public goods game*). Kurallar kısadır. Dört oyuncu her tur 10 jeton alır. Her biri dilediği kadarını ortak havuza katar. Havuzdaki jetonlar 1,6 ile çarpılır ve dört oyuncuya eşit bölünür. Havuza katılmayan jeton oyuncunun cebinde kalır.

Oyuncu *i*'nin havuza kattığı jeton sayısı k_i olsun. Bir turdaki kazancı şöyle yazılır:

$$u_i = 10 - k_i + 1{,}6 \cdot (k_1 + k_2 + k_3 + k_4) / 4 = 10 - k_i + 0{,}4 \cdot \Sigma k$$

Formüldeki 0,4, çarpanın (1,6) oyuncu sayısına (4) bölümüdür. Şekil 12.1 bir jetonun yolculuğunu gösterir. Oyuncu havuza 1 jeton kattığında o jeton 1,6 olur ve dörde bölünür; her oyuncuya 0,4 düşer. Katan oyuncu 1 jeton vermiş, 0,4 geri almıştır: neti −0,6'dır. Öteki üç oyuncunun her biri ise hiçbir şey vermeden 0,4 kazanmıştır; üçünün toplam kazancı +1,2'dir.



Şekil 12.1: Kamu Malı Oyunu'nda bir jetonun yolu. Havuza katılan 1 jeton 1,6 olur ve dört oyuncuya 0,4'er bölünür; katan oyuncunun neti −0,6, öteki üç oyuncunun toplam kazancı +1,2'dir.

Bu küçük hesap bölümün ana gerilimini taşır. Grubun bakış açısından her jeton katkısı değerlidir: verilen 1 jeton gruba toplam 1,6 döndürür. Bireyin bakış açısından ise her jeton katkısı zararlıdır: verilen 1 jeton katana yalnızca 0,4 döndürür. Herkes 10 katarsa havuz 40 olur, 64'e çıkar ve herkes 16 alır. Kimse katmazsa herkes 10 ile kalır. Grubun en iyisi tam katkıdır, ama bireyin en iyisi (ötekiler ne ya-

parsa yapsın) hiç katmamaktır. Aynı gerilim, grubun ortak çıkarına hizmet eden ama bireye maliyetli bir katkı gerektiren her durumda ortaya çıkar: ortak bir projede çalışmak, bir parkı temiz tutmak, açık kaynak bir yazılıma katkı vermek.

Oyuna başlamadan iki soru sorulabilir. Birincisi: dört kişilik bir grupta ilk turda ortalama kaç jeton katılır, on tur sonra kaç? İkincisi: katkısı 3 jetondan az olana 2 jeton ceza kesilirse (oyundaki "Cezalı katkı" senaryosu) katkılar nasıl değişir? Kuramın cevabı ile deneylerde gözlenen davranış belirgin biçimde ayrışır. İki cevap da 12.4'te hesaplanır. Aradaki kesimler, bu farkı açıklayan modelleri kurar.

12.1.2 Sınırlı rasyonellik

Klasik karar kuramında karar verici bütün seçenekleri bilir, her birinin sonuçlarını ve olasılıklarını hesaplar ve beklenen faydası en büyük olanı seçer. Herbert Simon 1955'te bu resmin gerçek insanlara uymadığını öne sürdü. Bilgimiz eksiktir, zamanımız kısıtlıdır, aklımızda aynı anda tutabildiğimiz seçenek sayısı azdır.

Tanım 12.1

Sınırlı rasyonellik (*bounded rationality*; Simon, 1955): karar vericinin bilgisi, zamanı ve hesaplama kapasitesi sınırlı olduğundan, tüm seçenekler arasından en iyisini aramak yerine belirli bir istek düzeyini karşılayan "yeterince iyi" bir seçeneği kabul etmesidir. Bu arama biçimine **yeterince iyiyi arama** (*satisficing*) denir.

Yaz stajı için şirket arayan bir öğrenciyi düşünelim. Tam rasyonel bir karar verici bütün şirketleri listeler, her birinin maaşını, öğrenme fırsatını, ulaşım süresini puanlar ve en yüksek toplamı seçer. Gerçek öğrenci ise birkaç şirkete başvurur ve "İstanbul'da olsun, en az asgari ücret versin, yazılım işi olsun" koşullarını sağlayan ilk teklifi kabul eder. Bu davranış hesap maliyetini düşürür; bedeli, belki de daha iyi bir teklifi kaçırmaktır.

Davranışsal araştırmanın en önemli bulgusu şudur: insanların rasyonel modelden sapmaları *rastgele değil sistematiktir*. Aynı hatalar farklı kişilerde, farklı ülkelerde, tekrar tekrar gözlenir. Rastgele hatalar ortalamada birbirini götürür ve modelleri bozmaz. Sistematik hatalar ise hep aynı yöne kayar; tam da bu yüzden ölçülebilir ve bir formülle modele eklenebilir. Bu bölümdeki modellerin hepsi bu fikre dayanır.

Sapmaların nedenleri dört başlıkta toplanabilir: bilgi eksikliği, zaman baskısı, duygular ve kısa yollar. Kısa yollara **sezgisel kurallar** (*heuristics*) denir: "herkes bu ürünü alıyorsa iyidir", "geçen sefer işe yaradıysa yine yarar" gibi hızlı ama bazen yanıltan kurallar. Sık görülen dört sistematik sapma Tablo 12.1'de özetlenmiştir.

Sapma	Tanım	Örnek
Aşırı güven	Kendi tahminlerine gereğinden çok güvenmek	Yatırımcıların aşırı işlem yapması

Sapma	Tanım	Örnek
Çerçeveleme etkisi	Aynı seçenek farklı sunulunca kararın değişmesi	"%90 başarı" ile "%10 başarısızlık" farklı algılanır
Kayıptan kaçınma	Eşit büyüklükteki kaybın kazançtan daha güçlü hissedilmesi	Panik satışı
Taban oranı ihmali	Yeni bir kanıtı yorumlarken ön olasılığı yok saymak	Pozitif test sonucunda hastalığın yaygınlığını unutmak

Tablo 12.1: Dört sistematik sapma ve günlük örnekleri.

Aşırı güven (*overconfidence*), insanların kendi bilgisinin doğruluğunu abartmasıdır. Borsada sık alım satım yapan bireysel yatırımcılar, işlem maliyetleri yüzünden ortalamada daha az kazanır; ama her biri kendi tahmininin piyasadan iyi olduğuna inanır.

Taban oranı ihmali en iyi bir sayıyla anlaşılır. Bir hastalık nüfusun %1'inde görülsün. Bir test hastaların %90'ında pozitif, sağlıklıların %9'unda yanlışlıkla pozitif çıksın. 1000 kişilik bir grupta 10 hasta vardır; bunların 9'u pozitif çıkar. 990 sağlıklının yaklaşık 89'u da pozitif çıkar. Pozitif çıkan 98 kişiden yalnızca 9'u gerçekten hastadır; yani pozitif sonuçtan sonra hasta olma olasılığı yaklaşık %9'dur. İnsanların çoğu ise "test %90 doğru" diye düşünüp bu olasılığı %90'a yakın tahmin eder: hastalığın yaygınlığını, yani ön olasılığı göz ardı eder. Bayes kuralının ön olasılıkla son olasılık arasında kurduğu bağ, görüldüğü gibi sezgiye epey ters gelebilir. Aynı Bayes hesabı bilgi şelalelerinde (12.6.2) yeniden kullanılır.

Çerçeveleme ve kayıptan kaçınma ise ayrı birer modelle açıklanır: beklenti kuramı.

12.1.3 Beklenti kuramı ve değer fonksiyonu

Klasik beklenen fayda kuramında bir kişi sonuçları *toplam servetine* göre değerlendirir. Serveti 100.000 TL olan biri için 1.000 TL kazanmak, servetin 100.000'den 101.000'e çıkmasıdır. Daniel Kahneman ve Amos Tversky 1979'da bunun insanların gerçek değerlendirme biçimine uymadığını gösterdi. İnsanlar sonuçları bir **referans noktasına** (*reference point*) göre kazanç ya da kayıp olarak algılar. Referans noktası çoğu zaman mevcut durumdur, ama bir beklenti ya da hedef de olabilir.

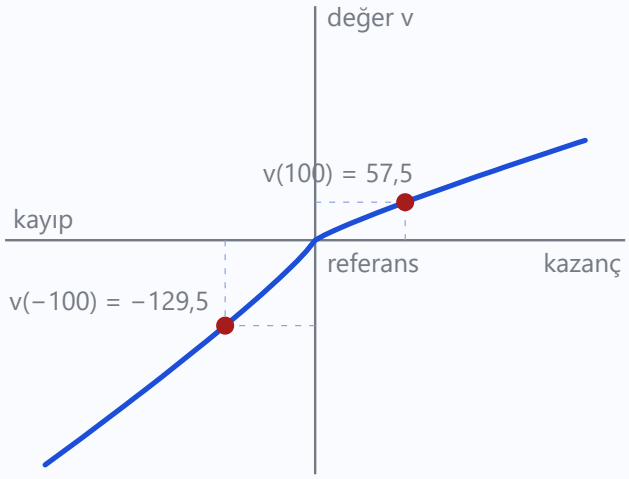
Tanım 12.2

Beklenti kuramı (*prospect theory*; Kahneman ve Tversky, 1979) risk altındaki bir seçeneği şöyle değerlendirir: her sonuç x, referans noktasına göre kazanç ($x \geq 0$) ya da kayıp ($x < 0$) olarak yazılır, bir **değer fonksiyonu** (*value function*) $v(x)$ ile değerlendirilir ve olasılıklarla ağırlıklandırılıp toplanır. Bu notta kullanılan değer fonksiyonu:

$$v(x) = x^{0,88} \ (x \geq 0), \quad v(x) = -2{,}25 \cdot (-x)^{0,88} \ (x < 0)$$

Buradaki $\lambda = 2{,}25$ sayısına **kayıptan kaçınma** (*loss aversion*) katsayısı denir. Üs 0,88 ve $\lambda = 2{,}25$, Tversky ve Kahneman'ın (1992) deneylerden tahmin ettiği değerlerdir.

Değer fonksiyonunun üç özelliği vardır ve her biri bir davranışı açıklar. Şekil 12.2 bu özellikleri tek bir eğri üzerinde gösterir.



Şekil 12.2: Beklenti kuramının değer fonksiyonu. Eğri referans noktasından geçer; kazanç tarafında içbükey, kayıp tarafında dışbükeydir ve kayıp tarafında daha diktir: $v(100) = 57{,}5$, $v(-100) = -129{,}5$.

1. Kazançta içbükeylik. $x^{0{,}88}$ fonksiyonu x'ten yavaş büyür. Üs 1'den küçük olduğu için sayı büyüdükçe artış hızı azalır. Sayılarla: $v(100) = 100^{0{,}88} \approx 57{,}5$ ve $v(200) = 200^{0{,}88} \approx 105{,}9$'dur. Tutar iki katına çıktığında değer 2 kat değil, yalnızca $2^{0{,}88} \approx 1{,}84$ kat artar. Eğri sağa doğru gittikçe yatıklaşır: ilk 100 TL'nin sevinci, ikinci 100 TL'ninkinden büyüktür. Bunun sonucu kazanç bölgesinde riskten kaçınmadır: kesin 100 TL, yarı yarıya 200 TL ya da 0 TL'lik bir piyangodan daha değerli bulunur, çünkü $57{,}5 > \tfrac{1}{2} \cdot 105{,}9 \approx 53{,}0$.

2. Kayıpta dışbükeylik. Kayıp tarafında da aynı azalan duyarlılık vardır: ilk 100 TL'lik kayıp, ikinci 100 TL'lik kayıptan daha acıdır. Ama bu kez sonuç ters yöndedir. Kesin 100 TL kayıp $v(-100) \approx -129{,}5$ değerindedir; yarı yarıya 200 TL kayıp ya da 0 TL kayıp ise $\tfrac{1}{2} \cdot v(-200) \approx \tfrac{1}{2} \cdot (-238{,}3) \approx -119{,}1$ değerindedir. Riskli seçenek daha az kötü hissedilir: kayıp bölgesinde insanlar risk sever. “Kaybı kesinleştirmektense kumar oynamak” davranışı buradan gelir; zarar eden bir hisseyi “belki geri döner” diye satmamak buna bir örnektir.

3. Kayıpta daha dik. Aynı büyüklükteki kayıp, kazançtan $\lambda = 2{,}25$ kat daha ağır gelir: $v(-100) = -2{,}25 \cdot 57{,}5 \approx -129{,}5$. Eğrinin referans noktasında bir kırılma vardır; sağdan gelirken eğim yumuşak, soldan gelirken diktir. Bu kırılma, sıfır beklenen değerli adil bahislerin bile reddedilmesini açıklar (12.1.4).

Referans noktasının etkisi tek bir örnekle görülebilir. 10.000 TL yılsonu ikramiyesi bekleyen bir çalışan 9.000 TL alırsa bunu 1.000 TL'lik bir kayıp gibi yaşar. Hiç ikramiye beklemeyen biri için aynı 9.000 TL büyük bir kazançtır. İki kişinin serveti aynı miktarda artmıştır; klasik kurama göre ikisi de aynı derecede memnun olmalıdır. Beklenti kuramına göre ise memnuniyet, sonucun referans noktasına göre nerede durduğuna bağlıdır.

Örnek 12.1

Değer fonksiyonunu birkaç noktada hesaplayalım. Hesap makinesinde $x^{0,88}$, x'in 0,88'inci kuvvetidir (Python'da `x ** 0.88`).

1. $v(10) = 10^{0,88} \approx 7,6$. On liralık kazanç yaklaşık 7,6 birim değer verir.
2. $v(100) = 100^{0,88} \approx 57,5$. Tutar 10 kat arttı, değer yaklaşık 7,6 kat arttı.
3. $v(1000) = 1000^{0,88} \approx 436,5$. Yine 10 kat tutar, yine yaklaşık 7,6 kat değer: kuvvet fonksiyonunda oran sabittir ($10^{0,88} \approx 7,6$).
4. $v(-100) = -2,25 \cdot 57,5 \approx -129,5$. Aynı büyüklükteki kayıp 2,25 kat ağırdır.

Bu fonksiyonda tutar 10 katına çıkınca değer her ölçekte aynı oranda (yaklaşık 7,6 kat) büyür. Kayıp ile kazanç arasındaki oran da her ölçekte 2,25'tir.

12.1.4 Kayıptan kaçınma: yazı-tura bahsi

Kayıptan kaçınmanın en temiz testi adil bir bahistir. Bir para atılır; yazı gelirse 100 TL kazanılır, tura gelirse 100 TL kaybedilir. Beklenen değer $½ \cdot 100 + ½ \cdot (-100) = 0$'dır. Risk nötr bir karar verici için bahse girmek ile girmemek aynıdır. Deneylerde ise insanların büyük çoğunluğu bu bahsi reddeder. Beklenti kuramı bunu şöyle açıklar.

Örnek 12.2

1. **Referansa göre sonuçlar.** Bahse girmemek referans noktasıdır (0). Bahis ½ olasılıkla +100, ½ olasılıkla −100 verir.
2. **Değerler.** $v(100) \approx 57,5$, $v(-100) \approx -129,5$.
3. **Beklenti değeri.** $½ \cdot 57,5 + ½ \cdot (-129,5) = 28,75 - 64,75 = -36,0$.
4. **Karşılaştırma.** Bahse girmemenin değeri $v(0) = 0$'dır. $-36,0 < 0$ olduğundan bahis reddedilir.

Beklenen değeri sıfır olan bahis, kaybın kazançtan ağır gelmesi yüzünden reddedilir.

Peki kazanç ne kadar olursa bahis kabul edilir? Kazanç G, kayıp yine 100 TL olsun. Bahsin kabul edilmesi için beklenti değeri sıfırdan büyük olmalıdır. Sınır, iki terimin eşitlendiği noktadır.

Örnek 12.3

1. **Eşitlik.** $½ \cdot v(G) + ½ \cdot v(-100) = 0$, yani $½ \cdot G^{0,88} = ½ \cdot 2,25 \cdot 100^{0,88}$.
2. **Sadeleştirme.** ½'ler sadeleşir; iki taraf $100^{0,88}$'e bölünür: $G^{0,88} / 100^{0,88} = (G/100)^{0,88} = 2,25$.
3. **Üssün kaldırılması.** İki tarafın $1/0,88$'inci kuvveti alınır: $G/100 = 2,25^{1/0,88} \approx 2,51$.

4. **Sonuç.** $G \approx 100 \cdot 2{,}51 \approx 251$ TL.

100 TL kaybetme riskini göze almak için karşılığında yaklaşık 251 TL kazanma şansı istenir. Bu oran tutardan bağımsızdır: 200 TL kaybı göze almak için yaklaşık 503 TL gerekir. Deneylerde de insanlar tipik olarak kaybın iki katından fazlasını ister.

Üçüncü adımdaki işlem, üslü denklemin tersini almaktır. $y^a = b$ ise $y = b^{1/a}$'dır; çünkü $(b^{1/a})^a = b$. Burada $a = 0{,}88$, $1/a \approx 1{,}136$ ve $2{,}25^{1{,}136} \approx 2{,}51$'dir. Değer fonksiyonu doğrusal olsaydı (üs 1) eşik tam $\lambda \cdot 100 = 225$ TL olurdu; içbükeylik büyük kazançları biraz daha az değerli kıldığı için eşik 251'e çıkar.

12.1.5 Olasılık ağırlığı

Beklenti kuramının ikinci bileşeni olasılıkların algılanma biçimidir. Klasik kuramda %1'lik olasılık tam olarak 0,01 ağırlık alır. Deneyler, insanların küçük olasılıkları olduğundan büyük, büyük olasılıkları olduğundan küçük hissettiğini gösterir. Bu algılanan ağırlığa **olasılık ağırlığı** (*probability weighting*) denir ve $w(p)$ ile gösterilir.

Olasılık p	0,01	0,50	0,99
Algılanan ağırlık w(p)	0,055	0,421	0,912

Tablo 12.2: Olasılık ağırlığı (Tversky ve Kahneman, 1992 tahmini).

Tablo 12.2'ye göre %1'lik bir olasılık yaklaşık %5,5 gibi hissedilir: beş kattan fazla abartılır. %50 yaklaşık %42, %99 yaklaşık %91 gibi algılanır: orta ve büyük olasılıklar küçümsenir. Kesinlik ise (p = 1) tam ağırlık alır; bu yüzden %99'dan %100'e geçiş, %50'den %51'e geçişten çok daha değerli bulunur.

Bu ağırlıklar, birbirine zıt görünen iki davranışın aynı kişide neden birlikte görüldüğünü açıklar. Piyango bileti alan biri, çok küçük bir kazanma olasılığını abartır: %0,01'lik şans ona daha büyük gelir. Aynı kişi evini sigortalatırken de küçük bir felaket olasılığını abartır ve beklenen hasardan fazla prim ödemeye razı olur. Beklenen değere dayanan model bu iki davranışı birlikte açıklayamaz: ona göre kişi ya riski sever ya riskten kaçar. Olasılık ağırlığıyla ikisi de aynı kaynaktan gelir: küçük olasılıkların abartılması.

Sık yapılan hata

"*w(p)* bir olasılıktır." Hayır, bir karar ağırlığıdır. Olasılıklar toplamı 1 eder; ağırlıklar etmez: $w(0{,}01) + w(0{,}99) = 0{,}055 + 0{,}912 = 0{,}967 \neq 1$. Ayrıca $w(0{,}5) = 0{,}421$'dir, 0,5 değildir. Ağırlıklar olasılık gibi işlem görmez; yalnızca kararın nasıl hissedildiğini ölçer.

12.1.6 Kod izleme: beklenti()

Kazanç tutarı değiştikçe bahsin değerinin nasıl değiştiğini kısa bir Python programıyla izleyelim. Program önce değer fonksiyonu `v(x)`'i, sonra yarı yarıya +G kazanç ve −K kayıp veren bir bahsin beklenti değerini tanımlar ve dört farklı kazanç tutarını dener.

```
def v(x):
    if x >= 0: return x ** 0.88
    return -2.25 * (-x) ** 0.88
def beklenti(G, K):      # ½: +G, ½: -K
    return 0.5 * v(G) + 0.5 * v(-K)
for G in [100, 200, 251, 300]:
    print(G, round(beklenti(G, 100), 1))
```

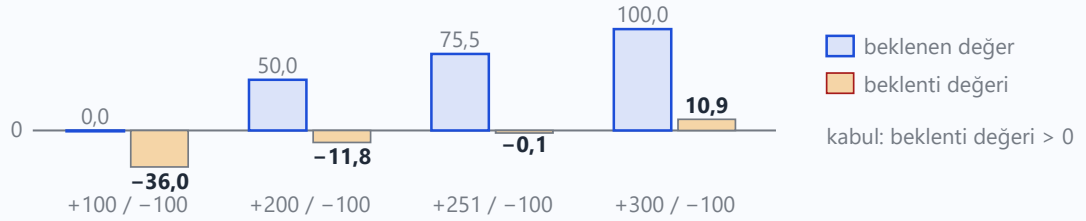
Kod satır satır şöyle çalışır:

- `v(x)`, Tanım 12.2'deki değer fonksiyonudur. `x >= 0` ise kazanç kolu `x ** 0.88` çalışır; değilse kayıp kolu `-2.25 * (-x) ** 0.88` çalışır. Kayıp kolunda `-x` pozitif bir sayıdır; kuvvet alınır, sonra −2,25 ile çarpılır.
- `beklenti(G, K)`, ½ olasılıkla +G ve ½ olasılıkla −K veren bahsin değeridir. Olasılık ağırlığı burada kullanılmaz; ½ doğrudan ağırlık alır.
- Döngü `G` için 100, 200, 251 ve 300 değerlerini dener; kayıp hep 100'dür. Her biri için değer bir ondalığa yuvarlanıp yazdırılır.

G	v(G)	v(−100)	Beklenti değeri	Beklenen değer	Karar
100	57,5	−129,5	−36,0	0	ret
200	105,9	−129,5	−11,8	+50	ret
251	129,3	−129,5	−0,1	+75,5	eşik
300	151,3	−129,5	+10,9	+100	kabul

Tablo 12.3: `beklenti(G, 100)` izlemesi. Beklenen değer *(G − 100)/2*'dir.

Program çıktısı dört satırdır: `100 -36.0`, `200 -11.8`, `251 -0.1`, `300 10.9`. Tablo 12.3 her satırın nasıl oluştuğunu gösterir. G = 100 iken bahis adildir ama beklenti değeri −36,0'dır. G = 200 iken beklenen değer +50'dir, yani risk nötr biri bahsi hemen kabul ederdi; beklenti değeri ise hâlâ −11,8'dir ve bahis reddedilir. G = 251'de değer neredeyse sıfırdır (−0,1): Örnek 12.3'teki eşik tam burasıdır. G = 300'de değer +10,9 olur ve bahis artık kabul edilir.



Şekil 12.3: Dört bahis için beklenen değer (mavi) ve beklenti değeri (turuncu). Beklenen değer her bahiste sıfır ya da pozitiftir; beklenti değeri ise ancak kazanç 251 TL'yi aştığında pozitif olur.

Şekil 12.3 iki ölçütü yan yana gösterir. Mavi çubuklar (beklenen değer) soldan sağa düzenli büyür. Turuncu çubuklar (beklenti değeri) ilk üç bahiste sıfırın altındadır. +200 / −100 bahsi, beklenen değeri +50 olduğu hâlde reddedilir. Bu farkı büyük ölçüde kayıptan kaçınma yaratır: $\lambda = 2{,}25$ iken kayıp o kadar ağır gelir ki kazanç 251 TL'yi aşmadan bahis kabul edilmez.

12.1.7 Bağış etkisi ve statüko yanlılığı

Kayıptan kaçınma yalnızca bahislerde değil, insanların sahip oldukları eşyaya biçtikleri değerde de görülür. Klasik kurama göre bir kupanın değeri, kupayı kimin elinde tuttuğuna bağlı değildir. Kahneman, Knetsch ve Thaler (1990) bu öngörüyü basit bir deneyle sınadı. Bir sınıftaki öğrencilerin yarısına rastgele birer kupa verildi. Kupa sahiplerine kupayı en az kaça satacakları, kupası olmayanlara en çok kaça alacakları soruldu.

Sonuç açıktı: kupa sahiplerinin istediği en düşük satış fiyatı, alıcıların ödemeye razı olduğu en yüksek fiyatın yaklaşık iki katıydı. Kupalar rastgele dağıtıldığı için iki grubun zevkleri ortalamada aynıdır. Klasik kurama göre kupaların yaklaşık yarısının el değiştirmesi gerekirdi; gerçekte beklenenden çok daha az takas oldu.

Tanım 12.3

Bağış etkisi (*endowment effect*): bir nesneye sahip olmanın, o nesneye biçilen değeri artırmasıdır. Sahibin isteyeceği en düşük satış fiyatı, aynı nesne için ödenmeye razı olunan en yüksek alış fiyatından sistematik olarak büyüktür.

Beklenti kuramı bu etkiyi referans noktasıyla açıklar. Kupa sahibinin referans noktası “kupam var” durumudur; kupayı vermek bir kayıptır. Kupası olmayanın referans noktası “kupam yok” durumudur; kupayı almak bir kazançtır. Kayıp λ kat ağır geldiği için satıcı, alıcının ödeyeceğinden fazlasını ister.

Örnek 12.4

Kupanın kullanım değeri 20 TL olsun. Hesabı sade tutmak için değer fonksiyonu doğrusal alınsın (üs 1), ama kayıptan kaçınma katsayısı $\lambda = 2{,}25$ kalsın.

1. **Sahip.** Kupayı vermek referans noktasından 20 TL'lik bir kayıptır ve $-2{,}25 \cdot 20 = -45$ hissettirir. Bu kaybı karşılamak için en az 45 TL ister.
2. **Alıcı.** Kupayı almak 20 TL'lik bir kazançtır: $+20$. En çok 20 TL öder.
3. **Oran.** $45 / 20 = 2{,}25 = \lambda$. Satış fiyatı ile alış fiyatı arasındaki oran tam kayıptan kaçınma katsayısıdır.
4. **Sonuç.** 20 ile 45 TL arasındaki hiçbir fiyatta iki taraf anlaşamaz: alıcı 20'den fazla vermez, satıcı 45'ten azını kabul etmez. Takas olmaz.

Deneyde ölçülen oran (yaklaşık 2) bu modelin öngördüğüne yakındır. Bağış etkisinin önemli bir sonucu, kaynakların kendiliğinden en çok değer verenlere akmamasıdır. Ekonomide Coase teoremi, işlem maliyetleri yoksa kaynakların başlangıçta kime verildiğinin verimliliği etkilemediğini söyler. Bağış etkisi bu sonucu zayıflatır: başlangıç dağılımı, takasın ne kadar olacağını belirler.

Aynı mekanizmanın günlük hayattaki karşılığı **statüko yanlılığıdır** (*status quo bias*). Varsayılan seçenekten ayrılmak da bir kayıp gibi algılanır. Emeklilik planlarında "katılmak isteyen başvursun" kuralı yerine "herkes otomatik katılır, isteyen ayrılsın" kuralı uygulanınca katılım oranı belirgin biçimde artar. İki kuralda seçenekler aynıdır; değişen yalnızca referans noktasıdır. Yazılım tasarımında varsayılan ayarların bu kadar önemli olmasının nedeni de budur: kullanıcıların çoğu varsayılanı değiştirmez.

Sık yapılan hata

"Sahibi kupayı daha çok sever." Hayır. Nesne aynıdır ve kupalar rastgele dağıtılmıştır; sahiplerin zevki ortalamada alıcılarınkinden farklı değildir. Değişen yalnızca referans noktasıdır: aynı kupa, sahip için bir kayıp, alıcı için bir kazanç olarak değerlendirilir.

12.1.8 Çerçeveleme etkisi

Referans noktası kişinin kendi durumundan gelebileceği gibi, seçeneklerin nasıl sunulduğundan da gelebilir. Tversky ve Kahneman (1981) bunu ünlü "Asya hastalığı" sorusuyla gösterdi. Katılımcılardan, 600 kişinin ölümüne yol açması beklenen bir salgına karşı iki programdan birini seçmeleri istendi. Bir gruba seçenekler kurtulanlar üzerinden, öteki gruba ölenler üzerinden anlatıldı.

Çerçeve	Kesin program	Riskli program	Çoğunluğun seçimi
Kazanç	A: 200 kişi kurtulur	B: 1/3 olasılıkla 600 kişi kurtulur, 2/3 olasılıkla kimse kurtulmaz	%72 A (kesin)
Kayıp	C: 400 kişi ölür	D: 1/3 olasılıkla kimse ölmez, 2/3 olasılıkla 600 kişi ölür	%78 D (riskli)

Tablo 12.4: Salgın sorusunun iki çerçevesi (Tversky ve Kahneman, 1981).

Tablo 12.4'teki dört program aslında iki programdır. A'da 200 kişi kurtulur, yani 400 kişi ölür: A ile C aynı sonuçtur. B'de 1/3 olasılıkla herkes kurtulur, 2/3 olasılıkla herkes ölür; D'de de öyle: B ile D aynı piyangodur. Dört programda da beklenen kurtulan sayısı 200'dür: $\frac{1}{3} \cdot 600 = 200$.

Tanım 12.4

Çerçeveleme etkisi (*framing effect*): sonuçları aynı olan seçenekler farklı biçimde (örneğin kazanç ya da kayıp olarak) sunulduğunda tercihin değişmesidir.

Beklenen fayda kuramına göre tercih değişmemelidir, çünkü iki çerçevedeki seçenekler aynıdır. Gözlenen ise çoğunluğun kesin seçenekten (%72) riskli seçeneğe (%78) dönmesidir. Beklenti kuramı bu dönüşü değer fonksiyonunun şekliyle açıklar. Kazanç çerçevesinde referans noktası "herkes ölür" durumudur ve kurtulanlar kazançtır. Kazanç bölgesinde değer fonksiyonu içbükeydir: kesin 200 kurtulan, 1/3 olasılıkla 600 kurtulandan değerli gelir. Kayıp çerçevesinde referans noktası "kimse ölmez" durumudur ve ölenler kayıptır. Kayıp bölgesinde fonksiyon dışbükeydir: kesin 400 ölü, 2/3 olasılıkla 600 ölüden daha kötü hissedilir.

Örnek 12.5

Değer fonksiyonu bu kez kişi sayısına uygulanır (olasılık ağırlığı kullanılmadan).

1. **Kazanç çerçevesi, A.** $v(200) = 200^{0{,}88} \approx 105{,}9$.
2. **Kazanç çerçevesi, B.** $\frac{1}{3} \cdot v(600) \approx \frac{1}{3} \cdot 278{,}5 \approx 92{,}8$. $105{,}9 > 92{,}8$: kesin program A seçilir.
3. **Kayıp çerçevesi, C.** $v(-400) = -2{,}25 \cdot 400^{0{,}88} \approx -438{,}5$.
4. **Kayıp çerçevesi, D.** $\frac{2}{3} \cdot v(-600) \approx \frac{2}{3} \cdot (-626{,}5) \approx -417{,}7$. $-417{,}7 > -438{,}5$: riskli program D seçilir.

Aynı fonksiyon iki çerçevede iki ayrı seçim üretir; çoğunluğun davranışı modelle uyumludur.

Çerçeveleme etkisinin uygulamadaki karşılıkları çoktur. "Bu ameliyatta hastaların %90'ı hayatta kalır" cümlesi, "%10'u ölür" cümlesinden daha fazla onay alır. Aynı fiyat farkı "nakit ödemede indirim" olarak sunulduğunda, "kartla ödemede ek ücret" olarak sunulduğundan daha az tepki çeker. Platform tasarlayan bir mühendis için çıkan ders açıktır: seçeneklerin sunuluş biçimi tarafsız değildir, kullanıcı davranışını değiştirir.

12.1.9 Beklenti kuramında sık hatalar

Beklenti kuramının kavramları birbirine yakın olduğu için sık karıştırılır. Aşağıdaki dört yanılgı, sınavlarda ve tartışmalarda en sık görülenlerdir.

Sık yapılan hata

- **"Kayıptan kaçınma, riskten kaçınmadır."** Hayır. Riskten kaçınma faydanın eğriliğinden, kayıptan kaçınma referans noktasındaki kırılmadan (λ) gelir. ±100 TL bahsi, *v* kazançta neredeyse doğrusal olsa bile λ yüzünden reddedilir. Üs 1 alınsa (eğrilik yok) bile beklenti değeri $½ \cdot 100 − ½ \cdot 225 = −62{,}5$ olur.
- **"*w(p)* bir olasılıktır."** Hayır, bir karar ağırlığıdır: $w(0{,}01) + w(0{,}99) = 0{,}967 \neq 1$ ve $w(0{,}5) = 0{,}421$.
- **"Referans noktası hep sıfırdır."** Referans; beklenti, statüko ya da hedef olabilir. 10.000 TL ikramiye bekleyene 9.000 TL bir kayıptır.
- **"Kayıp çerçevesinde herkes riski seçer."** %78 bir çoğunluk eğilimidir. Kuram ortalama davranışı açıklar, tek tek kişileri öngörmez; katılımcıların %22'si kayıp çerçevesinde de kesin programı seçti.

12.1.10 Kuantal tepki

Nash dengesi her oyuncunun her zaman en iyi tepkiyi verdiğini varsayar. En iyi tepki, öteki oyuncuların seçimi sabitken kazancı en büyük yapan stratejidir. Gerçek insanlar ise daha iyi seçeneği daha sık seçer, ama her zaman değil. Seçenekler arasındaki fark küçükse sık sık yanlış olanı seçerler; fark büyüdükçe hata azalır. Bir öğrenci, sınavda iki yakın şıktan birini seçerken yarı yarıya yanılabilir; şıklardan biri açıkça saçmaysa onu neredeyse hiç seçmez.

Bu davranışı modellemenin en yaygın yolu **logit seçimdir** (*logit choice*). Fikir şudur: her seçeneğe kazancıyla artan bir "çekim" verilir ve seçenekler çekimleriyle orantılı olasılıkla seçilir. Çekim olarak üstel fonksiyon kullanılır. $e \approx 2{,}718$ sabit bir sayıdır ve e^x, x büyüdükçe hızla büyür; her zaman pozitiftir.

Tanım 12.5

Logit seçim: oyuncunun her *a* seçeneğini, o seçeneğin beklenen kazancı *u(a)*'ya bağlı olarak

$$P(a) = e^{\lambda \cdot u(a)} \,/\, \Sigma_b \, e^{\lambda \cdot u(b)}$$

olasılığıyla seçmesidir. Payda bütün seçeneklerin çekimlerinin toplamıdır; böylece olasılıklar toplamı 1 eder. $\lambda \geq 0$ parametresi seçimin ne kadar keskin olduğunu ölçer: $\lambda = 0$ iken seçim tamamen rastgeledir, $\lambda \to \infty$ iken en iyi tepkiye dönüşür.

Buradaki λ, kayıptan kaçınmadaki $\lambda = 2{,}25$ ile ilgisizdir; iki modelde aynı harfin kullanılması yalnızca bir gelenektir. Logit seçimde λ'yı "dikkat" ya da "hassasiyet" gibi düşünmek yararlıdır. $\lambda = 0$ iken bütün üsler $e^0 = 1$ olur ve her seçenek eşit olasılık alır: oyuncu kazançlara hiç bakmıyordur. λ büyüdükçe ka-

zanç farklarının etkisi büyür; en yüksek kazançlı seçeneğin çekimi ötekileri bastırır ve sınırda bütün olasılık ona gider.

İki seçenek olduğunda formül sadeleşir. Seçenekler "iyi" ve "kötü" olsun, aradaki kazanç farkı $d = u(iyi) - u(kötü)$ olsun. Pay ve payda $e^{\lambda \cdot u(iyi)}$'ye bölünürse:

$$P(iyi) = 1 / (1 + e^{-\lambda \cdot d})$$

Bu, yalnızca kazanç farkına bağlı, S biçimli bir eğridir. Fark 0 iken olasılık ½'dir; fark büyüdükçe 1'e yaklaşır.

Örnek 12.6

İki seçenek, kazançlar 10 ve 8; fark $d = 2$. İyi seçeneğin olasılığı $P = 1 / (1 + e^{-2\lambda})$'dır.

1. $\lambda = 0$: $1/(1 + 1) = 0{,}500$.
2. $\lambda = 0{,}5$: $e^{-1} \approx 0{,}368$, $1/1{,}368 \approx 0{,}731$.
3. $\lambda = 1$: $e^{-2} \approx 0{,}135$, $1/1{,}135 \approx 0{,}881$.
4. $\lambda = 2$: $e^{-4} \approx 0{,}018$, $1/1{,}018 \approx 0{,}982$.
5. $\lambda \to \infty$: $e^{-2\lambda} \to 0$, olasılık 1'e gider: en iyi tepki.

λ	0	0,5	1	2	∞
P(kazanç 10)	0,500	0,731	0,881	0,982	1

Tablo 12.5: İki seçenek, kazançlar 10 ve 8 (fark 2): $P(\text{kazanç } 10) = 1/(1 + e^{-2\lambda})$.

Seçenek sayısı üç olduğunda da formül aynıdır. Kazançlar 4, 3 ve 1, $\lambda = 1$ olsun. Çekimler $e^4 \approx 54{,}6$, $e^3 \approx 20{,}1$ ve $e^1 \approx 2{,}7$; toplam yaklaşık 77,4'tür. Olasılıklar yaklaşık 0,705, 0,259 ve 0,035 olur. En iyi seçenek en sık seçilir; ikinci seçenek, kazancı yakın olduğu için hâlâ dörtte bir olasılık alır; açıkça kötü olan üçüncü seçenek neredeyse hiç seçilmez. Logit seçim, "hata yapılır ama büyük hata nadirdir" sezgisini böyle sayıya çevirir.

Oyunlarda bu fikir bir denge kavramına dönüşür. Her oyuncunun beklenen kazancı ötekilerin olasılıklarına bağlıdır; ötekiler de logit tepki veriyorsa olasılıklar birbirine bağlanır.

Tanım 12.6

Kuantal tepki dengesi (*quantal response equilibrium, QRE*; McKelvey ve Palfrey, 1995): her oyuncunun, öteki oyuncuların denge olasılıklarına göre hesaplanan beklenen kazançlara logit tepki verdiği olasılık profilidir. Başka bir deyişle, ötekilerin olasılıkları logit formülüne konunca her oyuncunun kendi olasılıkları geri elde edilir. $\lambda \to \infty$ iken QRE bir Nash dengesine gider.

QRE, Nash dengesini genelleştirir: Nash dengesi "sıfır hatalı" uç durumdur. Bu sayede QRE, deneylerde gözlenen ama Nash'in açıklayamadığı sistematik sapmaları tek bir parametreyle (λ) yakalayabilir. Bunun en çarpıcı örneği asimetrik Yazı-Tura oyunudur (12.1.11).

Hatalı davranışı modellemenin tek yolu bu değildir. Ayrı bir model olan **seviye-k düşünme** (*level-k reasoning*) hatayı değil akıl yürütme derinliğini sınırlar (Nagel, 1995). Seviye-0 oyuncu rastgele oynar. Seviye-1 oyuncu ötekilerin seviye-0 olduğunu varsayar ve onlara en iyi tepkiyi verir. Genel olarak seviye-k oyuncu ötekileri seviye k−1 sayar. Klasik "ortalamanın 2/3'ünü tahmin et" oyununda seviye-0 ortalama 50 tahmin eder; seviye-1, $\frac{2}{3} \cdot 50 \approx 33$; seviye-2, $\frac{2}{3} \cdot 33 \approx 22$ der. Nash dengesine (herkes 0) ancak sonsuz derinlikte ulaşılır; deneylerde tahminler çoğunlukla 33 ve 22 civarında yoğunlaşır. QRE "her adımda biraz hata", seviye-k ise "sınırlı sayıda adım" varsayar; iki model farklı deneyleri farklı başarıyla açıklar.

12.1.11 Asimetrik Yazı-Tura

Klasik Yazı-Tura Eşleştirme oyununda saf denge yoktur; tek denge karışıktır ve iki oyuncu da seçeneklerini ½ olasılıkla oynar. Goeree ve Holt (2001) bu oyunun asimetrik bir sürümünü inceledi: satır oyuncusunun bir hücredeki kazancı çok büyütüldü. Tablo 12.6'da hücrelerde önce satır, sonra sütun oyuncusunun kazancı yazılıdır.

Satır \ Sütun	Sol	Sağ
Üst	(320, 40)	(40, 80)
Alt	(40, 80)	(80, 40)

Tablo 12.6: Asimetrik Yazı-Tura. Hücrede (satır, sütun) kazancı.

Oyunda saf denge yoktur. (Üst, Sol)'da sütun Sağ'a kaçar (40 → 80); (Üst, Sağ)'da satır Alt'a kaçar (40 → 80); (Alt, Sağ)'da sütun Sol'a kaçar (40 → 80); (Alt, Sol)'da satır Üst'e kaçar (40 → 320). Döngü hiçbir hücrede durmaz. Karışık dengeyi bulmak için **kayıtsızlık ilkesi** kullanılır: karışık dengede bir oyuncu karıştırıyorsa, karıştırdığı her saf strateji ona aynı beklenen kazancı vermelidir. Aksi hâlde daha iyi olanı her zaman oynardı. Bu yüzden bir oyuncunun olasılığı, *öteki* oyuncuyu kayıtsız bırakacak biçimde belirlenir.

Örnek 12.7

$p = P(\text{Üst})$ ve $q = P(\text{Sol})$ olsun.

1. **Sütunu kayıtsız bırakan p.** Sütunun Sol'dan beklentisi $40p + 80(1 - p)$, Sağ'dan beklentisi $80p + 40(1 - p)$'dir. Eşitlenirse $80 - 40p = 40 + 40p$, yani $p = 1/2$.
2. **Satırı kayıtsız bırakan q.** Satırın Üst'ten beklentisi $320q + 40(1 - q) = 40 + 280q$, Alt'tan beklentisi $40q + 80(1 - q) = 80 - 40q$'dir. Eşitlenirse $320q = 40$, yani $q = 1/8$.

3. **Yorum.** Satır oyuncusunun Üst olasılığı 1/2'dir ve satırın kendi kazancı olan 320'ye hiç bağlı değildir: 320 yerine 120 ya da 1000 yazılsa da p = 1/2 kalır. Değişen yalnızca sütunun q olasılığıdır.

Bu sonuç ilk bakışta tuhaftır. Satır oyuncusu için (Üst, Sol) hücresinde çok büyük bir ödül vardır; sezgi, Üst'ü daha sık oynayacağını söyler. Nash dengesine göre ise oyuncunun kendi kazancı, kendi olasılığını hiç etkilemez. Deneyler sezginin tarafındadır: Goeree ve Holt'un (2001) deneylerinde satır oyuncuları büyük kazanca doğru açıkça kaydı ve Üst'ü yarıdan çok daha sık oynadı.

QRE bu "kendi kazancı etkisini" yakalar. Satırın Üst ile Alt arasındaki beklenen kazanç farkı $(40 + 280q) - (80 - 40q) = 320q - 40$, sütunun Sol ile Sağ arasındaki farkı $(80 - 40p) - (40 + 40p) = 40 - 80p$'dir. QRE'de iki logit denklemi aynı anda sağlanmalıdır:

$$p = 1 / (1 + e^{-\lambda(320q - 40)}), \quad q = 1 / (1 + e^{-\lambda(40 - 80p)})$$

Bu iki denklemin kapalı bir çözümü yoktur; değerler bilgisayarla sayısal olarak bulunur. Örneğin bir q tahmin edilir, ondan p, p'den de yeni bir q hesaplanır. Tahmin, iki q değeri tutarlı olana kadar ikiye bölme yöntemiyle düzeltilir. Sonuçlar Tablo 12.7'dedir.

λ	p = P(Üst)	q = P(Sol)
0	0,50	0,50
0,02	0,83	0,37
0,1	0,71	0,15
1	0,52	0,125
∞ (Nash)	0,50	0,125

Tablo 12.7: Asimetrik Yazı-Tura'da logit QRE.

Tablonun bir satırını elle denetleyelim. $\lambda = 0{,}02$ ve $q = 0{,}37$ iken satırın farkı $320 \cdot 0{,}37 - 40 = 78{,}4$'tür; $e^{-0{,}02 \cdot 78{,}4} = e^{-1{,}568} \approx 0{,}208$ ve $p = 1/1{,}208 \approx 0{,}83$ olur. $p = 0{,}83$ iken sütunun farkı $40 - 80 \cdot 0{,}83 = -26{,}4$'tür; $e^{0{,}528} \approx 1{,}70$ ve $q = 1/2{,}70 \approx 0{,}37$ olur. İki değer birbirini doğrular.

Tablo üç aşama gösterir. $\lambda = 0$ iken iki oyuncu da rastgele oynar. λ küçükken (0,02) satır oyuncusu 320'ye kapılıp Üst'ü sık oynar (0,83); sütun oyuncusu bunu sezip Sol'u biraz azaltır (0,37), ama tam tepki vermez. λ büyüdükçe satır oyuncusu, sütunun Sağ'a kaçtığını hesaba katar; Üst olasılığı 0,71'e, sonra 0,52'ye iner ve sınırda Nash'in 1/2'sine döner. Deneylerdeki davranış orta λ değerlerine benzer. QRE'nin gücü, Nash'in tamamen göz ardı ettiği "kendi kazancına tepki" etkisini, ek bir varsayım olmadan, yalnızca hata payıyla açıklamasıdır.

Sık yapılan hata

"Satırın büyük kazancı, karışık Nash'te satırın Üst olasılığını artırır." Hayır. Karışık Nash'te bir oyuncunun olasılığı *rakibinin* kazançlarından belirlenir, çünkü amacı rakibi kayıtsız bırakmaktır. 320 değişince değişen yalnızca q'dur (1/8). İnsanlar ise kendi kazancına da tepki verir; bu etki Nash'te yoktur, QRE'de vardır.

12.2 Sosyal tercihler

Önceki kısımdaki sapmalar kişinin kendi kazancını nasıl değerlendirdiğiyle ilgiliydi. Bu kısımdaki sapmalar ise kişinin *başkalarının* kazancını önemsemesiyle ilgilidir. Klasik modelde oyuncu yalnızca kendi parasını önemser; karşısındakinin ne aldığı, ona nasıl davranıldığı ya da sonucun adil olup olmadığı fayda fonksiyonuna girmez. Laboratuvar deneyleri bu varsayımın da sistematik biçimde bozulduğunu gösterir. Sosyal tercih modelleri, adalet, kıskançlık, suçluluk ve karşılıklılık gibi güdüleri fayda fonksiyonunun içine yazar; böylece oyuncular yine rasyoneldir, ama farklı bir şeyi maksimize ederler.

12.2.1 Diktatör ve ültimatom oyunları

Sosyal tercihlerin en yalın ölçüsü iki küçük oyundur.

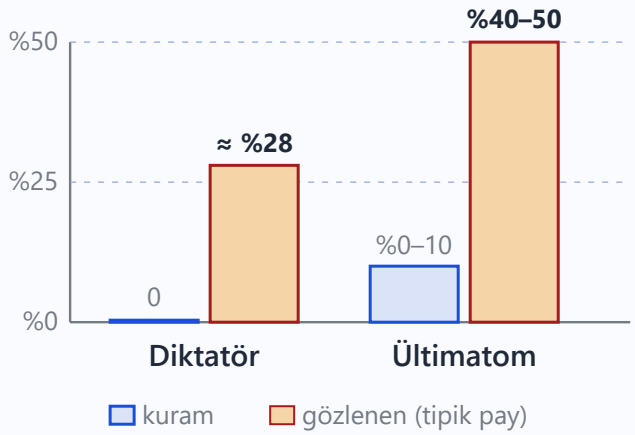
Tanım 12.7

Diktatör oyunu (*dictator game*): bir oyuncu (diktatör) 10 jetonu kendisiyle karşısındaki arasında böler; karşıdaki hiçbir şey yapamaz, bölüşüm olduğu gibi uygulanır.

Ültimatom oyunu (*ultimatum game*): teklif eden 10 jetondan s tanesini karşıya önerir. Yanıtlayan teklifi kabul ederse bölüşüm $(10 - s, s)$ olarak uygulanır; reddederse ikisi de 0 alır.

Yalnızca parayı önemseyen biri için iki oyunun çözümü kısadır. Diktatör oyununda vermenin hiçbir getirisi yoktur; diktatör 0 verir. Ültimatom oyununda yanıtlayan, pozitif her teklifi kabul etmelidir, çünkü 1 jeton bile 0'dan iyidir. Bunu bilen teklif eden en küçük pozitif teklifi (1 jeton) yapar. Bu sonuç geriye doğru çıkarımla bulunan alt oyun mükemmel dengedir.

Gözlenen davranış iki oyunda da bu öngörüden belirgin biçimde ayrılır. Yüzü aşkın diktatör oyunu çalışmasını birleştiren bir meta-analizde diktatörlerin verdiği pay ortalama yaklaşık %28'dir; azımsanmayacak bir kesim de hiç vermez (Engel, 2011). Ültimatom oyununda teklifler tipik olarak %40–50 arasındadır ve %20'nin altındaki teklifler sık sık reddedilir. Şekil 12.4 kuramla gözlemi karşılaştırır.



Şekil 12.4: Diktatör ve ültimatom oyunlarında karşıya verilen pay. Kuram (mavi) diktatörde 0, ültimatomda en küçük pozitif teklifi öngörür; gözlenen tipik paylar (turuncu) diktatörde yaklaşık %28, ültimatomda %40–50'dir.

İki oyun birlikte okunduğunda güdüler birbirinden ayrılabilir. Diktatör oyununda karşı tarafın hiçbir gücü yoktur; verilen pay ancak adalet duygusundan ya da özgecilikten (başkasının iyiliğini önemsemekten) gelebilir. Ültimatom oyununda ise teklif edenin ret korkusu vardır; diktatördeki %28'in üzerine eklenen pay büyük ölçüde stratejiktir, yani reddedilmemek içindir. Yanıtlayanın düşük teklifleri reddetmesi ise kendi parasından vazgeçmek demektir: 2 jetonu reddeden yanıtlayan 2 jeton kaybeder, ama teklif edeni 8 jetondan mahrum bırakır. Bu ret davranışı, yalnızca parayı önemseyen modelle açıklanamaz. Sosyal tercih modelleri bu güdüleri fayda fonksiyonunun içine yazarak hem verme hem ret davranışını açıklar.

12.2.2 Eşitsizlikten kaçınma: Fehr–Schmidt modeli

Ernst Fehr ve Klaus Schmidt'in (1999) modeli, insanların yalnızca kendi kazançlarını değil, kazançlar arasındaki farkı da önemsediğini varsayar. Fark iki yönde rahatsız edicidir: karşıdakinden az almak kıskançlık, fazla almak suçluluk duygusu yaratır.

Tanım 12.8

Eşitsizlikten kaçınma (*inequity aversion*; Fehr ve Schmidt, 1999). İki kişilik bir durumda oyuncu i'nin parası x_i, karşıdakinin parası x_j ise faydası

$$u_i = x_i - \alpha \cdot \max(x_j - x_i,\, 0) - \beta \cdot \max(x_i - x_j,\, 0), \quad 0 \le \beta \le \alpha,\ \beta < 1$$

biçimindedir. α geride kalmanın acısını (kıskançlık), β önde olmanın rahatsızlığını (suçluluk) ölçer.

Formüldeki *max(·, 0)* ifadeleri, iki ceza teriminden en çok birinin çalışmasını sağlar. Oyuncu gerideyse ($x_j > x_i$) yalnızca α terimi, öndeyse yalnızca β terimi sıfırdan farklıdır; eşitlikte ikisi de sıfırdır. Parametrelere konan kısıtların her birinin bir anlamı vardır:

- $\beta \le \alpha$: insanlar geride kalmaktan, önde olmaktan daha çok rahatsız olur.
- $\beta < 1$: önde olan biri, sırf eşitsizliği azaltmak için kendi parasını yakmaz. 1 jeton yakmak farkı 1 azaltır ve en çok $\beta < 1$ fayda kazandırır; yakılan 1 jetonun kaybı daha büyüktür.
- $\alpha = \beta = 0$: klasik, yalnızca parayı önemseyen oyuncu elde edilir. Model klasik modeli özel bir durum olarak içerir.

Modelin öngörüsü üç oyunda farklıdır; her oyunu ayrı ayrı çözelim.

Örnek 12.8

Ültimatom, yanıtlayan. Teklif $s < 5$ olsun; yanıtlayan s, teklif eden $10 - s$ alır ve yanıtlayan geridedir.

1. **Kabulün faydası.** Fark $(10 - s) - s = 10 - 2s$'dir. $u = s - \alpha(10 - 2s)$.
2. **Reddin faydası.** İkisi de 0 alır, fark yoktur: $u = 0$.
3. **Kabul koşulu.** $s - \alpha(10 - 2s) \ge 0 \Leftrightarrow s + 2\alpha s \ge 10\alpha \Leftrightarrow s \ge 10\alpha / (1 + 2\alpha)$.
4. **Sayısal değer.** $\alpha = 1$ ise $s \ge 10/3 \approx 3{,}33$; $\alpha = 2$ ise $s \ge 20/5 = 4$; $\alpha = 0$ ise $s \ge 0$ (her teklif kabul).

α büyüdükçe eşik yükselir, ama hiçbir zaman 5'i aşmaz: $10\alpha/(1 + 2\alpha) < 5$ her α için doğrudur. Eşit bölüşüm her yanıtlayan için kabul edilebilir.

Örnek 12.9

Diktatör. Diktatör $s \le 5$ verirse kendisi önde kalır (ya da eşittir); yalnızca β terimi çalışır.

1. **Fayda.** Diktatörün parası $10 - s$, fark $10 - 2s$: $u = (10 - s) - \beta(10 - 2s)$.
2. **Düzenleme.** $u = 10 - 10\beta - s + 2\beta s = 10 - 10\beta + s(2\beta - 1)$.
3. **Eğim.** u, s'in doğrusal bir fonksiyonudur ve eğimi $2\beta - 1$'dir. $\beta > 1/2$ ise eğim pozitiftir; diktatör verebildiği kadar, yani 5 verir (eşit böler). $\beta < 1/2$ ise eğim negatiftir; diktatör hiç vermez. $\beta = 1/2$ ise 0 ile 5 arasındaki her bölüşüm aynı faydayı verir.
4. **5'ten fazla vermek.** $s > 5$ verilirse diktatör geride kalır ve α terimi devreye girer. Hem parası azalır hem kıskançlık acısı başlar; bu yüzden 5'ten fazla vermez.

Örneğin $\beta = 0{,}6$ olan diktatör 5 verir ve $u = 5$ alır; $\beta = 0{,}3$ olan 0 verir ve $u = 10 - 3 = 7$ alır.

Bu sonuç, deneylerdeki iki kümeyi açıklar: diktatörlerin bir kısmı hiç vermez, bir kısmı eşit böler. Modelde ara değerler (ör. 2 ya da 3 vermek) ancak $\beta = 1/2$ sınır durumunda ortaya çıkar; gerçek verilerde görülen ara değerler, başka güdülerin (ör. iyi görünme isteği) de iş başında olduğunu düşündürür.

Kamu malı. Eşitsizlikten yeterince kaçınan bir oyuncu, ötekiler katkı yapıyorsa katkı yapar; çünkü katmazsa ötekilerden önde kalır ve β terimi onu rahatsız eder. Ötekiler katmıyorsa o da katmaz; çünkü tek başına katarsa geride kalır ve α terimi onu rahatsız eder. Bu davranışa **koşullu işbirlikçi** (*conditional cooperator*) denir. Laboratuvardaki kamu malı deneylerinin dinamiği (12.4) büyük ölçüde bu tipteki oyuncuların davranışından gelir.

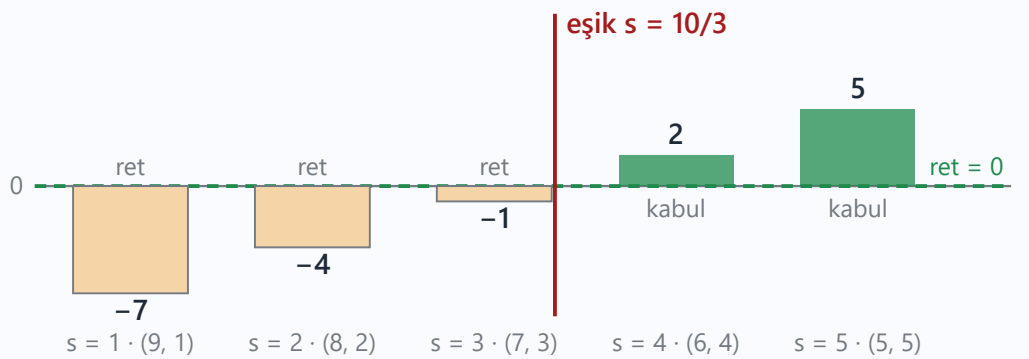
12.2.3 Ültimatomda kabul eşiği adım adım

$\alpha = 1$ olan bir yanıtlayan için kabul eşiği 10/3'tür. Teklifler tam sayı olduğunda hangilerinin kabul edileceğini görmek için teklifleri tek tek denetleyelim. Kabul faydası $u = s - 1 \cdot (10 - 2s) = 3s - 10$'dur ve reddin faydası 0'dır.

Teklif s	Bölüşüm (teklif eden, yanıtlayan)	Fark $10 - 2s$	Kabul faydası u	Karar
1	(9, 1)	8	$1 - 8 = -7$	ret
2	(8, 2)	6	$2 - 6 = -4$	ret
3	(7, 3)	4	$3 - 4 = -1$	ret (kıl payı)
4	(6, 4)	2	$4 - 2 = +2$	kabul
5	(5, 5)	0	5	kabul

Tablo 12.8: $\alpha = 1$ olan yanıtlayanın tam sayı tekliflere tepkisi.

Tablo 12.8 her satırda aynı karşılaştırmayı yapar. 1 jetonluk teklif, 8 jetonluk farkın acısını karşılamaz; fayda −7'dir ve ret (0) daha iyidir. Teklif 1 artınca yanıtlayanın parası 1 artar, fark 2 azalır; fayda toplam 3 artar. Bu yüzden fayda −7, −4, −1, +2, +5 diye üçer üçer yükselir ve 3 ile 4 arasında sıfırı keser. Şekil 12.5 aynı sayıları çubuklarla gösterir.



Şekil 12.5: $\alpha = 1$ olan yanıtlayanın kabul faydası. Çubuklar $s = 1, 2, 3$ için sıfırın altında (ret), $s = 4, 5$ için üstündedir (kabul); eşik $s = 10/3$'tür.

Eşik $10\alpha/(1+2\alpha) = 10/3 \approx 3{,}33$ olduğundan en küçük kabul edilen tam sayı teklif 4'tür. Teklif eden yanıtlayanın α değerini biliyorsa 4 önerir ve 6 alır; 3 önerseydi reddedilir ve 0 alırdı. Bu sonuç, teklif edenin kendisi yalnızca parayı önemsese bile geçerlidir. Düşük teklifin reddi bir hata değil, adalet duygusunun sonucudur; teklif eden de bunu hesaba katan rasyonel bir oyuncudur. Böylece model hem yanıtlayanın reddini hem de teklif edenin "cömert" görünen teklifini aynı çerçevede açıklar.

Sık yapılan hata

"Eşik 3,33 ise 3 jeton kabul edilir." Hayır. Kabul koşulu $s \geq 3{,}33$'tür; 3 bu koşulu sağlamaz ($u = -1$). Eşik tam sayı değilse en küçük kabul edilen teklif, eşiğin yukarı yuvarlanmışıdır. Ayrıca eşik formülü yalnızca $s < 5$ için geçerlidir; $s > 5$ tekliflerde yanıtlayan önde kalır ve β terimi çalışır.

12.2.4 Karşılıklılık ve güven oyunu

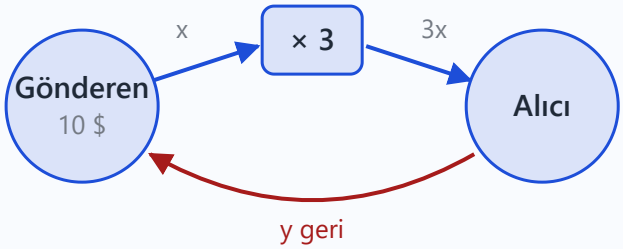
Eşitsizlikten kaçınma yalnızca sonuçların dağılımına bakar: kim ne kadar aldı? İnsanlar ise çoğu zaman sonuca değil, o sonuca yol açan *niyete* tepki verir. Bunu ölçen klasik deney güven oyunudur.

Tanım 12.9

Güven oyunu (*trust game*; Berg, Dickhaut ve McCabe, 1995): gönderen 10 doların istediği x kadarını alıcıya gönderir. Gönderilen miktar üçe katlanır; alıcı $3x$ alır ve bunun istediği y kadarını gönderene geri yollar. Gönderen $10 - x + y$, alıcı $3x - y$ kazanır.

Oyunun kurgusu dikkatle seçilmiştir. Gönderilen her dolar üçe katlandığı için, tam güven ve tam karşılıklılık durumunda iki taraf birlikte en çok kazanır: 10 dolar 30 dolar olur. Ama alıcının geri göndermek için hiçbir zorunluluğu yoktur. Deney tek seferlik ve tam gizlilik altında oynandı: katılımcılar birbirini tanımıyordu, birbirini bir daha görmeyecekti ve deneyi yapanlar bile kimin ne yaptığını bireysel olarak bilmiyordu. Böylece itibar ya da gelecekte karşılık görme gibi stratejik nedenler ortadan kaldırıldı.

Yalnızca parayı önemseyen oyuncular için çözüm geriye doğru çıkarımla bulunur. Son hamle alıcınındır; geri göndermek yalnızca kendi kazancını azaltır, bu yüzden $y = 0$'dır. Bunu öngören gönderen, gönderdiği her şeyi kaybedeceğini bilir; bu yüzden $x = 0$'dır. Kuramın öngörüsü: hiçbir şey gönderilmez, iki taraf da başlangıçtaki durumunda kalır.



Şekil 12.6: Güven oyunu. Gönderen *x* gönderir, miktar üçe katlanır, alıcı *y* geri yollar. Deneyde ortalama *x = 5,16* dolar, *3x = 15,48* dolar, *y = 4,66* dolardır.

Deneyde ise gönderenler ortalama 5,16 dolar gönderdi ve alıcılar ortalama 4,66 dolar geri yolladı (Şekil 12.6). Gönderenlerin büyük çoğunluğu bir miktar gönderdi; alıcıların bir kısmı gönderilenden fazlasını geri yolladı, bir kısmı hiçbir şey yollamadı. Ortalama bir gönderen 10 dolarla başlayıp *10 − 5,16 + 4,66 = 9,50* dolarla bitirdi; ortalama bir alıcı *15,48 − 4,66 = 10,82* dolar tuttu.

Tanım 12.10

Karşılıklılık (*reciprocity*; Rabin, 1993): bir kişinin, karşısındakinin *niyetine* yanıt vermesi; iyi niyetli bir davranışa iyilikle, kötü niyetli bir davranışa cezayla karşılık vermesidir.

Karşılıklılık ile eşitsizlikten kaçınma arasındaki fark bir düşünce deneyiyle görülebilir. Ültimatom oyununda yanıtlayana 2 jetonluk bir teklif gelsin. Birinci durumda teklifi karşıdaki kişi kendisi seçmiştir. İkinci durumda teklifi, iki tarafın da etkileyemediği bir piyango belirlemiştir. Eşitsizlikten kaçınma iki durumu ayırt edemez, çünkü iki durumda da sonuç (8, 2)'dir ve fayda aynıdır. Karşılıklılık ise ayırt eder: birinci durumda karşıdakinin bencil bir niyeti vardır ve ret bu niyeti cezalandırır; ikinci durumda cezalandırılacak bir niyet yoktur ve kabul daha olasıdır. Güven oyununda alıcının geri göndermesi de karşılıklılıkla açıklanır: gönderen güvendiğini göstermiştir ve alıcı bu iyi niyete yanıt verir.

Sık yapılan hata

“Güvenmek kârlıydı.” Ortalamada değildi. Geri dönen (4,66 dolar) gönderilenden (5,16 dolar) azdı; ortalama gönderen 0,50 dolar kaybetti. Buna rağmen çoğu kişi gönderdi. Deneyin şaşırtıcı sonucu güvenin kârlı olması değil, kârlı olmadığı hâlde bu kadar yaygın olmasıdır. Toplam açısından ise güven değer yarattı: gönderilen her dolar üçe katlandığı için iki tarafın toplam kazancı 10 dolardan ortalama *10 + 2 · 5,16 ≈ 20,3* dolara çıktı.

12.3 Kolektif kararlar ve ortak kaynaklar

Buraya kadar tek ya da iki kişinin kararları incelendi. Bundan sonra odakta, bir grubun birlikte ortaya çıkardığı sonuç vardır. Grup sonuçlarının temel sorunu, her bireyin kendi kararını verirken bunun öteki üyelere etkisini hesaba katmamasıdır. Bu kısım sorunu en klasik biçimiyle, ortak bir kaynağın aşırı kullanımıyla modeller; sonra çözümleri ve sonuçları değerlendirmek için kullanılan refah ölçütlerini ele alır.

12.3.1 Bireysel ve toplumsal çıkar

Bir grubun aldığı karara **kolektif karar** (*collective decision*) denir. Grup kararlarında sosyal etki, uyma baskısı ve ortak çıkarlar rol oynar. Topluluklar kararlarını çoğunluk oyuyla, **uzlaşıyla** (*consensus*, herkesin razı olduğu ortak karar) ya da daha karmaşık oy sistemleriyle alır. Ama bir grubun sonucu her zaman açık bir oylamayla belirlenmez. Çoğu zaman sonuç, her bireyin kendi başına verdiği küçük kararların toplamıdır: kimse “trafik sıkışıklığı olsun” diye oy vermez, ama herkesin kendi yolunu seçmesi sıkışıklığı üretir.

Bireysel çıkar kişinin kendi faydasını, **kolektif çıkar** grubun toplam faydasını maksimize eder. İkisi çoğu zaman çelişir. Üç örnek bu çelişkinin farklı biçimlerini gösterir:

- **Trafik.** Her sürücü kendine en kısa yolu seçer; toplam gecikme, merkezî bir planlamanın sağlayacağından büyük olur. Bu, bencil yönlendirme ve anarşinin bedeli konusunun temel örneğidir: iki yollu Pigou ağında bencil denge, toplam gecikmeyi optimuma göre 4/3 kat artırır.
- **İş grupları.** Herkes kendi işine odaklanır; kimsenin tek başına sahiplenmediği ortak proje aksar. Bu bir kamu malı sorunudur (12.4).
- **Ortak kaynak.** Balıkçılık, mera, yeraltı suyu, atmosfer: herkesin kullanabildiği ama kullanıldıkça azalan kaynaklar aşırı kullanılır.

Son örneğin klasik adı **ortak mallar trajedisidir** (*tragedy of the commons*). Garrett Hardin 1968'de yayımladığı makalede, herkese açık bir meraya hayvan salan çobanları örnek verdi. Her çoban için bir hayvan daha eklemek, kazancın tamamını kendisine getirir; meranın bozulmasının maliyeti ise bütün çobanlara bölünür. Bu yüzden her çoban hayvan eklemeye devam eder ve mera tükenir. Hardin'in sözlü argümanı, oyun kuramıyla sayısal bir modele çevrilebilir.

12.3.2 Ortak mallar trajedisi: sayısal model

Dört çoban ortak bir meraya hayvan salar. Çoban i'nin saldığı hayvan sayısı g_i, toplam hayvan sayısı $G = g_1 + g_2 + g_3 + g_4$ olsun. Mera kalabalıklaştıkça her hayvan daha az besi alır; hayvan başına değer $100 - G$ olsun. 100 hayvanda mera tamamen tükenir ve hayvanlar değersizleşir. Çobanın kazancı, hayvan sayısı çarpı hayvan başına değerdir:

$$u_i = g_i \cdot (100 - G)$$

Bu model Cournot modeline çok benzer: orada firmalar üretim miktarı seçer ve fiyat toplam üretimle düşer; burada çobanlar hayvan sayısı seçer ve hayvan başına değer toplam hayvanla düşer. Bu yüzden çözüm yöntemi de aynıdır: en iyi tepki türevle bulunur. Önce grubun en iyisi, sonra bireyin en iyisi hesaplanır.

Örnek 12.10

Sosyal optimum. Grubun toplam kazancı $\Sigma u_i = G \cdot (100 - G)$'dir; hangi çobanın kaç hayvanı olduğu toplamı değiştirmez.

1. **Açılım.** $G(100 - G) = 100G - G^2$.
2. **Türev.** $100G$'nin türevi 100, G^2'nin türevi $2G$'dir: türev $100 - 2G$.
3. **Sıfıra eşitleme.** $100 - 2G = 0 \Rightarrow G = 50$. İkinci türev -2 olduğundan bu bir tepedir.
4. **Değer.** Toplam kazanç $50 \cdot 50 = 2.500$. Eşit paylaşımda her çoban 12,5 hayvan salar ve $12{,}5 \cdot 50 = 625$ kazanır.

Optimum bir anlaşma olarak düşünülebilir: "herkes 12,5 hayvan salsın." Soru, bu anlaşmanın kendi kendini sürdürüp sürdürmediğidir. Bunun için bir çobanın, ötekiler anlaşmaya uyarken sapmak isteyip istemediğine bakılır.

Örnek 12.11

Tek çobanın sapması. Öteki üç çoban 12,5'er, toplam 37,5 hayvan salıyor. Çoban 1 g hayvan salarsa toplam $37{,}5 + g$ olur.

1. **Kazanç.** $u_1 = g \cdot (100 - 37{,}5 - g) = g \cdot (62{,}5 - g) = 62{,}5g - g^2$.
2. **Türev.** $62{,}5 - 2g = 0 \Rightarrow g = 31{,}25$.
3. **Sapanın kazancı.** Toplam hayvan $37{,}5 + 31{,}25 = 68{,}75$, hayvan başı değer $31{,}25$; kazanç $31{,}25 \cdot 31{,}25 \approx 976{,}6$. Anlaşmadaki 625'ten çok daha fazladır.
4. **Ötekilerin kazancı.** Her biri $12{,}5 \cdot 31{,}25 \approx 390{,}6$ alır; 625'ten 390,6'ya düşer.

Anlaşma kendi kendini sürdürmez: her çoban sapmak ister. Sapanın kazancı, ötekilerin kaybından gelir.

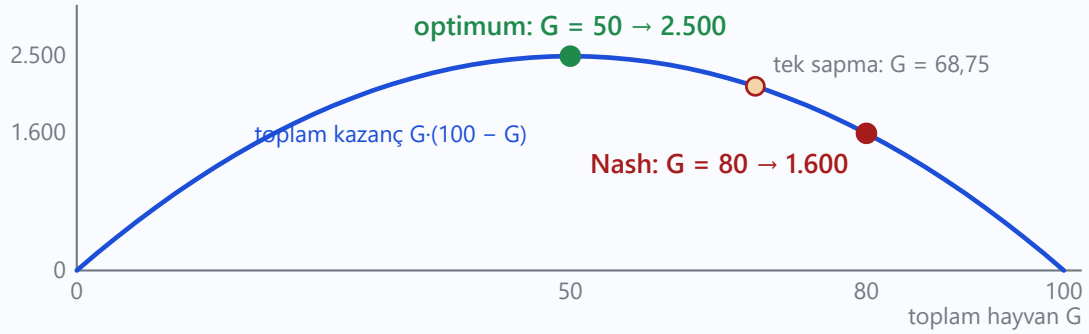
Herkes aynı mantıkla hareket ederse sonuç Nash dengesidir. Dengeyi bulmak için çoban i'nin genel en iyi tepkisi yazılır. Ötekilerin toplam hayvanı G_{-i} olsun:

$$u_i = g_i(100 - g_i - G_{-i}) \;\Rightarrow\; \text{türev } 100 - 2g_i - G_{-i} = 0 \;\Rightarrow\; g_i = (100 - G_{-i}) / 2$$

Örnek 12.12

Simetrik Nash dengesi.

1. **Simetri.** Dört çoban aynı durumdadır; dengede hepsi aynı g'yi seçer. O zaman $G_{-i} = 3g$'dir.
2. **Denklem.** $g = (100 - 3g)/2 \Rightarrow 2g = 100 - 3g \Rightarrow 5g = 100 \Rightarrow g^* = 20$.
3. **Toplam ve kazanç.** $G = 80$, hayvan başı değer 20; kişi başı kazanç $20 \cdot 20 = 400$, toplam 1.600.
4. **Kontrol.** Öteki üçü 60 hayvan salarken en iyi tepki $(100 - 60)/2 = 20$'dir. Kimse sapmak istemez ✓.



Şekil 12.7: Toplam kazanç $G(100 - G)$ eğrisi. Optimum $G = 50$'de 2.500'dür; tek çoban saptığında $G = 68{,}75$'e, herkes en iyi tepki verdiğinde (Nash) $G = 80$'e ve 1.600'e inilir.

Şekil 12.7 üç durumu aynı eğri üzerinde gösterir. Eğrinin tepesi optimumdur. Tek bir çobanın sapması toplamı tepenin sağına iter; herkes sapınca toplam hayvan 80'e çıkar ve grup kazancı 1.600'e düşer. Trajedi şudur: dengede toplam kazanç optimumun yalnızca %64'üdür ve her çoban, anlaşmaya uyulsaydı alacağı 625 yerine 400 alır. Herkes kendi açısından doğru hesap yapmıştır; sonuç herkes için kötüdür. Bu yapı, Mahkûmun İkilemi'nin çok oyunculu ve sürekli stratejili bir biçimidir.

Dengenin neden optimumdan uzak olduğu **dışsallık** (*externality*) kavramıyla açıklanır. Dışsallık, bir kişinin kararının başkalarına verdiği ve kendi hesabına girmeyen etkidir. Çoban bir hayvan daha eklediğinde hayvan başı değer 1 azalır ve bu azalma meradaki bütün hayvanlara yansır. Çoban bu kaybın yalnızca kendi hayvanlarına düşen kısmını hisseder; simetrik durumda dört çobandan biri olduğu için bu pay kabaca 1/4'tür. Geri kalan 3/4 öteki çobanların sırtına biner ve çobanın hesabına girmez. Her çoban maliyetin bir kısmını görmezden geldiği için herkes fazla hayvan salar.

Dengeye nasıl ulaşılır? Çobanlar sırayla en iyi tepki verirse süreç bu dengeye yakınsar. Başlangıçta mera boş olsun. Çoban 1, kimse yokken $(100 - 0)/2 = 50$ salar; çoban 2, 50'ye karşı 25; çoban 3, 75'e karşı 12,5; çoban 4, 87,5'e karşı 6,25 salar. Sonraki turlarda çoban 1 ötekilerin 43,75'ine karşı hayvanlarını azaltır, öteki çobanlar artırır. Turlar ilerledikçe dört sayı da 20'ye yaklaşır. Denge bu anlamda kararlıdır: yalnızca bir varsayım değil, makul bir öğrenme sürecinin vardığı noktadır.

12.3.3 Çoban sayısı arttıkça ve çözümler

Çoban sayısı n değiştiğinde aynı hesap genel olarak yapılabilir. Simetrik dengede $g = (100 - (n-1)g)/2$, yani $(n+1)g = 100$ ve

$$g^* = 100/(n+1), \quad G^* = 100n/(n+1)$$

n	g* (kişi başı)	G*	Toplam kazanç	Optimuma oran
1	50	50	2.500	%100
2	33,33	66,7	2.222	%88,9
4	20	80	1.600	%64
10	9,09	90,9	826	%33,1
100	0,99	99	98	%3,9

Tablo 12.9: Çoban sayısına göre denge. Optimum her durumda $G = 50$, toplam 2.500'dür.

Tablo 12.9 trajedinin çoban sayısıyla nasıl derinleştiğini gösterir. Tek çoban ($n = 1$) meranın tek sahibidir; bütün maliyeti kendisi hisseder ve optimumu seçer. İki çobanda kayıp küçüktür (toplam, optimumun %88,9'u). On çobanda grup, olası kazancın üçte birini bile alamaz. Yüz çobanda mera neredeyse tamamen tükenir ($G^* = 99$) ve toplam kazanç sıfıra yaklaşır: $n \to \infty$ iken $G^* \to 100$ ve toplam $\to 0$. Bunun nedeni dışsallığın büyümesidir: n çobanlı merada her çoban kendi verdiği zararın yalnızca kabaca $1/n$'ini hisseder. Büyük ve anonim gruplarda ortak kaynakların daha kolay tükenmesinin nedeni budur.

Trajediden çıkış için üç tür çözüm önerilmiştir.

1. Pigou vergisi. Ekonomist Arthur Pigou'nun adını taşıyan bu çözümde, dışsallık yaratan her birime bir vergi konur. Böylece bireyin hesabına, görmezden geldiği maliyet yeniden eklenir. Çoban hayvan başına t vergi öderse kazancı $u_i = g_i(100 - G - t)$ olur.

Örnek 12.13

1. **En iyi tepki.** Türev $100 - t - 2g_i - G_{-i} = 0 \Rightarrow g_i = (100 - t - G_{-i})/2$. Formül vergisiz durumla aynıdır; yalnızca 100 yerine $100 - t$ yazılmıştır.
2. **Simetrik denge (n = 4).** $g = (100 - t - 3g)/2 \Rightarrow g = (100 - t)/5$.
3. **Optimum hedefi.** $G = 4g = 50$ isteniyor, yani $g = 12{,}5$: $(100 - t)/5 = 12{,}5 \Rightarrow 100 - t = 62{,}5 \Rightarrow t = 37{,}5$.
4. **Kontrol.** Vergiyle her çoban 12,5 hayvan salar; çoban net $12{,}5 \cdot (100 - 50 - 37{,}5) = 156{,}25$ kazanır ve toplam vergi geliri $4 \cdot 12{,}5 \cdot 37{,}5 = 1.875$'tir. Çobanların net kazancı ile vergi gelirinin toplamı $4 \cdot 156{,}25 + 1.875 = 2.500$, yani optimumdur.

Genel n için aynı hesap $g = (100 - t)/(n + 1)$ ve optimum için $t = 100 - 50(n + 1)/n$ verir; örneğin $n = 9$ iken $t \approx 44{,}4$.

Kontroldeki sayılar Pigou vergisinin önemli bir özelliğini gösterir: vergi toplam değeri yok etmez, yalnızca yer değiştirir. Toplanan vergi hayvan sayısından bağımsız, eşit paylar hâlinde çobanlara geri dağıtılırsa her çoban 625 alır; herkes vergisiz dengeye göre daha iyi durumdadır. Verginin büyüklüğü de rastgele değildir: optimumda bir hayvan daha eklemek, öteki çobanların toplam 37,5 hayvanının her birinin değerini 1 düşürür. Vergi, optimumdaki bu dışsallığa (37,5) eşittir.

2. Kota ya da izin. Toplam hayvan sayısı doğrudan 50 ile sınırlanır ve haklar çobanlar arasında paylaştırılır. Haklar alınıp satılabilirse, hakkı en verimli kullanacak çobanlar onları satın alır. Balıkçılıktaki av kotaları ve karbon salımı izinleri bu çözümün örnekleridir. Kota, miktarı doğrudan belirler; vergi ise fiyatı belirler ve miktarı bireylerin kararına bırakır.

3. Yerel kurumlar. Elinor Ostrom'un çalışmaları, yerel toplulukların izleme, kademeli yaptırım ve ortak kurallarla kaynaklarını yüzyıllarca sürdürebildiğini gösterdi. Ostrom bu çalışmalarıyla 2009'da Nobel ekonomi ödülünü aldı. Bu üçüncü yolun ilkeleri 12.3.4'te verilir.

12.3.4 Ostrom'un tasarım ilkeleri

Hardin'e göre ortak mallar trajedisinden yalnızca iki çıkış vardı: ya devlet kaynağı merkezden denetler ya da kaynak özelleştirilip bölünür. Ostrom (1990) bu ikiliği sorguladı. İsviçre'nin Alp meralarından Japon köy ormanlarına, İspanya'nın sulama birliklerinden Filipinler'in sulama sistemlerine kadar, yüzyıllardır süren pek çok yerel yönetim düzenini inceledi. Bu düzenlerde kaynak ne devletindi ne de bölünmüştü; kullanıcılar ortak kurallarla kendi kendilerini yönetiyordu. Ostrom, başarılı örneklerin paylaştığı ilkeleri belirledi.

İlke	Anlamı
1. Açık sınırlar	Kimin, hangi kaynaktan yararlanacağı bellidir.
2. Yerel koşullara uygun kurallar	Kullanım ve katkı kuralları kaynağa göre biçilir.
3. Kurallara katılım	Kurallardan etkilenenler kuralları değiştirebilir.
4. İzleme	Kullanıcılara hesap veren gözetleyiciler vardır.
5. Kademeli yaptırım	İlk ihlale hafif, tekrarına ağır ceza verilir.
6. Ucuz çatışma çözümü	Anlaşmazlık hızlı ve yerelde çözülür.
7. Örgütlenme hakkı	Dış otoriteler yerel kuralları tanır.
8. İç içe yönetim	Büyük kaynaklarda katmanlı kurumlar vardır.

Tablo 12.10: Ostrom'un sekiz tasarım ilkesi (Ostrom, 1990).

İlkeler oyun kuramının diliyle okunabilir. Açık sınırlar oyuncu kümesini sabitler; yabancıların gelip kaynağı tüketmesini önler ve “kim kime karşı oynuyor” sorusunu netleştirir. Kurallara katılım, kuralların yerel bilgiyle yapılmasını ve meşru görülmesini sağlar. İzleme, tekrarlı oyunda sapmanın fark

edilmesini mümkün kılar; sapma görülmezse hiçbir ceza tehdidi işe yaramaz. Kademeli yaptırım iki işe yarar: küçük bir hata ya da yanlış anlama, topluluğu dağıtacak bir cezaya yol açmaz; tekrar eden ihlal ise giderek pahalılaşır. Ucuz çatışma çözümü, kuralların belirsiz kaldığı durumlarda anlaşmazlığın büyümeden çözülmesini sağlar. Bu mekanizmaların hepsi, tekrarlı oyunlarda işbirliğini sürdüren koşulları (gözlem, ceza, uzun ilişki) somut kurumlara çevirir.

Ostrom'un incelediği örneklerden biri Türkiye'dendir: Alanya'nın kıyı balıkçıları. 1970'lerde artan rekabet ve avlak yerleri için yaşanan kavgalar sonrasında balıkçılar kendi düzenlerini kurdu. Her sezon başında, avlanma hakkı olan balıkçılar listelenir ve adlandırılmış avlak yerleri kurayla dağıtılır. Sezon boyunca tekneler, balığın göç yönüne göre her gün bir sonraki avlağa geçerek sırayla yer değiştirir. Böylece herkes iyi ve kötü avlaklardan eşit yararlanır, kavga eden tekne sayısı azalır ve kuralın izlenmesi kolaydır: yanlış yerde avlanan tekneyi, o avlağın o günkü hak sahibi hemen fark eder. Kural devletten gelmemiştir, kaynak da bölünüp satılmamıştır; ama kurumsal düzen işler.

Sık yapılan hata

"Ortak malın çözümü ya devlet ya piyasadır." Ostrom'un yerel kurumları üçüncü bir yoldur. Kaynak kamulaştırılmadan ve özelleştirilmeden, kullanıcıların kendi koyduğu ve kendi izlediği kurallarla sürdürülebilir. Bu, her ortak kaynağın kendiliğinden yönetileceği anlamına da gelmez: ilkeler, başarılı örneklerde bulunan ve başarısızlarda eksik olan koşullardır.

12.3.5 Sosyal refah ölçütleri ve cehalet peçesi

Ortak mera örneğinde "optimum" sözcüğü toplam kazancı en büyük yapan durum için kullanıldı. Ama bir grubun sonuçlarını değerlendirmenin tek yolu toplam değildir. Bir politika toplamı artırırken birkaç kişiyi çok kötü durumda bırakabilir. Bu yüzden **sosyal refah** (*social welfare*) için birden çok ölçüt kullanılır.

Tanım 12.11

Bireylerin faydaları $u_1, \ldots, u_n$ olsun.

- **Faydacı ölçüt** (*utilitarian criterion*): $SW = \Sigma\, u_i$. Toplamı en büyük olan seçenek en iyidir.
- **Eşitlikçi ölçüt** (*egalitarian criterion*): $\min u_i$. En kötü durumdakinin faydası en büyük olan seçenek en iyidir.
- **Nash sosyal refahı** (*Nash social welfare*): $\Pi\, u_i$. Faydaların çarpımı en büyük olan seçenek en iyidir.

Üç ölçüt aynı seçenekleri farklı sıralayabilir. Ortak meradaki optimum ($G = 50$, toplam 2.500) faydacı ölçütün en iyisidir. Nash sosyal refahı arada bir yerde durur: çarpım, toplamı önemser ama bir faydanın sıfıra yaklaşmasını ağır cezalandırır, çünkü tek bir sıfır bütün çarpımı sıfırlar.

Hangi ölçütün seçileceği felsefi bir sorudur, ama karar kuramı bu soruyu netleştiren bir araç verir. Filozof John Rawls'un ünlü düşünce deneyinde, bir toplumun kuralları, kimin hangi konumda olacağı bilinmeden seçilir. Karar verici bir **cehalet peçesi** (*veil of ignorance*) ardındadır: zengin mi yoksul mu, sağlıklı mı hasta mı doğacağını bilmez. Bu durumda toplumsal seçim, belirsizlik altında bir bireysel karar problemine dönüşür. Her politika bir "eylem", her konum bir "doğa durumu"dur ve karar kuramının belirsizlik ölçütleri uygulanabilir.

- Her konuma eşit olasılık verip beklenen faydayı maksimize eden karar verici **Laplace kuralını** uygular. Beklenen fayda $(1/n)\Sigma u_i$ olduğundan bu kural faydacı ölçütle aynı politikayı seçer (Harsanyi, 1955).
- En kötü durumu en iyi yapmaya çalışan karar verici **maksimin** kuralını uygular ve eşitlikçi ölçüte varır (Rawls, 1971).

Örnek 12.14

Üç konumlu bir toplumda iki politika vardır: $P = (10, 6, 2)$ ve $R = (5, 5, 5)$. Sayılar her konumdaki kişinin faydasıdır.

1. **Laplace (Harsanyi).** P'nin ortalaması $18/3 = 6$, R'nin ortalaması 5. $6 > 5 \Rightarrow P$.
2. **Maksimin (Rawls).** P'nin en kötü konumu 2, R'ninki 5. $2 < 5 \Rightarrow R$.
3. **Nash sosyal refahı.** P'nin çarpımı $10 \cdot 6 \cdot 2 = 120$, R'ninki $5^3 = 125$. $125 > 120 \Rightarrow R$.

Aynı tablo, hangi belirsizlik ölçütü seçildiğine göre farklı politika önerir. Faydacı ölçüt toplamın büyüklüğüne, eşitlikçi ölçüt en kötü duruma bakar; Nash ölçütü bu örnekte eşitliğe yakın politikayı seçer.

Peçe argümanının gücü, adalet üzerine bir tartışmayı, herkesin üzerinde anlaşabileceği bir karar probleminin yapısına indirmesidir. Tartışma ortadan kalkmaz; yalnızca yer değiştirir: "Hangi politika adildir?" sorusu "Belirsizlik altında hangi karar kuralı makuldür?" sorusuna dönüşür. Riskten çok kaçınan biri maksimine, risk nötr biri Laplace kuralına yakın durur. Bu ölçütler, kaynakların ve görevlerin adil paylaşımını inceleyen algoritmik adalet çalışmalarında da temel araçlardır.

12.4 Kamu malı ve ceza

Ortak mera bir "alma" sorunudur: herkes ortak kaynaktan fazla alır. Kamu malı ise bir "verme" sorunudur: herkes ortak projeye az verir. İki sorunun yapısı aynıdır: bireyin kararı, kendi hesabına girmeyen bir etki yaratır. Bu kısım, bölümün başındaki Kamu Malı Oyunu'nu kuramsal olarak çözer, laboratuvar deneylerinin sonuçlarını sayısal bir modelle açıklar ve cezanın işbirliğini nasıl koruduğunu, nerede yetersiz kaldığını inceler.

12.4.1 Laboratuvarda kamu malı

Kamu malı (*public good*), bir kişinin yararlanmasının başkasının yararlanmasını azaltmadığı ve kimsenin yararlanmaktan dışlanamadığı maldır: sokak aydınlatması, temiz hava, ulusal savunma, açık kaynak yazılım. Dışlanamama, katkı yapmayanın da yararlanabilmesi demektir. Katkı yapmadan başkalarının katkısından yararlanan kişiye **beleşçi** (*free rider*), bu davranışa **beleşçilik** (*free riding*) denir.

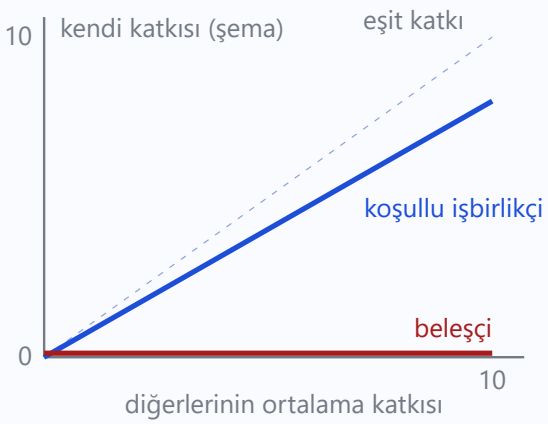
Kamu Malı Oyunu (12.1.1) bu yapının laboratuvar modelidir. Katılan her jeton katana 0,4, gruba toplam 1,6 döndürür. Kuramsal çözüm tek satırdır.

> **Teorem 12.1 (Katmamak baskındır)**
>
> Kazancı $u_i = 10 - k_i + 0{,}4 \cdot \Sigma k$ olan dört kişilik Kamu Malı Oyunu'nda, yalnızca kendi parasını önemseyen her oyuncu için $k_i = 0$ kesin baskın stratejidir. Tek Nash dengesinde herkes 0 katar ve 10 alır; herkesin 10 kattığı profilde ise herkes 16 alır.

İspat. Oyuncu i'nin katkısı k_i'yi 1 artırması, öteki katkılar ne olursa olsun, kazancını $-1 + 0{,}4 = -0{,}6$ değiştirir: 1 jeton cebinden çıkar ve havuzdan 0,4 geri döner. Kazanç k_i'nin azalan doğrusal bir fonksiyonu olduğundan en büyük değer $k_i = 0$'da alınır ve bu sonuç ötekilerin katkısına bağlı değildir. Her oyuncu baskın stratejisini oynarsa profil (0, 0, 0, 0) olur ve herkes 10 alır. Herkes 10 katarsa havuz 40, çarpılınca 64 olur; herkes 16 alır. □

Kuram açık bir öngörü verir: 0 katkı. Laboratuvar deneyleri ise farklı bir tablo gösterir (Ledyard, 1995):

- **İlk turlar:** bütçenin yaklaşık %40–60'ı katılır. Denekler ne tam işbirlikçi ne tam beleşçidir.
- **Tekrarla düşüş:** oyun aynı grupla on tur kadar tekrarlandığında katkılar tur tur azalır ve son turlarda beleşçilik düzeyine yaklaşır.
- **Kimler?** Fischbacher, Gächter ve Fehr (2001), deneklere öteki üyelerin ortalama katkısının her olası değeri için kendilerinin ne kadar katacağını sordu ve böylece her deneğin katkı şemasını çıkardı. Deneklerin yaklaşık yarısı **koşullu işbirlikçidir**: diğerleri ne kadar katarsa o kadar katmak ister. Yaklaşık üçte biri beleşçidir: diğerleri ne yaparsa yapsın 0 katar.



Şekil 12.8: Katkı şemaları. Beleşçi (kırmızı) her durumda 0 katar. Koşullu işbirlikçi (mavi), diğerlerinin ortalama katkısı arttıkça katkısını artırır, ama eşit katkı çizgisinin (kesikli) biraz altında kalır.

Şekil 12.8 iki tipin şemalarını gösterir. Düşüşün mekanizması bu şekilden okunur. Koşullu işbirlikçiler ortalamanın biraz altında katar; mavi çizgi kesikli çizginin altındadır. Beleşçiler ortalamayı aşağı çeker. Bir sonraki turda koşullu işbirlikçiler bu daha düşük ortalamaya bakar ve yine biraz altında katar. Katkı her tur biraz daha erir.

12.4.2 Kod izleme: kosullu_katki()

Model şöyledir: dört oyuncudan O1 hep 0 verir (beleşçi); O2, O3 ve O4 koşullu işbirlikçidir ve bu tur, diğerlerinin önceki turdaki ortalaması kadar verir. Başlangıçta (t = 0) üç işbirlikçi 10'ar jeton katmıştır. Fonksiyon her turun katkı listesini döndürür.

```
def kosullu_katki(c, tur):
  gecmis = [c]
  for t in range(tur):
    yeni = [0]                  # O1 hep beleşçi
    for i in range(1, len(c)):
      digerleri = sum(c) - c[i]
      yeni.append(digerleri / (len(c) - 1))
    c = yeni
    gecmis.append(c)
  return gecmis
```

Kod satır satır şöyle çalışır:

- `gecmis`, turların katkı listelerini saklar; ilk eleman başlangıç katkılarıdır.
- Her turda `yeni` listesi `[0]` ile başlar: O1 (indis 0) her zaman beleşçidir.

- İç döngü O2–O4 (indis 1, 2, 3) üzerinde gezinir. `digerleri = sum(c) - c[i]`, oyuncu `i` dışındaki üç oyuncunun önceki turdaki toplam katkısıdır. Bu toplam `len(c) - 1 = 3`'e bölünerek ortalama bulunur ve oyuncunun yeni katkısı olur.
- Tur sonunda `c` yeni katkılarla değiştirilir ve geçmişe eklenir. Bu sıra önemlidir: bir turdaki bütün oyuncular *önceki* turun katkılarına bakar; aynı turda birbirlerinin yeni kararlarını görmezler.

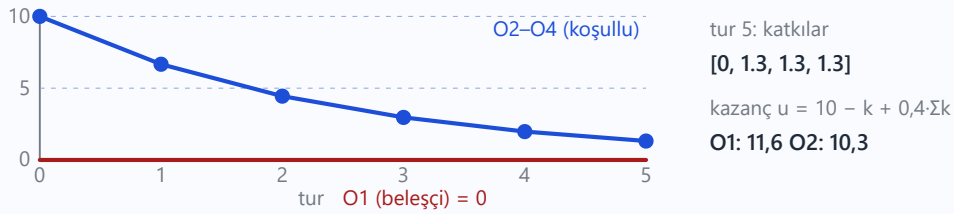
Tur	O2 için diğerlerinin toplamı (önceki tur)	Katkılar	O1 kazancı	O2 kazancı
0	—	[0, 10, 10, 10]	22,0	12,0
1	0 + 10 + 10 = 20 ⇒ 20/3 ≈ 6,7	[0, 6,7, 6,7, 6,7]	18,0	11,3
2	13,3 ⇒ 4,4	[0, 4,4, 4,4, 4,4]	15,3	10,9
3	8,9 ⇒ 3,0	[0, 3,0, 3,0, 3,0]	13,6	10,6
4	5,9 ⇒ 2,0	[0, 2,0, 2,0, 2,0]	12,4	10,4
5	4,0 ⇒ 1,3	[0, 1,3, 1,3, 1,3]	11,6	10,3

Tablo 12.11: `kosullu_katki([0, 10, 10, 10], 5)` izlemesi. Kazanç $u = 10 - k + 0{,}4 \cdot \Sigma k$.

Tablo 12.11 her turu ayrıntılı gösterir. t = 0'dan tur 1'e geçerken O2, diğerlerinin toplamına bakar: O1'in 0'ı, O3'ün 10'u ve O4'ün 10'u, toplam 20. Ortalama *20/3 ≈ 6,7*'dir. O3 ve O4 de aynı hesabı yapar. Böylece tur 1'de katkılar [0, 6,7, 6,7, 6,7] olur. Tur 2'de O2 için diğerlerinin toplamı *0 + 6,7 + 6,7 ≈ 13,3*, ortalama 4,4'tür; ve bu böyle sürer.

Örüntü genel olarak yazılabilir. İşbirlikçiler bir turda *x* katıyorsa, sonraki turda her biri için diğerlerinin toplamı *0 + x + x = 2x*, ortalaması *2x/3*'tür. Her tur katkı bir önceki değerin üçte ikisine iner (üçte bir oranında azalır):

$$x_t = 10 \cdot (2/3)^t: \quad 10 \to 6{,}7 \to 4{,}4 \to 3{,}0 \to 2{,}0 \to 1{,}3$$



Şekil 12.9: Koşullu işbirlikçilerin katkısı (mavi) beş turda 10'dan 1,3'e iner; beleşçi O1 (kırmızı) hep 0 verir. Tur 5'te kazançlar O1 için 11,6, O2 için 10,3'tür.

Şekil 12.9 düşüşü gösterir. Beş turda katkı 10'dan 1,3'e iner: tek bir beleşçi, çoğunluğun işbirliğini çözer. Kazanç sütunları da öğreticidir. Her turda beleşçi O1, işbirlikçilerden daha çok kazanır: t = 0'da 22'ye karşı 12. Beleşçilik her turda kârlıdır; işbirlikçilerin "cezası" ise katkılarını azaltmaktır. Ama bu ceza herkese, kendilerine de vurur: tur 5'te işbirlikçiler 12 yerine 10,3 alır. Bu basit model, laboratuvardaki düşüşün nitel özelliklerini yakalar: yüksek bir başlangıç, tur tur erime ve son turlarda beleşçilik düzeyine yaklaşma.

Sık yapılan hata

"Katkılar düştüğüne göre denekler beleşçiliği öğrendi." Bu modelde öğrenen kimse yoktur. İşbirlikçiler kuralını hiç değiştirmez: her tur diğerlerinin ortalaması kadar katar. Düşüş, koşullu işbirliği kuralı ile bir beleşçinin birleşiminden kendiliğinden doğar. Deneylerde de düşüşün önemli bir kısmının bu mekanizmadan geldiği düşünülür.

12.4.3 Cezalı ve cezasız: Fehr ve Gächter

Koşullu işbirlikçilerin elindeki tek silah katkılarını azaltmaksa, beleşçiyi cezalandırırken kendilerini de cezalandırırlar. Fehr ve Gächter (2000) deneklere ikinci bir silah verdi: doğrudan ceza.

Deneyin tasarımı şöyledir. 4 kişilik gruplar kamu malı oyunu oynar; katılan her jeton katana 0,4 jeton döndürür. Her turun sonunda katkılar (isimsiz olarak) açıklanır. Ardından her denek, kendi parasından ödeyerek diğer üyelere ceza puanı verebilir; ceza puanı alanın kazancı düşer. Her grup cezalı ve cezasız koşulda 10'ar tur oynar. Deney iki ayrı biçimde yapıldı: aynı kişilerin her tur birlikte oynadığı sabit gruplar ve her tur yeniden karıştırılan gruplar.

- **Cezasız koşul:** katkılar ilk turlarda yüksektir, sonra sıfıra doğru düşer. 12.4.2'deki dinamik burada da görülür.
- **Cezalı koşul:** katkılar yükselir; sabit gruplarda tam işbirliğine yaklaşır. Grup ortalamasından en çok aşağıda katan en çok cezayı alır.

En şaşırtıcı bulgu, her tur karışan gruplarla ilgilidir. Bu gruplarda aynı kişiyle bir daha karşılaşma olasılığı yok denecek kadar azdır. Ceza vermek, ceza verene hiçbir şey kazandırmaz: cezalandırılan kişinin gelecekteki davranışı, ceza verenin gelecekteki gruplarını etkilemez. Yine de denekler ceza verdi. Kendine hiçbir maddi getirisi olmadığı hâlde, normu çiğneyeni maliyetine katlanarak cezalandırma davranışına **özgeci ceza** (*altruistic punishment*) denir (Fehr ve Gächter, 2002).

Kuramın öngörüsü ise açıktır. Yalnızca parayı önemseyen biri için ceza vermek saf maliyettir; bu yüzden kimse ceza vermez. Bunu bilen herkes, ceza imkânı yokmuş gibi davranır ve yine 0 katar. Deneyde hem ceza verildi hem de katkılar yüksek kaldı: kuramın iki öngörüsü de tutmadı. Eşitsizlikten kaçınma ve karşılıklılık bu davranışı açıklar: beleşçi öndedir ve kötü niyetlidir; onu cezalandırmak eşitsizliği azaltır ve niyete karşılık verir.

12.4.4 Kod izleme: ceza()

Ceza beleşçiliği kârsız kılabilir mi? Aşağıdaki fonksiyon, katkılar ve ceza puanları verildiğinde oyuncuların kazançlarını hesaplar. Kural şudur: ceza veren her puan için 1 jeton öder, cezalı her puan için 3 jeton kaybeder. Bir ceza puanı cezalıya, verenin ödediğinin üç katı zarar verir; ceza etkilidir ama veren için de pahalıdır.

```
def ceza(c, p):          # p[i][j]: i'nin j'ye puanı
    n = len(c)
    pay = 1.6 * sum(c) / n
    u = [10 - c[i] + pay for i in range(n)]
    for i in range(n):
        for j in range(n):
            u[i] -= p[i][j]       # puan başı 1 öder
            u[j] -= 3 * p[i][j]   # cezalı 3 kaybeder
    return u
```

Kod satır satır şöyle çalışır:

- `p[i][j]`, oyuncu `i`'nin oyuncu `j`'ye verdiği ceza puanıdır; `p` bir puan matrisidir.
- `pay`, havuzdan her oyuncuya düşen paydır: toplam katkı 1,6 ile çarpılıp `n`'ye bölünür.
- `u`, ceza öncesi kazançlardır: elde kalan `10 - c[i]` artı havuz payı.
- İç içe döngü her (`i`, `j`) çifti üzerinde gezinir. Ceza veren `i`'nin kazancından puan sayısı kadar, cezalı `j`'nin kazancından puanın 3 katı düşülür.

Çağrı `ceza([0, 10, 10, 10], p)` olsun: O1 beleşçi, O2–O4 tam katkı yapıyor. Her işbirlikçi O1'e 2 ceza puanı verir; puan matrisinde `p[1][0] = p[2][0] = p[3][0] = 2`, öteki her şey 0'dır.

Adım	İşlem	u = [O1, O2, O3, O4]
Başlangıç	*pay = 1,6 · 30 / 4 = 12*; O1: *10 − 0 + 12*, ötekiler *10 − 10 + 12*	[22, 12, 12, 12]
i = 0	O1 kimseyi cezalandırmaz	[22, 12, 12, 12]
i = 1	O2, 2 puan için 2 öder; O1, 2 · 3 = 6 kaybeder	[16, 10, 12, 12]
i = 2	O3 de 2 öder; O1 6 daha kaybeder	[10, 10, 10, 12]
i = 3	O4 de 2 öder; O1 6 daha kaybeder	[4, 10, 10, 10]

Tablo 12.12: `ceza([0, 10, 10, 10], p)` izlemesi; her işbirlikçi O1'e 2 puan verir.



Şekil 12.10: Ceza sonrası kazançlar. Beleşçi O1 4 alır; ceza veren işbirlikçiler 10'ar alır. Herkes katsaydı herkes 16 alırdı (yeşil kesikli çizgi).

Tablo 12.12 ve Şekil 12.10 sonucu verir. Ceza öncesi beleşçi, katkı yapanlardan 10 fazla alıyordu (22'ye karşı 12). O3'ün cezası sonrasında (i = 2) beleşçinin üstünlüğü biter: O1 ile O2, O3 eşit 10'dadır. Üç ceza sonrasında beleşçi 4 alır. Katkı yapsaydı ve kimse ceza vermeseydi, herkes 10 katmış

olurdu ve O1 16 alırdı. Ceza tehdidi inandırıcıysa katmak kârlıdır: 16 > 4. Ama ceza verenler de 2'şer kaybetmiştir; ceza herkese maliyetlidir. Bu da yeni bir soru doğurur: cezayı kim verecek?

12.4.5 İkinci derece beleşçilik ve antisosyal ceza

Ceza beleşçiliği caydırır, ama ceza vermek de maliyetlidir. Aynı `ceza()` fonksiyonu üç farklı ceza düzeninde çalıştırılsın (Tablo 12.13).

Ceza verenler	O1	O2	O3	O4
Kimse	22	12	12	12
O2, O3, O4	4	10	10	10
O2, O3 (O4 vermez)	10	10	10	12

Tablo 12.13: Katkılar [0, 10, 10, 10]; ceza puanı 1 öder, 3 kaybettirir. Her ceza veren O1'e 2 puan verir.

Son satır önemlidir. O4 katkı yapmış ama ceza vermemiştir. O2 ve O3'ün cezası beleşçinin üstünlüğünü zaten silmiştir; O4 ise cezanın maliyetine katlanmadan bu faydadan yararlanır ve gruptaki en yüksek kazancı (12) alır. Ceza da bir kamu malıdır: maliyeti cezalandırana, faydası (beleşçinin caydırılması, gelecekte daha yüksek katkı) herkese düşer. Ceza vermeyen işbirlikçiye **ikinci derece beleşçi**, bu davranışa **ikinci derece beleşçilik** (*second-order free riding*) denir. Ceza vermeyenleri cezalandırmak bir çözüm gibi görünür, ama o ceza da maliyetlidir ve sorun bir üst düzeye taşınır: üçüncü derece beleşçilik.

Yalnızca parayı önemseyen oyuncular için sonuç geriye doğru çıkarımla bulunur. Son turda ceza vermek yalnızca maliyettir, çünkü cezanın caydıracağı bir gelecek tur yoktur; kimse vermez. Bunu bilen oyuncular son turda katkı yapmaz. Bir önceki turda da gelecek turların davranışı zaten belirlenmiştir; ceza yine yalnızca maliyettir. Bu mantık başa kadar yürür: tehdit inandırıcı değildir ve katkı 0'a iner. Deneylerde ise insanlar ceza verir (12.4.3): özgeci ceza, ikinci derece beleşçilik sorununu kısmen aşar.

Cezanın her yerde aynı biçimde işlemediği de görülmüştür. Herrmann, Thöni ve Gächter (2008), aralarında İstanbul'un da bulunduğu 16 şehirde aynı cezalı kamu malı deneyini yaptı. Bazı şehirlerde denekler yalnızca düşük katkı yapanları değil, *yüksek* katkı yapanları da cezalandırdı. Normlara uyanların cezalandırılmasına **antisosyal ceza** (*antisocial punishment*) denir. Antisosyal cezanın sık görüldüğü yerlerde cezanın işbirliğini artırıcı etkisi büyük ölçüde silindi. Yazarlar, yurttaşlık normlarının ve hukukun üstünlüğünün zayıf olduğu toplumlarda antisosyal cezanın daha sık olduğunu buldu. Antisosyal cezanın olası nedenlerinden biri intikamdır: önceki turda cezalandırılan beleşçi, kendisini cezalandırdığını düşündüğü yüksek katkı yapanlara karşılık verir.

Bu bulgunun dersi şudur: ceza bir kurumdur ve işleyişi, onu çevreleyen normlara bağlıdır. Bir platform ya da kurum tasarlarken "kullanıcılar birbirini cezalandırabilsin" kuralı tek başına işbirliği üretmez; kuralı kimin, kime karşı ve hangi gerekçeyle kullanacağı da önemlidir. Ostrom'un kademeli yaptırım ve hesap veren izleme ilkeleri (12.3.4), cezanın kötüye kullanılmasını sınırlayan kurumsal önlemler olarak da okunabilir.

Sık yapılan hata

"Ceza işbirliğini hep artırır." Ceza maliyetlidir ve ikinci derece beleşçilik doğurur; antisosyal ceza da etkisini silebilir. Ceza imkânının işbirliğini artırması, büyük ölçüde onu kullananların normlara uyanları değil, uymayanları hedef almasına bağlıdır.

12.4.6 Kamu Malı Oyunu'na dönüş

Bölümün başındaki iki soruya artık cevap verilebilir. Birinci soru, kuramın ve deneylerin katkı öngörüsüydü. Kuramın cevabı Teorem 12.1'de verildi: 0 jeton. Katılan her jeton katana yalnızca $1{,}6/4 = 0{,}4$ döndürür; diğerleri ne yaparsa yapsın, katmamak jeton başına 0,6 daha iyidir. Gerçekte ise tekrarlı deneylerde katkılar ilk turlarda bütçenin yaklaşık %40–60'ı kadardır ve turlar ilerledikçe azalır (Ledyard, 1995). Katılımcılar birbirini kendi parasından ödeyerek cezalandırabildiğinde katkılar yüksek kalır (Fehr ve Gächter, 2000).

İkinci soru, oyundaki "Cezalı katkı" senaryosuydu: katkısı 3 jetondan az olana 2 jeton ceza kesilir. Burada ceza oyunculardan değil kuralın kendisinden gelir; ceza verenin bir maliyeti yoktur. Bu kural, oyuncuların teşviklerini değiştirir.

Örnek 12.15

Kazanç $u_i = 10 - k_i + 0{,}4 \cdot \Sigma k - 2 \cdot [k_i < 3]$. Köşeli parantez, içindeki koşul doğruysa 1, değilse 0'dır.

1. **3'ten fazla katmak.** 3'ün üzerindeki her jeton, cezasız oyundaki gibi 0,6 kaybettirir; ceza durumu değişmez. 3'ten fazla katmak, 3 katmaktan kötüdür.
2. **1 ya da 2 katmak.** Hem jeton başına 0,6 kaybedilir hem 2 jetonluk ceza ödenir. 0 katmaktan kötüdür.
3. **0 ile 3'ün karşılaştırması.** 3 katmak, 0 katmaya göre $3 \cdot 0{,}6 = 1{,}8$ kaybettirir ama 2 jetonluk cezadan kurtarır. Net fark $2 - 1{,}8 = +0{,}2$: 3 katmak daha iyidir.
4. **Sonuç.** Diğerleri ne katarsa katsın 3 katmak en iyisidir: 3 baskın stratejidir. Dengede herkes 3 katar; havuz 12, çarpılınca 19,2, kişi başı pay 4,8; kazanç $10 - 3 + 4{,}8 = 11{,}8$.

Senaryo	Denge katkısı	Kişi başı kazanç
Temel	0	10
Cezalı (3'ten az katana 2 ceza)	3	11,8
Optimum (herkes 10)	10	16

Tablo 12.14: Kamu Malı Oyunu senaryoları.

Tablo 12.14 üç durumu karşılaştırır. Ceza kuralı, ortak meradaki Pigou vergisi gibi teşvikleri değiştiren bir kurumdur: dengeyi 0'dan 3'e taşır ve herkesin kazancını 10'dan 11,8'e çıkarır. Ama optimuma (10 katkı, 16 kazanç) götürmez. Kuralın etkisi, eşiğin ve cezanın büyüklüğüne bağlıdır. Ceza 1 jeton olsaydı, 3 katmak 1,8 kaybettirir ama yalnızca 1 jeton kurtarırdı; katmamak 0,8 daha iyi olurdu ve denge yine 0 olurdu. Genel kural şudur: eşik e ve ceza c ise, $c > 0{,}6 \cdot e$ olduğunda e katmak baskın olur. Tam katkıyı (10) zorlamak için cezanın 6'dan büyük olması gerekir.

12.5 Eşikli projeler

Kamu malı oyununda her jeton katkısı, havuzu biraz büyütür; katkı ile fayda arasındaki ilişki doğrusaldır. Pek çok gerçek projede ise fayda ancak belirli bir eşiğe ulaşıldığında ortaya çıkar. Bir köprü yarısı yapılınca işe yaramaz; bir kitlesel fonlama kampanyası hedefine ulaşmazsa ürün üretilmez; bir dilekçe yeterli imzayı toplamazsa gündeme alınmaz. Bu yapı, katkı sorununu bambaşka bir oyuna dönüştürür.

12.5.1 Katkı bir koordinasyon oyunu olur

Tanım 12.12

Eşikli kamu malı ya da **eşikli proje** (*threshold public good*): toplam katkı belirli bir eşiğe ulaşırsa herkesin bir fayda aldığı, ulaşmazsa katkıların boşa gittiği kamu malıdır. Kamu Malı Oyunu'nun eşikli proje senaryosunda 4 oyuncu 10'ar jetonla başlar; toplam katkı 22'ye ulaşırsa herkes 12 ek jeton alır, ulaşmazsa katkılar yanar:

$$u_i = 10 - k_i + 12 \cdot [\Sigma k \geq 22]$$

Bu oyunun dengeleri doğrusal kamu malı oyunundan çok farklıdır. Katmamak artık baskın değildir: eşiğin kıl payı kaçtığı bir durumda, eksik jetonu tamamlayan oyuncu 12 kazanır. Ama katmak da baskın değildir: eşiğe ulaşılamıyorsa her katkı boşa gider. Oyuncunun en iyi tepkisi, ötekilerin ne kattığına bağlıdır. Bu, oyunu bir **koordinasyon oyununa** dönüştürür: oyuncuların doğru kararı, ötekilerle aynı beklentiyi paylaşmalarına bağlıdır.

İki tür denge vardır.

Denge 1: kimse katmaz. Herkes 0 katıyorsa, bir oyuncunun tek başına katabileceği en fazla 10 jetondur ve eşik 22'ye ulaşılamaz. Katan yalnızca kaybeder. Kimse sapmak istemez; herkes 10 alır, toplam 40.

Denge 2: toplam tam 22. Toplam katkı tam 22 ise her katkıcı *belirleyicidir* (*pivotal*): 1 jeton bile çekse toplam 21'e iner, eşik kaçar ve 12 jetonu kaybeder. Fazladan katmak da anlamsızdır: eşik zaten tutuyor, fazlası boşa gider. Örneğin (6, 6, 5, 5) profilinde kazançlar (16, 16, 17, 17), toplam 66'dır.

Hangi denge oynanır? Kuram bunu söylemez. Sıfır dengesi kimseyi riske atmayan güvenli bir dengedir; eşik dengesi ise herkes için daha iyidir ama güven gerektirir. Katan oyuncu, diğerleri katmazsa katkısını yakar: 6 katan, diğerleri katmazsa 10 yerine 4'e düşer. Güven yoksa sıfır dengesi seçilebilir. Bu, Geyik Avı'ndaki ikilemin aynısıdır: işbirliği dengesi Pareto baskındır, ama yalnız kalan işbirlikçi büyük kayıp yaşar.

İkinci bir koordinasyon sorunu, eşiğe ulaşan dengelerin çokluğudur. (10, 10, 2, 0) katkı profili de dengedir; kazançlar (12, 12, 20, 22) olur. Toplamı tam 22 olan her profil dengedir ve bu profillerin hepsinde grubun toplam kazancı aynıdır: $40 - 22 + 4 \cdot 12 = 66$. Eşik dengeleri verimlidir, ama eşit olmayabilir. Kimin ne kadar katacağı bir pazarlık sorunudur: her oyuncu, başkalarının daha çok katmasını ister.

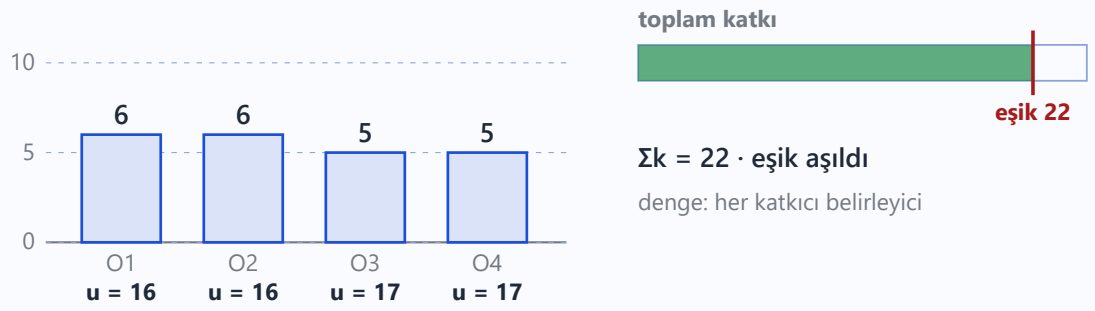
Gerçek hayattaki eşikli projelerde bu sorunları hafifleten bir mekanizma yaygındır: **iade**. Hedefe ulaşmayan kitlesel fonlama kampanyalarında para katkıcılara geri verilir. Bu durumda eşik tutmazsa katkı yanmaz; "katıp yakma" riski ortadan kalkar. Oyuncunun katkısı ancak eşik tuttuğunda harcanır, yani katkı ancak karşılığında 12 alındığında ödenir. Bu kural, sıfır dengesinin güvenlik avantajını siler ve koordinasyonu kolaylaştırır.

12.5.2 Hangi katkı profili dengedir?

Bir profilin denge olup olmadığını denetlemek için her oyuncunun tek başına sapmakla kazancını artırıp artıramayacağına bakılır. Eşikli projede iki tür sapma vardır: katkıyı azaltmak (eşik tutuyorsa) ve katkıyı artırmak (eşik tutmuyorsa). Beş profili sırayla denetleyelim.

Örnek 12.16

1. **(0, 0, 0, 0).** O1 tek başına 10 katsa toplam $10 < 22$ olur ve 10 jetonu yanar. Hiçbir oyuncu tek başına eşiğe ulaşamaz; kimse sapmak istemez. Nash dengesidir; herkesin kazancı 10.
2. **(6, 6, 5, 4).** Toplam 21, eşik kaçtı; kazançlar (4, 4, 5, 6). O4 5 katarsa toplam 22 olur ve $10 - 5 + 12 = 17 > 6$ alır. Denge değildir.
3. **(6, 6, 5, 5).** Toplam 22, kazançlar (16, 16, 17, 17). O1 5'e inerse toplam 21 olur ve O1 $10 - 5 = 5 < 16$ alır. Öteki oyuncular için de aynı geçerlidir; fazla katan da boşa öder. Nash dengesidir; toplam 66.
4. **(7, 6, 5, 5).** Toplam 23. O1 6'ya inerse toplam 22 olur, eşik yine tutar ve O1 1 jeton kazanır (15 yerine 16). Bu profil denge değildir.
5. **(10, 10, 2, 0).** Toplam 22. O3 çekilirse (0 katarsa) toplam 20 olur ve eşik kaçar: 20 yerine 10 alır. O1 ve O2 de belirleyicidir. O4 katarsa yalnızca kaybeder. Nash dengesidir; kazançlar (12, 12, 20, 22).



Şekil 12.11: (6, 6, 5, 5) katkı profili. Toplam katkı tam 22'dir ve eşik aşılır; kazançlar (16, 16, 17, 17). Her katkıcı belirleyicidir: biri 1 jeton çekse eşik kaçar.

Şekil 12.11 (6, 6, 5, 5) dengesini gösterir. Bu beş denetimden genel bir sonuç çıkar: bu oyunda Nash dengeleri tam olarak iki türdür. Ya herkes 0 katar, ya da toplam katkı tam 22'dir. Toplam 0 ile 22 arasındaysa eşik tutmuyordur; katkı yapan herhangi bir oyuncu katkısını 0'a çekerek kazanır, çünkü katkısı zaten boşa gitmektedir. Toplam 22'nin üstündeyse katkı yapan herhangi bir oyuncu katkısını 1 azaltır, eşik yine tutar ve 1 jeton kazanır. Geriye yalnızca toplamı 0 ve toplamı tam 22 olan profiller kalır; bu iki türün de denge olduğu 12.5.1'de gösterildi. Bütün olası profillerin bilgisayarla taranması da bu sonucu doğrular: her oyuncunun 0 ile 10 arasında katabildiği $11^4 = 14.641$ profilden toplamı tam 22 olan 850 profilin hepsi ve yalnızca bunlar, sıfır profiliyle birlikte dengedir. Verimli dengeler çoktur; kimin ne katacağı bir pazarlık ve koordinasyon sorunudur.

Sık yapılan hata

"Eşikli projede tek denge vardır." Hem sıfır katkı hem de toplamı tam 22 olan her profil dengedir. Ayrıca "eşit katkı (5,5'er)" özel bir denge değildir; tam sayı jetonlarla eşit bölüşüm bile mümkün değildir ve (10, 10, 2, 0) gibi çok eşitsiz profiller de dengedir.

12.6 Toplu bilgi

Önceki kısımlarda grup üyelerinin çıkarları çatışıyordu: biri fazla alınca ötekiler kaybediyordu. Bu kısımda ise herkesin çıkarı aynıdır: bir sorunun doğru cevabını bulmak. Sorun, her üyenin cevabı kısmen bilmesidir. Grubun bu dağınık bilgiyi bir araya getirip getiremeyeceği, bilginin nasıl paylaşıldığına bağlıdır. Üç model bu soruyu farklı açılardan ele alır: kalabalığın bilgeliği (tahminlerin ortalaması), bilgi şelaleleri (sırayla verilen ve birbirini izleyen kararlar) ve Condorcet jüri teoremi (çoğunluk oyu).

12.6.1 Kalabalığın bilgeliği

Francis Galton 1906'da İngiltere'de bir hayvan fuarında ilginç bir yarışmaya rastladı. Ziyaretçiler, kesilip parçalanacak bir öküzün ağırlığını tahmin etmek için bilet alıyordu. Galton yarışmadan sonra 787 geçerli tahmini topladı ve inceledi. Tahminlerin ortancası (sıralandığında ortaya düşen tahmin)

1.207 libre, öküzün gerçek ağırlığı 1.198 libreydi: hata %1'den az (yaklaşık %0,75). Katılımcıların çoğu hayvancılıktan anlamayan sıradan ziyaretçilerdi; ama tahminlerinin birleşimi şaşırtıcı derecede isabetliydi (Galton, 1907).

Bu olgu **kalabalığın bilgeliği** (*wisdom of crowds*) adıyla bilinir. Temel açıklama, bağımsız hataların birbirini götürmesidir. Biri ağırlığı fazla, öteki az tahmin eder; hatalar farklı yönlerde ve birbirinden bağımsızsa, ortalamada büyük ölçüde sıfırlanır. Scott Page bu sezgiyi kesin bir eşitliğe çevirdi.

Teorem 12.2 (Çeşitlilik tahmin teoremi)

Gerçek değer θ, n kişinin tahminleri $x_1, \ldots, x_n$ ve ortalamaları $\bar{x}$ olsun. O zaman

$$(\bar{x} - \theta)^2 = (1/n)\,\Sigma\,(x_i - \theta)^2 - (1/n)\,\Sigma\,(x_i - \bar{x})^2$$

Sözle: kalabalığın hatası = bireysel hata karelerinin ortalaması − tahminlerin çeşitliliği (Page, 2007). Çeşitlilik, tahminlerin kendi ortalamalarından sapma karelerinin ortalamasıdır.

İspat. Her bireysel hata iki parçaya ayrılır: $x_i - \theta = (x_i - \bar{x}) + (\bar{x} - \theta)$. Karesi alınır: $(x_i - \theta)^2 = (x_i - \bar{x})^2 + 2(x_i - \bar{x})(\bar{x} - \theta) + (\bar{x} - \theta)^2$. Bütün i'ler için toplanıp n'ye bölünür. Ortadaki terimde $(\bar{x} - \theta)$ her i için aynıdır ve dışarı çıkar; geriye $\Sigma(x_i - \bar{x})$ kalır. Bu toplam sıfırdır, çünkü ortalamadan sapmaların toplamı her zaman sıfırdır: $\Sigma x_i - n \cdot \bar{x} = 0$. Son terim n kez toplanıp n'ye bölününce $(\bar{x} - \theta)^2$ olur. Böylece bireysel hata ortalaması = çeşitlilik + kalabalığın hatası elde edilir; düzenlenince teorem çıkar. □

Örnek 12.17

Gerçek değer 50; beş kişinin tahminleri 40, 45, 55, 60, 70.

1. **Ortalama.** $(40 + 45 + 55 + 60 + 70)/5 = 270/5 = 54$.
2. **Kalabalığın hatası.** $(54 - 50)^2 = 16$.
3. **Bireysel hata kareleri.** $(x - 50)^2$: 100, 25, 25, 100, 400. Toplam 650, ortalama 130.
4. **Çeşitlilik.** $(x - 54)^2$: 196, 81, 1, 36, 256. Toplam 570, ortalama 114.
5. **Denetim.** $130 - 114 = 16$ ✓.

Kalabalığın hatası (16), ortalama bir üyenin hatasından (130) çok küçüktür; aradaki fark tam olarak çeşitliliktir (114).

Teoremin sonuçları güçlüdür. Çeşitlilik hiçbir zaman negatif olmadığından, kalabalığın hatası hiçbir zaman ortalama bir üyenin hatasından büyük değildir: kalabalık, ortalama bir üyesinden her zaman en az onun kadar iyi tahmin eder. Çeşitlilik ne kadar büyükse kazanç o kadar büyüktür. Daha küçük

bir örnek de aynı ilişkiyi gösterir: gerçek değer 10, tahminler 8, 12, 16 olsun. Ortalama 12, kalabalığın hatası 4'tür. Bireysel hata kareleri 4, 4, 36 (ortalama 44/3), çeşitlilik 16, 0, 16'dır (ortalama 32/3); fark *12/3 = 4* ✓.

Teorem bir eşitliktir ve her zaman doğrudur; ama kalabalığın gerçekten bilge olması iki koşula bağlıdır: **bağımsızlık** ve **çeşitlilik**. Tahminler birbirinden bağımsız değilse hatalar birbirini götürmez, aynı yönde birikir. Herkes birbirini izlerse tahminler birbirine yaklaşır, çeşitlilik çöker ve kalabalık, tek bir üye kadar yanılabilir. Bu çöküş, bilgi şelalelerinde rasyonel bireyler arasında bile ortaya çıkar (12.6.2).

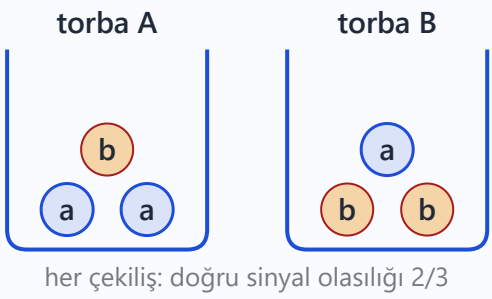
Sık yapılan hata

"Kalabalık, en iyi üyesinden iyidir." Teorem bunu söylemez. Kalabalık, *ortalama* bir üyeden en az onun kadar iyidir; en iyi üyeden kötü olabilir. Örnekte 45 ve 55 tahmin edenlerin hatası 25'tir; kalabalığınki 16 ile onlardan iyidir, ama gerçek değeri tam bilen biri olsaydı onun hatası 0 olurdu.

12.6.2 Bilgi şelaleleri

Yeni bir şehirde iki restoran yan yanadır. Birinde kuyruk var, öteki boştur. Çoğu kişi kuyruğa girer: "bu kadar insan yanılıyor olamaz". Ama kuyruktaki insanların her biri de aynı şeyi düşünmüş olabilir. İlk birkaç kişi rastlantıyla o restorana girmiş, sonrakiler onları izlemiş olabilir. Bu durumda kuyruk, restoranın kalitesi hakkında neredeyse hiçbir bilgi taşımaz. Bikhchandani, Hirshleifer ve Welch (1992) bu olguyu kesin bir modelle inceledi.

Model. İki torba vardır. Torba A'da 2 a topu ve 1 b topu, torba B'de 1 a topu ve 2 b topu bulunur (Şekil 12.12). Torbalardan biri yazı-turayla seçilir; kimse hangisinin seçildiğini bilmez. Sırayla her kişi torbadan gizlice bir top çeker, topu görür, geri koyar. Sonra öncekilerin tahminlerini (toplarını değil) duyar ve torbanın A mı B mi olduğunu yüksek sesle tahmin eder. Doğru tahmin ödüllendirilir.



Şekil 12.12: İki torba. Torba A'daki topların 2/3'ü a, torba B'dekilerin 2/3'ü b'dir; her çekiliş torbanın kendi rengini 2/3 olasılıkla gösterir.

Her top, 2/3 olasılıkla torbanın kendi rengindedir; bu yüzden topa **sinyal** denir: torba hakkında gürültülü ama yararlı bir bilgi taşır. Her kişinin sorusu aynıdır: eldeki bilgilere göre torbanın A olma olasılığı nedir?

Bayes hesabı. Başlangıçta iki torba eşit olasılıklıdır: $P(A) = 1/2$. Bu ön olasılıktır. Bir a sinyali görüldükten sonra son olasılık Bayes kuralıyla hesaplanır:

$$P(A \mid a) = P(a \mid A) \cdot P(A) \,/\, [P(a \mid A) \cdot P(A) + P(a \mid B) \cdot P(B)] = (\tfrac{2}{3} \cdot \tfrac{1}{2}) \,/\, (\tfrac{2}{3} \cdot \tfrac{1}{2} + \tfrac{1}{3} \cdot \tfrac{1}{2}) = 2/3$$

Hesabı sadeleştirmenin yolu olasılık oranlarıyla düşünmektir. A lehine oran başlangıçta 1'e 1'dir. Bir a sinyali A'da 2/3, B'de 1/3 olasılıklıdır; bu yüzden her a, A lehine oranı 2 katına çıkarır. Her b ise oranı yarıya indirir. Bağımsız sinyallerde sıra önemli değildir; yalnızca a'ların b'lerden ne kadar fazla olduğu önemlidir. a'ların b'lerden f fazla olduğu durumda oran 2^f'e 1'dir ve

$$P(A) = 2^f \,/\, (2^f + 1)$$

Örnek 12.18

1. **f = 0** (sinyal yok ya da a ve b sayısı eşit): $P(A) = 1/(1 + 1) = 1/2$.
2. **f = 1** (bir a fazla): $P(A) = 2/3$.
3. **f = 2** (iki a fazla): $P(A) = 4/5 = 0{,}8$.
4. **f = −2** (iki b fazla): $P(A) = 1/(4 + 1) = 0{,}2$.
5. **a, a, b sinyalleri** (f = 1): uzun yoldan, $P(A) = (\tfrac{2}{3} \cdot \tfrac{2}{3} \cdot \tfrac{1}{3}) \,/\, (\tfrac{2}{3} \cdot \tfrac{2}{3} \cdot \tfrac{1}{3} + \tfrac{1}{3} \cdot \tfrac{1}{3} \cdot \tfrac{2}{3}) = (4/27)/(4/27 + 2/27) = 2/3$. Kısa yol aynı sonucu verir.

Kişiler ne yapar? Birinci kişinin yalnızca kendi sinyali vardır; a görürse A, b görürse B der. Onun tahmini, sinyalini tam olarak ele verir. İkinci kişi birinci kişinin tahmininden onun sinyalini çıkarır ve kendi sinyaliyle birleştirir. İki sinyal aynıysa onu söyler. Farklıysa $f = 0$ olur, iki torba eşit olasılıklıdır; model gereği eşitlikte kişi kendi sinyalini izler. Böylece ikinci kişinin tahmini de sinyalini ele verir.

Şelale. İlk iki kişi a görüp A dediyse, üçüncü kişi öncekilerin tahminlerinden iki a çıkarır: $f = 2$. Kendi sinyali b ise $f = 2 - 1 = 1$ olur ve $P(A) = 2/3 > 1/2$. Kendi sinyalini yok sayıp A demek rasyoneldir. Kendi sinyali a olsaydı da A diyecekti. Demek ki üçüncü kişinin tahmini, sinyali ne olursa olsun A'dır ve sinyali hakkında hiçbir bilgi taşımaz. Dördüncü kişi, üçüncünün A demesinden hiçbir şey öğrenemez; o da üçüncüyle tam aynı durumdadır ve aynı nedenle A der. Bundan sonraki herkes A der.

Tanım 12.13

Bilgi şelalesi (*information cascade*): bir kişinin, öncekilerin davranışından çıkardığı bilgiyi kendi özel bilgisinden daha güçlü bulup kendi bilgisini yok sayarak öncekileri izlemesi ve bu yüzden davranışının sonrakilere yeni bilgi taşımamasıdır. İki torba modelinde öncekilerin tahminlerinden çıkarılan fark $|f| = 2$'ye ulaşınca şelale başlar.

Şelalelerin iki önemli özelliği vardır. Birincisi, **yanlış olabilirler**. Gerçek torba B olsa bile, ilk iki kişinin a çekme olasılığı $\frac{1}{3} \cdot \frac{1}{3} = 1/9$'dur; bu durumda bütün kalabalık yanlış A şelalesine girer ve sonradan kaç kişi b görürse görsün düzelmez. İkincisi, **kırılgandırlar**. Şelale yalnızca iki sinyale dayanır; yüzlerce kişinin A demesi, aslında iki topun bilgisidir. Kamuya açık tek bir bilgi (örneğin herkesin görebileceği iki b topu) onu bozabilir. Moda akımlarının, ürün yorumlarının ve finansal paniklerin ani dönüşleri bu kırılganlıkla açıklanır: uzun süren bir eğilim, küçük bir yeni bilgiyle tamamen tersine dönebilir.

Model laboratuvarda da sınandı. Anderson ve Holt (1997) bu torba deneyini öğrencilerle yaptı. Şelalelerin oluşabildiği durumların (yani tahminlerden çıkarılan farkın 2'ye ulaştığı durumların) çoğunda şelaleler gerçekten oluştu: denekler kendi sinyallerine aykırı olsa da öncekileri izledi.

12.6.3 Kod izleme: selale()

Şelale modeli birkaç satırlık bir fonksiyonla yazılabilir. Fonksiyon sinyal dizisini alır ve her kişinin tahminini döndürür. `f` değişkeni, öncekilerin tahminlerinden çıkarılan a − b farkıdır.

```
def selale(sinyaller):
    f = 0                  # tahminlerden çıkan a - b
    tahmin = []
    for s in sinyaller:
        if f >= 2: t = 'A'     # A şelalesi
        elif f <= -2: t = 'B'  # B şelalesi
        else:
            t = 'A' if s == 'a' else 'B'
            f += 1 if s == 'a' else -1
        tahmin.append(t)
    return tahmin
```

Kod satır satır şöyle çalışır:

- `f` başlangıçta 0'dır: henüz hiç tahmin yoktur.
- `f >= 2` ise A şelalesi vardır: kişi kendi sinyali ne olursa olsun A der. `f <= -2` ise B şelalesi vardır: kişi B der. Bu iki durumda `f` değişmez, çünkü şelaledeki tahmin sonrakilere bilgi taşımaz.

- Aksi hâlde ($|f| \le 1$) kişi kendi sinyalini izler. f = 1 iken kendi a'sıyla f = 2, kendi b'siyle f = 0 olur ve eşitlikte kendi sinyalini izler; f = −1 durumu simetriktir. Bu durumda tahmin sinyali ele verir; sonraki kişiler onu çıkarabilir, bu yüzden `f` güncellenir.

Kişi	Sinyal	Öncekilerden f	Kendi sinyaliyle P(A)	Tahmin	Sonraki f
1	a	0	2/3	A	1
2	a	1	0,8	A	2
3	b	2	2/3	A (şelale)	2
4	b	2	2/3	A	2
5	b	2	2/3	A	2
6	b	2	2/3	A	2

Tablo 12.15: `selale(['a', 'a', 'b', 'b', 'b', 'b'])` izlemesi; gerçek torba B.



Şekil 12.13: Altı kişilik şelale. İlk iki kişi a görüp A der; f = 2 olunca şelale başlar ve sonraki dört kişi b görse de kendi sinyalini yok sayıp A der. Gerçek torba B'dir.

Tablo 12.15 ve Şekil 12.13 izlemeyi gösterir. Kişi 1 a görür; f = 0 olduğundan kendi sinyalini izler ve A der. Tahmini sinyali ele verir: f = 1, sonraki kişi için $P(A) = 2/3$. Kişi 2 de a görür; kendi a'sıyla $P(A) = 0{,}8$ olur ve A der; f = 2. Kişi 3 b görür; ama öncekilerden çıkardığı iki a ile kendi b'si birleşince $P(A) = 2/3$ olur ve A der. Şelale başlamıştır; f artık değişmez. Kişi 4, 5 ve 6 da b görür ve aynı hesapla A der; onların tahminleri de bilgi taşımaz.

Fonksiyon `['A', 'A', 'A', 'A', 'A', 'A']` döndürür: herkes A dedi, oysa 6 sinyalin 4'ü b'dir. Sinyaller birleştirilseydi $f = 2 - 4 = -2$ ve $P(A) = 0{,}2$ olurdu; doğru cevap B bulunurdu. Her kişi, elindeki bilgiyle Bayes'e göre doğru davrandı; sorun, bilginin birikmemesidir. Dört b sinyali vardı, ama hiçbiri tahminlere yansımadığı için kayboldu.

Sık yapılan hata

"Şelaleyi izleyenler irrasyoneldir" ya da "sosyal baskıyla uyarlar." Modelde kimse sosyal baskı hissetmez ve herkes Bayes'e göre doğru hesap yapar. Kişi 3'ün A demesi, eldeki bilgiyle en iyi tahmindir. Sorun bireylerde değil, bilginin paylaşılma biçimindedir: kişiler sinyallerini değil yalnızca tahminlerini açıklar.

12.6.4 Condorcet jüri teoremi

Şelalelerde kişiler sırayla karar verir ve birbirini izler. Bir jüride ise herkes aynı anda ve birbirinden bağımsız oy verir; karar çoğunlukla alınır. Ortada tek bir doğru cevap vardır (sanık suçlu mu, değil mi?). Soru, çoğunluk oyunun bu doğru cevabı bulma olasılığının jüri büyüdükçe nasıl değiştiğidir. Bu soru, tercihlerin döngüye girdiği Condorcet paradoksundan farklıdır: paradoks "grubun tercihi tutarlı mı?" diye sorar; jüri teoremi ise "grup doğruyu buluyor mu?" diye sorar.

Teorem 12.3 (Condorcet jüri teoremi)

n (tek sayıda) seçmen bağımsız oy verir ve her biri $p > 1/2$ olasılıkla doğruyu seçerse, çoğunluğun doğru olma olasılığı n ile artar ve $n \to \infty$ iken 1'e gider (Condorcet, 1785). $p < 1/2$ ise tersine: kalabalık büyüdükçe çoğunluk neredeyse kesin yanılır.

Çoğunluğun doğru olma olasılığını hesaplamak için **binom olasılığı** kullanılır. n seçmenden tam k tanesinin doğru oy verme olasılığı, doğru oy veren k kişiyi seçmenin yol sayısı ile tek bir seçimin olasılığının çarpımıdır:

$$P(\text{tam } k \text{ doğru}) = C(n, k) \cdot p^k \cdot (1 - p)^{n-k}$$

Burada $C(n, k)$, n kişiden k kişi seçme yollarının sayısıdır (örneğin $C(5, 3) = 10$). p^k, seçilen k kişinin hepsinin doğru olması; $(1 - p)^{n-k}$, geri kalanların hepsinin yanlış olmasıdır; bağımsızlık sayesinde olasılıklar çarpılır. Doğru oy sayısı $n/2$'den büyükse çoğunluk doğrudur. Bu yüzden

$$P_n = \Sigma_{k > n/2}\, C(n, k) \cdot p^k \cdot (1 - p)^{n-k}$$

Örnek 12.19

Üç kişilik jüri, $p = 0{,}6$. Çoğunluk için en az 2 doğru oy gerekir.

1. **Üçü de doğru.** $C(3, 3) \cdot 0{,}6^3 = 0{,}216$.
2. **Tam ikisi doğru.** Yanlış olan kişi 3 farklı biçimde seçilebilir: $C(3, 2) \cdot 0{,}6^2 \cdot 0{,}4 = 3 \cdot 0{,}36 \cdot 0{,}4 = 0{,}432$.
3. **Toplam.** $P_3 = 0{,}216 + 0{,}432 = 0{,}648$.

Tek bir seçmen %60 olasılıkla doğrudur; üç kişilik jürinin çoğunluğu %64,8 olasılıkla doğrudur.

n	p = 0,6	p = 0,4
1	0,600	0,400
3	0,648	0,352

n	p = 0,6	p = 0,4
5	0,683	0,317
11	0,753	0,247
51	0,926	0,074
101	0,979	0,021

Tablo 12.16: Çoğunluğun doğru karar verme olasılığı P_n.

Tablo 12.16 teoremin iki yüzünü gösterir. *p = 0,6* iken her seçmen yalnızca biraz bilgilidir, ama 101 kişilik bir çoğunluk %97,9 olasılıkla doğrudur. *p = 0,4* iken tablo simetrik olarak tersine döner: 101 kişide çoğunluk yalnızca %2,1 olasılıkla doğrudur. Bunun nedeni büyük sayılar yasasıdır: seçmen sayısı arttıkça doğru oyların oranı *p*'ye yaklaşır. *p > 1/2* ise bu oran yarıyı geçer ve çoğunluk doğrudur; *p < 1/2* ise yarının altında kalır ve çoğunluk yanlıştır. Tablodaki iki sütunun toplamı her satırda 1'dir, çünkü *p = 0,4*'teki doğru oy sayısı, *p = 0,6*'daki yanlış oy sayısıyla aynı dağılımı izler.

12.6.5 Kod izleme: juri()

Binom toplamı kısa bir fonksiyonla hesaplanır. Python'un `math.comb(n, k)` fonksiyonu *C(n, k)*'yi verir.

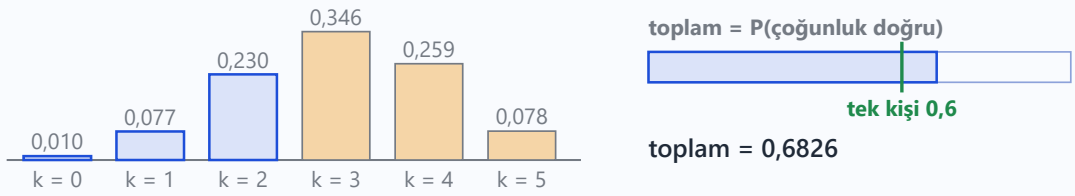
```
from math import comb
def juri(n, p):
    toplam = 0
    for k in range(n // 2 + 1, n + 1):
        terim = comb(n, k) * p**k * (1 - p)**(n - k)
        toplam += terim
    return toplam
print(round(juri(5, 0.6), 4))
```

Kod satır satır şöyle çalışır:

- `range(n // 2 + 1, n + 1)`, çoğunluk sağlayan doğru oy sayılarını üretir. `n // 2` tam sayı bölmesidir; `n = 5` için 2'dir, yani döngü `k = 3, 4, 5` değerlerini alır.
- Her `k` için `terim`, tam `k` kişinin doğru olma olasılığıdır ve `toplam`'a eklenir.
- Son satır `juri(5, 0.6)`'yı dört ondalığa yuvarlayıp yazdırır.

k	C(5, k)	$0{,}6^k$	$0{,}4^{5-k}$	terim	toplam
3	10	0,2160	0,1600	0,3456	0,3456
4	5	0,1296	0,4000	0,2592	0,6048
5	1	0,0778	1	0,0778	0,6826

Tablo 12.17: `juri(5, 0.6)` izlemesi.



Şekil 12.14: Beş kişilik jüride ($p = 0{,}6$) doğru oy sayısının dağılımı. Turuncu çubuklar (k = 3, 4, 5) çoğunluğun doğru olduğu durumlardır ve toplamları 0,6826'dır; tek bir seçmenin olasılığı 0,6'dır.

Tablo 12.17 üç terimi tek tek verir; Şekil 12.14 ise bütün dağılımı gösterir. En olası sonuç tam 3 doğru oydur (0,3456). k = 0, 1, 2 (yanlış çoğunluk) durumlarının olasılıkları 0,010, 0,077 ve 0,230'dur ve toplamları 0,3174'tür; iki toplam 1 eder. Program `0.6826` yazdırır: $P_5 = 0{,}6826 > 0{,}6$. Beş bağımsız ve orta yetkinlikte seçmen, tek bir seçmenden daha iyi karar verir.

12.6.6 Jüri teoremi ne zaman çöker?

Teoremin iki varsayımı vardır: oylar bağımsızdır ve her seçmenin doğruyu bulma olasılığı yarıdan büyüktür. Üçüncü, örtük bir varsayım da seçmenlerin bildiklerini dürüstçe oylamasıdır. Bunlardan biri bozulunca teoremin sonucu da bozulur.

Bağımlılık. 5 kişilik bir jüride 4 kişi bir kanaat önderinin oyunu izlerse, çoğunluk her zaman o kişinin oyudur ve doğru olma olasılığı 0,6'dır. Bağımsız oylarda 0,683'tü. Bağımlılık jüriyi tek bir seçmene indirir; kalabalığın avantajı kaybolur. Bilgi şelaleleri ve ortak bilgi kaynakları (herkesin aynı haberi okuması) bağımsızlığı bozan başlıca mekanizmalardır.

Yetkinlik. $p < 1/2$ olan bir konuda, örneğin herkesi aynı yöne yanıltan bir ipucu varsa, kalabalık büyüdükçe yanılgı kesinleşir: $p = 0{,}4$ iken 101 kişide çoğunluk yalnızca %2,1 olasılıkla doğrudur. Kalabalığı büyütmek bilgiyi değil, ortak yanılgıyı güçlendirir.

Stratejik oy. Seçmen, oyunun yalnızca sonucu değiştirebildiği, yani belirleyici olduğu durumda önem taşıdığını hesaba katarsa, kendi sinyaline aykırı oy vermesi rasyonel olabilir (Austen-Smith ve Banks, 1996). Örneğin oybirliği kuralında bir jüri üyesinin oyu, ancak öteki bütün üyeler “suçlu” dediğinde belirleyicidir. Bu durumda öteki üyelerin hepsinin suçluluk işareti gördüğü sonucu çıkar; bu bilgi, kendi “masum” işaretini bastıracak kadar güçlü olabilir. Bu mantıkla jüri üyeleri “suçlu” oyuna kayabilir ve oybirliği kuralı, masumu mahkûm etme olasılığını artırabilir (Feddersen ve Pesendorfer, 1998). Masumları korumak için tasarlanan bir kural, stratejik davranış yüzünden tersine işleyebilir.

Kalabalığın bilgeliği, şelaleler ve jüri teoremi aynı dersi verir: bir grubun bilgisi, üyelerinin bağımsız bilgisinin bir araya gelebildiği ölçüde değerlidir. Ortalama alma, sıralı karar ve çoğunluk oyu, bilgiyi birleştirmenin üç farklı mekanizmasıdır; üçü de bağımsızlık bozulduğunda başarısız olur.

Sık yapılan hata

"Jüri teoremi ile Condorcet paradoksu aynıdır." İkisi de Condorcet'nin adını taşır ama farklı sorulara cevap verir. Teorem, bir doğru cevabın varlığını kabul eder ve çoğunluğun onu bulma olasılığıyla ilgilidir. Paradoks ise doğru cevap aramaz; bireysel tercihler tutarlıyken çoğunluk tercihlerinin döngüye girebileceğini (A > B, B > C, C > A) gösterir.

12.7 Kolektif kararda sık hatalar ve örnek problemler

Bölümün ikinci yarısındaki modellerin sonuçları birbirine benzediği için kolayca, bazen de yanlış biçimde genellenir. Aşağıdaki liste en sık görülen yanılgıları ve doğrularını bir arada verir.

Sık yapılan hata

- **"Kalabalık her zaman bilgedir."** Bağımsızlık ve $p > 1/2$ gerekir. Şelalede altı kişinin altısı da yanıldı; $p = 0{,}4$ iken büyük jüri neredeyse kesin yanılır.
- **"Şelaleyi izleyenler irrasyoneldir."** Her biri Bayes'e göre doğru davrandı; sorun, bilginin birikmemesidir.
- **"Ceza işbirliğini hep artırır."** Ceza maliyetlidir, ikinci derece beleşçilik doğurur; antisosyal ceza etkisini silebilir.
- **"Eşikli projede tek denge vardır."** Hem sıfır katkı hem de toplamı tam 22 olan her profil dengedir.
- **"Ortak malın çözümü ya devlet ya piyasadır."** Ostrom'un yerel kurumları üçüncü bir yoldur.
- **"Jüri teoremi ile Condorcet paradoksu aynıdır."** Teorem doğruyu bulma olasılığıyla, paradoks tercihlerin döngüsüyle ilgilidir.

Aşağıdaki beş problem bölümün araçlarını birlikte kullanır. Her biri, notta çözülen bir örneğin bir parametresini değiştirir; çözüm yolu tanıdıktır, ama sayılar yeniden hesaplanır.

Örnek 12.20

Soru. Beklenti kuramı parametreleriyle ($v(x) = x^{0{,}88}$, $\lambda = 2{,}25$) +300 / −150 yazı-tura bahsi kabul edilir mi?

Çözüm. $v(300) = 300^{0{,}88} \approx 151{,}3$, $v(-150) = -2{,}25 \cdot 150^{0{,}88} \approx -2{,}25 \cdot 82{,}2 \approx -185{,}0$. Beklenti değeri $\tfrac{1}{2} \cdot 151{,}3 - \tfrac{1}{2} \cdot 185{,}0 \approx 75{,}7 - 92{,}5 \approx -16{,}8 < 0$: bahis reddedilir. Beklenen değeri +75 olduğu hâlde reddedilmesinin nedeni, kazancın kabul eşiğinin ($150 \cdot 2{,}51 \approx 377$ TL) altında kalmasıdır.

Örnek 12.21

Soru. Ortak mallar modelinde ($u_i = g_i(100 - G)$) $n = 9$ çoban için denge G^* ve toplam kazanç nedir? Optimumu sağlayan vergi kaçtır?

Çözüm. $g^* = 100/(n + 1) = 100/10 = 10$, $G^* = 90$; hayvan başı değer 10, toplam $90 \cdot 10 = 900$ (optimumun %36'sı). Vergiyle $g = (100 - t)/10$; optimum $G = 50$ için $g = 50/9$, yani $(100 - t)/10 = 50/9 \Rightarrow 100 - t = 55{,}6 \Rightarrow t \approx 44{,}4$.

Örnek 12.22

Soru. Kamu Malı Oyunu'nun cezalı senaryosunda ceza 2 yerine 1 jeton olsaydı denge katkısı kaç olurdu?

Çözüm. 3 jeton katmak $3 \cdot 0{,}6 = 1{,}8$ kaybettirir, ceza ise yalnızca 1 jetondur. Diğerleri ne katarsa katsın, katmamak $1{,}8 - 1 = 0{,}8$ daha iyidir; 1 ya da 2 katmak hem jeton kaybettirir hem cezadan kurtarmaz. Denge katkısı 0'dır; ceza kuralı bu büyüklükte etkisizdir.

Örnek 12.23

Soru. Fehr–Schmidt modelinde $\alpha = 2$ olan yanıtlayan 10 jetondan en az kaçını ister?

Çözüm. $s \geq 10\alpha/(1 + 2\alpha) = 20/5 = 4$. Denetim: $s = 4$ için $4 - 2 \cdot (10 - 8) = 0$; kabul ile ret aynı faydayı verir ve kabul koşulu (≥ 0) sağlanır. $s = 3$ için $3 - 2 \cdot 4 = -5 < 0$, ret. En az 4 jeton ister.

Örnek 12.24

Soru. Condorcet jüri teoreminde $p = 0{,}7$ iken 3 kişilik jürinin çoğunluğu hangi olasılıkla doğrudur?

Çözüm. $P_3 = 0{,}7^3 + 3 \cdot 0{,}7^2 \cdot 0{,}3 = 0{,}343 + 0{,}441 = 0{,}784$. Tek seçmenin 0,7'lik doğruluğu, üç kişilik çoğunlukta 0,784'e çıkar.

12.8 Özet

- **Sınırlı rasyonellik**: bilgi, zaman ve hesaplama sınırlı olduğundan insanlar "yeterince iyi"yi arar. Sapmalar rastgele değil sistematiktir; bu yüzden modellenebilir.
- **Beklenti kuramı**: sonuçlar referans noktasına göre kazanç ya da kayıp olarak değerlendirilir. Değer fonksiyonu kazançta içbükey, kayıpta dışbükey ve kayıpta $\lambda \approx 2{,}25$ kat diktir. ±100 TL bahsi reddedilir (−36,0); kabul eşiği ≈ 251 TL'dir. Olasılık ağırlığı küçük olasılıkları abartır ($w(0{,}01) \approx 0{,}055$).

- **Bağış etkisi ve çerçeveleme** aynı kökten gelir: referans noktası değişince aynı nesne ya da aynı program farklı değerlendirilir. Kupada satış/alış oranı λ'dır; salgın sorusunda %72 kesin seçenekten %78 riskli seçeneğe dönüş görülür.
- **Kuantal tepki**: logit seçimde iyi seçenek daha sık seçilir; $\lambda = 0$ rastgele, $\lambda \to \infty$ en iyi tepkidir. QRE, asimetrik Yazı-Tura'da Nash'in 1/2'si yerine büyük kazanca kaymayı (λ = 0,02'de 0,83) yakalar.
- **Sosyal tercihler**: diktatörde ortalama pay yaklaşık %28, ültimatomda teklifler %40–50'dir. Fehr–Schmidt modelinde $\alpha = 1$ olan yanıtlayan 10/3'ün altındaki teklifi reddeder; diktatör $\beta > 1/2$ ise eşit böler. Güven oyununda ortalama 5,16 $ gönderilir, 4,66 $ döner: karşılıklılık niyete yanıt verir.
- **Ortak mallar**: 4 çobanlı dengede $g^* = 20$, toplam 1.600; optimum 2.500 (%64). n arttıkça dışsallık büyür ve toplam sıfıra gider. Pigou vergisi ($t = 37{,}5$), kota ve Ostrom'un sekiz ilkesine dayanan yerel kurumlar çözüm yollarıdır.
- **Refah ölçütleri**: faydacı, eşitlikçi ve Nash sosyal refahı farklı politikalar seçebilir. Cehalet peçesi ardında Laplace kuralı faydacı ölçüte (Harsanyi), maksimin eşitlikçi ölçüte (Rawls) götürür.
- **Kamu malı ve ceza**: kuramda katmamak baskındır; deneylerde katkılar %40–60'tan başlayıp erir. Koşullu işbirliği ve tek bir beleşçi bu düşüşü üretir (10 → 1,3). Ceza işbirliğini korur ama maliyetlidir; ikinci derece beleşçilik ve antisosyal ceza doğurabilir. Kural olarak konan ceza dengeyi 0'dan 3'e taşır.
- **Eşikli projeler**: katkı bir koordinasyon oyununa dönüşür; sıfır dengesi ile toplamı tam 22 olan her profil dengedir. Eşik dengeleri verimli (toplam 66) ama eşitsiz olabilir; iade koordinasyonu kolaylaştırır.
- **Toplu bilgi**: çeşitlilik teoremi (kalabalığın hatası = bireysel hata − çeşitlilik); şelalede bilgi birikmez (altı kişinin altısı yanılır); jüri teoremi bağımsız ve yetkin seçmenlerde çoğunluğu güçlendirir (5 kişi, $p = 0{,}6 \Rightarrow 0{,}683$). Üç mekanizma da bağımsızlığa dayanır.

Model	Klasik öngörü	Gözlem / modelin öngörüsü
Beklenti kuramı	±100 bahsine kayıtsız	Reddedilir; eşik ≈ 251 TL
Kuantal tepki	Satır Üst'ü 1/2 oynar	Büyük kazanca kayar (0,83)
Fehr–Schmidt	Her pozitif teklif kabul	α = 1 iken 4'ten azı ret
Ortak mallar	—	Denge 1.600, optimum 2.500
Kamu malı	0 katkı	%40–60'tan başlayıp erir
Ceza	Kimse cezalandırmaz	Özgeci ceza; katkı yüksek kalır
Eşikli proje	—	Sıfır dengesi ve toplamı 22 olan dengeler
Şelale	—	İki sinyal herkesin tahminini belirler
Jüri teoremi	—	p > 1/2 ise çoğunluk 1'e gider

Tablo 12.18: Bölümün modelleri bir bakışta.

12.9 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

sınırlı rasyonellik *bounded rationality* · yeterince iyiyi arama *satisficing* · sezgisel kurallar *heuristics* · aşırı güven *overconfidence* · risk altında karar *decision under risk* · beklenti kuramı *prospect theory* · referans noktası *reference point* · değer fonksiyonu *value function* · kayıptan kaçınma *loss aversion* · olasılık ağırlığı *probability weighting* · bağış etkisi *endowment effect* · statüko yanlılığı *status quo bias* · çerçeveleme etkisi *framing effect* · logit seçim *logit choice* · kuantal tepki dengesi *quantal response equilibrium* · seviye-k düşünme *level-k reasoning* · diktatör oyunu *dictator game* · ültimatom oyunu *ultimatum game* · eşitsizlikten kaçınma *inequity aversion* · güven oyunu *trust game* · karşılıklılık *reciprocity* · kolektif karar *collective decision* · uzlaşı *consensus* · ortak mallar trajedisi *tragedy of the commons* · dışsallık *externality* · sosyal optimum *social optimum* · Pigou vergisi *Pigouvian tax* · sosyal refah *social welfare* · faydacı ölçüt *utilitarian criterion* · eşitlikçi ölçüt *egalitarian criterion* · Nash sosyal refahı *Nash social welfare* · cehalet peçesi *veil of ignorance* · Laplace ölçütü *Laplace criterion* · maksimin *maximin* · kamu malı *public good* · kamu malı oyunu *public goods game* · beleşçi *free rider* · beleşçilik *free riding* · koşullu işbirlikçi *conditional cooperator* · özgeci ceza *altruistic punishment* · ikinci derece beleşçilik *second-order free riding* · antisosyal ceza *antisocial punishment* · eşikli kamu malı (eşikli proje) *threshold public good* · kalabalığın bilgeliği *wisdom of crowds* · çeşitlilik tahmin teoremi *diversity prediction theorem* · Bayes kuralı *Bayes' rule* · bilgi şelalesi *information cascade* · Condorcet jüri teoremi *Condorcet jury theorem*

12.10 Kendini Dene

1. Beklenti kuramı parametreleriyle ($v(x) = x^{0,88}$, $\lambda = 2,25$) yarı yarıya +150 TL ya da −50 TL veren bahis kabul edilir mi? 50 TL kaybı göze almak için en az ne kadar kazanç gerekir?

   Cevap

   $v(150) = 150^{0,88} \approx 82,2$, $v(-50) = -2,25 \cdot 50^{0,88} \approx -2,25 \cdot 31,3 \approx -70,4$. Beklenti değeri $\tfrac{1}{2} \cdot 82,2 - \tfrac{1}{2} \cdot 70,4 \approx 5,9 > 0$: kabul edilir. Eşik $G = 50 \cdot 2,25^{1/0,88} \approx 50 \cdot 2,51 \approx 125,7$ TL'dir; 150 bu eşiğin üstündedir. Oran (≈ 2,51) kayıp tutarından bağımsızdır.

2. Fehr–Schmidt modelinde 10 jetonluk ültimatom oyunu oynanıyor ve teklifler tam sayıdır. $\alpha = 0,5$

olan yanıtlayan en az kaç jetonu kabul eder? $\alpha = 3$ olan yanıtlayan için cevap nedir?

Cevap

Eşik $10\alpha/(1 + 2\alpha)$'dır. $\alpha = 0{,}5$: $5/2 = 2{,}5$; $s = 2$ için $2 - 0{,}5 \cdot 6 = -1$ (ret), $s = 3$ için $3 - 0{,}5 \cdot 4 = +1$ (kabul): en az 3. $\alpha = 3$: $30/7 \approx 4{,}29$; $s = 4$ için $4 - 3 \cdot 2 = -2$ (ret), $s = 5$ eşit bölüşümdür ve kabul edilir: en az 5. α ne kadar büyük olursa olsun eşik 5'i aşmaz.

3. Üç çoban ortak meraya hayvan salıyor; hayvan başı değer $60 - G$, çoban i'nin kazancı $g_i(60 - G)$. Simetrik Nash dengesi, sosyal optimum, dengenin optimuma oranı ve optimumu sağlayan hayvan başı vergi nedir?

Cevap

En iyi tepki $g_i = (60 - G_{-i})/2$. Simetrik: $g = (60 - 2g)/2 \Rightarrow 4g = 60 \Rightarrow g^* = 15$, $G^* = 45$; hayvan başı değer 15, kişi başı 225, toplam 675. Kontrol: diğerleri 30 iken $(60 - 30)/2 = 15$. Optimum: $G(60 - G)$ en büyük $\Rightarrow G = 30$, toplam 900. Oran $675/900 = \%75$. Vergiyle $g = (60 - t)/4$, $G = 3(60 - t)/4 = 30 \Rightarrow t = 20$.

4. Kamu Malı Oyunu'nda (4 oyuncu, 10'ar jeton, çarpan 1,6) kural "5 jetondan az katana 2 jeton ceza" olsaydı denge katkısı ne olurdu? Ceza 4 jeton olsaydı denge katkısı ve kişi başı kazanç ne olurdu?

Cevap

5 katmak, 0 katmaya göre $5 \cdot 0{,}6 = 3$ jeton kaybettirir. Ceza 2 ise kurtarılan ceza 3'ten azdır: katmamak 1 jeton daha iyidir, denge katkısı 0. Ceza 4 ise $4 > 3$: 5 katmak 1 jeton daha iyidir; 5'ten fazlası jeton başına 0,6 kaybettirir, 1–4 arası hem kaybettirir hem cezadan kurtarmaz. 5 baskındır. Herkes 5 katar: havuz $20 \cdot 1{,}6 = 32$, kişi başı 8, kazanç $10 - 5 + 8 = 13$.

5. Eşikli projede (4 oyuncu, 10'ar jeton, eşik 22, ödül 12) (9, 9, 4, 0) profili Nash dengesi midir? Kazançlar nedir? (5, 5, 5, 5) profili için cevap nedir?

Cevap

(9, 9, 4, 0): toplam 22, kazançlar (13, 13, 18, 22). Katkı yapan biri 1 azaltırsa toplam 21 olur ve eşik kaçar (ör. O3 3 katarsa 7 alır < 18); fazladan katmak boşa ödemedir; O4 katarsa yalnızca kaybeder. Nash dengesidir; toplam 66. (5, 5, 5, 5): toplam 20, eşik kaçar ve herkes 5 alır. O1 7 katarsa toplam 22 olur ve $10 - 7 + 12 = 15 > 5$ alır: denge değildir.

6. İki torbalı şelale modelinde gerçek torba A'dır. Beş kişinin sırayla gördüğü sinyaller a, b, b, b, a. Her

kişinin tahmini ne olur? Beş sinyal birleştirilseydi $P(A)$ kaç olurdu?

Cevap

Kişi 1: a ⇒ A, f = 1. Kişi 2: b görür; öncekiyle birlikte f = 0, $P(A) = 1/2$; eşitlikte kendi sinyalini izler ⇒ B, f = 0. Kişi 3: b ⇒ $P(A) = 1/3$, B, f = −1. Kişi 4: b ⇒ kendi b'siyle f = −2, $P(A) = 1/5$, B; öncekilerden çıkan f = −2 olur ve B şelalesi başlar. Kişi 5: a görür; öncekilerden −2, kendi a'sıyla −1, $P(A) = 1/3 < 1/2$ ⇒ B. Tahminler A, B, B, B, B. Beş sinyal birleştirilseydi 2 a, 3 b ⇒ f = −1, $P(A) = 1/3$; bu örnekte sinyallerin kendisi de yanıltıcıdır ve bilgiyi birleştirmek bile doğru cevabı vermezdi.

7. Her seçmen bağımsız olarak $p = 0{,}8$ olasılıkla doğru oy veriyor. 3 ve 5 kişilik jürilerde çoğunluğun doğru olma olasılığı kaçtır? 5 kişilik jüride dört üye beşincinin oyunu kopyalarsa ne olur?

Cevap

$P_3 = 0{,}8^3 + 3 \cdot 0{,}8^2 \cdot 0{,}2 = 0{,}512 + 0{,}384 = 0{,}896$. $P_5 = 10 \cdot 0{,}8^3 \cdot 0{,}2^2 + 5 \cdot 0{,}8^4 \cdot 0{,}2 + 0{,}8^5 = 0{,}2048 + 0{,}4096 + 0{,}32768 \approx 0{,}942$. Dört üye kopyalarsa beş oy hep aynıdır; çoğunluk tek kişinin oyudur ve olasılık 0,8'e düşer.

8. Üç konumlu bir toplumda politikaların faydaları $P = (9, 4, 2)$, $Q = (4, 4, 4)$, $R = (6, 5, 3)$. Faydacı ölçüt (Laplace), eşitlikçi ölçüt (maksimin) ve Nash sosyal refahı hangi politikayı seçer?

Cevap

Ortalamalar $15/3 = 5$, $12/3 = 4$, $14/3 \approx 4{,}67$ ⇒ faydacı ölçüt P'yi seçer. En kötü konumlar 2, 4, 3 ⇒ maksimin Q'yu seçer. Çarpımlar $9 \cdot 4 \cdot 2 = 72$, $4^3 = 64$, $6 \cdot 5 \cdot 3 = 90$ ⇒ Nash sosyal refahı R'yi seçer. Üç ölçüt üç farklı politika seçer.

12.11 İleri okuma

- D. Kahneman, *Thinking, Fast and Slow* (2011): beklenti kuramını, kayıptan kaçınmayı, çerçevelemeyi ve sezgisel kuralları kuramın kurucularından biri, teknik olmayan bir dille anlatır.
- C. F. Camerer, *Behavioral Game Theory: Experiments in Strategic Interaction* (2003): diktatör, ültimatom ve güven oyunlarından kuantal tepki ve seviye-k modellerine kadar davranışsal oyun kuramının deneysel bulgularını kapsamlı biçimde derler.
- E. Ostrom, *Governing the Commons* (1990): ortak kaynakların yerel yönetimine ilişkin vaka çalışmalarını (Alanya balıkçıları dahil) ve tasarım ilkelerinin ortaya çıkışını ayrıntılı verir.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 16: bilgi şelalelerini bu bölümdeki torba modeliyle, Bayes hesabını adım adım göstererek anlatır.

12.12 Kaynaklar

1. L. R. Anderson, C. A. Holt, "Information cascades in the laboratory", *American Economic Review* 87(5), 1997.
2. D. Austen-Smith, J. S. Banks, "Information aggregation, rationality, and the Condorcet jury theorem", *American Political Science Review* 90(1), 1996.
3. J. Berg, J. Dickhaut, K. McCabe, "Trust, reciprocity, and social history", *Games and Economic Behavior* 10(1), 1995.
4. S. Bikhchandani, D. Hirshleifer, I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades", *J. Political Economy* 100(5), 1992.
5. Marquis de Condorcet, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, 1785.
6. C. Engel, "Dictator games: A meta study", *Experimental Economics* 14(4), 2011.
7. T. Feddersen, W. Pesendorfer, "Convicting the innocent", *American Political Science Review* 92(1), 1998.
8. E. Fehr, S. Gächter, "Cooperation and punishment in public goods experiments", *American Economic Review* 90(4), 2000.
9. E. Fehr, S. Gächter, "Altruistic punishment in humans", *Nature* 415, 2002.
10. E. Fehr, K. M. Schmidt, "A theory of fairness, competition, and cooperation", *Quarterly J. Economics* 114(3), 1999.
11. U. Fischbacher, S. Gächter, E. Fehr, "Are people conditionally cooperative? Evidence from a public goods experiment", *Economics Letters* 71(3), 2001.
12. F. Galton, "Vox populi", *Nature* 75, 1907.
13. J. K. Goeree, C. A. Holt, "Ten little treasures of game theory and ten intuitive contradictions", *American Economic Review* 91(5), 2001.
14. G. Hardin, "The tragedy of the commons", *Science* 162, 1968.
15. J. C. Harsanyi, "Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility", *J. Political Economy* 63, 1955.
16. B. Herrmann, C. Thöni, S. Gächter, "Antisocial punishment across societies", *Science* 319, 2008.
17. D. Kahneman, J. L. Knetsch, R. H. Thaler, "Experimental tests of the endowment effect and the Coase theorem", *J. Political Economy* 98(6), 1990.
18. D. Kahneman, A. Tversky, "Prospect theory: An analysis of decision under risk", *Econometrica* 47(2), 1979.
19. J. O. Ledyard, "Public goods: A survey of experimental research", J. H. Kagel, A. E. Roth (ed.), *The Handbook of Experimental Economics*, Princeton University Press, 1995.
20. R. D. McKelvey, T. R. Palfrey, "Quantal response equilibria for normal form games", *Games and Economic Behavior* 10(1), 1995.
21. R. Nagel, "Unraveling in guessing games: An experimental study", *American Economic Review* 85(5), 1995.
22. E. Ostrom, *Governing the Commons*, Cambridge University Press, 1990.
23. S. E. Page, *The Difference*, Princeton University Press, 2007.

24. M. Rabin, “Incorporating fairness into game theory and economics”, *American Economic Review* 83(5), 1993.

25. J. Rawls, *A Theory of Justice*, Harvard University Press, 1971.

26. H. A. Simon, “A behavioral model of rational choice”, *Quarterly J. Economics* 69, 1955.

27. A. Tversky, D. Kahneman, “The framing of decisions and the psychology of choice”, *Science* 211, 1981.

28. A. Tversky, D. Kahneman, “Advances in prospect theory: Cumulative representation of uncertainty”, *J. Risk and Uncertainty* 5, 1992.

KISIM V

Dinamikler ve Gelecek

Bölüm 13–14

KISIM V · BÖLÜM 13

Öğrenme ve Adaptasyon

Oyun kuramının klasik sorusu “dengede ne olur?” sorusudur. Bu soru, oyuncuların oyunu bildiğini ve dengeyi hesaplayabildiğini varsayar. Gerçekte oyuncular çoğu zaman kazanç tablosunu bilmez, dengeyi hesaplayacak gücü de yoktur; yalnızca oynar, sonucu görür ve bir sonraki seferde davranışını biraz değiştirir. Bu bölüm, böyle basit güncelleme kurallarının uzun vadede nereye gittiğini inceler. En iyi tepki dinamiği ve kurgusal oyun rakibin geçmişine bakar; Hedge algoritması pişmanlığı küçültür; çok kollu haydut algoritmaları keşif ile sömürü arasında denge kurar; Q-öğrenme ödülü durumlar arasında geriye yayar; replikatör dinamiği başarılı stratejinin nüfusta yayılmasını modeller; tekrarlanan Mahkûmun İkilemi de işbirliğinin nasıl öğrenilip korunabildiğini gösterir.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum13`

ÖĞRENME ÇIKTILARI

Dinamikler
En iyi tepki dinamiği ve kurgusal oyunu adım adım çalıştırmak; ne zaman dengeye yakınsadıklarını açıklamak.

Pişmanlık
Dışsal pişmanlığı hesaplamak; çarpımsal ağırlıklar (Hedge) algoritmasının pişmanlıksız olduğunu yorumlamak.

Keşif–sömürü
Çok kollu haydut probleminde ε-açgözlü ve UCB algoritmalarını izlemek ve karşılaştırmak.

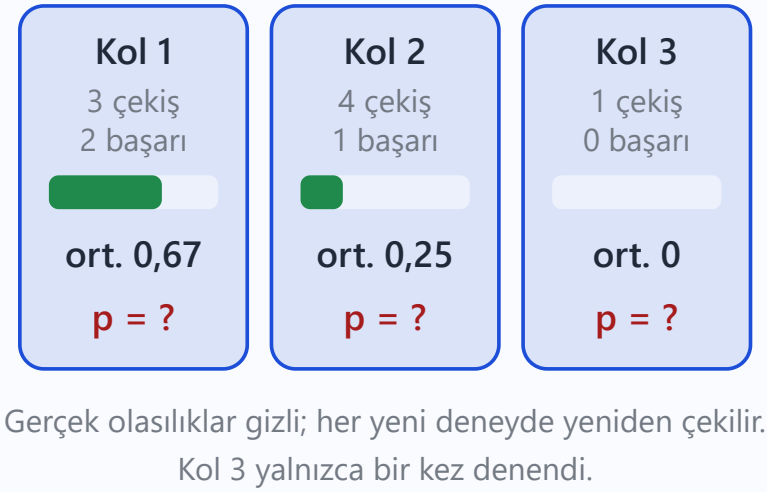
Evrim ve tekrar
Replikatör dinamiğini ve tekrarlanan Mahkûmun İkilemi'nde Tit-for-Tat'ın başarısını açıklamak.

13.1 Giriş

Denge analizi oyunları çoğunlukla "dışarıdan" çözer: kazanç matrisi verilir, denge hesaplanır ve oyuncuların o dengeyi oynayacağı kabul edilir. Bu bölümde bakış açısı değişir. Oyuncuların her biri bir algoritmadır. Her turda bir eylem seçer, sonucu gözler ve kuralını günceller. Soru artık "denge nedir?" değil, "bu güncelleme kuralı uzun vadede nereye gider?" sorusudur.

13.1.1 Bir oyunla başlangıç: çok kollu haydut

Bölümün açılış oyunu bir kumar makinesidir. Makinenin üç kolu vardır. Her kol, gizli bir olasılıkla 1 puan verir, yoksa 0 verir. Bir kolu çekince yalnızca o kolun sonucu görülür; öteki kolların o turda ne vereceği öğrenilemez. Amaç, belirli sayıda çekişte toplam ödülü en büyük yapmaktır. İngilizcede tek kollu kumar makinesine "tek kollu haydut" denir; bu yüzden birden çok kolu olan probleme **çok kollu haydut** (*multi-armed bandit*) adı verilir.



Şekil 13.1: Açılış durumu. Kol 1 üç kez çekilmiş ve iki kez başarı vermiş (ortalama 0,67); Kol 2 dört çekişte bir başarı (0,25); Kol 3 yalnız bir kez çekilmiş ve başarısız olmuş (0). Gerçek olasılıklar gizlidir.

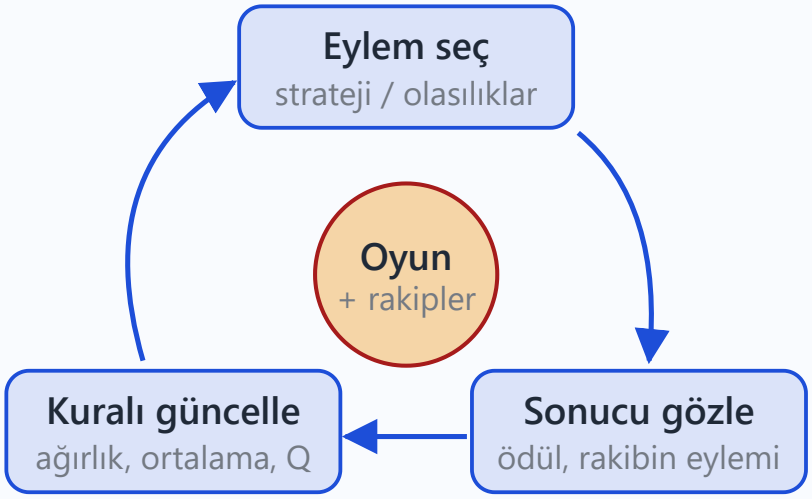
Şekil 13.1'de sekiz çekişten sonraki durum görülür. Kol 1'in gözlenen ortalaması $2/3 \approx 0{,}67$ ile en yüksektir. Kol 3 ise tek denemede başarısız olmuştur. İki makul düşünce çatışır. Birincisine göre en iyi görünen kol Kol 1'dir ve kalan çekişler ona harcanmalıdır. İkincisine göre Kol 3 hakkında neredeyse hiçbir şey bilinmemektedir; gerçek olasılığı yüksek olsa bile tek bir başarısız deneme şaşırtıcı değildir. Kol 3'ü yeniden denemek bilgi kazandırır, ama bu çekiş o anki en iyi tahminden vazgeçmek demektir.

Bu gerilimin adı **keşif–sömürü ikilemi**dir (*exploration–exploitation*). **Keşif**, az bilinen seçenekleri bilgi toplamak için denemektir. **Sömürü**, o ana kadar en iyi görünen seçeneği kullanmaktır. Yalnızca sömüren bir oyuncu yanlış kola takılabilir; yalnızca keşfeden bir oyuncu ise öğrendiğini hiç kullan-

maz. Örneğin yüz çekişlik bir oyunda kaç çekişin keşfe ayrılacağı sorusunun tek bir doğru cevabı yoktur; cevap kolların ne kadar farklı olduğuna ve oyunun ne kadar süreceğine bağlıdır. Bu soruya iki algoritma (ε-açgözlü ve UCB1) sayısal cevaplar verir; açılıştaki durumun çözümü 13.4.5'tedir.

13.1.2 Neden öğrenme ve adaptasyon?

Önce iki terimi ayıralım. **Öğrenme**, geçmiş deneyimden bir strateji geliştirmektir. **Adaptasyon** (uyum), rakip ya da çevre değişince stratejiyi güncellemektir. İkisi çoğu zaman birlikte görülür: bir teklif algoritması dünkü açık artırmalardan öğrenir, bugün rakipler değişince yeniden uyum sağlar.



Şekil 13.2: Öğrenme döngüsü. Oyuncu bir eylem (ya da olasılık dağılımı) seçer, oyun ve rakipler bir sonuç üretir, oyuncu sonucu gözler ve kuralını (ağırlık, ortalama ya da Q değeri) günceller.

Şekil 13.2 bu bölümdeki bütün algoritmaların ortak iskeletini gösterir. Algoritmalar yalnız iki noktada ayrılır: neyi gözledikleri ve gözlemi nasıl kullandıkları. Öğrenme modellerine neden ihtiyaç duyulduğunu üç gözlem açıklar.

1. **Denge hesaplamak zor olabilir.** Genel oyunlarda karışık Nash dengesini bulmak PPAD-tam bir problemdir. PPAD-tam, büyük oyunlarda hızlı (polinom zamanlı) bir çözüm yolu bilinmeyen bir problem sınıfıdır. Bilgisayarlar bile dengeyi hızlı bulamıyorsa, oyuncuların dengeyi kafadan hesapladığını varsaymak zordur. O hâlde oyuncular dengeyi nasıl “bulur”?
2. **Oyuncular oyunu bilmiyor olabilir.** Kazanç matrisi çoğu zaman bilinmez. Bir reklamveren, rakiplerinin değerlemelerini görmez; bir sürücü, öteki sürücülerin rota seçimlerini görmez. Oyuncu yalnızca oynayıp sonucu görür.
3. **Asıl soru basit kuralların gücüdür.** Basit, yerel güncelleme kuralları uzun vadede dengeye, ya da en azından iyi bir performansa götürür mü? Cevap kurala ve oyuna göre değişir: bazı kurallar bazı oyunlarda dengeye gider, bazıları döner, bazıları dengeye gitmese bile “pişmanlıksız” olur.

Örnekler günlük hayattan tanıdıktır. Satranç ve poker yapay zekâları kendilerine karşı milyonlarca oyun oynayarak öğrenir. Reklam açık artırmalarındaki teklif algoritmaları her gün sonuçlara bakıp tekliflerini ayarlar. Navigasyon uygulamaları trafiğe göre rota değiştirir; her sürücünün rotası öteki sürücülerin trafiğini etkilediği için bu da bir oyundur.

13.1.3 Öğrenme modellerine genel bakış

Bölümde altı öğrenme modeli incelenir. Tablo 13.1 bunları iki soruya göre karşılaştırır: oyuncu neyi gözler ve nasıl günceller?

Model	Oyuncu neyi gözler?	Nasıl günceller?
En iyi tepki dinamiği	Rakibin son eylemi	Ona en iyi tepkiyi verir
Kurgusal oyun (*fictitious play*)	Rakibin tüm geçmişi	Frekanslara en iyi tepki
Pişmanlığı en aza indirme (Hedge)	Her eylemin kazancı	İyi eylemlerin ağırlığını çarpımsal artırır
Çok kollu haydut (ε-açgözlü, UCB)	Yalnızca seçtiği eylemin ödülü	Ortalama tahmini + keşif
Pekiştirmeli öğrenme (Q-öğrenme)	Ödül ve yeni durum	$Q(s, a)$ değerlerini günceller
Replikatör dinamiği	Nüfustaki ortalama başarı	Başarılı strateji yayılır

Tablo 13.1: Öğrenme modelleri: oyuncunun gözlediği bilgi ve güncelleme kuralı.

Tablo yukarıdan aşağı okunduğunda bir eğilim görülür. İlk iki modelde oyuncu rakibi izler: rakibin ne oynadığını bilir ve buna göre en iyi tepkiyi hesaplar. Bunun için oyuncunun kendi kazanç tablosunu bilmesi gerekir. Hedge'de oyuncu rakibi değil, kendi eylemlerinin kazancını izler; rakibin ne yaptığını bilmesine gerek kalmaz. Haydut ve Q-öğrenmede bilgi daha da azalır: yalnızca seçilen eylemin ödülü görülür. Son satırda ise bakış bireyden nüfusa kayar; tek bir oyuncunun "düşünmesi" yoktur, başarılı davranış çoğalır.

Modeller arasındaki bir başka fark, **tam geri bildirim** ile **kısmi geri bildirim** ayrımıdır. Tam geri bildirimde oyuncu, oynamadığı eylemlerin de o turda ne kazandıracağını öğrenir. Örneğin bir yatırımcı hangi hisseyi alırsa alsın akşam bütün hisselerin getirisini görür. Kısmi geri bildirimde yalnızca seçilen eylemin sonucu görülür. Örneğin hastasına bir ilaç veren hekim, öteki ilacın o hastada ne yapacağını hiçbir zaman öğrenemez. Hedge tam geri bildirim için, haydut algoritmaları kısmi geri bildirim için tasarlanmıştır; keşif gereksinimi bu farktan doğar.

> **Sık yapılan hata**
>
> "Öğrenme" sözcüğünü yalnızca yapay sinir ağlarıyla özdeşleştirmek. Bu bölümdeki algoritmaların çoğu birkaç satırlık sayaç ve ağırlık güncellemesidir. Öğrenmeyi tanımlayan şey karmaşıklık değil, davranışın gözlenen sonuçlara göre değişmesidir.

13.2 En iyi tepki ve kurgusal oyun

En basit öğrenme kuralları, Nash dengesinin tanımından doğrudan çıkar. Nash dengesi, her oyuncunun ötekilerin seçimine en iyi tepki verdiği profildir. O hâlde oyuncular sırayla en iyi tepki verirse, bir noktada herkesin en iyi tepki verdiği bir profile, yani dengeye varılabilir. Bu kısım, fikrin ne zaman işlediğini, ne zaman döngüye girdiğini ve döngünün geçmişi hatırlayarak nasıl kırılabildiğini inceler.

Bir ön bilgiyi hatırlatalım. Oyuncu i'nin, ötekilerin s_{-i} seçimine **en iyi tepkisi** (*best response*), kazancını en yüksek yapan stratejidir: $BR_i(s_{-i}) = \operatorname{argmax}_{s_i} u_i(s_i, s_{-i})$. İki kişilik bir matris oyununda satır oyuncusunun en iyi tepkisi, rakibin sütununda ilk sayıların en büyüğünü veren satırdır; sütun oyuncusunun en iyi tepkisi, rakibin satırında ikinci sayıların en büyüğünü veren sütundur. Bir profil, iki oyuncu da birbirine en iyi tepki veriyorsa Nash dengesidir.

13.2.1 En iyi tepki dinamiği adım adım

Tanım 13.1

En iyi tepki dinamiği (*best-response dynamics*): bir başlangıç profilinden başlanır. Oyuncular sırayla, rakiplerinin *şu anki* eylemine en iyi tepki verecek biçimde kendi eylemlerini değiştirir. Hiçbir oyuncu eylemini değiştirmek istemediğinde dinamik durur.

Dinamik durduğu anda her oyuncu ötekilerin eylemine en iyi tepki vermektedir; bu, Nash dengesinin tanımıdır. Demek ki en iyi tepki dinamiği *durursa* bir saf Nash dengesinde durur. Asıl soru, dinamiğin durup durmadığıdır. Önce durduğu bir örneğe bakalım.

	X	**Y**	**Z**
A	**(4, 3)**	(0, 2)	(1, 1)
B	(2, 1)	**(3, 3)**	(0, 2)
C	(1, 4)	(2, 0)	(3, 2)

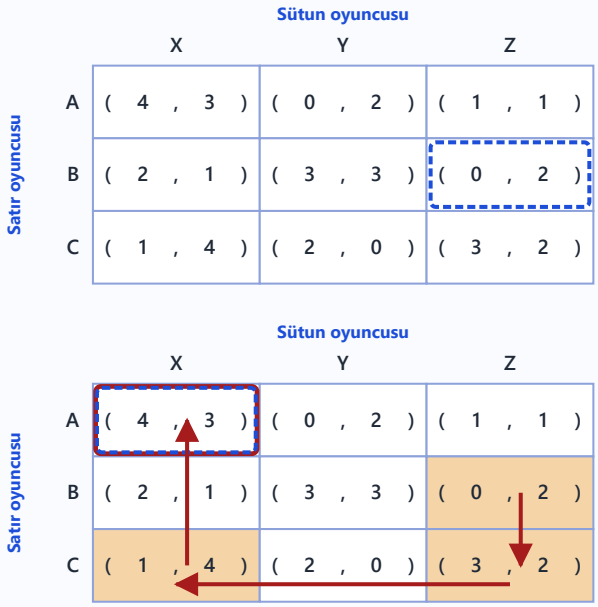
Tablo 13.2: Üçe üç örnek oyun. Hücrede (u_1, u_2): önce satır, sonra sütun oyuncusunun kazancı. Yeşil hücreler iki saf Nash dengesidir.

Örnek 13.1

Tablo 13.2'deki oyunda dinamik (B, Z) profilinden başlasın. Önce satır oyuncusu, sonra sütun oyuncusu hareket etsin.

1. **Satır oyuncusu Z'ye karşı:** Z sütununda ilk sayılar A: 1, B: 0, C: 3'tür. En büyüğü 3 olduğundan satır oyuncusu C'ye geçer. Profil (C, Z) olur.
2. **Sütun oyuncusu C'ye karşı:** C satırında ikinci sayılar X: 4, Y: 0, Z: 2'dir. En büyüğü 4 olduğundan sütun oyuncusu X'e geçer. Profil (C, X) olur.
3. **Satır oyuncusu X'e karşı:** X sütununda ilk sayılar A: 4, B: 2, C: 1'dir. Satır oyuncusu A'ya geçer. Profil (A, X) olur.
4. **Sütun oyuncusu A'ya karşı:** A satırında ikinci sayılar X: 3, Y: 2, Z: 1'dir. Sütun oyuncusu zaten X'i seçmektedir (3 > 2, 1); değiştirmez.
5. **Satır oyuncusu yeniden X'e karşı:** A zaten en iyisidir (4 > 2, 1). Kimse değiştirmek istemez; dinamik (A, X)'te durur ve (A, X) bir Nash dengesidir.

Aynı dinamik (C, Y)'den başlasaydı sonuç farklı olurdu. Satır oyuncusu Y'ye karşı A: 0, B: 3, C: 2 görür ve B'ye geçer. Sütun oyuncusu B'ye karşı X: 1, Y: 3, Z: 2 görür ve Y'de kalır. Dinamik (B, Y) dengesinde durur.



Şekil 13.3: En iyi tepki dinamiği. Solda başlangıç profili (B, Z) kesikli çerçeveyle gösterilir. Sağda dinamiğin izlediği yol: (B, Z) → (C, Z) → (C, X) → (A, X). Turuncu hücreler yol üzerindeki profillerdir; dinamik (A, X) Nash dengesinde durur.

Şekil 13.3'te okların her biri tek bir oyuncunun değişimidir. Dikey oklar satır oyuncusunun (yalnız satır değişir, sütun sabit), yatay oklar sütun oyuncusunun hamlesidir. Örnek iki önemli gözlem verir. Birincisi, dinamik bir dengede durdu; her adımda yalnızca bir oyuncunun yerel bir hesap yapması yetti, kimse dengeyi "çözmedi". İkincisi, *hangi* dengeye varıldığı başlangıca bağlıdır. Birden çok denge olan oyunlarda öğrenme kuralı, bir denge seçim mekanizması gibi çalışır: (B, Z)'den başlayan dinamik (A, X)'e, (C, Y)'den başlayan dinamik (B, Y)'ye yerleşir.

Sık yapılan hata

- **İki oyuncuyu aynı anda güncellemek:** Bu örnekte oyuncular sırayla hareket eder. İkisi aynı anda en iyi tepki verseydi farklı bir yol izlenirdi; örneğin (B, Z)'de satır C'ye, sütun da B'ye karşı en iyi tepki olan Y'ye aynı anda geçer ve (C, Y)'ye varılır.
- **Yanlış sayıya bakmak:** Satır oyuncusu her zaman ilk sayıyı, sütun oyuncusu ikinci sayıyı büyütmek ister. Sütun oyuncusunun hesabında satır oyuncusunun kazancına bakmak en sık hatadır.

13.2.2 En iyi tepki dinamiği ne zaman yakınsar?

Her oyunda dinamik durmaz. En basit karşı örnek **Yazı-Tura Eşleştirme** oyunudur. İki oyuncu aynı anda bir madeni paranın Yazı (Y) ya da Tura (T) yüzünü gösterir. Yüzler aynıysa *Eşleştiren* kazanır (1 alır, rakip −1); farklıysa *Ayrıştıran* kazanır.

Eşl. \ Ayr.	Y	T
Y	(1, −1)	(−1, 1)
T	(−1, 1)	(1, −1)

Tablo 13.3: Yazı-Tura Eşleştirme. Satır: Eşleştiren, sütun: Ayrıştıran. Hücrede (Eşleştiren, Ayrıştıran) kazancı.

Örnek 13.2

1. (Y, Y)'de Eşleştiren kazanmaktadır ve memnundur. Ayrıştıran farklı yüz ister ve T'ye geçer: (Y, T).
2. (Y, T)'de Ayrıştıran memnundur. Eşleştiren aynı yüzü ister ve T'ye geçer: (T, T).
3. (T, T)'de Ayrıştıran Y'ye geçer: (T, Y).
4. (T, Y)'de Eşleştiren Y'ye geçer: yeniden (Y, Y). Döngü baştan başlar.

(Y, Y) → (Y, T) → (T, T) → (T, Y) → (Y, Y) → … dizisi sonsuza kadar sürer.



Şekil 13.4: Yazı-Tura Eşleştirme'de en iyi tepki dinamiği. Oklar dört profil arasında saat yönünde bir döngü oluşturur; hiçbir hücrede durulmaz.

Şekil 13.4'te görüldüğü gibi her adımda bir oyuncu mutlu, öteki mutsuzdur. Oyunun saf Nash dengesi yoktur; dört hücrenin her birinde kaybeden oyuncunun sapma isteği vardır. Bu yüzden dinamik hiçbir hücrede duramaz. Oyunun tek dengesi karışıktır: iki oyuncu da Y ve T'yi ½ olasılıkla oynar. En iyi tepki dinamiği ise yalnızca saf eylemler arasında atlar; hiçbir zaman “yarı yarıya” bir davranış üretmez, bu yüzden karışık dengeye de varamaz.

Dinamiğin hangi oyunlarda kesinlikle durduğu sorusunun iyi bilinen bir cevabı vardır: **potansiyel oyunlar** (*potential games*).

Tanım 13.2

Bir oyun, bütün oyuncular için ortak bir Φ fonksiyonu bulunabiliyorsa **potansiyel oyundur**: herhangi bir oyuncu tek başına eylemini değiştirdiğinde kendi kazancındaki değişim, Φ'deki değişime eşittir.

$$u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \Phi(s_i', s_{-i}) - \Phi(s_i, s_{-i})$$

Potansiyel fonksiyonu, bütün oyuncuların bencil hesaplarını tek bir “ortak puan tablosunda” toplar. Bir oyuncunun kârlı sapması, ortak puanı da aynı miktarda artırır. Tıkanıklık oyunları, yani her sürücünün bir yol seçtiği ve yolun maliyetinin onu kullanan sürücü sayısına bağlı olduğu trafik oyunları, potansiyel oyunların en önemli örneğidir (Monderer ve Shapley, 1996).

Teorem 13.1 (Potansiyel oyunlarda yakınsama)

Sonlu bir potansiyel oyunda en iyi tepki dinamiği (ve daha genel olarak her adımda kazancı kesin artıran herhangi bir tek taraflı değişim dizisi) sonlu sayıda adımda bir saf Nash dengesinde durur.

İspat. Dinamikte her adım bir *kârlı* sapmadır: hareket eden oyuncunun kazancı kesin olarak artar. Potansiyel tanımına göre bu artış Φ'deki artışa eşittir; demek ki her adımda Φ kesin olarak büyür. Oyun sonlu olduğu için strateji profillerinin sayısı sonludur, dolayısıyla Φ'nin alabileceği değerler de sonludur. Kesin artan bir dizi sonlu bir küme içinde sonsuza kadar süremez; aynı profile iki kez dönmek de olanaksızdır, çünkü dönüldüğünde Φ eski değerini alırdı. Bu yüzden dinamik sonlu adımda durur. Durduğu profilde hiçbir oyuncunun kârlı sapması yoktur; bu profil bir saf Nash dengesidir. □

İspattan bir yan sonuç çıkar: Yazı-Tura Eşleştirme potansiyel oyun olamaz. Olsaydı, döngüdeki dört adımın her biri Φ'yi artırırdı ve dört adım sonra aynı profile dönüldüğünde Φ başladığı değerden büyük olurdu; bu bir çelişkidir. Döngüye giren bir dinamik, potansiyel bulunmadığının kanıtıdır.

Örnek 13.3

İki arkadaş aynı yazılımı kullanmak ister. İkisi de A'yı seçerse 2'şer, ikisi de B'yi seçerse 1'er kazanır; farklı seçerlerse 0 alırlar. Bu bir koordinasyon oyunudur ve potansiyeli $\Phi(A, A) = 2$, $\Phi(B, B) = 1$, $\Phi(A, B) = \Phi(B, A) = 0$ olarak seçilebilir: her tek taraflı değişimde değişen oyuncunun kazanç farkı Φ farkına eşittir.

1. Başlangıç (A, B), $\Phi = 0$. Satır oyuncusu B'ye karşı A'da 0, B'de 1 alır ve B'ye geçer: (B, B), $\Phi = 1$.
2. (B, B)'de sütun oyuncusu B'de 1, A'da 0 alır; değiştirmez. Satır oyuncusu da değiştirmez. Dinamik (B, B) dengesinde durur.

Φ 0'dan 1'e çıktı ve orada kaldı. Dinamik daha yüksek potansiyelli (A, A) dengesine gitmedi: yakınsama garantisi bir dengeye varılacağını söyler, en iyi dengeye varılacağını söylemez.

Saf dengesi olmayan oyunlar için çare, yalnız son eyleme değil, rakibin bütün geçmişine tepki vermektir. Bu fikir kurgusal oyunu doğurur.

13.2.3 Kurgusal oyun: Yazı-Tura Eşleştirme

Tanım 13.3

Kurgusal oyun (*fictitious play*): her turda oyuncu, rakibin o ana kadarki eylemlerinin frekanslarını sayar ve rakibin bu frekanslarla oynayan bir karışık strateji kullandığını "kurgular". Sonra bu karışık stratejiye en iyi tepkiyi oynar.

Adındaki "kurgu", oyuncunun rakibi hakkında yaptığı basitleştirici varsayımdır: rakip sanki sabit bir zarla oynuyormuş, zarın olasılıkları da geçmiş frekanslar kadarmış gibi davranılır. Örneğin rakip yedi turun beşinde T oynadıysa, onun $P(Y) = 2/7$, $P(T) = 5/7$ ile oynadığı varsayılır. En iyi tepki dinamiğinden farkı, tek bir son gözlem yerine bütün geçmişin kullanılmasıdır. Bu yüzden kurgusal oyunun inancı yavaş değişir: tek bir yeni gözlem, uzun bir geçmişin içinde küçük bir ağırlık taşır.

Yazı-Tura Eşleştirme'de bu kural çok sade bir hâl alır. Eşleştiren aynı yüzü istediği için, rakibin geçmişinde hangi yüz çoğunluktaysa onu oynar. Ayrıştıran farklı yüz istediği için, rakibin geçmişindeki çoğunluğun tersini oynar. Eşitlik olursa ikisi de Y oynar. İlk turda geçmiş olmadığı için ikisi de keyfî olarak Y oynar.

Örnek 13.4

"Çoğunluğu eşle" kuralının bir en iyi tepki olduğu beklenen kazançla doğrulanabilir. 8. turda Eşleştiren, Ayrıştıran'ın önceki yedi turda 2 kez Y, 5 kez T oynadığını görür; inancı $P(Y) = 2/7$'dir.

1. Eşleştiren Y oynarsa: $2/7 \cdot 1 + 5/7 \cdot (-1) = -3/7$.
2. Eşleştiren T oynarsa: $2/7 \cdot (-1) + 5/7 \cdot 1 = 3/7$.
3. $3/7 > -3/7$ olduğundan en iyi tepki T'dir; T, rakibin geçmişindeki çoğunluktur.

Genel olarak inanç $P(Y) = q$ ise Y'nin beklenen kazancı $2q - 1$, T'ninki $1 - 2q$'dur. $q > 1/2$ ise Y, $q < 1/2$ ise T daha iyidir; $q = 1/2$'de ikisi eşittir ve kural Y'yi seçer.

Kural on tur boyunca uygulandığında Tablo 13.4 elde edilir. Tabloda "geçmiş" sütunları o turdan *önceki* turları sayar; "oran" sütunları ise o turu da içerir.

t	Ayr. geçmişi Y/T	Eşleştiren	Eşl. geçmişi Y/T	Ayrıştıran	Eşl. Y oranı	Ayr. Y oranı
1	0 / 0	Y	0 / 0	Y	1,00	1,00
2	1 / 0	Y	1 / 0	T	1,00	0,50
3	1 / 1	Y	2 / 0	T	1,00	0,33
4	1 / 2	T	3 / 0	T	0,75	0,25
5	1 / 3	T	3 / 1	T	0,60	0,20
6	1 / 4	T	3 / 2	T	0,50	0,17
7	1 / 5	T	3 / 3	Y	0,43	0,29
8	2 / 5	T	3 / 4	Y	0,38	0,38
9	3 / 5	T	3 / 5	Y	0,33	0,44
10	4 / 5	T	3 / 6	Y	0,30	0,50

Tablo 13.4: Kurgusal oyun, Yazı-Tura Eşleştirme, on tur. "Ayr. geçmişi" Eşleştiren'in gördüğü, "Eşl. geçmişi" Ayrıştıran'ın gördüğü Y/T sayılarıdır (önceki turlar). Oranlar o tur dahil Y oynama oranıdır.

Tablonun ilk birkaç satırı ayrıntılı okunmaya değer.

- **t = 1:** Geçmiş yoktur; iki oyuncu da keyfî olarak Y oynar.
- **t = 2:** Eşleştiren rakip geçmişinde 1 / 0 görür; çoğunluk Y'dir, Y oynar. Ayrıştıran da rakip geçmişinde 1 / 0 görür; çoğunluğun tersi olan T'yi oynar.
- **t = 3:** Eşleştiren 1 / 1 görür; eşitlikte Y oynar. Ayrıştıran 2 / 0 görür ve T oynar.

- **t = 4:** Eşleştiren 1 / 2 görür; çoğunluk artık T'dir ve Eşleştiren T'ye geçer. Ayrıştıran 3 / 0 görür, T'de kalır. Eşleştiren'in Y oranı *3/4 = 0,75*'e düşer.
- **t = 7:** Ayrıştıran 3 / 3 görür; eşitlikte Y'ye geçer. Böylece Eşleştiren'in T'ye kaymasına üç tur gecikmeyle cevap verir.

Eylemler hâlâ döner: önce (Y, Y), sonra (Y, T), (T, T) ve (T, Y). Ancak en iyi tepki dinamiğinden önemli bir fark vardır. Her aşama birkaç tur sürer, çünkü inancın tersine dönmesi için geçmişteki çoğunluğun değişmesi gerekir. Oranlar ise sıçramaz, yavaş yavaş değişir: on turun sonunda Eşleştiren'in Y oranı 0,30, Ayrıştıran'ınki 0,50'dir.

Sık yapılan hata

- **Kendi geçmişine bakmak:** Kurgusal oyunda oyuncu *rakibin* geçmişini sayar. Eşleştiren kendi Y sayısına göre karar vermez.
- **Bu turu geçmişe katmak:** t. turun kararı, t. turdan önceki turlara dayanır. Bu turun eylemi henüz bilinmemektedir.
- **Eylemlerin dengeye gittiğini sanmak:** Dengeye yaklaşan şey *frekanslardır*; her turda oynanan eylem hâlâ saf bir Y ya da T'dir.

13.2.4 Kod izleme: fictitious()

Aşağıdaki fonksiyon, Yazı-Tura Eşleştirme'de kurgusal oyunu *T* tur oynatır. Eylemler sayıyla kodlanır: 0 = Y, 1 = T.

```
def fictitious(T):
    ne, na = [0, 0], [0, 0]   # Y/T sayaçları
    a, b = 0, 0               # 1. tur: ikisi de Y
    for t in range(1, T + 1):
        ne[a] += 1            # eşleştirenin geçmişi
        na[b] += 1            # ayrıştıranın geçmişi
        # eşleştiren: ayrıştıranın çoğunluğunu eşle
        a = 0 if na[0] >= na[1] else 1
        # ayrıştıran: eşleştirenin çoğunluğundan ayrış
        b = 1 if ne[0] > ne[1] else 0
    return ne, na
```

Kodun parçaları şöyledir.

- `ne` ve `na` sayaç listeleridir. `ne[0]` Eşleştiren'in o ana kadar kaç kez Y, `ne[1]` kaç kez T oynadığını tutar; `na` aynı bilgiyi Ayrıştıran için tutar.
- `a` ve `b` bu turda oynanacak eylemlerdir; ilk turda ikisi de 0 (Y) olarak başlatılır.
- Döngünün başında bu turun eylemleri sayaçlara eklenir; yani tur oynanmış olur.

- Ardından bir sonraki turun eylemleri hesaplanır. Eşleştiren, Ayrıştıran'ın Y sayısı T sayısından küçük değilse (eşitlik dahil) Y oynar: `a = 0 if na[0] >= na[1] else 1`.
- Ayrıştıran, Eşleştiren'in Y sayısı T sayısından kesin büyükse T oynar, yoksa Y: `b = 1 if ne[0] > ne[1] else 0`. Eşitlikte koşul yanlış olur ve Y seçilir; bu, tablodaki "eşitlikte Y" kuralıdır.

Çağrı `fictitious(6)` olsun. Tablo 13.5 her turun sonunda sayaçların durumunu ve bir sonraki turun kararını gösterir.

Tur	Oynanan (a, b)	Kazanan	ne	na	Sonraki tur
1	(Y, Y)	Eşleştiren	1 / 0	1 / 0	(Y, T)
2	(Y, T)	Ayrıştıran	2 / 0	1 / 1	(Y, T)
3	(Y, T)	Ayrıştıran	3 / 0	1 / 2	(T, T)
4	(T, T)	Eşleştiren	3 / 1	1 / 3	(T, T)
5	(T, T)	Eşleştiren	3 / 2	1 / 4	(T, T)
6	(T, T)	Eşleştiren	3 / 3	1 / 5	—

Tablo 13.5: `fictitious(6)` izlemesi. Sayaçlar tur oynandıktan sonraki değerlerdir (Y/T).

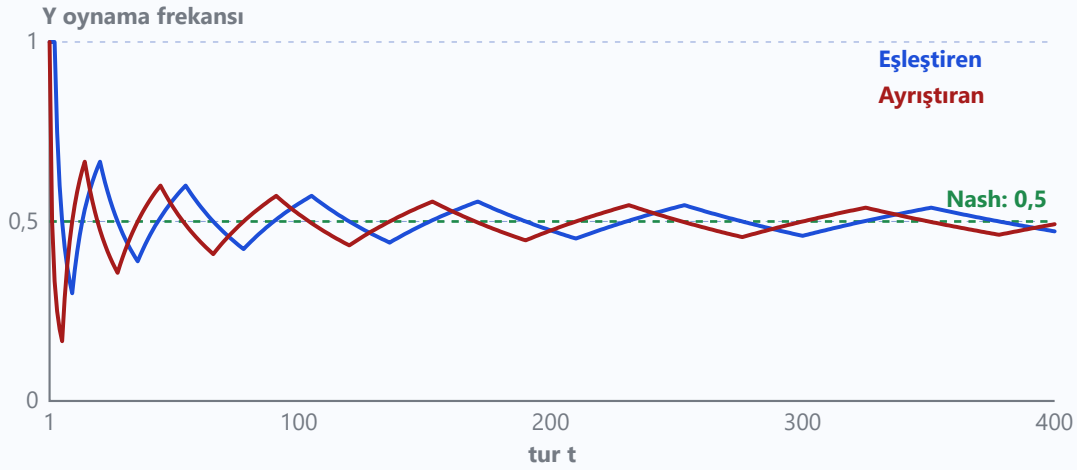
İzlemenin kritik anları 2. ve 3. turlardır. 2. tur sonunda Ayrıştıran'ın sayaçları 1 / 1 eşittir; Eşleştiren eşitlikte Y oynamayı sürdürür. 3. tur sonunda Ayrıştıran'ın sayaçları 1 / 2 olur ve Eşleştiren T'ye geçer. Ayrıştıran ise 6. tur sonuna kadar Eşleştiren'in geçmişinde Y çoğunluğunu görür ve T oynamaya devam eder. Fonksiyon `ne = [3, 3]`, `na = [1, 5]` döndürür. Y oranları *3/6 = 0,50* ve *1/6 ≈ 0,17*'dir; bunlar Tablo 13.4'ün 6. satırıyla aynıdır.



Şekil 13.5: `fictitious(6)` çağrısının sonu. Solda 6. turda oynanan (T, T) hücresi, sağda iki oyuncunun Y/T sayaçları: Eşleştiren 3 / 3 (Y oranı 0,50), Ayrıştıran 1 / 5 (Y oranı 0,17).

13.2.5 Uzun vade: frekanslar dengeye yakınsar

Kurgusal oyun dört yüz tur boyunca sürdürüldüğünde eylemler döngüler hâlinde değişmeye devam eder: (Y, Y) → (Y, T) → (T, T) → (T, Y) → (Y, Y) … Ancak her döngü bir öncekinden uzundur. Geçmiş büyüdükçe çoğunluğu tersine çevirmek için daha çok tur gerekir. Frekanslar ise karışık dengeye, yani ½'ye yaklaşır: dört yüz turda Eşleştiren'in Y oranı 0,472, Ayrıştıran'ınki 0,492'dir.



Şekil 13.6: Kurgusal oyunda dört yüz tur boyunca Y oynama frekansı. Mavi: Eşleştiren, kırmızı: Ayrıştıran. Salınımların genliği küçülür ve iki eğri Nash dengesi değeri 0,5'e yaklaşır.

Şekil 13.6'da salınımların neden söndüğü sezilebilir. Bir oyuncunun frekansı ½'den uzaklaştığında, rakibin en iyi tepkisi onu cezalandıracak yöne döner ve bu sapma bir süre sonra düzelir. Örneğin Eşleştiren'in Y oranı ½'nin üstündeyken Ayrıştıran T oynar; Eşleştiren de sonunda T'ye geçer ve kendi Y oranını düşürür. Geçmiş uzadıkça tek bir turun frekansa etkisi $1/t$ kadar küçülür, bu yüzden salınımın genliği de küçülür.

Bu davranış her oyunda görülmez. Kurgusal oyunda frekansların dengeye yakınsaması şu oyun sınıflarında kanıtlanmıştır:

- **İki kişilik sıfır toplamlı oyunlar** (Robinson, 1951). Yazı-Tura Eşleştirme bu sınıftadır.
- **2×2 oyunlar**, yani iki oyuncunun ikişer stratejisi olan oyunlar (Miyasawa, 1961).
- **Potansiyel oyunlar** (Monderer ve Shapley, 1996).

Buna karşılık Shapley (1964), 3×3'lük genel toplamlı bir oyun kurdu. Bu oyunda eylemler giderek uzayan döngülerle döner ve frekanslar hiçbir dengeye yaklaşmaz. Shapley örneği, "her oyunda öğrenme dengeye götürür" beklentisinin yanlış olduğunu gösteren klasik sonuçtur.

Sık yapılan hata

"Frekanslar dengeye yakınsıyor, demek ki oyuncular dengeyi oynuyor" sonucunu çıkarmak. Her turda oynanan eylemler saftır ve döngü sürer; yalnızca uzun dönem ortalaması (½, ½)'ye yaklaşır. Bir gözlemci yalnızca son turlara bakarsa denge değil, uzun bir Y ya da T serisi görür.

13.3 Pişmanlık ve Hedge

En iyi tepki dinamiği ve kurgusal oyun, oyuncunun kazanç tablosunu bildiğini ve rakibin eylemlerini gözlediğini varsayar. Burada bu varsayım zayıflatılır. Oyuncu rakibi modellemez; yalnızca her turdan sonra kendi eylemlerinin ne kazandırdığını (ya da kazandıracağını) görür. Rakip bir insan, bir algoritma ya da kötü niyetli bir düşman olabilir. Bu koşullarda "iyi oynamak" ne demektir? Cevap, pişmanlık kavramıdır.

13.3.1 Dışsal pişmanlık

Gündelik dilde pişmanlık, geriye bakınca "keşke şunu yapsaydım" demektir. Öğrenme kuramındaki tanım bu duygunun sayısal hâlidir. Oyun bittikten sonra geriye bakılır ve şu soru sorulur: baştan sona hep aynı eylem oynansaydı, en iyi durumda ne kazanılırdı? Bu sayı ile gerçekte kazanılan arasındaki fark pişmanlıktır.

Tanım 13.4

T tur boyunca t. turda i eyleminin kazancı $g_t(i)$, algoritmanın seçtiği eylem a_t olsun. **Dışsal pişmanlık** (*external regret*), geriye dönük en iyi sabit eylemin toplamı ile algoritmanın toplamı arasındaki farktır:

$$R_T = \max_i \sum_{t=1}^{T} g_t(i) - \sum_{t=1}^{T} g_t(a_t)$$

Algoritma rastgele seçim yapıyorsa ikinci toplam beklenen kazançla hesaplanır. T büyüdükçe $R_T/T \to 0$ oluyorsa algoritma **pişmanlıksızdır** (*no-regret*).

Tanımın üç ayrıntısı önemlidir. Birincisi, karşılaştırma *sabit* bir eylemle yapılır; her turda sonradan en iyisini seçen "kâhinle" yapılmaz. Kâhinle karşılaştırma hiçbir algoritmanın kazanamayacağı bir yarış olurdu. İkincisi, en iyi sabit eylem ancak oyun bittikten sonra bilinir; algoritma oyun sırasında bunu bilmez. Üçüncüsü, pişmanlıksızlık ortalama bir ölçüttür. R_T büyüyebilir, ama T'den yavaş büyümelidir. Örneğin $R_T = 2\sqrt{T}$ ise $R_T/T = 2/\sqrt{T}$ sıfıra gider; tur başına kayıp zamanla yok olur. $R_T = T/2$ ise tur başına kayıp ½'de takılı kalır.

Örnek 13.5

İki eylemli üç turluk bir oyunda kazançlar $g_1 = (1, 0)$, $g_2 = (0, 1)$, $g_3 = (1, 0)$ olsun. Bir algoritma sırasıyla B, A, A oynasın.

1. Algoritmanın toplamı: 1. turda B ile 0, 2. turda A ile 0, 3. turda A ile 1. Toplam 1.
2. Sabit eylemlerin toplamı: hep A ile $1 + 0 + 1 = 2$, hep B ile $0 + 1 + 0 = 1$. En iyisi 2'dir.
3. Dışsal pişmanlık $R_3 = 2 - 1 = 1$.

Her turda en iyisini seçen bir kâhin 3 kazanırdı; ama pişmanlık tanımında bu 3 değil, en iyi sabit eylemin 2'si kullanılır.

13.3.2 "Lideri izle" neden başarısız olur?

Pişmanlığı küçültmek için akla gelen ilk kural şudur: o ana kadar toplam kazancı en büyük olan eylemi oyna. Bu kurala **lideri izle** (*follow the leader*) denir. Kurgusal oyuna çok benzer; geçmişe en iyi tepkidir. Ne yazık ki kötü seçilmiş bir kazanç dizisine karşı bu kural felakettir. Tablo 13.6 bunun bir örneğidir; eşitlikte A seçilir.

t	Önceki toplam A / B	g(A)	g(B)	Lideri izle	Kazanç
1	0,0 / 0,0	0,5	0,0	A	0,5
2	0,5 / 0,0	0,0	1,0	A	0,0
3	0,5 / 1,0	1,0	0,0	B	0,0
4	1,5 / 1,0	0,0	1,0	A	0,0
5	1,5 / 2,0	1,0	0,0	B	0,0
6	2,5 / 2,0	0,0	1,0	A	0,0

Tablo 13.6: Lideri izle, iki eylem, altı tur. "Önceki toplam", o turdan önceki birikmiş kazançlardır.

Örnek 13.6

1. **t = 1:** Toplamlar eşit (0 / 0); eşitlikte A seçilir. A o tur 0,5 verir. Algoritmanın toplamı 0,5.
2. **t = 2:** Lider A'dır (0,5 > 0). Ama bu tur A 0, B 1 verir; algoritma 0 alır.
3. **t = 3:** Toplamlar 0,5 / 1,0; lider B'dir. Bu tur B 0, A 1 verir; algoritma yine 0 alır.
4. **t = 4, 5, 6:** Aynı örüntü sürer. Lider her turda tam o turun kaybedenidir; algoritmanın toplamı 0,5'te kalır.
5. **Sabit eylemler:** hep A ile $0{,}5 + 0 + 1 + 0 + 1 + 0 = 2{,}5$; hep B ile $0 + 1 + 0 + 1 + 0 + 1 = 3$. En iyi sabit eylem B'dir.
6. **Pişmanlık:** $R_6 = 3 - 0{,}5 = 2{,}5$, yani yaklaşık $T/2$.

Dizinin hilesi, ilk turdaki 0,5'lik küçük farktır. Bu fark liderin her turda "bir adım geride" kalmasını sağlar: kazançlar 1 ve 0 arasında gidip geldiği için bir önceki turda öne geçen eylem, bu tur 0 verir. Dizi uzatıldıkça algoritmanın toplamı 0,5'te kalır, en iyi sabit eylemin toplamı ise yaklaşık $T/2$ olur. Pişmanlık doğrusal büyür: $R_T/T \to 1/2$, sıfıra gitmez. Lideri izle pişmanlıksız değildir.

Sorunun kaynağı kuralın **deterministik** olmasıdır. Kuralı bilen bir rakip, algoritmanın hangi eylemi seçeceğini önceden hesaplayabilir ve o eyleme 0, ötekine 1 kazanç verebilir. Bu durumda algoritma her tur 0 alır. İki eylemin toplam kazancı her turda 1 olduğu için T turda iki eylemin toplamları T eder

ve bunlardan büyük olanı en az $T/2$'dir. Demek ki *her* deterministik algoritmanın kötü durumda pişmanlığı en az $T/2$'dir. Pişmanlıksız olmanın tek yolu rastgele oynamaktır: rakip olasılıkları bilse bile hangi eylemin çıkacağını bilemez.

Sık yapılan hata

"Geçmişte en iyi olan gelecekte de en iyidir" varsayımı. Kazançlar sabit bir dağılımdan geliyorsa lideri izle iyi çalışır. Oysa oyunlarda kazançları rakipler belirler. Kazançları bir rakip seçiyorsa geçmişe tam güvenmek, sömürülmeye davetiye çıkarmaktır.

13.3.3 Çarpımsal ağırlıklar (Hedge)

Lideri izlemenin iki zayıflığı vardır: kesin seçim yapar ve küçük bir farka tam güvenir. **Çarpımsal ağırlıklar** (*multiplicative weights*) algoritması, ya da yaygın adıyla **Hedge**, iki zayıflığı birden giderir. Liderin seçilme olasılığı yüksektir ama kesin değildir; fark küçükse olasılıklar da birbirine yakın kalır.

Tanım 13.5

Hedge algoritması, n eylemin her birine bir ağırlık atar ve $\eta > 0$ **öğrenme oranı** (*learning rate*) parametresini kullanır:

1. Başlangıçta her eyleme aynı ağırlık verilir: $w_i = 1$.
2. Her turda eylem i, $p_i = w_i / \Sigma_j w_j$ olasılığıyla oynanır.
3. Tur sonunda bütün eylemlerin kazancı görülür ve ağırlıklar güncellenir: $w_i \leftarrow w_i \cdot e^{\eta \cdot g_t(i)}$.

Güncelleme kuralı çarpımsaldır: kazancı 1 olan eylemin ağırlığı e^{η} katına çıkar, kazancı 0 olan eylemin ağırlığı değişmez. Örneğin $\eta = \ln 2$ seçilirse $e^{\eta} = 2$ olur ve kazancı 1 olan eylemin ağırlığı tam iki katına çıkar. $\eta = 0{,}5$ için çarpan $e^{0{,}5} \approx 1{,}65$'tir.

Güncellemeler üst üste uygulandığında ağırlık, eylemin toplam kazancının üstel bir fonksiyonu olur. G_i, i eyleminin o ana kadarki toplam kazancı ise $w_i = e^{\eta \cdot G_i}$'dir, çünkü $e^{\eta \cdot g_1} \cdot e^{\eta \cdot g_2} \cdot \ldots = e^{\eta \cdot (g_1 + g_2 + \ldots)}$'dir. Bu yazım, Hedge'in "yumuşak" bir lideri izle olduğunu açıkça ortaya koyar. Liderin G_i değeri en büyük olduğu için en büyük olasılığı o alır. Ama iki eylemin toplamları arasındaki fark Δ ise olasılıklarının oranı $e^{\eta \cdot \Delta}$'dır; küçük fark küçük üstünlük demektir. η çok büyürse oran sonsuza gider ve Hedge lideri izlemeye dönüşür; η sıfıra yaklaşırsa bütün olasılıklar eşitlenir ve Hedge rastgele oynamaya dönüşür.

Teorem 13.2 (Hedge'in pişmanlık sınırı)

Kazançlar *[0, 1]* aralığında ve *n* eylem varsa Hedge'in dışsal pişmanlığı

$$R_T \le (\ln n)/\eta + \eta\, T$$

ile sınırlıdır. $\eta = \sqrt{(\ln n / T)}$ seçilirse $R_T \le 2\sqrt{(T \ln n)}$ olur; bu yüzden $R_T/T \to 0$'dır ve Hedge pişmanlıksızdır.

Sınırdaki $\ln n$, n'nin doğal logaritmasıdır; $n = 3$ için $\ln 3 \approx 1{,}1$'dir. Logaritma çok yavaş büyür: eylem sayısı bin olsa bile $\ln 1000 \approx 6{,}9$'dur. Bu, Hedge'in çok sayıda eylem arasında bile hızlı öğrenebildiğini gösterir.

Sınırın iki terimi η'nın iki ayrı etkisini ölçer. İlk terim $(\ln n)/\eta$, **öğrenmenin yavaşlığının** bedelidir: η küçükse algoritma en iyi eylemi geç fark eder. İkinci terim ηT, **aşırı tepkinin** bedelidir: η büyükse algoritma her dalgalanmaya fazla tepki verir ve lideri izle gibi sömürülebilir hâle gelir. En iyi η, iki terimin toplamını en küçük yapan değerdir. Toplam $f(\eta) = (\ln n)/\eta + \eta T$'nin türevi $f'(\eta) = -(\ln n)/\eta^2 + T$'dir. Türevi sıfıra eşitlemek $\eta^2 = (\ln n)/T$, yani $\eta = \sqrt{(\ln n / T)}$ verir. Bu değer yerine konunca iki terim de $\sqrt{(T \ln n)}$'ye eşit olur:

$$(\ln n)/\sqrt{(\ln n / T)} = \sqrt{(T \ln n)} \text{ ve } \sqrt{(\ln n / T)} \cdot T = \sqrt{(T \ln n)},$$

toplamları da $2\sqrt{(T \ln n)}$'dir.

Örnek 13.7

İki eylem ($n = 2$, $\ln 2 \approx 0{,}693$) ile oynanan iki oyunu karşılaştıralım.

1. $T = 100$: $\eta = \sqrt{(0{,}693/100)} \approx 0{,}083$. Sınır $2\sqrt{(100 \cdot 0{,}693)} = 2\sqrt{69{,}3} \approx 16{,}7$. Tur başına pişmanlık en fazla $16{,}7/100 \approx 0{,}17$.
2. $T = 10.000$: $\eta \approx 0{,}0083$. Sınır $2\sqrt{6931} \approx 166{,}5$. Tur başına pişmanlık en fazla $0{,}017$.

Tur sayısı 100 katına çıkınca toplam sınır yalnız 10 katına çıktı ($\sqrt{100} = 10$); tur başına sınır ise 10'da birine indi. Pişmanlıksızlığın anlamı budur.

Lideri izlemeyi sömüren dizi Hedge'e verilince sınırın somut anlamı görülür. Tablo 13.6'daki altı turluk dizi $\eta = 0{,}5$ ile Hedge'e verilsin. 1. turda olasılıklar eşittir ve beklenen kazanç $0{,}5 \cdot 0{,}5 = 0{,}25$'tir. 2. turda A'nın ağırlığı $e^{0{,}25} \approx 1{,}28$, B'ninki 1'dir; $p(A) = 1{,}28/2{,}28 \approx 0{,}56$ ve beklenen kazanç $0{,}44 \cdot 1 \approx 0{,}44$'tür. Sonraki turlarda olasılıklar 0,56 ile 0,44 arasında gidip gelir ve her tur yaklaşık 0,44 kazanılır. Altı tur sonunda Hedge'in beklenen toplamı yaklaşık 2,44, pişmanlığı $3 - 2{,}44 \approx 0{,}56$'dır. Lideri izle aynı dizide 2,5 pişmanlık biriktirmişti. Hedge küçük farka tam güvenmediği için rakibin hilesi işe yaramaz.

Bölümün geri kalanında üç gözlem tekrar tekrar karşımıza çıkar:

- **Rastgelelik şart.** Lideri izle deterministik olduğu için rakip onu sömürür; Hedge karışık oynar.
- **η bir ayar düğmesidir.** Büyük η hızlı tepki verir ama gürültüye duyarlıdır; küçük η yavaş ama kararlıdır.
- **Oyunlarda denge.** Herkes pişmanlıksız oynarsa zaman ortalaması kaba korelasyonlu dengeye, sıfır toplamlı oyunlarda minimaks stratejisine yaklaşır.

13.3.4 Kod izleme: hedge()

Aşağıdaki fonksiyon, her turun kazanç vektörünü içeren `G` listesi ve öğrenme oranı `eta` ile Hedge'i çalıştırır ve algoritmanın toplam beklenen kazancını döndürür.

```
import math
def hedge(G, eta):
    w = [1.0] * len(G[0])
    top = 0.0
    for g in G:                          # her tur
        W = sum(w)
        p = [x / W for x in w]
        top += sum(pi * gi for pi, gi in zip(p, g))
        w = [x*math.exp(eta*gi) for x, gi in zip(w, g)]
    return top
```

- `w` ağırlık listesidir; eylem sayısı kadar 1,0 ile başlar.
- Her turda önce ağırlıkların toplamı `W` hesaplanır, sonra her ağırlık `W`'ye bölünerek olasılıklar `p` bulunur.
- `top`, beklenen kazançların toplamıdır. Bu turun beklenen kazancı $\Sigma\, p_i \cdot g_i$'dir; `zip(p, g)` iki listeyi eleman eleman eşleştirir.
- Son satır güncellemedir: her ağırlık $e^{\eta \cdot g_i}$ ile çarpılır.

Üç uzman (eylem) A, B, C olsun ve $\eta = 0{,}5$ seçilsin. İlk iki turda kazançlar $g = [1, 0, 0{,}5]$, sonraki üç turda $g = [0, 1, 0{,}5]$'tir. Yani ilk iki tur A iyi, sonraki üç tur B iyidir; C her zaman ortadadır. Kullanılan çarpanlar: kazanç 1 için $e^{0{,}5} \approx 1{,}65$, kazanç 0,5 için $e^{0{,}25} \approx 1{,}28$, kazanç 0 için 1.

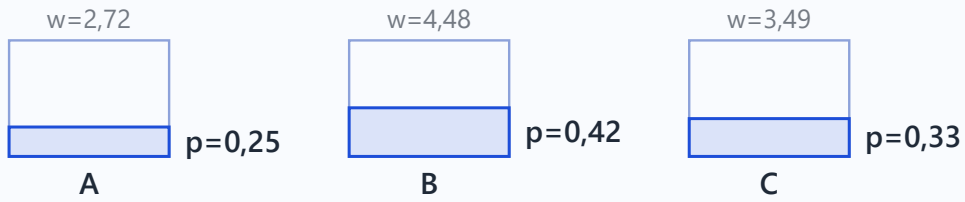
Tur	w (A, B, C)	W	p	g	Σ p·g	top	yeni w
1	1; 1; 1	3,00	0,33; 0,33; 0,33	1; 0; 0,5	0,50	0,50	1,65; 1,00; 1,28
2	1,65; 1,00; 1,28	3,93	0,42; 0,25; 0,33	1; 0; 0,5	0,58	1,08	2,72; 1,00; 1,65
3	2,72; 1,00; 1,65	5,37	0,51; 0,19; 0,31	0; 1; 0,5	0,34	1,42	2,72; 1,65; 2,12
4	2,72; 1,65; 2,12	6,48	0,42; 0,25; 0,33	0; 1; 0,5	0,42	1,84	2,72; 2,72; 2,72
5	2,72; 2,72; 2,72	8,15	0,33; 0,33; 0,33	0; 1; 0,5	0,50	2,34	2,72; 4,48; 3,49

Tablo 13.7: `hedge(G, 0.5)` izlemesi. *w* o turun başındaki ağırlıklar, "yeni w" tur sonundaki güncellemedir.

Örnek 13.8

1. **Tur 1:** $W = 3$, olasılıklar eşit (0,33). Beklenen kazanç $⅓ \cdot 1 + ⅓ \cdot 0 + ⅓ \cdot 0{,}5 = 0{,}50$. Güncelleme: A $1 \cdot 1{,}65$, B değişmez, C $1 \cdot 1{,}28$.
2. **Tur 2:** $W = 1{,}65 + 1 + 1{,}28 = 3{,}93$. $p(A) = 1{,}65/3{,}93 \approx 0{,}42$: A'nın payı büyüdü. Beklenen kazanç $0{,}42 \cdot 1 + 0{,}33 \cdot 0{,}5 \approx 0{,}58$; `top` 1,08 olur. A'nın ağırlığı $1{,}65 \cdot 1{,}65 \approx 2{,}72 = e^{1}$'e çıkar.
3. **Tur 3:** $W = 5{,}37$, $p(A) \approx 0{,}51$. Ama bu tur kazanan B'dir: beklenen kazanç $0{,}19 \cdot 1 + 0{,}31 \cdot 0{,}5 \approx 0{,}34$. Algoritma geçmişe uyduğu için bir tur geç kalır.
4. **Tur 4:** B'nin ve C'nin ağırlıkları artmıştır; $p \approx (0{,}42;\ 0{,}25;\ 0{,}33)$. Beklenen kazanç 0,42. Tur sonunda üç ağırlık da $e \approx 2{,}72$'ye eşitlenir, çünkü her eylemin toplam kazancı 2'dir.
5. **Tur 5:** Olasılıklar yeniden eşittir; beklenen kazanç 0,50, `top` 2,34. Son ağırlıklar $(2{,}72;\ 4{,}48;\ 3{,}49) = (e^{1},\ e^{1{,}5},\ e^{1{,}25})$.

Sabit eylemlerin toplamları A: 2, B: 3, C: 2,5'tir. En iyi sabit eylem B'dir ve pişmanlık $3 - 2{,}34 = 0{,}66$'dır.



Şekil 13.7: `hedge()` izlemesinin sonu. Beş tur sonunda ağırlıklar A: 2,72, B: 4,48, C: 3,49; bir sonraki turun olasılıkları 0,25, 0,42 ve 0,33 olurdu.

Şekil 13.7 Hedge'in ağırlıkları nasıl "hatırladığını" gösterir. Son ağırlıklar yalnızca toplam kazançlara bağlıdır: $w_i = e^{0{,}5 \cdot G_i}$. Sıralama da toplam kazançların sıralamasıyla aynıdır: B (3) > C (2,5) > A (2). Kazançların hangi sırayla geldiği ağırlıkları değiştirmez; yalnızca algoritmanın yol boyunca ne kazandığını değiştirir.

Sık yapılan hata

- **Ağırlığı olasılık sanmak:** 2,72 gibi bir ağırlık olasılık değildir; olasılık, ağırlığın toplam W'ye bölünmesiyle bulunur.
- **Kazancı ağırlığa eklemek:** Hedge'de güncelleme çarpımsaldır ($w \cdot e^{\eta g}$), toplamsal değildir ($w + \eta g$).
- **Beklenen kazancı gerçekleşen kazanç sanmak:** `top` beklenen kazançların toplamıdır. Gerçek bir oyunda her tur bir eylem çekilir ve gerçekleşen kazanç bu değer çevresinde dalgalanır.

13.3.5 Rakip de öğrenirse

Şimdiye kadar kazanç dizisi dışarıdan verildi. Oyunlarda ise kazançları rakibin davranışı belirler ve rakip de öğrenebilir. Bu farkı görmek için Taş-Kağıt-Makas oyununda iki senaryoyu karşılaştıralım. Kazanan 1, berabere kalan ½, kaybeden 0 alır; böylece kazançlar *[0, 1]* aralığındadır. Oyuncu 1 Hedge ile öğrenir ve $\eta = 0{,}1$'dir. Öğrenen oyuncu her turda eylemlerinin *beklenen* kazancıyla günceller.

Senaryo 1: sabit rakip. Rakip her turda Taş'ı 0,5, Kağıt'ı 0,3, Makas'ı 0,2 olasılıkla oynar. Oyuncu 1'in her eyleminin beklenen kazancı hesaplanabilir.

Örnek 13.9

1. **Taş:** Taş'a karşı berabere (½), Kağıt'a karşı kaybeder (0), Makas'a karşı kazanır (1). Beklenen kazanç $0{,}5 \cdot ½ + 0{,}3 \cdot 0 + 0{,}2 \cdot 1 = 0{,}25 + 0{,}20 = 0{,}45$.
2. **Kağıt:** Taş'ı yener, Kağıt'la berabere, Makas'a yenilir: $0{,}5 \cdot 1 + 0{,}3 \cdot ½ + 0 = 0{,}65$.
3. **Makas:** Taş'a yenilir, Kağıt'ı yener, Makas'la berabere: $0 + 0{,}3 \cdot 1 + 0{,}2 \cdot ½ = 0{,}40$.
4. En iyi tepki Kağıt'tır. Her tur Kağıt'ın ağırlığı Taş'ınkinden $e^{0{,}1 \cdot 0{,}20}$, Makas'ınkinden $e^{0{,}1 \cdot 0{,}25}$ kat daha fazla artar. Eşit ağırlıklarla başlayan Hedge'de n güncellemeden sonra $p(\text{Kağıt}) = 1 / (1 + e^{-0{,}02\,n} + e^{-0{,}025\,n})$ olur.
5. 100. turda bu olasılık yaklaşık 0,82, 300. turda 0,997'dir.

Sabit rakibe karşı Hedge en iyi tepkiye gider. Bu, pişmanlıksızlığın doğal bir sonucudur: rakip değişmiyorsa en iyi sabit eylem en iyi tepkidir ve pişmanlığı küçük tutmanın tek yolu onu giderek daha sık oynamaktır.

Senaryo 2: öğrenen rakip. Şimdi rakip de Hedge kullanır ve iki oyuncu aynı anda günceller. Oyuncu 1 (0,5; 0,3; 0,2) olasılıklarıyla, oyuncu 2 eşit olasılıklarla başlar. Oyuncu 1 Kağıt'a kaydıkça oyuncu 2 için Makas değerlenir; oyuncu 2 Makas'a kaydıkça oyuncu 1 için Taş değerlenir. Döngü böyle sürer ve hedef sürekli kayar.

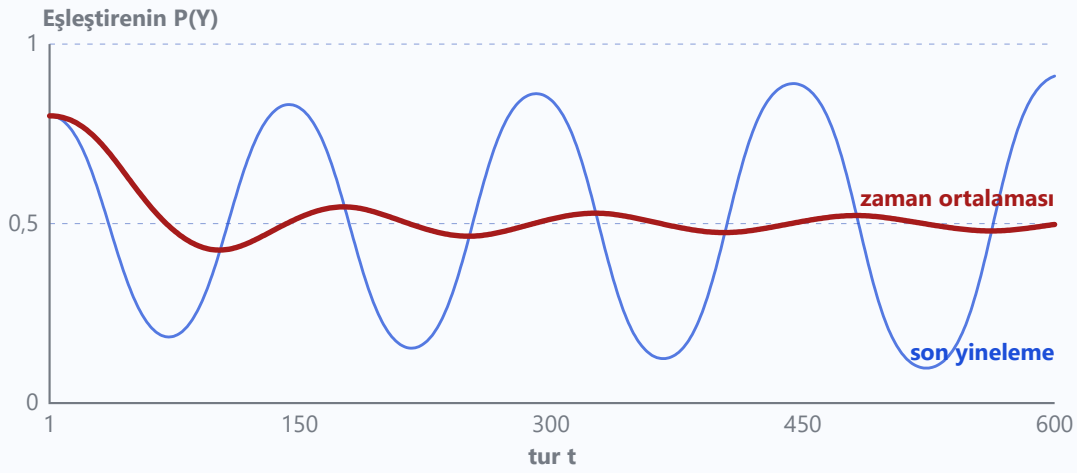
tur	100	200	800	1000
o turdaki P(Taş)	0,19	0,46	0,14	0,19
ortalama P(Taş)	0,35	0,31	0,33	0,34

Tablo 13.8: İki Hedge oyuncusu ($\eta = 0{,}1$), Taş-Kağıt-Makas. Oyuncu 1'in Taş olasılığı: o turdaki değer ve o tura kadarki ortalama.

Tablo 13.8 iki farklı davranış gösterir. O turdaki olasılık (son yineleme) dönmeye devam eder: 0,19'dan 0,46'ya çıkar, 0,14'e iner. Bin turda bile durulmaz. Zaman ortalaması ise ⅓'e, yani oyunun tek Nash dengesi olan (⅓, ⅓, ⅓)'ün Taş bileşenine yaklaşır. Öğrenen bir rakibe karşı "hedefe varmak" anlamsızdır, çünkü hedef rakibin davranışıyla birlikte hareket eder. Kararlı olan şey ortalamadır.

13.3.6 Hedge kendine karşı

Aynı olgu iki eylemli bir oyunda daha net görülür. Yazı-Tura Eşleştirme'de iki oyuncu da Hedge ($\eta = 0{,}1$) kullansın. Eşleştiren'in başlangıç ağırlıkları (Y, T) = (0,8; 0,2), yani $P(Y) = 0{,}8$; Ayrıştıran'ınkiler (0,5; 0,5)'tir. Kazanan 1, kaybeden 0 alır ve her turda iki oyuncu da eylemlerinin beklenen kazancıyla aynı anda günceller.



Şekil 13.8: Yazı-Tura Eşleştirme'de iki Hedge oyuncusu. Mavi ince eğri Eşleştiren'in o turdaki $P(Y)$ değeri (son yineleme), kırmızı kalın eğri zaman ortalamasıdır. Son yineleme büyük salınımlar yapar; ortalama 0,5'e yerleşir.

Şekil 13.8'de mavi eğri dengeye oturmaz; 0,1 ile 0,9 arasında giderek büyüyen salınımlar yapar. Mekanizma kurgusal oyundakine benzer. Eşleştiren Y'ye ağırlık verince Ayrıştıran için T değerlenir; Ayrıştıran T'ye kayınca Eşleştiren için T değerlenir ve döngü böyle sürer. Başlangıçtan bu yana $P(Y)$ değerlerinin ortalaması olan kırmızı eğri ise minimaks stratejisi olan ½'ye yaklaşır: 600 turda 0,498'dir.

Bu gözlemin kuramsal bir karşılığı vardır. İki kişilik sıfır toplamlı bir oyunda iki oyuncu da pişmanlıksız bir algoritma kullanırsa, ortalama stratejileri minimaks stratejilerine, ortalama kazanç da oyunun değerine yaklaşır. Von Neumann'ın minimaks teoreminin, yani sıfır toplamlı oyunlarda “maks-min = min-maks” eşitliğinin, öğrenme yoluyla yapılan ispatı bu gözleme dayanır. İspatın fikri şudur: iki oyuncunun da pişmanlığı sıfıra gidiyorsa, ikisi de ortalamada rakibinin ortalama stratejisine karşı en iyi sabit eylem kadar kazanır; bu iki ifade birleştirilince maks-min ile min-maks arasındaki fark sıfıra sıkışır.

Sık yapılan hata

“Pişmanlıksız algoritmalar dengeye yakınsar” cümlesini son yineleme için okumak. Kuram yalnızca *zaman ortalaması* için yakınsama verir. Şekil 13.8'de son yineleme, 600 turda bile denge değerinden çok uzaktır.

13.3.7 Pişmanlıktan dengeye

Sıfır toplamlı oyunlar dışında pişmanlıksız öğrenmenin zaman ortalaması nereye gider? Cevap, Nash dengesinden daha geniş iki denge kavramıdır. Bu kavramları anlatmanın en kolay yolu bir **öneri makinesi** düşünmektir. Makine her turda bir olasılık dağılımından bir eylem profili çeker ve her oyuncuya yalnızca kendi eylemini söyler. Kavşaktaki trafik ışığı böyle bir makinedir: bir sürücüye "geç", ötekine "dur" der.

Tanım 13.6

Eylem profilleri üzerindeki bir σ dağılımı,

- her oyuncu için, *önerisini gördükten sonra* öneriye uymak sapmaktan kötü değilse **korelasyonlu dengedir** (*correlated equilibrium*, CE);
- her oyuncu için, *öneriyi görmeden* "önerinin ne olduğuna bakmaksızın hep a' oynarım" demek beklenen kazancı artırmıyorsa **kaba korelasyonlu dengedir** (*coarse correlated equilibrium*, CCE).

İki tanım arasındaki fark sapma anındadır. CE'de oyuncu öneriyi gördükten sonra karar verir; bu yüzden "Kır önerisi gelince Düz git" gibi öneriye bağlı sapmalar da hesaba katılır. CCE'de oyuncu, öneriyi görmeden önce bütünüyle makineden ayrılıp sabit bir eyleme geçmeyi düşünür. CE'nin koşulları her öneri için ayrı ayrı sağlandığından, ortalamada da sağlanır; bu yüzden her CE bir CCE'dir. Tersi doğru değildir: CCE daha zayıf, yani daha geniş bir kavramdır.



Şekil 13.9: Denge kavramlarının iç içe kümeleri. Saf Nash dengesi her oyunda olmayabilir; karışık Nash dengesi her sonlu oyunda vardır ama bulmak PPAD-tamdır; korelasyonlu denge (CE) iç pişmanlıksız dinamiklerle, kaba korelasyonlu denge (CCE) Hedge gibi pişmanlıksız dinamiklerle bulunur.

Şekil 13.9'daki iç içe yapı şöyle okunur. Her saf Nash dengesi bir karışık Nash dengesidir (olasılıkları 0 ya da 1 olan). Her karışık Nash dengesi bir CE'dir: oyuncular bağımsız zarlar attığında, önerisini gören oyuncu ötekiler hakkında yeni bir şey öğrenmez ve dengede zaten en iyi tepki vermektedir. Her CE de bir CCE'dir. Dışa doğru gidildikçe kavramlar genişler ama hesaplanmaları kolaylaşır: CE ve CCE doğrusal programlamayla hızlı bulunur, karışık Nash dengesi ise PPAD-tamdır.

Teorem 13.3 (Pişmanlıksız öğrenme ve CCE)

Bir oyunu tekrar tekrar oynayan bütün oyuncuların dışsal pişmanlığı için $R_T/T \le \varepsilon$ ise, T tur boyunca oynanan profillerin **zaman ortalaması dağılımı** (her profilin oynanma oranı) bir ε-CCE'dir. Daha güçlü bir pişmanlık türünü, **iç pişmanlığı**, küçülten algoritmalar CE'ye götürür (Foster ve Vohra, 1997; Hart ve Mas-Colell, 2000; Blum ve Mansour, 2007).

İspat. Oynanan profiller $s^1, \dots, s^T$ ve σ bunların zaman ortalaması dağılımı olsun: σ'dan bir profil çekmek, $1, \dots, T$ arasından rastgele bir tur seçip o turun profilini almak demektir. Oyuncu i'nin t. turdaki kazancı $g_t(a) = u_i(a, s_{-i}^t)$'dir. Pişmanlık koşulu her sabit a' için

$$\sum_t u_i(a', s_{-i}^t) - \sum_t u_i(s^t) \le R_T \le \varepsilon T$$

der. İki tarafı T'ye bölmek toplamları ortalamaya çevirir. Ortalama ise σ altındaki beklenen değerdir: $E_{s\sim\sigma}[u_i(a', s_{-i})] - E_{s\sim\sigma}[u_i(s)] \le \varepsilon$. Bu eşitsizlik, "öneriyi görmeden hep a' oynamak" sapmasının en fazla ε kazandırdığını söyler; CCE tanımının ε payıyla sağlanmasıdır. □

İç pişmanlık (*internal / swap regret*), "a oynadığım her turda onun yerine a' oynasaydım ne kazanırdım?" sorusunu her (a, a') çifti için ayrı ayrı sorar. Bu, öneriye bağlı sapmanın öğrenme dilindeki karşılığıdır: "makine a dediğinde a' oyna". Dışsal pişmanlık bütün turları tek bir sabit eylemle karşılaştırdığı için daha kaba bir ölçüttür ve daha kaba bir dengeye götürür.

Örnek 13.10

Kavşaktaki iki sürücünün Tavuk oyununu düşünelim. Kır (dur) ve Düz (geç) eylemleri vardır; kazançlar (Kır, Kır) = (0, 0), (Kır, Düz) = (−1, 1), (Düz, Kır) = (1, −1), (Düz, Düz) = (−10, −10)'dur. Trafik ışığı yarı yarıya (Kır, Düz) ya da (Düz, Kır) önerir.

1. **"Kır" önerisini alan sürücü** ötekine kesin "Düz" denildiğini bilir. Uyarsa −1, Düz'e saparsa −10 alır. Uymak daha iyidir.
2. **"Düz" önerisini alan sürücü** ötekine kesin "Kır" denildiğini bilir. Uyarsa 1, Kır'a saparsa 0 alır. Uymak daha iyidir.
3. İki öneride de uymak en iyisidir: ışık bir korelasyonlu dengedir. Her sürücünün beklenen kazancı $\tfrac{1}{2}\cdot(-1) + \tfrac{1}{2}\cdot 1 = 0$'dır.

4. CCE koşulu da kendiliğinden sağlanır. Örneğin “hep Düz” sapması, ötekine yarı yarıya Kır ve Düz denildiği için $½ \cdot 1 + ½ \cdot (-10) = -4{,}5$ verir; bu 0'dan kötüdür. “Hep Kır” sapması $½ \cdot 0 + ½ \cdot (-1) = -0{,}5$ verir; bu da 0'dan kötüdür.

Işığa uymamak kimseye kazanç sağlamaz. Pişmanlıksız öğrenme bu tür dengeleri merkezî bir hesaplayıcı olmadan, yalnızca oyuncuların kendi pişmanlıklarını küçültmesiyle bulur.

Sık yapılan hata

- **CE ile CCE'yi karıştırmak:** Hedge yalnızca dışsal pişmanlığı küçültür ve CCE'ye götürür; CE için iç pişmanlığı küçülten algoritmalar gerekir.
- **Ortalama dağılımı her oyuncunun karışık stratejisi sanmak:** CCE profiller üzerinde *ortak* bir dağılımdır; oyuncuların eylemleri birbirine bağlı (korelasyonlu) olabilir. Bu dağılım her zaman iki bağımsız karışık stratejinin çarpımı olarak yazılamaz.

13.4 Çok kollu haydut

Hedge her turdan sonra *bütün* eylemlerin kazancını görür. Açılıştaki kumar makinesinde ise yalnızca çekilen kolun sonucu görülür. Bu küçük fark büyük bir sonuç doğurur: bilgi artık bedava değildir. Bir kol hakkında bilgi toplamanın tek yolu onu çekmektir; o kol kötüyse bu çekiş ödülden kayıptır. Çok kollu haydut problemi, bu bilgi–ödül alışverişinin en sade modelidir.

13.4.1 Problem, pişmanlık ve keşif–sömürü ikilemi

Tanım 13.7

Çok kollu haydut (*multi-armed bandit*): K kollu bir kumar makinesinde kol i, ortalaması μ_i olan ve oyuncunun bilmediği bir dağılımdan rastgele ödül verir. Her turda bir kol çekilir ve yalnızca onun ödülü görülür (kısmi geri bildirim). En iyi kolun ortalaması $\mu^* = \max_i \mu_i$ olmak üzere T turdaki pişmanlık

$$R_T = T \cdot \mu^* - \Sigma_{t=1}^{T} \mu_{a_t}$$

ile ölçülür; burada a_t, t. turda çekilen koldur.

Bu pişmanlık tanımı Hedge'deki dışsal pişmanlıkla aynı fikri taşır: en iyi sabit seçenekle (burada en iyi kolla) karşılaştırma. Farkı, gerçekleşen ödüllerin değil ortalamaların kullanılmasıdır. Bu sayede şans faktörü ayıklanır: kötü kolu çekip şans eseri 1 almak pişmanlığı azaltmaz, çünkü hesapta o kolun

ortalaması kullanılır. Her tur, çekilen kolun en iyi koldan ne kadar geride olduğu, yani $\mu^* - \mu_{a_t}$ kadar pişmanlık ekler.

Problemin bütün zorluğu **keşif ve sömürü ikilemi**nde toplanır. Keşif, yeni ya da az denenmiş kollar hakkında bilgi toplar. Sömürü, o ana kadar en iyi görüneni kullanır. Yalnızca sömüren bir algoritma, şanssız bir başlangıçtan sonra yanlış kola takılabilir ve hiçbir zaman kurtulamaz. Yalnızca keşfeden bir algoritma ise bütün kolları eşit çeker ve öğrendiğini hiç kullanmaz; ödülü boşa harcar.

Haydut algoritmaları her kol için iki sayı tutar: kolun kaç kez çekildiği N_i ve gözlenen ödüllerin ortalaması Q_i. Ortalama, her yeni ödül geldiğinde baştan hesaplanmaz; **artımlı ortalama** formülüyle güncellenir:

$$Q \leftarrow Q + (r - Q)/N$$

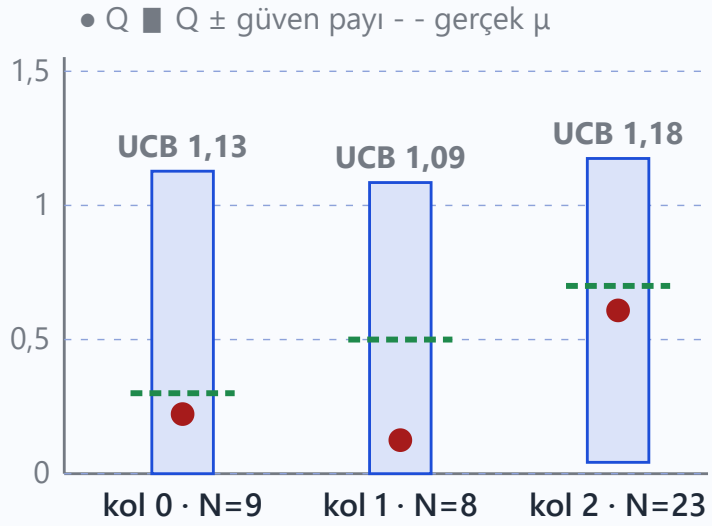
Burada N, yeni ödül de sayıldıktan sonraki çekiş sayısıdır. Formülün doğruluğu kısa bir hesapla görülür. Önceki $N - 1$ ödülün ortalaması Q ise toplamları $(N - 1)Q$'dur. Yeni ödül eklenince yeni ortalama $((N - 1)Q + r)/N = (NQ - Q + r)/N = Q + (r - Q)/N$ olur. Formülün yorumu da açıktır: tahmin, "sürpriz" $r - Q$ yönünde $1/N$ kadar kayar. Çok çekilmiş bir kolun tahmini tek bir gözlemle az değişir.

Örnek 13.11

Bir kolun ödülleri sırayla 1, 1, 1, 0, 1 olsun.

1. $N = 1$: $Q = 0 + (1 - 0)/1 = 1$.
2. $N = 2$: $Q = 1 + (1 - 1)/2 = 1$. $N = 3$: yine 1.
3. $N = 4$: $Q = 1 + (0 - 1)/4 = 0{,}75$.
4. $N = 5$: $Q = 0{,}75 + (1 - 0{,}75)/5 = 0{,}75 + 0{,}05 = 0{,}80$.

Sonuç, beş ödülün doğrudan ortalamasıyla ($4/5 = 0{,}80$) aynıdır.



Şekil 13.10: UCB1 algoritması, gerçek ortalamalar $\mu = (0{,}3;\ 0{,}5;\ 0{,}7)$, 40 tur sonra. Kırmızı noktalar tahmin Q, mavi çubuklar $Q \pm$ *güven payı*, yeşil kesikli çizgiler gerçek μ. Az çekilen kolların (kol 0: 9, kol 1: 8 çekiş) aralığı geniştir; çok çekilen kol 2'nin (23 çekiş) aralığı dardır.

Şekil 13.10, UCB1 algoritmasının (13.4.3) temel fikrini gösterir. Az denenen kolun tahmini belirsizdir, bu yüzden güven aralığı geniştir. Aralığın üst ucu, yani "bu kolun olabileceği en iyi değer", bu kolu yeniden denemeye değer kılar. Tahmini düşük olsa bile kol 1'in üst ucu 1,09'dur ve kol 2'nin 1,18'ine yakındır.

Haydut probleminin uygulama alanı oyun kuramının çok ötesine uzanır: hangi reklamın ya da haberin gösterileceğine karar veren öneri sistemleri, web sitelerinde iki tasarımın karşılaştırıldığı A/B testleri, hangi tedavinin daha çok hastaya verileceğinin denemeler sırasında ayarlandığı klinik deneyler ve bir ürünün farklı fiyatlarla denendiği fiyat denemeleri.

13.4.2 Kod izleme: ε-açgözlü

En basit haydut algoritması **ε-açgözlü**dür (*ε-greedy*). Her turda küçük bir ε olasılıkla rastgele bir kol çeker (keşif); geri kalan $1 - \varepsilon$ olasılıkla en büyük Q'lu kolu çeker (sömürü). "Açgözlü" adı, sömürü adımının o anki en iyi tahmine körü körüne güvenmesinden gelir.

```
def egreedy(mu, T, eps, rng):
    N, Q = [0] * len(mu), [0.0] * len(mu)
    for t in range(1, T + 1):
        if rng.random() < eps:          # keşif
            a = rng.randrange(len(mu))
        else:                           # sömürü
            a = Q.index(max(Q))
        r = 1 if rng.random() < mu[a] else 0
        N[a] += 1
        Q[a] += (r - Q[a]) / N[a]
    return N, Q
```

- `mu`, kolların gerçek ortalamalarıdır. Algoritma bu listeyi yalnızca ödül üretmek için kullanır; karar verirken `mu`'ya bakmaz.
- `rng.random()` 0 ile 1 arasında bir rastgele sayı üretir. Sayı `eps`'ten küçükse keşif yapılır ve `rng.randrange(len(mu))` ile rastgele bir kol seçilir.
- Sömürüde `Q.index(max(Q))` en büyük tahmini veren kolu bulur. Eşitlik varsa listedeki *ilk* kol seçilir.
- Ödül, kolun ortalamasıyla belirlenen bir para atışıdır: yeni bir rastgele sayı `mu[a]`'dan küçükse ödül 1, değilse 0'dır.
- Son iki satır sayaç ve artımlı ortalama güncellemesidir.

Çağrı `egreedy([0.3, 0.5, 0.7], 8, 0.2, random.Random(11))` olsun. Kodda kollar 0'dan numaralanır: kol 0, 1, 2, açılış oyunundaki Kol 1, 2, 3'e karşılık gelir. Rastgele sayı üreteci 11 tohumuyla başlatıldığı için çağrı her çalıştırıldığında aynı sonucu verir; aşağıdaki izleme bu yüzden tekrarlanabilir.

t	Rastgele sayı	Karar	Kol	r	N	Q
1	0,452	sömürü (eşitlikte ilk)	0	0	[1, 0, 0]	[0,00; 0,00; 0,00]
2	0,924	sömürü (eşitlikte ilk)	0	0	[2, 0, 0]	[0,00; 0,00; 0,00]
3	0,508	sömürü (eşitlikte ilk)	0	0	[3, 0, 0]	[0,00; 0,00; 0,00]
4	0,185	keşif	2	1	[3, 0, 1]	[0,00; 0,00; 1,00]
5	0,614	sömürü	2	1	[3, 0, 2]	[0,00; 0,00; 1,00]
6	0,447	sömürü	2	1	[3, 0, 3]	[0,00; 0,00; 1,00]
7	0,539	sömürü	2	0	[3, 0, 4]	[0,00; 0,00; 0,75]
8	0,634	sömürü	2	1	[3, 0, 5]	[0,00; 0,00; 0,80]

Tablo 13.9: `egreedy` izlemesi ($\varepsilon = 0{,}2$, tohum 11). "Rastgele sayı" keşif kararında kullanılan ilk sayıdır.

Örnek 13.12

1. **t = 1–3:** Rastgele sayılar 0,2'den büyüktür; üçü de sömürüdür. Bütün tahminler 0 olduğu için eşitlikte ilk kol, yani kol 0 seçilir. Kol 0 üç kez de 0 verir; tahmini 0'da kalır ve eşitlik bozulmaz.
2. **t = 4:** Rastgele sayı 0,185 < 0,2: keşif. Rastgele seçilen kol 2'dir ve 1 verir. *N[2] = 1*, *Q[2] = 1,00*.
3. **t = 5–6:** Sömürü; en büyük tahmin artık kol 2'dedir. İki çekiş de 1 verir; *Q[2]* 1'de kalır.
4. **t = 7:** Kol 2 bu kez 0 verir: *Q[2] = 1 + (0 − 1)/4 = 0,75*.
5. **t = 8:** Kol 2 1 verir: *Q[2] = 0,75 + (1 − 0,75)/5 = 0,80*.

Fonksiyon *N = [3, 0, 5]* döndürür. Sekiz turun pişmanlığı *8 · 0,7 − (3 · 0,3 + 5 · 0,7) = 5,6 − 4,4 = 1,2*'dir; bunun tamamı kol 0'a harcanan ilk üç turdan gelir.



Şekil 13.11: ε-açgözlü izlemesinin sonu. Kol 0 üç kez çekilmiş, tahmini 0; kol 1 hiç çekilmemiş; kol 2 beş kez çekilmiş, tahmini 0,80. Yeşil kesikli çizgiler gerçek ortalamalardır (0,3; 0,5; 0,7).

İzleme ε-açgözlünün iki yüzünü birden gösterir. Olumsuz yüzü: ilk üç tur kol 0'a takıldı, çünkü hepsi 0 ödül verdi ve eşitlik ilk kolu seçtirdi. Keşif olmasaydı algoritma sonsuza kadar kol 0'da kalabilirdi; kol 0 bir kez 1 verse bile tahmini öteki kolların 0'ından büyük olur ve sömürü onu seçmeye devam ederdi. Olumlu yüzü: tek bir keşif en iyi kolu buldu. Şekil 13.11'de bir eksiklik de görülür: kol 1 hiç denenmedi. Sekiz turda bu önemli değildir, ama ε-açgözlünün keşfi kör bir rastgeleliktir; az bilinen kolları özellikle aramaz.

13.4.3 Kod izleme: UCB1

ε-açgözlünün keşfi "kör"dür: keşif turunda her kol eşit olasılıkla seçilir, en iyi olduğu zaten bilinen kol da, hiç denenmemiş kol da. **UCB1** (*Upper Confidence Bound*, üst güven sınırı) keşfi yönlendirir. Her kolun bir puanı vardır:

$$UCB_i = Q_i + \sqrt{(2 \ln t / N_i)}$$

Burada t şu anki tur numarası, N_i kolun çekilme sayısıdır. Her turda puanı en büyük kol çekilir. Hiç çekilmemiş bir kolun puanı sonsuz sayılır; bu yüzden ilk turlarda her kol bir kez denenir.

Puanın iki terimi vardır. İlk terim Q_i sömürüdür: iyi görünen kol yüksek puan alır. İkinci terim $\sqrt{(2\ \ln t\ /\ N_i)}$ **güven payı**dır ve keşfi temsil eder. Pay, N_i büyüdükçe küçülür: çok çekilmiş kolun tahminine güvenilir. Pay, t büyüdükçe çok yavaş (logaritmik olarak) büyür: uzun süre çekilmeyen bir kolun payı yavaş yavaş artar ve sonunda kol yeniden denenir. Bu fikre **belirsizlik karşısında iyimserlik** denir: her kol, verilerle çelişmeyen en iyimser değeriyle değerlendirilir. Kötü bir kol iyimser değerlendirmeyle seçilirse kısa sürede gerçek yüzünü gösterir ve payı küçülür; iyi bir kol ise iyimser değerlendirmeyi hak eder.

```
import math
def ucb(mu, T, rng):
    K = len(mu)
    N, Q = [0] * K, [0.0] * K
    for t in range(1, T + 1):
        b = [math.inf if n == 0 else
             math.sqrt(2*math.log(t)/n) for n in N]
        a = max(range(K), key=lambda i: Q[i] + b[i])
        r = 1 if rng.random() < mu[a] else 0
        N[a] += 1
        Q[a] += (r - Q[a]) / N[a]
    return N, Q
```

- `b`, her kolun güven payıdır. `math.inf`, hiç çekilmemiş kolun sonsuz payıdır.
- `max(range(K), key=lambda i: Q[i] + b[i])`, $Q + b$ puanı en büyük olan kolu seçer; eşitlikte küçük indisli kol kazanır.
- Ödül ve güncelleme ε-açgözlüdeki gibidir.

Çağrı `ucb([0.3, 0.5, 0.7], 8, random.Random(3))` olsun. Tablo 13.10 her turdaki payları, puanları ve seçimi verir.

t	pay b	UCB = Q + b	Kol	r	Q (sonra)	N (sonra)
1	∞, ∞, ∞	∞, ∞, ∞	0	1	1,00; 0; 0	1, 0, 0
2	1,18; ∞; ∞	2,18; ∞; ∞	1	0	1,00; 0,00; 0	1, 1, 0
3	1,48; 1,48; ∞	2,48; 1,48; ∞	2	1	1,00; 0,00; 1,00	1, 1, 1
4	1,67; 1,67; 1,67	2,67; 1,67; 2,67	0	0	0,50; 0,00; 1,00	2, 1, 1
5	1,27; 1,79; 1,79	1,77; 1,79; 2,79	2	1	0,50; 0,00; 1,00	2, 1, 2
6	1,34; 1,89; 1,34	1,84; 1,89; 2,34	2	1	0,50; 0,00; 1,00	2, 1, 3
7	1,39; 1,97; 1,14	1,89; 1,97; 2,14	2	1	0,50; 0,00; 1,00	2, 1, 4
8	1,44; 2,04; 1,02	1,94; 2,04; 2,02	1	0	0,50; 0,00; 1,00	2, 2, 4

Tablo 13.10: `ucb` izlemesi (tohum 3). Pay ve puanlar seçimden önceki değerlerdir; Q ve N seçimden sonraki değerlerdir.

Örnek 13.13

1. **t = 1, 2, 3:** Hiç denenmemiş kolların payı sonsuzdur; kollar sırayla birer kez çekilir. Ödüller 1, 0, 1'dir.
2. **t = 4:** Her kol bir kez çekildi: pay $\sqrt{(2\ \ln 4 / 1)} = \sqrt{2{,}77} \approx 1{,}67$, üç kolda aynıdır. Puanlar $1 + 1{,}67 = 2{,}67$, $0 + 1{,}67 = 1{,}67$, $2{,}67$. Kol 0 ile kol 2 eşittir; küçük indisli kol 0 seçilir ve 0 verir. $Q[0] = 0{,}50$.
3. **t = 5:** Kol 0 iki kez çekildi: payı $\sqrt{(2\ \ln 5 / 2)} = \sqrt{1{,}61} \approx 1{,}27$. Ötekilerin payı $\sqrt{(2\ \ln 5)} \approx 1{,}79$. Puanlar 1,77; 1,79; 2,79. Kol 2 seçilir ve 1 verir.
4. **t = 6, 7:** Kol 2 kazanmaya devam eder. Ama her çekişte payı küçülür (1,34 → 1,14), kol 1'in payı ise büyür (1,89 → 1,97).
5. **t = 8:** Kol 1'in payı $\sqrt{(2\ \ln 8 / 1)} = \sqrt{4{,}16} \approx 2{,}04$'e ulaşmıştır; puanı 2,04'tür. Kol 2'nin puanı $1 + \sqrt{(4{,}16/4)} = 1 + 1{,}02 = 2{,}02$'dir. Kol 1 kıl payı öne geçer ve yeniden denenir.

Fonksiyon $N = [2, 2, 4]$ döndürür. Kol 1'in tek denemesi 0 vermiş ve tahmini 0'dır; yine de 8. turda yeniden denenir. Keşif ayrı bir parametreye gerek kalmadan, belirsizliğin kendisinden doğar.



Şekil 13.12: UCB1 izlemesinin sonu: kol 0 iki kez (Q = 0,50), kol 1 iki kez (Q = 0), kol 2 dört kez (Q = 1,00) çekildi.

İki izlemeyi karşılaştırmak öğreticidir. ε-açgözlü kol 1'i hiç denemedi; UCB1 ise her kolu denedi ve tahmini düşük kolu bile, uzun süre çekilmeyince yeniden ziyaret etti. UCB1 sekiz turda daha çok keşif yaptığı için en iyi olmayan kolları daha sık çekti (üçe karşı dört tur). Ama bu turlardan ikisi orta kola (kol 1) gittiği için pişmanlığı yine $8 \cdot 0{,}7 - (2 \cdot 0{,}3 + 2 \cdot 0{,}5 + 4 \cdot 0{,}7) = 5{,}6 - 4{,}4 = 1{,}2$'dir. Kısa vadede iki algoritma aynı yerdedir; fark uzun vadede ortaya çıkar.

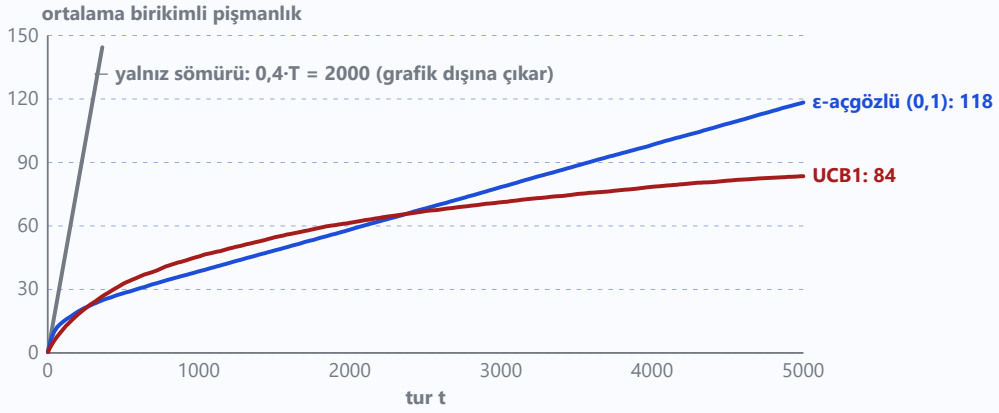
Sık yapılan hata

- **Payı yanlış t ile hesaplamak:** t, şu anki tur numarasıdır (1'den başlar), kolun çekilme sayısı değildir. $t = 8$'de bütün kolların payında $\ln 8$ kullanılır.
- **En büyük Q'yu seçmek:** UCB1 en büyük tahmini değil, en büyük *üst sınırı* seçer. 8. turda en büyük tahmin kol 2'dedir (1,00) ama seçilen kol 1'dir.

- **ln yerine $\log_{10}$ kullanmak:** Formüldeki *ln* doğal logaritmadır; $\ln 8 \approx 2{,}08$, $\log_{10} 8 \approx 0{,}90$'dır.

13.4.4 Karşılaştırma: 5000 tur, 100 tekrar

Sekiz turluk izlemeler algoritmaların nasıl çalıştığını gösterir, ama hangisinin daha iyi olduğunu göstermez. Ödüller rastgele olduğu için tek bir deney yanıltıcıdır. Karşılaştırma şöyle yapılır: kollar yine $\mu = (0{,}3;\ 0{,}5;\ 0{,}7)$'dir; her algoritma 5000 tur çalıştırılır; deney 0'dan 99'a kadar 100 farklı tohumla tekrarlanır ve birikimli pişmanlıkların ortalaması alınır. ε-açgözlü bu karşılaştırmada $\varepsilon = 0{,}1$ ile çalışır.



Şekil 13.13: 5000 tur, 100 tekrar ortalaması birikimli pişmanlık. Gri: yalnız sömürü ($0{,}4 \cdot T$, 5000 turda 2000; grafiğin dışına çıkar). Mavi: ε-açgözlü ($\varepsilon = 0{,}1$), 5000 turda 118. Kırmızı: UCB1, 5000 turda 84.

Şekil 13.13 üç farklı büyüme hızı gösterir.

- **Yalnız sömürü** ($\varepsilon = 0$): bütün tahminler 0 başladığı için ilk tur kol 0'dır. Kol 0 bir kez 1 verirse tahmini pozitif olur ve öteki kollar hiç denenmez; hep 0 verirse de eşitlikte yine kol 0 seçilir. Algoritma her durumda kol 0'a takılır. Her tur $0{,}7 - 0{,}3 = 0{,}4$ pişmanlık ekler: $R_T = 0{,}4 \cdot T$, doğrusal.
- **ε-açgözlü**: en iyi kolu bulduktan sonra bile turların %10'unda keşif yapar. Keşif turunda kollar eşit olasılıkla seçilir ve ortalama kayıp $(0{,}4 + 0{,}2 + 0)/3 = 0{,}2$'dir. Demek ki her tur yaklaşık $0{,}1 \cdot 0{,}2 = 0{,}02$ pişmanlık ekler; 5000 turda bu yaklaşık 100 eder. Başlangıçtaki öğrenme kayıplarıyla birlikte ortalama 118'dir. Pişmanlık doğrusal büyür, ama eğimi küçüktür.
- **UCB1**: başta daha çok keşif yapar. Pişmanlığı yaklaşık 270. ile 2.370. tur arasında ε-açgözlününkinden yüksektir. Sonra daha düşük kalır, çünkü UCB1'in pişmanlığı **logaritmik**, yani çok yavaş büyür (Auer, Cesa-Bianchi ve Fischer, 2002). Kötü bir kolun güven payı $\sqrt{(2 \ln t / N)}$ olduğundan, o kolun yeniden denenmesi için N'nin ancak $\ln t$ ile orantılı büyümesi gerekir; kötü kollara harcanan çekiş sayısı da bu yüzden yalnızca $\ln t$ kadar artar.

Logaritmik büyümenin ne kadar yavaş olduğu sayılarla görülebilir: $\ln 1000 \approx 6{,}9$, $\ln 1.000.000 \approx 13{,}8$'dir. Tur sayısı bin katına çıkınca logaritma yalnızca iki katına çıkar. Doğrusal pişmanlıkta ise tur sayısı bin katına çıkınca pişmanlık da bin katına çıkar. Uzun oyunlarda bu fark belirleyicidir.

ε bir ayar parametresidir ve bu bölümde üç farklı değer kullanılmıştır: kısa izlemede keşif görünsün diye 0,2, uzun karşılaştırmada 0,1, Çok Kollu Haydut oyununda 0,15. Küçük ε daha az keşif, ama yanlış bir başlangıçtan sonra daha geç düzeltme demektir. Sabit ε'la logaritmik pişmanlık elde edilemez; bunun için ε'un zamanla küçültülmesi gerekir.

Sık yapılan hata

Grafikte 1000. turda ε-açgözlünün daha az pişmanlığı olduğunu görüp "ε-açgözlü daha iyi" sonucuna varmak. Karşılaştırmanın cevabı oyunun uzunluğuna bağlıdır. Kısa oyunlarda basit bir ε-açgözlü iyi iş görebilir; uzun oyunlarda logaritmik pişmanlıklı algoritma doğrusal pişmanlıklı olanı mutlaka geçer.

13.4.5 Oyuna dönüş: keşfin bedeli

Açılıştaki oyuna kuramın cevabıyla dönülebilir. Kollar $\mu = (0{,}3;\ 0{,}5;\ 0{,}7)$ iken 100 turda farklı stratejilerin beklenen ödülü ve pişmanlığı Tablo 13.11'de verilmiştir.

Strateji	Ödül	Pişmanlık
hep rastgele (yalnız keşif)	50	20
0,5'lik kola bağlanmak	50	20
ε-açgözlü, ε = 0,15 (en iyiyi bulduktan sonra)	67	3
hep en iyi kol	70	0

Tablo 13.11: 100 turda beklenen ödül ve pişmanlık, $\mu = (0{,}3;\ 0{,}5;\ 0{,}7)$.

Örnek 13.14

1. **Hep rastgele:** Her tur kollar eşit olasılıkla seçilir; tur başına beklenen ödül $(0{,}3 + 0{,}5 + 0{,}7)/3 = 0{,}5$. 100 turda 50; pişmanlık $70 - 50 = 20$.
2. **0,5'lik kola bağlanmak:** Tur başına 0,5; yine 50 ve 20.
3. **ε-açgözlü (ε = 0,15), en iyi kolu bulduktan sonra:** Tur başına $0{,}85 \cdot 0{,}7 + 0{,}15 \cdot 0{,}5 = 0{,}595 + 0{,}075 = 0{,}67$. 100 turda 67; pişmanlık 3. Buradaki 0,5, keşif turunda rastgele seçilen kolun beklenen ödülüdür.
4. **Hep en iyi kol:** 70; bu, gerçek ortalamaları bilen birinin ulaşabileceği sınırdır.

Tablo, keşfin bir bedeli olduğunu açıkça gösterir. En iyi kolu bilmek 3 puan değerindedir; ε-açgözlü bu 3 puanı, en iyi kolu bulduktan sonra da keşfe devam ettiği için kaybeder. UCB1 keşfi az denenen kollara yönelttiği için bu bedeli zamanla küçültür; pişmanlığı logaritmik büyür.

Açılıştaki soru. Şekil 13.1'deki durumda Kol 3'ün tek başarısız çekişi çok az bilgi verir. Sıradaki tur $t = 9$'dur ve $2 \ln 9 \approx 4{,}39$'dur. UCB1 puanları şöyledir:

- Kol 1 ($Q = 0{,}67$, $N = 3$): $0{,}67 + \sqrt{(4{,}39/3)} = 0{,}67 + 1{,}21 \approx 1{,}88$.
- Kol 2 ($Q = 0{,}25$, $N = 4$): $0{,}25 + \sqrt{(4{,}39/4)} = 0{,}25 + 1{,}05 \approx 1{,}30$.
- Kol 3 ($Q = 0$, $N = 1$): $0 + \sqrt{4{,}39} \approx 2{,}10$.

UCB1 Kol 3'ü seçer. Tek bir başarısızlık, bir kolu bir kenara atmak için yeterli kanıt değildir.

İnsanlar nasıl karar verir? Dört kollu bir haydut deneyinde insan deneklerin seçimleri, belirsizliğe bonus veren UCB benzeri keşiften çok, değer farkına göre olasılıklı seçimle açıklandı (Daw vd., 2006). Bu seçim kuralına **softmax seçimi** denir: bir kolun seçilme olasılığı, tahmini değerinin üstel fonksiyonuyla orantılıdır. Bu kural, Hedge'deki $p_i \propto e^{\eta \cdot G_i}$ ile aynı biçimdedir.

> **Sık yapılan hata**
>
> Tek bir deneyle algoritmaları sıralamak. Ödüller rastgeledir; şanslı bir tohumla ε-açgözlü UCB1'i, şanssız bir tohumla UCB1 ε-açgözlüyü kolayca geçebilir. Karşılaştırma her zaman birçok deneyin ortalamasıyla yapılır.

13.5 Pekiştirmeli öğrenme

Haydut probleminde her tur birbirinden bağımsızdır: bugün hangi kol çekilirse çekilsin, yarın makine aynı makinedir. Pek çok gerçek problemde ise bugünkü eylem yarının durumunu değiştirir. Satrançta bir hamle tahtayı değiştirir; bir robotun adımı onun konumunu değiştirir. Bu kısım, eylemlerin gelecekteki durumu da etkilediği öğrenme problemlerini ele alır.

13.5.1 Hayduttan pekiştirmeli öğrenmeye: Q-öğrenme

Haydut, tek durumlu bir problemdir. **Pekiştirmeli öğrenmede** (*reinforcement learning*, RL) ise eylem, ödülün yanında bir sonraki **durumu** da belirler. Durum, ajanın bulunduğu konumdur: tahtadaki dizilim, bir labirentteki hücre ya da bir robotun eklem açıları. Bir eylemin değeri artık yalnızca hemen aldığı ödülle ölçülemez; vardığı durumun gelecekte sağlayacağı ödüller de hesaba katılmalıdır. Hedefe tek adım uzaklıktaki bir durum, ödül vermese bile değerlidir.

Q-öğrenme (*Q-learning*; Watkins, 1989), her durum–eylem çifti için bir değer tahmini tutar: $Q(s, a)$, "s durumunda a eylemi yapılır, sonra en iyi biçimde devam edilirse toplam ne kazanılır?" sorusunun cevabıdır. Ajan s durumunda a eylemini yapıp r ödülünü alarak s' durumuna geçtiğinde tahmini şöyle günceller:

Tanım 13.8

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a) \right]$$

α **öğrenme oranı**dır: yeni bilgiye ne kadar güvenildiğini belirler. γ **iskonto faktörü**dür: gelecekteki ödülün bugünkü değerini belirler ($0 \le \gamma < 1$). Köşeli parantezin içindeki ilk iki terim, $r + \gamma \cdot \max Q(s', \cdot)$, güncellemenin **hedef** değeridir.

Formülün yapısı haydudun artımlı ortalamasıyla aynıdır: "yeni tahmin = eski tahmin + adım × (hedef − eski tahmin)". Fark hedeftedir. Haydutta hedef yalnızca gözlenen ödüldü. Q-öğrenmede hedef, gözlenen ödül ile varılan durumun en iyi değerinin iskontolu toplamıdır. $\max_{a'} Q(s', a')$ terimi, s''ye varıldıktan sonra en iyi eylemin yapılacağı varsayımını taşır. Bu, Q-öğrenmenin kendi tahminlerini kullanarak öğrenmesi demektir: hedefin bir kısmı da bir tahmindir.

γ'nın anlamı bir örnekle netleşir. $\gamma = 0{,}9$ ise bir adım sonra gelecek 1 puan bugün 0,9, iki adım sonra gelecek 1 puan $0{,}9^2 = 0{,}81$ değerindedir. γ sıfıra yakınsa ajan yalnızca hemen gelen ödüle bakar (miyop davranır); 1'e yakınsa uzak geleceği de önemser.

Örnek 13.15

$Q(s, a) = 2$, $r = 1$, $\gamma = 0{,}9$, $\max Q(s', \cdot) = 4$, $\alpha = 0{,}5$ olsun.

1. Hedef: $r + \gamma \cdot \max Q(s', \cdot) = 1 + 0{,}9 \cdot 4 = 1 + 3{,}6 = 4{,}6$.
2. Fark (hata): $4{,}6 - 2 = 2{,}6$. Tahmin hedefin altında kalmıştır.
3. Güncelleme: $Q \leftarrow 2 + 0{,}5 \cdot 2{,}6 = 2 + 1{,}3 = 3{,}3$.

Tahmin hedefe doğru yarı yol ($\alpha = 0{,}5$) ilerledi. $\alpha = 1$ olsaydı tahmin doğrudan hedefe (4,6) atlardı; $\alpha = 0{,}1$ olsaydı yalnızca 2,26'ya çıkardı.

Çoklu ajan ortamında Q-öğrenmenin önemli bir zayıflığı ortaya çıkar. Tek ajanlı Q-öğrenmenin yakınsama kanıtı, geçişlerin ve ödüllerin zamanla değişmediği, yani **durağan** bir ortam varsayar. Çoklu ajan ortamında ise bir ajanın gördüğü sonuçlar öteki ajanların davranışına bağlıdır. Öteki ajanlar da öğrendikçe oyunun kuralları ajanın gözünde sürekli değişir: dün iyi olan bir eylem bugün kötü olabilir. Ortam **durağan değildir** (*non-stationary*) ve tek ajanlı yakınsama garantileri geçerliliğini yitirir. Bu sorunu yönetmenin bir yolu kendine karşı oyundur (AlphaZero): ajan kendi kopyalarına karşı oynar; böylece rakip de ajanla birlikte gelişir.

13.5.2 Kod izleme: q_learn()

Ortam dört hücreli bir koridordur: s_0, s_1, s_2, s_3. Her hücrede iki eylem vardır: sol ve sağ. s_3 hedeftir; oraya varan geçiş 1 ödül verir ve epizot (bir deneme) biter. Öteki geçişlerin ödülü 0'dır. Fonksiyon, önceden kaydedilmiş epizotları (her biri (s, a, r, s') dörtlülerinin listesi) sırayla işler.

```
def q_learn(episodes, alpha, gamma):
    Q = [[0.0, 0.0] for s in range(4)]  # [sol, sağ]
    for ep in episodes:
        for s, a, r, s2 in ep:
            best = 0 if s2 == 3 else max(Q[s2])
            hedef = r + gamma * best
            Q[s][a] += alpha * (hedef - Q[s][a])
    return Q
```

- `Q`, dört satırlı bir tablodur; her satırda [sol, sağ] eylemlerinin değeri tutulur. Eylem 0 sol, 1 sağdır.
- `best`, varılan durumun en iyi değeridir. Hedefe varılmışsa (`s2 == 3`) epizot biter ve gelecek yoktur; değer 0 alınır.
- `hedef`, $r + \gamma \cdot best$'tir; son satır Q-öğrenme güncellemesidir.

Çağrı `q_learn(ep, 0.5, 0.9)` olsun. `ep` üç epizottan oluşur. İlk ikisinde ajan s_0'dan başlayıp üç kez sağa gider. Üçüncüsünde s_1'de bir kez sola gider, sonra yeniden sağa ilerler.

Epizot	Geçiş	r	best	hedef	Güncelleme
1	s0 → sağ → s1	0	0,000	0,000	Q(s0,sağ) = 0 + 0,5·(0 − 0) = 0,000
1	s1 → sağ → s2	0	0,000	0,000	Q(s1,sağ) = 0,000
1	s2 → sağ → s3	1	0 (hedef)	1,000	Q(s2,sağ) = 0 + 0,5·(1 − 0) = 0,500
2	s0 → sağ → s1	0	0,000	0,000	Q(s0,sağ) = 0,000
2	s1 → sağ → s2	0	0,500	0,450	Q(s1,sağ) = 0 + 0,5·(0,450 − 0) = 0,225
2	s2 → sağ → s3	1	0 (hedef)	1,000	Q(s2,sağ) = 0,5 + 0,5·(1 − 0,5) = 0,750
3	s0 → sağ → s1	0	0,225	0,203	Q(s0,sağ) = 0 + 0,5·(0,203 − 0) = 0,101
3	s1 → sol → s0	0	0,101	0,091	Q(s1,sol) = 0 + 0,5·(0,091 − 0) = 0,046
3	s0 → sağ → s1	0	0,225	0,203	Q(s0,sağ) = 0,101 + 0,5·(0,203 − 0,101) = 0,152
3	s1 → sağ → s2	0	0,750	0,675	Q(s1,sağ) = 0,225 + 0,5·(0,675 − 0,225) = 0,450
3	s2 → sağ → s3	1	0 (hedef)	1,000	Q(s2,sağ) = 0,75 + 0,5·(1 − 0,75) = 0,875

Tablo 13.12: `q_learn` izlemesi ($\alpha = 0{,}5$, $\gamma = 0{,}9$). Her satır bir geçiş ve bir güncellemedir.

Örnek 13.16

1. **Epizot 1:** İlk iki geçişte ödül 0 ve varılan durumların değeri 0'dır; hiçbir şey öğrenilmez. Yalnızca son geçiş ödül 1 getirir ve *Q(s2,sağ) = 0,5* olur. Ödül bilgisi şimdilik yalnızca hedefin bir adım öncesine ulaştı.
2. **Epizot 2:** *s1*'den *s2*'ye geçişte *best = max Q(s2, ·) = 0,5*'tir. Hedef *0 + 0,9 · 0,5 = 0,45*, güncelleme *0,5 · 0,45 = 0,225*. Ödül bir adım daha geriye yayıldı. *s0* hâlâ 0'dır, çünkü onun güncellemesi *Q(s1, ·)* artmadan önce yapıldı.
3. **Epizot 3, ilk geçiş:** *best = 0,225*, hedef *0,9 · 0,225 ≈ 0,203*, *Q(s0,sağ) ≈ 0,101*. Ödül bilgisi başlangıç hücresine ulaştı.

4. **Sola sapma:** *s1*'den sola gidilince *best = max Q(s0, ·) = 0,101*, hedef 0,091, *Q(s1,sol) ≈ 0,046*. Sola gitmek de biraz değer kazandı, çünkü sol, değeri artık sıfır olmayan *s0*'a götürür.
5. **Kalan geçişler:** Ajan yeniden sağa ilerler ve her güncelleme değerleri biraz daha artırır: *Q(s0,sağ) ≈ 0,152*, *Q(s1,sağ) = 0,450*, *Q(s2,sağ) = 0,875*.



Şekil 13.14: Üç epizottan sonra Q tablosu. Her hücrede sol (←) ve sağ (→) eylemlerinin değeri: *s0*: 0,000 / 0,152; *s1*: 0,046 / 0,450; *s2*: 0,000 / 0,875. *s3* hedeftir (r = 1, epizot biter).

Şekil 13.14 Q-öğrenmenin temel davranışını özetler: ödül hedeften geriye doğru yayılır. *Q(s2,sağ) = 0,875 → Q(s1,sağ) = 0,450 → Q(s0,sağ) = 0,152*. Her hücrede *Q(sağ) > Q(sol)* olduğundan, her durumda en büyük Q'lu eylemi seçen **açgözlü politika** "hep sağa" der. Ajan bu politikayı ödülü yalnızca son adımda gördüğü hâlde öğrenmiştir.

Aynı epizotlar sonsuza kadar tekrarlansaydı değerler nereye giderdi? Hep sağa gidildiğinde *Q(s2,sağ)*'nin hedefi her seferinde 1'dir, dolayısıyla *Q(s2,sağ) → 1*. *Q(s1,sağ)*'nin hedefi *0,9 · Q(s2,sağ)*'dir ve *0,9*'a gider. *Q(s0,sağ)* da $0{,}9^2 = 0{,}81$'e gider. Değerler, hedefe uzaklıkla γ'nın kuvvetleri biçiminde azalır: uzaktaki ödül iskonto edilir.

Sık yapılan hata

- **Hedef durumda max Q kullanmak:** Epizot hedefte biter; hedefin geleceği yoktur, *best = 0*'dır.
- **Güncelleme sırasını karıştırmak:** Bir epizot içinde güncellemeler geçiş sırasıyla yapılır. *s0* güncellenirken *Q(s1, ·)* henüz o epizodun eski değerindedir; bu yüzden ödül her epizotta yalnızca bir adım geriye yayılır.
- **Hedef yerine farkı eklemek:** $Q \leftarrow Q + \alpha \cdot hedef$ yanlıştır; eklenen, $\alpha \cdot (hedef - Q)$'dur.

13.5.3 Q-öğrenmenin ötesi: yöntemler ve kilometre taşları

Buraya kadarki algoritmalar geniş bir ailenin en sade üyeleridir. Ailenin öteki önemli üyelerinden bazıları şunlardır:

- **Thompson örneklemesi** (*Thompson sampling*; Thompson, 1933): haydut için Bayesçi bir yöntemdir. Her kol için "ortalaması şu olabilir" biçiminde bir olasılık dağılımı tutar; bu dağılım gözlemlerle güncellenmiş bir son olasılık dağılımıdır. Her turda her kolun dağılımından bir örnek çe-

kilir ve en büyük örneği veren kol oynanır. Belirsiz bir kol bazen yüksek bir tahmin verdiği için kendiliğinden keşfedilir; UCB1'deki iyimserliğin olasılıklı bir karşılığıdır.

- **Kazan-kal, kaybet-değiş** (*Win-Stay Lose-Shift*, WSLS; Nowak ve Sigmund, 1993): Mahkûmun İkilemi için bir kuraldır. 3 ya da 5 puan alındıysa aynı hamle tekrarlanır; 0 ya da 1 alındıysa hamle değiştirilir. Tek bir hatadan sonra iki turda işbirliğine döner.
- **EXP3**: yalnızca çekilen kolun ödülünü gören Hedge'dir; çekişmeli (*adversarial*) haydut için tasarlanmıştır. Ödülleri kötü niyetli bir rakip seçse bile pişmanlığı T'den yavaş büyür (Auer, Cesa-Bianchi, Freund ve Schapire, 2002).
- **SARSA ve DQN**: Q-öğrenmenin akrabalarıdır. DQN, Q tablosu yerine bir sinir ağı kullanır; bu sayede tabloya sığmayacak kadar çok durumu olan problemlerde (örneğin ekran görüntüsünden Atari oyunları) çalışır (Mnih vd., 2015).
- **CFR** (*counterfactual regret minimization*, karşı-olgusal pişmanlık minimizasyonu): pişmanlığa dayalı bir poker yöntemidir. Oyunun her karar noktasında ayrı bir pişmanlık tutar ve bunları küçültür (Zinkevich vd., 2007).

Yıl	Sistem	Yöntem
1992	TD-Gammon (tavla)	kendine karşı oyun
2017–19	Libratus, Pluribus (poker)	CFR (pişmanlık)
2018	AlphaZero (satranç, Go)	kendine karşı oyun

Tablo 13.13: Oyunlarda öğrenmenin kilometre taşları.

Oyunlar pekiştirmeli öğrenme için iyi bir laboratuvardır: kurallar açıktır, ödül nettir (kazanmak ya da kaybetmek), oyun milyonlarca kez tekrarlanabilir. Tablo 13.13 iki yaklaşımın öne çıktığını gösterir. Tam bilgili oyunlarda (tavla, satranç, Go) kendine karşı oyun başarılı oldu. Eksik bilgili oyunlarda, özellikle pokerde, rakibin kartları görülmediği için pişmanlık tabanlı yöntemler öne çıktı. Pokerdeki başarı bu bölümün kuramıyla doğrudan bağlantılıdır: CFR, iki oyunculu sıfır toplamlı bir oyunda pişmanlığı küçülttüğü için zaman ortalaması minimaks stratejisine yaklaşır.

13.6 Evrimsel dinamik

Şimdiye kadarki bütün modellerde öğrenen bir birey vardı: sayaç tutan, ağırlık güncelleyen, tahmin yapan bir oyuncu. Evrimsel oyun kuramında birey hiçbir şey öğrenmez. Her birey doğuştan tek bir strateji taşır ve onu ömrü boyunca oynar. Değişen, nüfusun bileşimidir: başarılı stratejiyi taşıyan bireyler daha çok yavru bırakır ve o strateji yayılır. Rasyonellik yoktur, seçilim vardır. Şaşırtıcı olan, bu “akılsız” sürecin çoğu zaman Nash dengesinin öngördüğü yere varmasıdır.

13.6.1 Replikatör dinamiği ve ESS

Model şöyledir. Büyük bir nüfusta bireyler rastgele ikili eşleşir ve aralarında bir oyun oynar. Her birey tek bir strateji oynar ve onu yavrularına aktarır. Bir stratejinin başarısı, onu oynayan bireyin nüfustan rastgele seçilen bir rakibe karşı ortalama kazancıyla ölçülür.

Tanım 13.9

Nüfusta i stratejisinin payı x_i olsun. f_i, i stratejisinin ortalama kazancıdır ve **uygunluk** (*fitness*) adını alır; $\bar{f} = \Sigma_j \, x_j \, f_j$ nüfusun ortalama kazancıdır. Kesikli **replikatör dinamiği** (*replicator dynamics*) her nesilde payları şöyle günceller:

$$\text{yeni pay} = \text{eski pay} \times f_i / \bar{f}, \text{ yani } x_i' = x_i \cdot f_i / \bar{f}$$

Formül, yavru sayısının uygunlukla orantılı olduğu varsayımından çıkar. Nüfusta i stratejisini oynayan n_i birey olsun ve her biri f_i yavru bıraksın. Yeni nesilde $n_i \, f_i$ birey i'yi oynar. Yeni pay, bunun toplam yavru sayısına oranıdır: $n_i \, f_i \, / \, \Sigma_j \, n_j \, f_j$. Pay ve paydayı toplam nüfus n'ye bölmek $x_i \, f_i \, / \, \Sigma_j \, x_j \, f_j = x_i \, f_i \, / \, \bar{f}$ verir.

Formülün yorumu tek cümledir: **ortalamanın üstünde kazanan strateji yayılır**. $f_i > \bar{f}$ ise $f_i/\bar{f} > 1$ ve pay artar; $f_i < \bar{f}$ ise pay azalır. Paylar her zaman toplamı 1 olan bir dağılım olarak kalır. Formülün anlamlı olması için uygunlukların pozitif olması gerekir: negatif bir uygunluk negatif yavru sayısı demek olurdu. Bu yüzden burada her kazanca sabit bir **taban** (3) eklenir. Tabanın eklenmesi stratejilerin sıralamasını ve dinlenme noktalarını değiştirmez; yalnızca dinamiğin hızını etkiler. (Sürekli zamanlı sürümde $\dot{x}_i = x_i(f_i - \bar{f})$ yazılır; burada kesikli biçim yeterlidir.)

Evrimsel bakışın denge kavramı **evrimsel kararlı strateji**dir (*evolutionarily stable strategy*, ESS; Maynard Smith ve Price, 1973).

Tanım 13.10

Bir x stratejisi, nüfusun tamamı x oynarken ortaya çıkan ve başka bir y stratejisi oynayan küçük bir mutant grubu nüfusu istila edemiyorsa **evrimsel kararlıdır**. Biçimsel olarak her $y \neq x$ için

$$u(x, x) > u(y, x) \text{ ya da } [u(x, x) = u(y, x) \text{ ve } u(x, y) > u(y, y)]$$

olmalıdır; burada $u(a, b)$, a oynayan bireyin b oynayan rakibe karşı kazancıdır.

Koşulun iki parçası mutantın neden yayılamadığını açıklar. Mutantlar azken hemen her eşleşmeleri yerleşik x'lerle olur. İlk koşul, mutantın bu eşleşmelerde yerleşiklerden kötü kazandığını söyler; mutant hemen geriler. İlk koşulda eşitlik varsa belirleyici olan seyrek mutant–mutant eşleşmeleridir:

ikinci koşul, mutantla karşılaşınca yerleşiğin mutanttan iyi kazandığını söyler. Her ESS bir simetrik Nash dengesidir (ilk koşulun eşitsizliği ≥ olarak her zaman sağlanır); ama her Nash dengesi ESS değildir.

Örnek oyun **Şahin–Güvercin**'dir. Bireyler değeri $V = 4$ olan bir kaynak için karşılaşır. Şahin her zaman kavga eder; Güvercin gösteriş yapar ama kavga başlarsa çekilir. Kavga etmenin maliyeti $C = 6$'dır.

- İki şahin kavga eder: biri kazanır (V), biri yaralanır ($-C$); ortalama $(V - C)/2 = (4 - 6)/2 = -1$.
- Şahin güvercine karşı kaynağı alır: $V = 4$; güvercin 0 alır.
- İki güvercin kaynağı paylaşır: $V/2 = 2$.

Birey \ Rakip	Şahin	Güvercin
Şahin	(−1, −1)	(4, 0)
Güvercin	(0, 4)	(2, 2)

Şekil 13.15: Şahin–Güvercin oyunu ($V = 4, C = 6$). Satır: birey, sütun: rakip; hücrede (birey, rakip) kazancı.

Şekil 13.15'teki oyunda hiçbir saf strateji ESS değildir. Bütün nüfus güvercinse, bir şahin mutant her eşleşmede 4 alır, güvercinler ise 2 alır; şahinler yayılır. Bütün nüfus şahinse, bir güvercin mutant 0 alır, şahinler ise −1 alır; güvercinler yayılır. İki uçta da azınlıkta olan strateji avantajlıdır. Bu yüzden nüfus bir karışımda dengelenmelidir.

13.6.2 Şahin–Güvercin adım adım

Şahin payı x, güvercin payı $1 - x$ olsun. Bir şahin, rakibi x olasılıkla şahin, $1 - x$ olasılıkla güvercin olan bir eşleşmeye girer. Taban 3 eklenmiş uygunluklar:

$$f_Ş = 3 + (-1) \cdot x + 4 \cdot (1 - x) = 7 - 5x, f_G = 3 + 0 \cdot x + 2 \cdot (1 - x) = 5 - 2x$$

Nüfus ortalaması $\bar{f} = x \cdot f_Ş + (1 - x) \cdot f_G$, yeni şahin payı $x' = x \cdot f_Ş / \bar{f}$'dir.

Örnek 13.17

Başlangıçta şahinler nüfusun %10'u olsun ($x = 0{,}100$).

1. Uygunluklar: $f_Ş = 7 - 5 \cdot 0{,}1 = 6{,}50, f_G = 5 - 2 \cdot 0{,}1 = 4{,}80$.
2. Ortalama: $\bar{f} = 0{,}1 \cdot 6{,}50 + 0{,}9 \cdot 4{,}80 = 0{,}65 + 4{,}32 = 4{,}97$.

3. Yeni pay: $x' = 0{,}1 \cdot 6{,}50 / 4{,}97 = 0{,}65 / 4{,}97 \approx 0{,}131$.

4. $f_Ş > \bar{f}$ olduğu için şahinler arttı. Bir sonraki nesilde $x = 0{,}131$ ile aynı hesap yapılır: $f_Ş = 6{,}35$, $f_G = 4{,}74$, $\bar{f} \approx 4{,}95$, $x' \approx 0{,}168$.

nesil	x (şahin payı)	$f_Ş$	f_G	$\bar{f}$	yeni x = $x \cdot f_Ş / \bar{f}$
0	0,100	6,50	4,80	4,97	0,131
1	0,131	6,35	4,74	4,95	0,168
2	0,168	6,16	4,66	4,92	0,210
3	0,210	5,95	4,58	4,87	0,257
4	0,257	5,72	4,49	4,80	0,306
5	0,306	5,47	4,39	4,72	0,354
6	0,354	5,23	4,29	4,62	0,401
7	0,401	5,00	4,20	4,52	0,443

Tablo 13.14: Şahin–Güvercin'de replikatör dinamiği ($V = 4$, $C = 6$, taban 3), $x_0 = 0{,}1$.

Tablo 13.14 iki eğilimi birlikte gösterir. Şahin payı her nesil artar, çünkü her satırda $f_Ş > \bar{f}$'dır. Ama şahinlerin avantajı giderek küçülür: $f_Ş$ 6,50'den 5,00'e, $f_Ş - f_G$ farkı 1,70'ten 0,80'e iner. Şahinler çoğaldıkça birbirleriyle daha sık karşılaşır ve pahalı kavgalara girer. Artış yavaşlar; nüfus bir dinlenme noktasına yaklaşır.

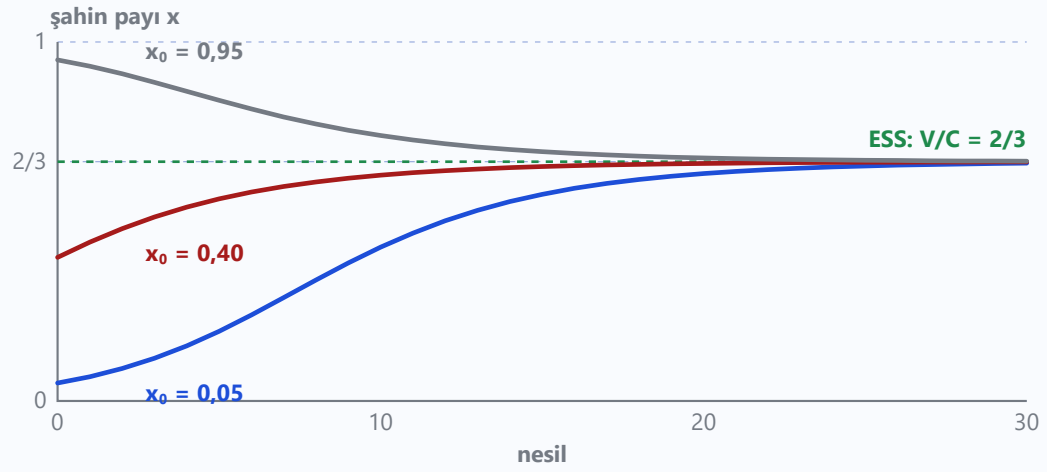
13.6.3 Her başlangıçtan ESS'ye

Dinlenme noktası, iki stratejinin eşit kazandığı paydır; o noktada $f_Ş = f_G = \bar{f}$ olur ve pay değişmez:

$$7 - 5x = 5 - 2x \Rightarrow 2 = 3x \Rightarrow x^* = 2/3$$

Genel olarak (taban eklenmeden) $(V - C)/2 \cdot x + V(1 - x) = (V/2)(1 - x)$ denklemi $x^* = V/C$ verir; burada $4/6 = 2/3$'tür. Bu değer, oyunun **simetrik karışık Nash dengesiyle** aynıdır: karışık dengede bir oyuncunun iki stratejisi arasında kayıtsız olması gerekir, bu da tam olarak $f_Ş = f_G$ koşuludur.

Dinlenme noktası kararlıdır. $x < 2/3$ iken $7 - 5x > 5 - 2x$, yani şahinler ortalamanın üstünde kazanır ve artar. $x > 2/3$ iken eşitsizlik tersine döner ve şahinler azalır. Her iki yönden de nüfus $2/3$'e itilir.



Şekil 13.16: Üç farklı başlangıçtan replikatör dinamiği: $x_0 = 0{,}05$ (mavi), 0,40 (kırmızı), 0,95 (gri). Otuz nesilde üç eğri de ESS $x^* = V/C = 2/3$'e yaklaşır.

Örnek 13.18

$x^* = 2/3$ karışımının ESS koşulunu sağladığını taban eklenmemiş kazançlarla doğrulayalım.

1. **İlk koşul:** x^*'a karşı şahin $\tfrac{2}{3} \cdot (-1) + \tfrac{1}{3} \cdot 4 = 2/3$, güvercin $\tfrac{2}{3} \cdot 0 + \tfrac{1}{3} \cdot 2 = 2/3$ kazanır. Karışık dengede her mutant x^*'a karşı aynı kazancı alır: $u(y, x^*) = u(x^*, x^*) = 2/3$. Eşitlik vardır; ikinci koşula bakılır.
2. **Şahin mutant:** $u(x^*, Ş) = \tfrac{2}{3} \cdot (-1) + \tfrac{1}{3} \cdot 0 = -2/3$, $u(Ş, Ş) = -1$. $-2/3 > -1$: sağlanır.
3. **Güvercin mutant:** $u(x^*, G) = \tfrac{2}{3} \cdot 4 + \tfrac{1}{3} \cdot 2 = 10/3$, $u(G, G) = 2$. $10/3 > 2$: sağlanır.

Her iki mutanta karşı da koşul sağlanır; $x^* = 2/3$ bir ESS'dir.

Şekil 13.16 bu oyunda her iç başlangıcın ESS'ye gittiğini gösterir. Her oyun bu kadar uysal değildir. **Taş-Kağıt-Makas**'ta replikatör dinamiği dengenin etrafında döner: taşlar çoğalınca kağıtlar avantaj kazanır, kağıtlar çoğalınca makaslar, makaslar çoğalınca taşlar; nüfus bir yere yerleşmez. **Geyik Avı**'nda ise dinamik başlangıca göre iki saf dengeden birine gider: işbirlikçiler (geyik avcıları) başta yeterince çoksa geyik dengesi, azsa tavşan dengesi oluşur.

Sık yapılan hata

- **Payı uygunlukla çarpıp bölmeyi unutmak:** $x \cdot f_Ş$ tek başına yeni pay değildir; $\bar{f}$'ya bölünmeden payların toplamı 1 olmaz.
- **$\bar{f}$'yı iki uygunluğun basit ortalaması sanmak:** $\bar{f} = x \cdot f_Ş + (1 - x) \cdot f_G$ ağırlıklı bir ortalamadır; $(f_Ş + f_G)/2$ değildir.

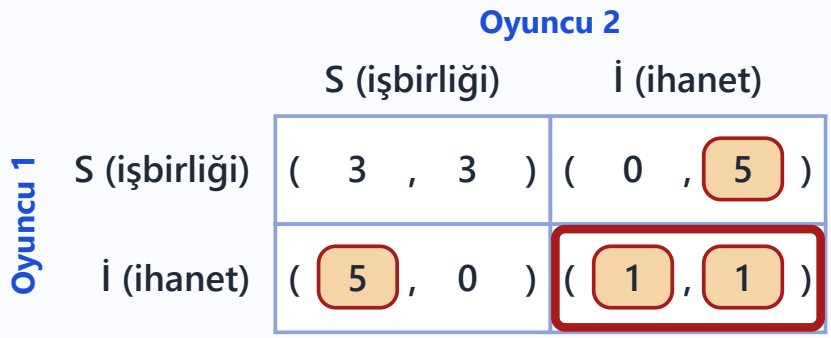
- **"Şahin her eşleşmede kazanıyor, öyleyse herkes şahin olur" demek:** Şahin güvercine karşı kazanır, ama şahinler çoğaldıkça şahin–şahin kavgaları ortalamayı düşürür. Uygunluk nüfusun bileşimine bağlıdır.

13.7 Tekrarlanan oyunlar

Mahkûmun İkilemi tek seferlik bir oyun olarak oyun kuramının en karamsar sonuçlarından birini verir: iki oyuncu da ihanet eder ve ikisi de işbirliğinden kötü durumda kalır. Oysa gerçek hayatta aynı kişilerle, aynı firmalarla, aynı komşularla tekrar tekrar karşılaşırız. Bu kısım, tekrarın işbirliğini nasıl mümkün kıldığını, hangi stratejilerin başarılı olduğunu ve öğrenen bir nüfusta işbirliğinin nasıl yayıldığını inceler.

13.7.1 Tekrarlanan Mahkûmun İkilemi ve kinci tetik

Burada Mahkûmun İkilemi puanlarla yazılır; yüksek puan iyidir. S (sessiz kalmak) işbirliği, İ ihanettir.



Şekil 13.17: Mahkûmun İkilemi puanları. (S, S) = (3, 3), (S, İ) = (0, 5), (İ, S) = (5, 0), (İ, İ) = (1, 1). İşaretli sayılar en iyi tepkilerdir; tek Nash dengesi (İ, İ)'dir.

Şekil 13.17'deki dört sayının adları vardır: **cazibe** 5 (rakip işbirliği yaparken ihanet etmek), **karşılıklı işbirliği** 3, **karşılıklı ihanet** 1, **enayi** 0 (rakip ihanet ederken işbirliği yapmak). Sıralama cazibe 5 > karşılıklı işbirliği 3 > karşılıklı ihanet 1 > enayi 0'dır. İhanet baskın stratejidir: rakip S oynarsa İ 5, S 3 verir; rakip İ oynarsa İ 1, S 0 verir.

Sonlu ve bilinen sayıda tekrar sonucu değiştirmez. Oyunun tam 10 tur süreceğini iki oyuncu da biliyorsa, son tur bir tek seferlik oyundur ve iki taraf da ihanet eder. 9. turda oyuncular son turun ne olacağını bilir; 9. turdaki davranış 10. turu etkilemez, o hâlde 9. tur da tek seferlik oyun gibidir ve ihanet edilir. Bu **geriye doğru çıkarım** ilk tura kadar sürer: her turda ihanet.

Durum, oyunun ne zaman biteceği bilinmediğinde değişir. Oyun her turdan sonra δ olasılıkla bir tur daha sürsün (ve $1-\delta$ olasılıkla bitsin). δ'ya **iskonto faktörü** denir; bir sonraki turun bugünkü değerini ölçer. Bu duruma **geleceğin gölgesi** de denir: gelecek ne kadar olasıysa, bugünkü davranışın üzeri-

ne o kadar uzun bir gölge düşer. Oyunun beklenen süresi $1/(1-\delta)$ turdur; örneğin $\delta = 0{,}9$ için ortalama 10 tur.

Tanım 13.11

Kinci tetik stratejisi (*grim trigger*, Grim): ilk turda işbirliği yapmak; rakip bir kez bile ihanet ederse o andan itibaren sonsuza kadar ihanet etmektir.

İki oyuncu da kinci tetik oynarken birinin sapmasının kârlı olup olmadığını hesaplayalım. Hesapta **geometrik seri** kullanılır. $S = a + a\delta + a\delta^2 + \dots$ toplamında her terim bir öncekinin δ katıdır; bu yüzden $S = a + \delta S$ yazılabilir ve $S = a/(1-\delta)$ bulunur ($0 \le \delta < 1$ iken).

Teorem 13.4 (Kinci tetikle işbirliği)

Puanlar cazibe 5, karşılıklı işbirliği 3, karşılıklı ihanet 1 iken iki oyuncunun da kinci tetik oynaması ancak ve ancak $\delta \ge 1/2$ ise bir dengedir. Genel olarak cazibe c ise eşik $\delta^* = (c-3)/(c-1)$'dir.

İspat. **Uymak:** İki taraf da kinci tetiğe uyarsa her tur karşılıklı işbirliği oynanır ve oyuncu her tur 3 alır. Beklenen toplam $3 + 3\delta + 3\delta^2 + \dots = 3/(1-\delta)$'dir.

Sapmak: Oyuncu bu tur ihanet ederse cazibe 5'i alır. Rakip kinci tetik oynadığı için sonraki her turda ihanet eder; o turlarda oyuncunun en iyi tepkisi de ihanettir ve karşılıklı ihanetle 1 alır. Toplam $5 + \delta + \delta^2 + \dots = 5 + \delta \cdot 1/(1-\delta) = 5 + \delta/(1-\delta)$'dir. (Daha sonraki bir turda sapmak aynı hesabı o turdan başlatır; sonuç değişmez.)

Karşılaştırma: Uymak ≥ sapmak koşulu $3/(1-\delta) \ge 5 + \delta/(1-\delta)$'dır. İki taraf pozitif $(1-\delta)$ ile çarpılınca eşitsizlik yön değiştirmez: $3 \ge 5(1-\delta) + \delta = 5 - 4\delta$. Buradan $4\delta \ge 2$, yani $\delta \ge 1/2 = (5-3)/(5-1)$ çıkar. Cazibe c ile aynı adımlar $3 \ge c(1-\delta) + \delta$, yani $\delta(c-1) \ge c - 3$ verir.

Cezanın kendisi: Sapmadan sonra iki tarafın da hep ihanet etmesi, tek turluk oyunun Nash dengesi olan (İ, İ)'nin tekrarıdır. Ceza evresinde hiçbir oyuncunun tek başına S'ye dönmesi kârlı değildir (1 yerine 0 alır); bu yüzden ceza tehdidi inandırıcıdır. □

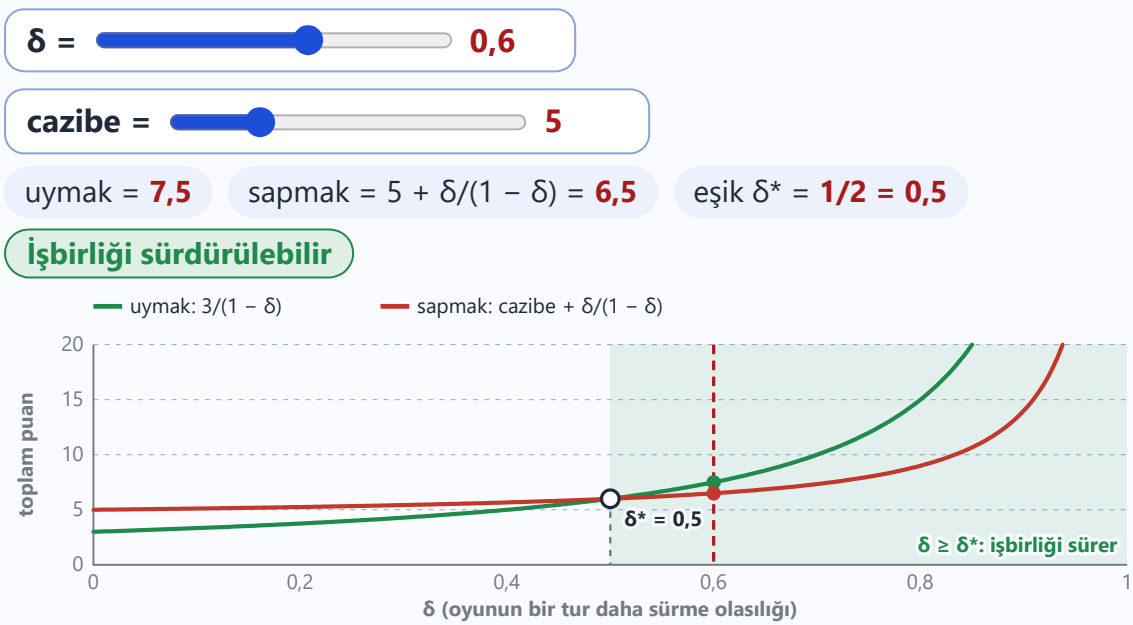
Örnek 13.19

1. $\delta = 0{,}9$**:** uymak $3/0{,}1 = 30$; sapmak $5 + 0{,}9/0{,}1 = 5 + 9 = 14$. $30 > 14$: işbirliği sürer.
2. $\delta = 0{,}3$**:** uymak $3/0{,}7 \approx 4{,}29$; sapmak $5 + 0{,}3/0{,}7 \approx 5 + 0{,}43 = 5{,}43$. Sapmak daha iyidir: ihanet.
3. **Eşikte,** $\delta = 0{,}5$**:** uymak $3/0{,}5 = 6$; sapmak $5 + 0{,}5/0{,}5 = 6$. İki seçenek eşittir.

Bu sonuç çok daha genel bir olgunun özel hâlidir. **Halk teoremi** (*folk theorem*) şunu söyler: δ yeterince büyükse, her oyuncuya ihanetle garantileyebileceğinden (burada 1'den) fazlasını veren pek çok ortalama kazanç, tekrarlanan oyunun bir dengesinde sürdürülebilir. Adı, sonucu ilk kimin kanıtladığının bilinmemesinden ve sonucun uzmanlar arasında bir "halk bilgisi" gibi dolaşmasından gelir. Teoremin iyi ve kötü bir yüzü vardır. İyi yüzü: işbirliği mümkündür. Kötü yüzü: çok sayıda denge vardır ve kuram hangisinin oynanacağını söylemez.

13.7.2 İşbirliği hangi δ'da sürer?

Eşiğin cazibeye nasıl bağlı olduğu, cazibeyi değiştirerek görülebilir. Uymak $3/(1-\delta)$, sapmak $\text{cazibe} + \delta/(1-\delta)$ değerindedir; uymak ≥ sapmak ancak ve ancak $\delta \geq \delta^* = (\text{cazibe}-3)/(\text{cazibe}-1)$.



Şekil 13.18: Etkileşimli kinci tetik grafiği. Yeşil eğri uymanın değeri $3/(1-\delta)$, kırmızı eğri sapmanın değeri $\text{cazibe} + \delta/(1-\delta)$. İki eğrinin kesiştiği nokta eşik δ^*'dır; sağındaki yeşil bölgede işbirliği sürer. Kaydırıcılar δ'yı ve cazibeyi değiştirir. Varsayılan değerler $\delta = 0{,}6$ ve cazibe 5'tir: uymak 7,5, sapmak 6,5, eşik 0,5.

Şekil 13.18'deki grafikte iki eğri $\delta = 0$'da 3 ve cazibe değerinden başlar. Sapma eğrisi daha yüksekten başlar ama daha yavaş yükselir, çünkü sapan oyuncu gelecekte her tur 3 yerine 1 alır. Gelecek ne kadar değerliyse (δ büyükse) kinci tetik tehdidi o kadar caydırıcıdır. Cazibe arttıkça işbirliği için gereken eşik de yükselir: cazibe 4 iken $\delta^* = 1/3$, cazibe 9 iken $\delta^* = 6/8 = 0{,}75$'tir. İhanetin anlık kazancı büyüdükçe, onu caydırmak için geleceğin daha olası olması gerekir.

13.7.3 On turluk maçlar

Kinci tetik işbirliğini korur ama çok serttir: tek bir ihanet sonsuza kadar cezalandırılır. Daha yumuşak ve çok ünlü bir strateji **Tit-for-Tat**'tır (TFT, "kısasa kısas"): ilk turda işbirliği yapar, sonra rakibin son hamlesini tekrarlar. Karşılaştırma için beş strateji tanımlayalım:

- **Hep İşbirliği** (Hep-S) ve **Hep İhanet** (Hep-İ): rakibe bakmadan hep aynı hamle.
- **Tit-for-Tat** (TFT): S ile başlar, sonra rakibin son hamlesini kopyalar.
- **Kinci** (Grim): S ile başlar; rakip bir kez ihanet ederse hep İ oynar.
- **Şüpheci TFT** (ŞTFT): İ ile başlar, sonra rakibin son hamlesini kopyalar.



Şekil 13.19: Beş on turluk maç. Yeşil S (işbirliği), kırmızı İ (ihanet). Yukarıdan aşağı: TFT – Hep İşbirliği (30 – 30), TFT – Hep İhanet (9 – 14), TFT – Şüpheci TFT (25 – 25), Kinci – Şüpheci TFT (13 – 13), Hep İhanet – Hep İşbirliği (50 – 0).

Şekil 13.19'daki maçların puanları şöyle oluşur:

1. **TFT – Hep İşbirliği:** TFT işbirliği yapana işbirliğiyle karşılık verir. Her tur (S, S): iki taraf da $10 \cdot 3 = 30$.

2. **TFT – Hep İhanet:** İlk tur TFT S, rakip İ: TFT 0, rakip 5. Sonraki dokuz tur TFT rakibin İ'sini kopyalar ve (İ, İ) sürer: her tur 1'er. TFT *0 + 9 = 9*, rakip *5 + 9 = 14*. TFT ilk turda sömürülür, sonra hep ihanet eder.
3. **TFT – Şüpheci TFT:** Şüpheci TFT ihanetle başlar. İki TFT birbirini "yankılar": TFT rakibin İ'sini kopyalarken rakip TFT'nin S'sini kopyalar; S/İ dönüşümlü sürer. Her oyuncu beş tur 5, beş tur 0 alır: ikisi de 25.
4. **Kinci – Şüpheci TFT:** 1. tur Kinci S, ŞTFT İ: 0 – 5. 2. tur Kinci affetmez ve İ oynar, ŞTFT Kinci'nin S'sini kopyalar: 5 – 0. Sonraki sekiz tur ikisi de İ: 8'er. Toplam 13 – 13.
5. **Hep İhanet – Hep İşbirliği:** Saf işbirlikçi tamamen sömürülür: *10 · 5 = 50*'ye *0*.

Bu maçlardan dikkat çekici bir özellik çıkar: **TFT bir maçı asla kazanamaz**. TFT hiçbir zaman ilk ihaneti yapmaz ve rakibin her ihanetine en fazla bir ihanetle cevap verir; bu yüzden bir maçta rakibinden fazla puan toplayamaz. En iyi durumda berabere kalır. Yine de turnuvayı kazanır (13.7.5).

13.7.4 Kod izleme: match()

Maçları oynatan kod kısadır. `PAY` sözlüğü, iki hamlenin birleşimini (örneğin `"Sİ"`) iki oyuncunun puanlarına eşler. Stratejiler, kendi geçmişini ve rakibin geçmişini alan küçük fonksiyonlardır.

```
PAY = {"SS": (3, 3), "Sİ": (0, 5),
       "İS": (5, 0), "İİ": (1, 1)}
tft = lambda ben, rakip: rakip[-1] if rakip else "S"
stft = lambda ben, rakip: rakip[-1] if rakip else "İ"
def match(s1, s2, T):
    h1, h2, p1, p2 = [], [], 0, 0
    for t in range(T):
        a, b = s1(h1, h2), s2(h2, h1)
        u, v = PAY[a + b]
        h1.append(a); h2.append(b)
        p1, p2 = p1 + u, p2 + v
    return p1, p2
```

- `tft`: rakibin geçmişi boş değilse rakibin son hamlesini (`rakip[-1]`) döndürür, boşsa `"S"`. `stft` aynı kuraldır ama boş geçmişte `"İ"` döndürür.
- `match`: her turda iki strateji kendi bakış açısından çağrılır. Birinci oyuncu `s1(h1, h2)`, ikinci oyuncu `s2(h2, h1)` ile çağrılır; böylece her strateji önce kendi, sonra rakibin geçmişini alır.
- `PAY[a + b]`, iki hamle harfini birleştirip puan çiftini bulur. Hamleler geçmişlere eklenir ve puanlar toplanır.

t	TFT	ŞTFT	Gerekçe	PAY	Puanlar
0	S	İ	geçmiş boş: TFT S, ŞTFT İ ile açar	Sİ = (0, 5)	0 – 5
1	İ	S	ikisi de rakibin son hamlesini kopyalar	İS = (5, 0)	5 – 5
2	S	İ	kopyalama	Sİ = (0, 5)	5 – 10

t	TFT	ŞTFT	Gerekçe	PAY	Puanlar
3	İ	S	kopyalama	İS = (5, 0)	10 – 10
4	S	İ	kopyalama	Sİ = (0, 5)	10 – 15
5	İ	S	kopyalama	İS = (5, 0)	15 – 15

Tablo 13.15: `match(tft, stft, 6)` izlemesi.



Şekil 13.20: `match(tft, stft, 6)` sonucu: TFT S İ S İ S İ, Şüpheci TFT İ S İ S İ S; puanlar 15 – 15.

Fonksiyon `(15, 15)` döndürür. Şekil 13.20 bir yankı döngüsünü gösterir: tek bir ilk ihanet, iki kopyalayıcı strateji arasında sonsuza kadar süren bir "Sİ / İS" dönüşümü başlatır. Tur başına ortalama puan *(0 + 5)/2 = 2,5*'tir; karşılıklı işbirliğinde 3 olurdu. Döngüyü kırabilecek tek şey, iki stratejiden birinin bir kez "affetmesi"dir; TFT ve ŞTFT bunu yapmaz.

13.7.5 Mini Axelrod turnuvası

Beş strateji, kendi ikizi dahil herkesle on turluk birer maç yapsın. Tablo 13.16 her satırdaki stratejinin, sütundaki rakibe karşı on turda topladığı puanı verir.

Strateji \ rakip	Hep-S	Hep-İ	TFT	Kinci	ŞTFT	Toplam	Sıra
Hep İşbirliği	30	0	30	30	27	117	2
Hep İhanet	50	10	14	14	10	98	4
Tit-for-Tat	30	9	30	30	25	124	1
Kinci (Grim)	30	9	30	30	13	112	3
Şüpheci TFT	32	10	25	13	10	90	5

Tablo 13.16: Mini Axelrod turnuvası. Satırdaki stratejinin sütundaki rakibe karşı 10 turda topladığı puan. Hep-S: Hep İşbirliği, Hep-İ: Hep İhanet, ŞTFT: Şüpheci TFT.

Örnek 13.20

Tablonun iki hücresini ayrıntılı hesaplayalım.

1. **Hep İşbirliği – Şüpheci TFT = 27:** 1. tur Hep-S S, ŞTFT İ: Hep-S 0 alır. 2. turdan itibaren ŞTFT, Hep-S'nin S'sini kopyalar ve dokuz tur (S, S) oynanır: *9 · 3 = 27*. Toplam *0 + 27 = 27*. Aynı maçta ŞTFT *5 + 27 = 32* alır; bu, ŞTFT satırının ilk hücresidir.
2. **Hep İhanet – TFT = 14:** 1. tur Hep-İ 5 alır; sonra dokuz tur (İ, İ): 9. Toplam 14. Kinci'ye karşı da aynıdır.

3. **Toplamlar:** TFT $30 + 9 + 30 + 30 + 25 = 124$; Hep İhanet $50 + 10 + 14 + 14 + 10 = 98$.

Sıralama: Tit-for-Tat (124), Hep İşbirliği (117), Kinci (112), Hep İhanet (98), Şüpheci TFT (90). **TFT hiçbir maçı kazanmadan birinci olur.** Hep İhanet ise her maçı kazanır ya da berabere bitirir, ama sondan ikinci olur. Bunun açıklaması puanların nereden geldiğindedir. Hep İhanet tek tek maçları kazanır ama kazandığı maçlarda az puan toplar: işbirlikçilerle karşılaşınca bir kez sömürür, sonra karşılıklı ihanete düşer ve tur başına 1 alır. TFT ise işbirlikçilerle uzun karşılıklı işbirliği kurar ve tur başına 3 toplar. Turnuva, rakibi yenmeyi değil, toplam puanı ödüllendirir.

Turnuva tablosu evrimsel bir hikâyeye de dönüştürülebilir. Başlangıçta her strateji nüfusun beşte birini oluştursun ve her stratejinin uygunluğu, nüfustan rastgele seçilen bir rakibe karşı ortalama puanı olsun: TFT $124/5 = 24{,}8$, Hep-S 23,4, Kinci 22,4, Hep-İ 19,6, ŞTFT 18,0. Nüfus ortalaması $\bar{f} = 541/25 = 21{,}64$'tür. Replikatör kuralıyla bir nesil sonra paylar TFT $0{,}2 \cdot 24{,}8/21{,}64 \approx 0{,}229$, Hep-S $\approx 0{,}216$, Kinci $\approx 0{,}207$, Hep-İ $\approx 0{,}181$, ŞTFT $\approx 0{,}166$ olur. Ortalamanın üstünde kazanan üç "iyi niyetli" strateji yayılır; ihanetle başlayanlar geriler.

13.7.6 Axelrod turnuvaları ve dersleri

Mini turnuva gerçek bir deneyin küçük bir kopyasıdır. Siyaset bilimci Robert Axelrod, 1980'de iki bilgisayar turnuvası düzenledi. Oyun kuramcıları, matematikçiler ve bilgisayar programcıları tekrarlanan Mahkûmun İkilemi için program gönderdi: ilk turnuvaya 14, ikincisine 62 program katıldı. İkisini de Anatol Rapoport'un dört satırlık Tit-for-Tat programı kazandı. İkinci turnuvanın katılımcıları ilk turnuvanın sonuçlarını biliyor ve TFT'yi yenmeye çalışıyordu; yine de başaramadılar (Axelrod, 1984).

Axelrod başarılı stratejilerin dört ortak özelliğini sıraladı:

- **İyi niyetli:** İlk ihaneti yapmaz. Mini turnuvada iyi niyetli üç strateji (TFT, Hep-S, Kinci) ilk üç sırayı aldı.
- **Misilleme yapan:** İhaneti cezasız bırakmaz. Hep İşbirliği misilleme yapmadığı için Hep İhanet karşısında 0 aldı.
- **Affedici:** Rakip işbirliğine dönerse o da döner. Kinci affetmediği için Şüpheci TFT'ye karşı yalnız 13 aldı; TFT 25 aldı.
- **Anlaşılır:** Rakip, kuralını kolayca çözebilir. TFT'nin kuralı birkaç turda anlaşılır; rakip ona karşı işbirliğinin en iyisi olduğunu çabuk öğrenir.

TFT'nin zayıf noktası **gürültüdür**, yani bir oyuncunun yanlışlıkla istemediği hamleyi yapma olasılığıdır. Gürültülü bir ortamda iki TFT, tek bir hatadan sonra sonsuz bir misilleme döngüsüne girer.



Şekil 13.21: İki TFT, 4. turda A'nın tek bir hatası (S yerine İ). Sonrasında sonsuz S/İ yankısı başlar: on turda 29 – 24 (hatasız maçta 30 – 30).

Şekil 13.21'deki maçta ilk üç tur (S, S)'dir: 9'ar puan. 4. turda A yanlışlıkla İ oynar: A 5, B 0 alır. 5. turda B, A'nın İ'sini kopyalar; A ise B'nin S'sini kopyalar: A 0, B 5. Bundan sonra iki oyuncu sırayla birbirine ihanet eder. 4.–10. turlar arasındaki yedi turda A dört kez (4, 6, 8, 10), B üç kez (5, 7, 9) 5 alır. Toplamlar $9 + 20 = 29$ ve $9 + 15 = 24$'tür; iki oyuncunun toplamı 60'tan 53'e düşer. Hatayı yapan A bile kaybeder: yankı döngüsünde tur başına ortalama 2,5 alır, bu da karşılıklı işbirliğinin 3'ünden azdır. Çare olarak **cömert TFT** (*generous Tit-for-Tat*) önerilir: rakibin ihanetini bazen (belli bir olasılıkla) affeder ve yankı döngüsünü kırar; gürültülü ortamda TFT'den daha iyi sonuç verir. Kazan-kal, kaybet-değiş (WSLS) de tek hatadan iki turda işbirliğine döner.

Son bir ders: **başarı ortama bağlıdır**. Hiçbir strateji her rakip havuzunda en iyi değildir. Rakip havuzu değişirse sıralama değişir. Örneğin havuz yalnızca Hep İhanet'lerden oluşsaydı en iyi strateji Hep İhanet olurdu; TFT her maçta ilk turda sömürülürdü. Axelrod bu yüzden evrimsel turnuvalar da düzenledi: her nesilde stratejiler önceki turnuva başarılarıyla orantılı olarak çoğalır (replikatör dinamiği). Bu turnuvalarda saldırgan stratejiler önce, sömürecek saf işbirlikçiler varken çoğalır; kurbanları azalınca kendileri de geriler ve TFT gibi iyi niyetli, misilleme yapan stratejiler nüfusa yerleşir.

Sık yapılan hata

- **"TFT maçları kazanır" demek:** TFT hiçbir maçı kazanamaz; en iyi ihtimalle berabere kalır. Turnuvayı yüksek toplam puanla kazanır.
- **Kendi ikiziyle maçı unutmak:** Mini turnuvada her strateji kendi ikiziyle de oynar; Hep İhanet'in 10 ve ŞTFT'nin 10 puanı bu maçlardan gelir.
- **Sonlu ve bilinen tekrarda işbirliği beklemek:** Turnuvadaki 10 tur, stratejilerin "bilmediği" bir sınırdır; programlar sona doğru davranış değiştirmez. Tur sayısı oyunculara bilinseydi geriye doğru çıkarım her turda ihanet önerirdi.

13.8 Tartışma: öğrenme, insan ve etik

Bölümün sonuçları üç tartışma sorusu etrafında toplanabilir.

Öğrenme her zaman optimuma götürür mü? Hayır. Bölüm boyunca birçok karşı örnek görüldü. En iyi tepki dinamiği Yazı-Tura'da döner; kurgusal oyun Shapley'nin 3×3 örneğinde yakınsamaz; iki Hedge oyuncusunun son yinelemesi salınmaya devam eder. Çevre ve rakipler de değişir: öğrenen

rakip hedefi kaydırır. Yanlış bir model de kötü bir yakınsamaya yol açar: yalnızca sömüren bir haydut algoritması yanlış kola takılır. Bu yüzden öğrenme kuramında güvenilir hedef çoğu zaman "dengeye varmak" değil, **pişmanlıksızlıktır**: rakip ne yaparsa yapsın, geriye bakınca hiçbir sabit stratejinin çok daha iyi olmayacağı garantisi.

İnsanlar algoritmalar gibi mi öğrenir? Kısmen. Benzerlikler açıktır: insanlar da deneme-yanılmayla öğrenir ve geçmiş deneyimlerine göre davranışlarını değiştirir. Farklar da vardır: insanlar risk algısı, duygular ve sosyal normlarla hareket eder; algoritmalar ise tanımlı bir ödülü izler. Haydut deneyindeki bulgu (Daw vd., 2006), insanların keşif davranışının UCB gibi bir algoritmadan çok softmax seçimine benzediğini göstermiştir.

Uyarlanabilir sistemlerde etik riskler nelerdir? Öğrenen sistemler veriyi olduğu gibi öğrenir. Verideki önyargı, sistemin kararlarına taşınır (veri önyargısı). Tıklama sayısını en büyük yapmayı öğrenen bir öneri sistemi, kullanıcıyı manipüle eden içerikleri öne çıkarabilir (tıklama tuzağı). Kullanıcı davranışından öğrenmek mahremiyet sorunları doğurur. Daha incelikli bir risk de **algoritmik fiyat koordinasyonudur**: aynı pazarda fiyat belirleyen öğrenen botlar, açıkça anlaşmadan yüksek fiyatlarda "örtük bir anlaşmaya" varabilir. Bu, tekrarlanan Mahkûmun İkilemi'ndeki işbirliğinin tüketici aleyhine çalışan bir hâlidir: kinci tetik benzeri cezalandırma stratejilerini öğrenen botlar, rakibin indirimini fiyat savaşıyla cezalandırarak yüksek fiyatı korur.

13.9 Özet

- **En iyi tepki dinamiğinde** oyuncular sırayla rakibin son eylemine en iyi tepki verir. Dinamik durursa bir saf Nash dengesinde durur; hangi dengeye varıldığı başlangıca bağlıdır. Sonlu **potansiyel oyunlarda** her kârlı sapma ortak potansiyeli artırdığı için dinamik mutlaka durur. Yazı-Tura Eşleştirme gibi saf dengesi olmayan oyunlarda döner.
- **Kurgusal oyunda** oyuncu rakibin bütün geçmişinin frekanslarına en iyi tepki verir. Yazı-Tura'da eylemler giderek uzayan döngülerle döner, frekanslar ise (½, ½)'ye yaklaşır (400 turda 0,472 ve 0,492). Yakınsama sıfır toplamlı (Robinson), 2×2 (Miyasawa) ve potansiyel oyunlarda garantilidir; Shapley'nin 3×3 örneğinde yoktur.
- **Dışsal pişmanlık**, geriye dönük en iyi sabit eylemle algoritma arasındaki toplam farktır: $R_T = \max_i \Sigma\, g_t(i) - \Sigma\, g_t(a_t)$. $R_T/T \to 0$ ise algoritma pişmanlıksızdır. Lideri izle gibi deterministik kurallar sömürülür (pişmanlık $\approx T/2$).
- **Hedge** ağırlıkları $w_i \leftarrow w_i \cdot e^{\eta \cdot g_t(i)}$ ile günceller ve ağırlıklarla orantılı olasılıkla oynar. $R_T \le (\ln n)/\eta + \eta T$; $\eta = \sqrt{(\ln n / T)}$ ile $R_T \le 2\sqrt{(T \ln n)}$.
- Sabit rakibe karşı Hedge en iyi tepkiye gider; öğrenen rakibe karşı son yineleme döner ama **zaman ortalaması** dengeye yaklaşır. Herkes pişmanlıksız oynarsa zaman ortalaması **kaba korelasyonlu dengeye** (CCE), iç pişmanlık küçültülürse **korelasyonlu dengeye** (CE), sıfır toplamlı oyunlarda minimaksa yaklaşır.

- **Çok kollu haydutta** yalnızca çekilen kolun ödülü görülür ve keşif–sömürü ikilemi doğar. **ε-açgözlü** ε olasılıkla rastgele keşfeder; sabit ε ile pişmanlığı doğrusal büyür. **UCB1**, $Q + \sqrt{(2 \ln t / N)}$ puanı en büyük kolu seçer; keşfi az denenen kollara yöneltir ve pişmanlığı logaritmik büyür (5000 turda 84'e 118).
- **Q-öğrenme** $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \cdot \max Q(s', \cdot) - Q(s, a)]$ ile durum–eylem değerlerini güncelleyer; ödül hedeften geriye yayılır. Çoklu ajan ortamında ortam durağan olmadığı için tek ajanlı garantiler geçerliliğini yitirir.
- **Replikatör dinamiğinde** ortalamanın üstünde kazanan strateji yayılır: $x_i' = x_i \, f_i \, / \, \bar{f}$. Şahin–Güvercin'de her iç başlangıç **ESS** $x^* = V/C$'ye gider; bu, simetrik karışık Nash dengesidir.
- **Tekrarlanan Mahkûmun İkilemi'nde** kinci tetik işbirliğini $\delta \geq (\text{cazibe} - 3)/(\text{cazibe} - 1)$ iken korur (puanlar 5, 3, 1, 0 ile $\delta \geq 1/2$). Mini Axelrod turnuvasında Tit-for-Tat hiçbir maçı kazanmadan birinci olur; başarılı stratejiler iyi niyetli, misilleme yapan, affedici ve anlaşılırdır. Gürültüde cömert TFT ve WSLS daha iyidir.

Yöntem	Açıklama	Sonuç
En iyi tepki dinamiği	Son eyleme en iyi tepki	Potansiyel oyunda dengeye; aksi hâlde döngü
Kurgusal oyun	Geçmiş frekanslara en iyi tepki	Sıfır toplamlıda frekanslar dengeye
Hedge	$w \leftarrow w \cdot e^{\eta g}$	$R_T = O(\sqrt{T \ln n})$; zaman ortalaması → CCE
ε-açgözlü / UCB	Keşif–sömürü dengesi	UCB: logaritmik pişmanlık
Q-öğrenme	$Q \leftarrow Q + \alpha(r + \gamma \cdot \max Q' - Q)$	Ödül hedeften geriye yayılır
Replikatör	Başarılı strateji yayılır	Şahin–Güvercin: $x^* = V/C$
Tekrarlanan MI	Geleceğin gölgesi	$\delta \geq 1/2$ ise işbirliği; TFT güçlü

Tablo 13.17: Bölümün yöntemleri bir bakışta.

13.10 Kavram listesi

Terimlerin İngilizce karşılıkları ve öteki bölümlerdeki kullanımları terim sözlüğünde bulunur.

en iyi tepki dinamiği *best-response dynamics* · potansiyel oyun *potential game* · kurgusal oyun *fictitious play* · dışsal pişmanlık *external regret* · iç pişmanlık *internal / swap regret* · lideri izle *follow the leader* · çarpımsal ağırlıklar (Hedge) *multiplicative weights* · öğrenme oranı *learning rate* · korelasyonlu denge *correlated equilibrium* · kaba korelasyonlu denge *coarse correlated equilibrium* · çok kollu haydut *multi-armed bandit* · keşif–sömürü *exploration–exploitation* · ε-açgözlü *ε-greedy* · üst güven sınırı (UCB) *upper confidence bound* · Thompson örneklemesi *Thompson sampling* · EXP3 (çekişmeli haydut) *adversarial bandit* · pekiştirmeli öğrenme *reinforcement learning* · Q-öğrenme *Q-learning* · iskonto faktörü *discount factor* · durağan olmayan *non-stationary* · karşı-olgusal pişmanlık minimizasyonu (CFR) *counterfactual regret minimization*

replikatör dinamiği *replicator dynamics* | uygunluk *fitness* | evrimsel kararlı strateji (ESS) *evolutionarily stable strategy* | kinci tetik stratejisi *grim trigger strategy* | halk teoremi *folk theorem* | Tit-for-Tat *Tit-for-Tat* | şüpheci / cömert TFT *suspicious / generous Tit-for-Tat* | kazan-kal, kaybet-değiş *win-stay lose-shift*

13.11 Kendini Dene

1. Hücrelerde (u_1, u_2): (Üst, Sol) = (2, 1), (Üst, Sağ) = (0, 0), (Alt, Sol) = (0, 0), (Alt, Sağ) = (1, 2). Önce satır, sonra sütun oyuncusu hareket eden en iyi tepki dinamiği (Üst, Sağ)'dan başlarsa nerede durur? (Alt, Sol)'dan başlarsa? Bu oyun potansiyel oyun mudur?

   Cevap

   (Üst, Sağ)'dan: satır oyuncusu Sağ'a karşı Üst 0, Alt 1 görür ve Alt'a geçer: (Alt, Sağ). Sütun oyuncusu Alt'a karşı Sol 0, Sağ 2 görür; Sağ'da kalır. Dinamik (Alt, Sağ)'da durur. (Alt, Sol)'dan: satır oyuncusu Sol'a karşı Üst 2, Alt 0 görür ve Üst'e geçer: (Üst, Sol). Sütun oyuncusu Üst'e karşı Sol 1, Sağ 0 görür; kalır. Dinamik (Üst, Sol)'da durur. Varılan denge başlangıca bağlıdır. Oyun potansiyel oyundur: $\Phi(\text{Üst, Sol}) = 2$, $\Phi(\text{Üst, Sağ}) = 1$, $\Phi(\text{Alt, Sol}) = 0$, $\Phi(\text{Alt, Sağ}) = 2$ seçilirse her tek taraflı değişimde değişen oyuncunun kazanç farkı Φ farkına eşittir. Örneğin Sağ sütununda satır oyuncusu Üst'ten Alt'a geçince kazancı 0'dan 1'e, Φ 1'den 2'ye çıkar; Üst satırında sütun oyuncusu Sağ'dan Sol'a geçince kazancı 0'dan 1'e, Φ 1'den 2'ye çıkar.

2. İki eylemli Hedge'de $\eta = \ln 3$, başlangıç ağırlıkları 1. Üç turun kazançları $(1, 0)$, $(0, 1)$, $(1, 0)$'dır. Her turun olasılıkları, algoritmanın toplam beklenen kazancı ve dışsal pişmanlık nedir?

   Cevap

   $e^{\ln 3} = 3$: kazancı 1 olan eylemin ağırlığı üç katına çıkar. Tur 1: $w = (1, 1)$, $p = (½, ½)$, beklenen kazanç 0,5; güncelleme $w = (3, 1)$. Tur 2: $p = (3/4, 1/4)$, kazanç $¼ \cdot 1 = 0{,}25$; $w = (3, 3)$. Tur 3: $p = (½, ½)$, kazanç 0,5; son ağırlıklar $(9, 3)$. Toplam $0{,}5 + 0{,}25 + 0{,}5 = 1{,}25$. Sabit eylemler: A = 2, B = 1. Pişmanlık $2 - 1{,}25 = 0{,}75$. 2. turda A'ya fazla güvenildi ve kazanan B çıktı.

3. Dört eylemli ($n = 4$) bir problemde Hedge 400 tur oynanacaktır. Sınırı en küçük yapan η, pişmanlık sınırı ve tur başına pişmanlık sınırı nedir? Tur sayısı dört katına çıkarsa (1600 tur) bu sayılar nasıl değişir?

   Cevap

   $\ln 4 \approx 1{,}386$. $\eta = \sqrt{(1{,}386/400)} \approx 0{,}059$. Sınır $2\sqrt{(400 \cdot 1{,}386)} = 2\sqrt{554{,}5} \approx 47{,}1$; tur başına $47{,}1/400 \approx 0{,}118$. 1600 turda η yarıya iner (≈ 0,029), toplam sınır $\sqrt{4} = 2$ katına çıkar (≈ 94,2), tur başına sınır yarıya iner (≈ 0,059).

4. İki kollu bir haydutta gerçek ortalamalar $\mu = (0{,}6;\ 0{,}4)$'tür. İlk beş turda kol 0 dört kez çekilmiş ($Q = 0{,}8$), kol 1 bir kez çekilmiş ($Q = 0{,}5$). (a) $t = 6$'da UCB1 hangi kolu seçer? (b) Seçimden sonra çekiş sayıları $N = (4,\ 2)$ olursa altı turun pişmanlığı kaçtır?

   Cevap

   (a) $2 \ln 6 \approx 3{,}58$. Kol 0: $0{,}8 + \sqrt{(3{,}58/4)} = 0{,}8 + 0{,}95 \approx 1{,}75$. Kol 1: $0{,}5 + \sqrt{3{,}58} = 0{,}5 + 1{,}89 \approx 2{,}39$. UCB1 kol 1'i seçer: tahmini düşük ama belirsizliği büyüktür. (b) $R = 6 \cdot 0{,}6 - (4 \cdot 0{,}6 + 2 \cdot 0{,}4) = 3{,}6 - 3{,}2 = 0{,}4$. Pişmanlık gerçek ortalamalarla hesaplanır; gözlenen Q değerleri kullanılmaz.

5. Q-öğrenmede $Q(s, a) = 0{,}5$, $r = 2$, $\gamma = 0{,}5$, $\max Q(s', \cdot) = 3$, $\alpha = 0{,}2$'dir. Hedef ve yeni $Q(s, a)$ nedir? s' bir hedef durum olsaydı (epizot bitseydi) sonuç ne olurdu?

   Cevap

   Hedef $2 + 0{,}5 \cdot 3 = 3{,}5$. Yeni değer $0{,}5 + 0{,}2 \cdot (3{,}5 - 0{,}5) = 0{,}5 + 0{,}6 = 1{,}1$. s' hedef olsaydı gelecek değer 0 alınırdı: hedef 2, yeni değer $0{,}5 + 0{,}2 \cdot 1{,}5 = 0{,}8$.

6. Şahin–Güvercin'de $V = 2$, $C = 8$, taban 3'tür. Şahin payı x iken uygunluklar nedir? ESS nedir? $x = 0{,}5$'ten bir nesil sonra şahin payı kaç olur?

   Cevap

   Kazançlar: ŞŞ $(2 - 8)/2 = -3$, ŞG 2, GŞ 0, GG 1. $f_Ş = 3 - 3x + 2(1 - x) = 5 - 5x$, $f_G = 3 + 1 \cdot (1 - x) = 4 - x$. Dinlenme noktası $5 - 5x = 4 - x \Rightarrow x^* = 1/4 = V/C$. $x = 0{,}5$'te $f_Ş = 2{,}5$, $f_G = 3{,}5$, $\bar{f} = 3$; $x' = 0{,}5 \cdot 2{,}5/3 \approx 0{,}417$. Şahinler ortalamanın altında kazandığı için azalır ve 0,25'e doğru iner. Kavga pahalı olduğunda şahin payı küçüktür.

7. Bir tekrarlanan Mahkûmun İkilemi'nde puanlar cazibe 7, karşılıklı işbirliği 4, karşılıklı ihanet 2, enayi 0'dır. Kinci tetikle işbirliğinin sürmesi için gereken en küçük δ kaçtır?

   Cevap

   Uymak $4/(1 - \delta)$, sapmak $7 + 2\delta/(1 - \delta)$. $(1 - \delta)$ ile çarpınca $4 \geq 7(1 - \delta) + 2\delta = 7 - 5\delta$, yani $5\delta \geq 3$ ve $\delta \geq 3/5$. Genel kural: $\delta^* = (\text{cazibe} - \text{işbirliği})/(\text{cazibe} - \text{ihanet}) = (7 - 4)/(7 - 2)$. Karşılıklı ihanetin puanı yükseldiği için ceza daha hafiftir; eşik buna göre ayarlanır.

8. Kazan-kal, kaybet-değiş (WSLS; ilk turda S oynar, 3 ya da 5 aldıysa aynı hamleyi tekrarlar, 0 ya da 1 aldıysa hamlesini değiştirir) altı tur boyunca Hep İhanet'e karşı oynuyor. Hamle dizisi ve puanlar

nedir? TFT aynı maçta ne alırdı?

Cevap

Tur 1: WSLS S, rakip İ: 0 – 5; WSLS 0 aldığı için değiştirir. Tur 2: İ – İ: 1 – 1; 1 aldığı için yine değiştirir. Dizi S İ S İ S İ olur: WSLS $0 + 1 + 0 + 1 + 0 + 1 = 3$, Hep İhanet $5 + 1 + 5 + 1 + 5 + 1 = 18$. TFT ilk turdan sonra hep ihanet ederdi: $0 + 5 \cdot 1 = 5$'e $5 + 5 = 10$. WSLS gürültüye karşı TFT'den güçlüdür ama koşulsuz ihanetçiye karşı her iki turda bir sömürülür.

13.12 İleri okuma

- T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory* (2016), Ders 16–18: pişmanlıksız öğrenme, Hedge'in pişmanlık sınırının ispatı, korelasyonlu dengeler ve minimaks teoreminin öğrenmeyle ispatı bilgisayar bilimi bakışıyla anlatılır.
- T. Lattimore ve C. Szepesvári, *Bandit Algorithms* (2020): ε-açgözlü, UCB, Thompson örneklemesi ve EXP3 algoritmalarının ayrıntılı analizini veren kapsamlı bir kaynaktır.
- R. Axelrod, *The Evolution of Cooperation* (1984): bilgisayar turnuvalarının hikâyesini, Tit-for-Tat'ın başarısını ve evrimsel turnuvaları matematik bilgisi gerektirmeden anlatır.
- D. Easley ve J. Kleinberg, *Networks, Crowds, and Markets* (2010), Bölüm 7: evrimsel oyun kuramını, ESS'yi ve Şahin–Güvercin benzeri örnekleri giriş düzeyinde sunar.

13.13 Kaynaklar

1. P. Auer, N. Cesa-Bianchi, P. Fischer, “Finite-time analysis of the multiarmed bandit problem”, *Machine Learning* 47, 2002.
2. P. Auer, N. Cesa-Bianchi, Y. Freund, R. E. Schapire, “The nonstochastic multiarmed bandit problem”, *SIAM Journal on Computing* 32(1), 2002.
3. R. Axelrod, *The Evolution of Cooperation*, Basic Books, 1984.
4. A. Blum, Y. Mansour, “From external to internal regret”, *JMLR* 8, 2007.
5. N. Cesa-Bianchi, G. Lugosi, *Prediction, Learning, and Games*, Cambridge, 2006.
6. N. D. Daw, J. P. O’Doherty, P. Dayan, B. Seymour, R. J. Dolan, “Cortical substrates for exploratory decisions in humans”, *Nature* 441, 2006.
7. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge, 2010 (Bölüm 7: evrimsel oyun kuramı).
8. D. Foster, R. Vohra, “Calibrated learning and correlated equilibrium”, *Games and Economic Behavior* 21, 1997.
9. D. Fudenberg, D. Levine, *The Theory of Learning in Games*, MIT Press, 1998.
10. S. Hart, A. Mas-Colell, “A simple adaptive procedure leading to correlated equilibrium”, *Econometrica* 68(5), 2000.
11. J. Hofbauer, K. Sigmund, *Evolutionary Games and Population Dynamics*, Cambridge, 1998.

12. A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017 (Bölüm 7: evrimsel denge; Bölüm 18: uyarlanabilir karar verme).

13. T. Lattimore, C. Szepesvári, *Bandit Algorithms*, Cambridge, 2020.

14. J. Maynard Smith, G. R. Price, "The logic of animal conflict", *Nature* 246, 1973.

15. K. Miyasawa, "On the convergence of the learning process in a 2×2 non-zero-sum two-person game", Princeton Research Memo 33, 1961.

16. V. Mnih vd., "Human-level control through deep reinforcement learning", *Nature* 518, 2015.

17. D. Monderer, L. S. Shapley, "Potential games", *Games and Economic Behavior* 14(1), 1996.

18. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge, 2007 (Bölüm 4: öğrenme, pişmanlık ve denge).

19. M. Nowak, K. Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game", *Nature* 364, 1993.

20. J. Robinson, "An iterative method of solving a game", *Annals of Mathematics* 54(2), 1951.

21. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge, 2016 (Ders 16–18).

22. L. S. Shapley, "Some topics in two-person games", *Advances in Game Theory*, 1964.

23. W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples", *Biometrika* 25, 1933.

24. C. Watkins, *Learning from Delayed Rewards*, doktora tezi, Cambridge Üniversitesi, 1989.

25. C. Watkins, P. Dayan, "Q-learning", *Machine Learning* 8, 1992.

26. M. Zinkevich, M. Johanson, M. Bowling, C. Piccione, "Regret minimization in games with incomplete information", *NeurIPS*, 2007.

KISIM V · BÖLÜM 14

Gelecek Yönelimler

Dersin önceki bölümleri oyunları, dengeleri ve mekanizmaları çoğunlukla insanlar arasında, sabit kurallar altında ve elle çözülebilir boyutlarda inceledi. Bugün aynı sorular yeni bir ortamda yeniden soruluyor. Oyuncuların bir kısmı artık kullanıcı adına konuşan yazılım ajanlarıdır. Mekanizmalar elle tasarlanmak yerine veriden öğreniliyor. Bazı kurallar ise kimsenin sonradan değiştiremediği, herkese açık kodla uygulanıyor. Bu not, dersin araçlarının bu üç alanda nasıl kullanıldığını ve nerede yetersiz kaldığını sayısal örneklerle anlatır: ajanlar arası pazarlık ve algoritmik danışıklılık, veriden fiyat öğrenme ve mahremiyet, işlem ücreti mekanizmaları ve işlem sıralama gücü, son olarak da büyük ölçekte hesaplama sorunları. Deney protokolü 14.2.8'de özetlenir; 14.6'daki tablo dersin bütün bölümlerini birleştiren haritaya bağlantı verir.

Kod (Python): `github.com/sercankulcu/algorithmic-game-theory-python/Bolum14`

ÖĞRENME ÇIKTILARI

Yeni oyuncular
Yapay zekâ ajanlarının pazarlık ve fiyatlandırmada yarattığı oyun kuramı sorularını tanımlamak.

Veriden tasarım
Mekanizmanın veriden öğrenilmesini ve örneklem karmaşıklığı kavramını sayısal örnekle açıklamak.

Zincir üstü mekanizmalar
İşlem ücreti açık artırmalarını, taban ücret kuralını ve MEV (işlem sıralama gücünden elde edilen değer) kavramını yorumlamak.

Dersi birleştirmek
Dersin bütün bölümlerini tek haritada ilişkilendirmek.

14.1 Giriş

Bu bölüm yeni bir kuram kurmaz; önceki bölümlerde öğrenilen araçları yeni ortamlara taşır. Geriye doğru çıkarım, tekrarlanan oyunlarda tetik stratejisi, teşvik uyumluluğu, gelir maksimizasyonu, ikinci fiyat fikri, oylama kuralları ve anarşinin bedeli burada yeniden karşımıza çıkar. Değişen, oyuncuların

ve kuralların doğasıdır. Bu yüzden her başlıkta iki soru birlikte sorulur: dersin aracı bu durumda ne söyler ve hangi varsayım artık tutmaz?

Bölüm hızla değişen alanlara değinir. Belirli ürünler, sürümler ve güncel uygulamalar birkaç yıl içinde değişir. Bu yüzden not, örnekleri ürün adlarına değil kalıcı ilkelere bağlar: sabır pazarlık gücüdür, aynı veride seçilen ve ölçülen bir şey başarısını abartır, herkesin gördüğü bir sıralama gücü değer taşır. Bu ilkeler, uygulamanın biçimi değişse de geçerli kalır.

14.1.1 Bir oyunla başlangıç: yapay zekâ ajan turnuvası

Bölüm küçük bir turnuvayla açılır. Dört basit ajan seçilen bir oyunu birbirine karşı oynar. Her maç 60 ya da 120 tur sürer. Tam turnuvada her ajan her ajanla bir kez karşılaşır; dört ajan için bu, $4 \cdot 3 / 2 = 6$ maç demektir. Sıralama, ajanların bütün maçlarda topladığı kazanca göre yapılır.

Ajanlar şunlardır:

- **Rastgele:** her turda iki hamleden birini yarı yarıya olasılıkla seçer.
- **Hep İşbirliği:** her turda işbirliği yapar; rakibe hiç bakmaz.
- **Hep İhanet:** her turda ihanet eder.
- **En İyi Tepki:** rakibin şimdiye kadar en sık yaptığı hamleyi gelecek hamle olarak tahmin eder ve ona en iyi tepkiyi verir. Geçmiş yokken, yani ilk turda, işbirliği yapar.



Şekil 14.1: Turnuvanın eşleşme tablosu. Mahkûmun İkilemi'nin ilk seviyesinde her maç 60 tur sürer; her ajan diğer üç ajanla birer kez oynar, toplam 6 maç yapılır. Soru işaretli hücreler oynanacak maçlardır.

Şekil 14.1'de eşleşmeler görülür. Turnuvanın merak uyandıran iki sorusu vardır. Birincisi, bu dört ajandan hangisi birinci olur? İkincisi, bir oyunda birinci olan ajan başka bir oyunda da birinci olur mu? İlk sorunun cevabı oyunun yapısından hesaplanabilir; ikinci sorunun cevabı ise bölümün ana

derslerinden biridir. İki soru da tekrarlanan oyunlar ve en iyi tepki kavramlarına dayanır ve 14.2.7'de sayılarla cevaplanır.

Bu turnuvanın önemi, ajanların "akıllı" olup olmamasından gelmez. Ajanlar çok basittir. Önemli olan, bir ajanın başarısının yalnız kendi kuralına değil rakiplerine ve oynanan oyuna bağlı olmasıdır. Bu, oyun kuramının temel gözlemidir ve yazılım ajanları yaygınlaştıkça daha da önem kazanır.

14.1.2 Gelişen bir alan

Algoritmik oyun kuramı, bilgisayar bilimi ile ekonominin kesişiminde doğmuş, uygulamaları hızla artan bir alandır. Endüstride ve akademide otonom araçlar, çevrim içi platformlar ve enerji ağları gibi alanlarda kullanılır. Bu büyümeyi üç teknolojik eğilim besler:

- **Yapay zekâ ve makine öğrenmesi:** karar verenlerin bir kısmı artık öğrenen yazılımlardır.
- **Büyük veri ve gerçek zamanlı karar:** fiyatlar, teklifler ve rotalar saniyeler içinde güncellenir.
- **Blokzincir ve akıllı sözleşmeler:** kurallar herkese açık kodla otomatik uygulanır.

Bu eğilimlerden dört araştırma ekseni doğar. Her biri dersin bir bölümüyle doğrudan ilgilidir.

Eksen	Soru	Dersteki karşılığı
Adalet ve etik	Mekanizmalar adil ve şeffaf olabilir mi?	Adalet ölçütleri, adil bölüşüm
Dinamik ve öğrenen sistemler	Uyarlanabilir otonom ajanlar nasıl davranır?	Öğrenme dinamikleri, pişmanlık
Büyük ağlar	Milyonlarca oyunculu stratejik etkileşim nasıl çözümlenir?	Tıkanıklık oyunları, anarşinin bedeli
Teşvik ve politika	Sosyal refah ve sürdürülebilirlik için hangi kurallar tasarlanmalı?	Mekanizma tasarımı, dışsallık fiyatlaması

Tablo 14.1: Alanın dört ekseni ve dersteki karşılıkları.

Bu bölümün amacı, dersin araçlarının bu yeni alanlarda nasıl kullanıldığını ve nerede yetersiz kaldığını göstermektir. Yetersizlik de öğreticidir. Örneğin klasik mekanizma tasarımı, tasarımcının ilan ettiği kurala sonradan bağlı kalacağını varsayar. Akıllı sözleşmeler bu varsayımı kodla sağlar ama yeni bir sorun yaratır: bekleyen işlemler herkese görünür ve bu görünürlük istismar edilebilir. Benzer biçimde klasik kuram oyuncuların rasyonel olduğunu varsayar. Dil modeli tabanlı ajanlar ise sorunun nasıl çerçevelendiğine duyarlı olabilir ve bu varsayımdan sapabilir.

14.1.3 Uygulama alanları

Oyun kuramının bugün en çok kullanıldığı dört alan şöyle özetlenebilir.

Finans
: Algoritmik ticaret ve portföy yönetimi. Alım-satım algoritmaları birbirine karşı oynar: bir algoritmanın kârı, ötekilerin emirlerine ve hızına bağlıdır. Bu, tek kişilik bir optimizasyon değil bir oyundur.

Trafik ve lojistik
: Otonom araç ağları ve rota seçimi. Her araç kendi süresini kısaltmak ister; bu bencil yönlendirme, ağa yol eklemenin herkesi yavaşlattığı Braess paradoksu gibi sonuçlar doğurabilir.

Çevrim içi platformlar
: Öneri sistemleri, reklam açık artırmaları ve sosyal ağ analizi. Arama sonuçlarının yanındaki reklam yuvaları, her arama için ayrı bir açık artırmayla satılır.

Enerji şebekeleri
: Yük dengeleme, talep yanıtı ve elektrik piyasaları. Tüketiciler fiyat sinyaline göre tüketimlerini kaydırır; üreticiler piyasaya teklif verir.

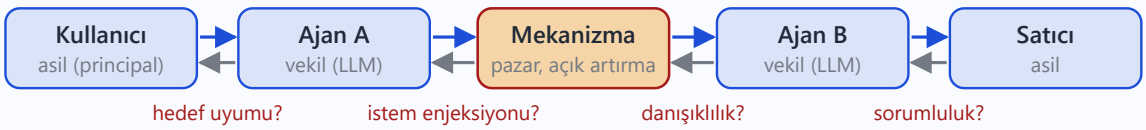
Bu alanların ortak noktası, karar verenlerin giderek daha çok yazılım olmasıdır. Bir alım-satım algoritması, bir rota uygulaması ya da bir reklam teklif aracı, sahibinin adına oynayan bir ajandır. Bölümün geri kalanı bu gözlemden yola çıkar.

14.2 Yapay zekâ ajanları

Bu kısım, yazılım ajanlarının oyuncu olarak ortaya çıkmasıyla gündeme gelen soruları inceler: ajanların yarattığı yeni sorunlar, ajanlar arası pazarlık (sonlu ve sonsuz ufuk) ve fiyatlandırma algoritmalarının danışıklılığı. Giriş turnuvasının sonuçları da bu araçlarla açıklanır.

14.2.1 Stratejik yapay zekâ ajanları

Büyük dil modeli tabanlı ajanlar, kullanıcı adına alışveriş, pazarlık, planlama ve kaynak paylaşımı gibi işleri üstlenebilir. Bu durumda ajan artık yalnızca bir araç değildir; hedefi, bilgisi ve stratejisi olan bir aktördür. Yani oyun kuramı anlamında bir **oyuncudur**. Oyuncunun ne istediği, ne bildiği ve neyi seçebildiği sorulabilir. Bu da oyun kuramının eski sorularını yeniden gündeme getirir.



Şekil 14.2: İki ajanın ortadaki bir mekanizmada (pazar, açık artırma) karşılaştığı yapı. Kullanıcı ve satıcı asildir; ajanlar onların vekilidir. Kırmızı etiketler her bağlantıdaki riski gösterir: hedef uyumu, istem enjeksiyonu, danışıklılık ve sorumluluk.

Şekil 14.2 tipik yapıyı gösterir. Bir kullanıcı ajan A'yı, bir satıcı ajan B'yi görevlendirir. İki ajan bir mekanizmada, örneğin bir pazarda ya da açık artırmada, karşılaşır. Bu zincirin her halkasında ayrı bir oyun kuramı sorusu vardır.

Tanım 14.1

Asil–vekil sorunu (*principal–agent problem*): Bir taraf (asil) bir işi başka bir tarafa (vekil) devreder. Vekilin hedefi asilinkinden farklıysa ve asil vekilin her hareketini izleyemiyorsa, vekil asilin değil kendi hedefinin yararına davranabilir. Asilin sorunu, vekilin hedefini kendi hedefiyle uyumlu kılan bir teşvik ya da denetim düzeni kurmaktır.

Bu sorun yapay zekâ ajanlarından çok eskidir. Ev sahibi ile emlakçı, hissedar ile yönetici, hasta ile doktor arasındaki ilişkiler aynı yapıdadır. Örneğin komisyonla çalışan bir emlakçı, evi biraz daha yüksek fiyata satmak için haftalarca uğraşmak istemeyebilir; çünkü fiyattaki artışın küçük bir kısmı kendisine gelir, uğraşın tamamı ise kendisinedir. Yazılım ajanlarında sorun dört biçimde ortaya çıkar.

Hedef uyumu
: Ajanın hedefi kullanıcınınkiyle aynı mı? Ajanın hedefi, kullanıcının söylediği sözlerden ya da eğitim verisinden çıkarılır. Kullanıcı “en ucuzunu bul” der ama aslında hızlı teslimatı da önemsiyor olabilir. Ajan söylenen hedefi kelimesi kelimesine izlerse kullanıcının gerçek faydasını kaçırabilir. Bu, asil–vekil sorununun en doğrudan biçimidir.

Rasyonellik varsayımı
: Klasik oyun kuramı oyuncuların kazançlarını maksimize ettiğini varsayar. Dil modeli tabanlı bir ajanın davranışı ise sorunun nasıl sunulduğuna, yani **çerçevelemeye** (*framing*) duyarlı olabilir. Aynı oyunun iki farklı anlatımında farklı hamle seçebilir. Bu nedenle Nash dengesi, bu ajanların davranışı için her zaman iyi bir tahmin olmayabilir. Davranışsal oyun kuramında insanlar için yapılan gözlemler burada ajanlar için de geçerlidir.

Manipülasyon
: Ajan, karşı tarafın yazdığı metni okur. Karşı ajan mesajına gizli talimatlar ekleyerek ajanı kendi lehine yönlendirmeye çalışabilir. Buna **istem enjeksiyonu** (*prompt injection*) denir. Oyun kuramı açısından bu, karşı tarafın yalnız hamle seçmekle kalmayıp rakibin karar kuralını da etkileyebildiği yeni bir strateji türüdür.

Kimlik ve sorumluluk
: Ajan kimin adına konuşuyor, bir hata olursa kim sorumlu, yapılan bir anlaşma nasıl doğrulanır? Mekanizma tasarımı, oyuncuların kim olduğunun ve bir sözleşmenin uygulanabilir olduğunun bilindiğini varsayar. Bu varsayımlar ajanlar için ayrıca kurulmalıdır.

Bu dört soru bir arada şunu söyler: ajanlar oyun kuramını gereksiz kılmaz, aksine onun sorularını daha önemli hâle getirir. Ajanın hangi oyunu oynadığı, kimin adına oynadığı ve oyunun kurallarının neler olduğu açıkça modellenmelidir. Pazarlık ve fiyatlandırma modelleri bu modellemenin iki somut örneğidir.

14.2.2 Ajanlar arası pazarlık: sonlu ufuk

İki ajanın bir miktarı paylaşması gerektiğini düşünelim. Örneğin alıcı ajanı ile satıcı ajanı, bir işlemden doğan 100 TL'lik kazancın nasıl bölüşüleceğini pazarlık eder. Pazarlık sırayla yapılan önerilerle ilerler.

Tanım 14.2

Sırayla teklif pazarlığı (*alternating-offers bargaining*): İki oyuncu bir miktarı paylaşır. Tek numaralı turlarda A, çift numaralı turlarda B bir paylaşım önerir. Öteki taraf öneriyi kabul ederse oyun biter ve paylaşım uygulanır. Reddederse sıradaki tura geçilir. Her gecikme, paylaşılacak değeri bir **iskonto faktörü** (*discount factor*) δ ile çarpar. T turdan oluşan bir oyunda son öneri de reddedilirse iki taraf da 0 alır.

$\delta = 0{,}9$, her turda değerin %10 azalması demektir. Bunun iki yorumu vardır. Birincisi, gerçekten bir kayıp vardır: örneğin bozulan bir ürün ya da kaçan bir fırsat. İkincisi, oyuncular sabırsızdır: bir tur sonra alınacak 100 TL, bugün alınacak 90 TL kadar değerlidir. İki yorumda da δ büyüdükçe oyuncu daha sabırlıdır.

Oyun sonlu olduğu için **geriye doğru çıkarım** (*backward induction*) ile çözülür. Son turdan başlanır ve her turda "öneren, karşı tarafı kabul ile red arasında kayıtsız bırakacak en küçük payı önerir" kuralı uygulanır. Karşı taraf reddederse bir sonraki turda ne alacaksa, bunun bugünkü değeri onun için alt sınırdır.

Örnek 14.1

100 TL paylaşılacak, $\delta = 0{,}9$, oyun dört tur sürüyor. A tek (1 ve 3), B çift (2 ve 4) turlarda öneriyor.

1. **Tur 4 (B önerir, son tur).** A reddederse ikisi de 0 alır. A için kabul ile red arasında fark yoktur; kayıtsız kalan tarafın kabul ettiği varsayılır. B kendine 100, A'ya 0 önerir: *(A, B) = (0, 100)*.
2. **Tur 3 (A önerir).** B reddederse tur 4'te 100 alır. Bunun bugünkü değeri $0{,}9 \cdot 100 = 90$'dır. A, B'ye en az 90 önermelidir; fazlasını önermesi gereksizdir. Paylaşım *(10, 90)*.
3. **Tur 2 (B önerir).** A reddederse tur 3'te 10 alır; bugünkü değeri $0{,}9 \cdot 10 = 9$. B, A'ya 9 önerir: *(9, 91)*.
4. **Tur 1 (A önerir).** B reddederse tur 2'de 91 alır; bugünkü değeri $0{,}9 \cdot 91 = 81{,}9$. A, B'ye 81,9 önerir ve kendine $100 - 81{,}9 = 18{,}1$ bırakır: *(18,1; 81,9)*.

B tur 1'deki öneriyi hemen kabul eder; pazarlık gecikmeden biter ve değer kaybolmaz. Bu sonuç bir **alt oyun mükemmel dengedir** (*subgame perfect equilibrium*): her turda, o turdan başlayan oyunda bile herkes en iyi hamlesini yapar.



Şekil 14.3: Dört turluk pazarlıkta geriye doğru çıkarım. Sağdan sola her adımda karşı tarafın bekleme değeri 0,9 ile çarpılır. Tur 4'te B her şeyi alır; tur 1'de dengede A 18,1, B 81,9 alır ve öneri hemen kabul edilir.

Şekil 14.3 çözümü sağdan sola okur. İlk bakışta şaşırtıcı bir sonuç vardır: A ilk öneriyi yaptığı hâlde payın yalnızca %18,1'ini alır. Nedeni son söz hakkının B'de olmasıdır. Tur 4'te B her şeyi alabilir ve bu tehdit geriye doğru bütün turlara yayılır. Her tur geriye gidildiğinde son sözün avantajı δ ile küçülür ama tamamen kaybolmaz.

Aynı hesap farklı tur sayıları için yapılabilir. Hesabı kısaltan bir gözlem vardır. x_T, T turluk oyunda ilk önerenin (A'nın) payı olsun. İlk turda A, karşı tarafı bekleme değeri kadar tatmin etmelidir. B için ikinci turdan başlayan oyun, ilk önerenin kendisi olduğu $T - 1$ turluk bir oyundur; orada B, x_{T-1} payını alır. Bu yüzden:

$$x_1 = 100, \quad x_T = 100 - \delta \cdot x_{T-1}$$

Bu kural tek turluk oyunla başlar: tek tur varsa A her şeyi önerir ve B kabul eder (red ikisine de 0 verir). Kural art arda uygulanınca Tablo 14.2 elde edilir.

Tur sayısı T	1	2	3	4	5	6	7	8	∞
A'nın payı	100	10	91	18,1	83,7	24,7	77,8	30	52,6

Tablo 14.2: Tur sayısına göre ilk önerenin (A'nın) dengedeki payı, 100 TL ve $\delta = 0{,}9$.

Tabloda her değer bir öncekinden hesaplanır. Örneğin $x_2 = 100 - 0{,}9 \cdot 100 = 10$, $x_3 = 100 - 0{,}9 \cdot 10 = 91$, $x_4 = 100 - 0{,}9 \cdot 91 = 18{,}1$, $x_5 = 100 - 0{,}9 \cdot 18{,}1 = 83{,}71$. Değerler salınır: tur sayısı tekse son söz A'dadır ve A büyük pay alır; çiftse son söz B'dedir ve A küçük pay alır. $T = 1$ durumu **ültimatom oyunudur**: tek bir öneri vardır, karşı taraf ya kabul eder ya da ikisi de 0 alır. Kuramsal tahmin, önerenin hemen hemen her şeyi aldığıdır.

Salınımın genliği tur sayısı arttıkça küçülür ve değerler 52,6 civarına yaklaşır. Bu sınır 14.2.3'te türetilir. Ajan tasarımı için iki ders çıkar. Birincisi, **sabır** (δ'nın büyüklüğü) ve **son söz hakkı** pazarlık gücüdür. İkincisi, son tarih son teklifin kimde olduğunu belirlediği için **ufuk bilgisi de stratejiktir**. Bir ajan, pazarlığın ne zaman biteceğini bilmekle ya da bunu belirleyebilmekle güç kazanır.

Sık yapılan hata

- **İlk önereni hep güçlü sanmak:** Sonlu ufukta güç, son önerendedir. $T = 4$'te ilk öneren A yalnızca 18,1 alır.
- **Bekleme değerini iskontosuz yazmak:** B'nin tur 3'teki alt sınırı 100 değil, $0{,}9 \cdot 100 = 90$'dır; bir tur sonraki 100, bugün 90 eder.
- **Geriye doğru çıkarıma baştan başlamak:** Hesap son turdan başlar. Son turda ne olacağını bilmeden ilk turdaki öneri bulunamaz.

14.2.3 Sonsuz ufuk: Rubinstein çözümü

Pazarlığın bir son turu yoksa geriye doğru çıkarım başlayamaz; başlangıç noktası olan son tur yoktur. Rubinstein (1982), bu durumda da tek bir alt oyun mükemmel denge olduğunu gösterdi. Çözüm **durağan stratejilerle** (*stationary strategies*) bulunur. Durağan strateji, geçmişe bakmadan her turda aynı teklifi yapmaktır. A önerirken kendine hep x^* payını, B önerirken A'ya hep y^* payını bırakır. Payları 1'e (bütün miktara) oranla yazalım.

Dengenin iki koşulu vardır.

1. **B, A'nın önerisinde kayıtsızdır.** A'nın önerisi B'ye $1 - x^*$ bırakır. B reddederse bir sonraki turda kendisi önerir ve $1 - y^*$ alır; bunun bugünkü değeri $\delta_B(1 - y^*)$'dır. A, B'ye tam bu kadar bırakır: $1 - x^* = \delta_B \cdot (1 - y^*)$.
2. **A, B'nin önerisinde kayıtsızdır.** B'nin önerisi A'ya y^* bırakır. A reddederse bir sonraki turda x^* alır; bugünkü değeri $\delta_A\, x^*$'dır. Buradan $y^* = \delta_A \cdot x^*$.

İkinci denklem birinciye konur: $1 - x^* = \delta_B(1 - \delta_A\, x^*) = \delta_B - \delta_A\delta_B\, x^*$. x^* içeren terimler bir tarafa toplanır: $x^* - \delta_A\delta_B\, x^* = 1 - \delta_B$, yani $x^*(1 - \delta_A\delta_B) = 1 - \delta_B$. Her iki taraf $1 - \delta_A\delta_B$'ye bölünür.

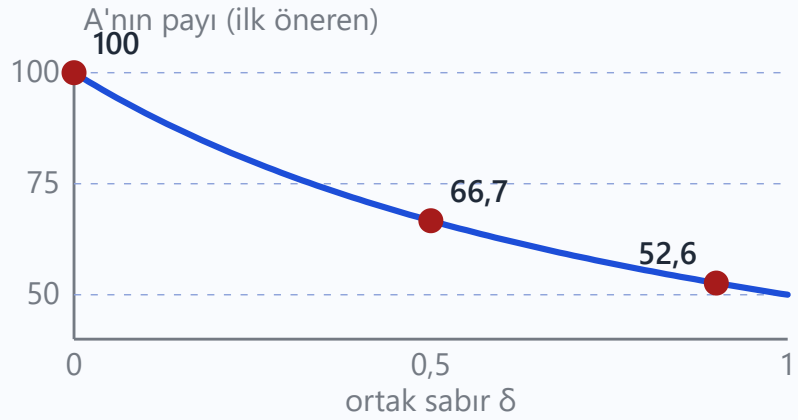
Teorem 14.1 (Rubinstein, 1982)

Sonsuz ufuklu sırayla teklif pazarlığında, iskonto faktörleri δ_A, $\delta_B < 1$ ise tek alt oyun mükemmel denge vardır. İlk öneren A'nın payı

$$x^* = (1 - \delta_B) \,/\, (1 - \delta_A\delta_B)$$

olur ve öneri ilk turda kabul edilir. Simetrik sabırda ($\delta_A = \delta_B = \delta$) $x^* = 1/(1 + \delta)$'dır.

Simetrik formül genel formülden şöyle çıkar: $1 - \delta^2 = (1 - \delta)(1 + \delta)$ olduğu için $(1 - \delta)/(1 - \delta^2) = 1/(1 + \delta)$. $\delta = 0{,}9$ için $100/1{,}9 \approx 52{,}6$ TL bulunur. Bu, Tablo 14.2'deki salınımın yaklaştığı değerdir. Bağlantı açıktır: sonlu ufuktaki $x_T = 100 - \delta\, x_{T-1}$ kuralının sabit noktası, yani $x = 100 - \delta x$ denkleminin çözümü, $x = 100/(1 + \delta)$'dır. Ufuk uzadıkça son turun etkisi her adımda δ ile çarpılarak söner ve pay bu sabit noktaya yaklaşır.



Şekil 14.4: Simetrik sabırda ilk önerenin payı $1/(1+\delta)$. Hiç sabır yoksa ($\delta = 0$) öneren her şeyi alır; $\delta = 0{,}5$'te 66,7, $\delta = 0{,}9$'da 52,6 alır. δ 1'e yaklaştıkça pay yarıya iner.

Şekil 14.4 ilk önerenin avantajını gösterir. Sabırsız oyuncular arasında (küçük δ) ilk öneri çok değerlidir; çünkü karşı taraf reddederse bir sonraki turu beklemek ona pahalıya mal olur. Sabır arttıkça bekleme ucuzlar ve ilk öneren avantajı küçülür. Çok sabırlı oyuncular yaklaşık eşit paylaşır.

Sabırlar farklıysa sonuç daha çarpıcıdır. Tablo 14.3, formülün dört örnekte verdiği payları gösterir.

δ_A	δ_B	Hesap	A	B
0,9	0,9	0,1 / 0,19	52,6	47,4
0,9	0,8	0,2 / 0,28	71,4	28,6
0,8	0,9	0,1 / 0,28	35,7	64,3
0,99	0,9	0,1 / 0,109	91,7	8,3

Tablo 14.3: Rubinstein paylaşımı (100 TL). A ilk önerendir.

Örnek 14.2

$\delta_A = 0{,}9$ ve $\delta_B = 0{,}8$ olsun. A'nın payı şöyle bulunur.

1. Kesrin payı: $1 - \delta_B = 1 - 0{,}8 = 0{,}2$.
2. Kesrin paydası: $\delta_A\delta_B = 0{,}72$, yani $1 - 0{,}72 = 0{,}28$.
3. $x^* = 0{,}2 / 0{,}28 \approx 0{,}714$: A 71,4 TL, B 28,6 TL alır.
4. Denetim: B'nin önerisinde A'ya kalan $y^* = \delta_A x^* = 0{,}9 \cdot 0{,}714 \approx 0{,}643$'tür. B'nin A'nın önerisini kabul ile red arasında kayıtsız olması gerekir: $\delta_B(1 - y^*) = 0{,}8 \cdot 0{,}357 \approx 0{,}286 = 1 - x^*$ ✓.

Roller aynı kalıp sabırlar yer değiştirirse ($\delta_A = 0{,}8$, $\delta_B = 0{,}9$) A ilk öneren olduğu hâlde yalnızca 35,7 alır. Sabır farkı, ilk öneri avantajından daha güçlüdür.

Tablonun son satırı uç bir durumdur: $\delta_A = 0{,}99$ çok sabırlı bir A'yı temsil eder ve A payın %91,7'sini alır. Ajan tasarımı açısından ders şudur: **sabır güçtür**. Gerçekten aceleci (düşük δ'lı) bir ajan daha az pay alır. Eksik bilgide durum biraz daha inceliklidir: karşı taraf ajanın gerçek sabrını bilmiyorsa, aceleci *görünmek* de rakibin inancını değiştirerek benzer bir etki yapar. Bu yüzden bir pazarlık ajanının, kullanıcının son tarihini ya da aciliyetini karşı tarafa sızdırmaması önemlidir.

Sık yapılan hata

Sonlu ve sonsuz ufku karıştırmak. Sonlu ufukta son öneren güçlüdür: $T = 4$'te A yalnızca 18,1 alır. Sonsuz ufukta ise son tur yoktur ve ilk öneren biraz öndedir: 52,6. Bir pazarlığın son tarihinin olup olmadığı, kimin güçlü olduğunu tamamen değiştirir.

14.2.4 Fiyatlandırma algoritmaları danışabilir mi?

Çevrim içi satıcıların çoğu fiyatlarını algoritmalarla belirler. Bu algoritmalar birbirinin fiyatını izler ve kendi fiyatını buna göre günceller. Burada rekabet hukukunu ilgilendiren bir soru doğar: birbirleriyle hiç konuşmayan algoritmalar, rakiplerin anlaşarak yaptığı gibi yüksek fiyatta buluşabilir mi?

Tanım 14.3

Danışıklılık (*collusion*), rakiplerin rekabet etmek yerine fiyatı ortak çıkarları için rekabetçi düzeyin üstünde tutmasıdır. Bu sonuca açık bir anlaşma olmadan, fiyatlandırma algoritmalarının etkileşimiyle ulaşılmasına **algoritmik danışıklılık** (*algorithmic collusion*) denir.

Model şöyledir. İki firma aynı ürünü satar. İkisi de tekel fiyatını koyarsa toplam kâr 100 olur ve her biri 50 alır. Biri fiyatı biraz kırarsa, Bertrand rekabetinde olduğu gibi ucuz olan bütün pazarı alır: o dönem 100 kazanır, öteki 0. Fiyatlar sonunda maliyete kadar inerse kâr sıfırlanır. Tek dönemlik oyun Mahkûmun İkilemi'dir: karşı taraf ne yaparsa yapsın fiyat kırmak daha kârlıdır, denge ise iki firmaya da 0 bırakır.

Oyun her dönem tekrarlanırsa durum değişir. Firmalar şu stratejiyi kullanabilir:

Tanım 14.4

Kinci tetik stratejisi (*grim trigger strategy*): rakip fiyat kırana kadar yüksek (tekel) fiyatı koymak; rakip bir kez fiyat kırarsa sonsuza dek rekabetçi fiyata geçmektir.

Bu strateji işbirliğini bir tehditle korur. Sapmak bugün kazandırır ama geleceğin bütün kârlarını yok eder. Sapmanın kârlı olup olmadığı, firmanın geleceğe ne kadar değer verdiğine, yani iskonto faktörü δ'ya bağlıdır.

İki yolun bugünkü değerlerini karşılaştıralım. İşbirliği yolunda firma her dönem 50 alır. Bugünkü değeri bir geometrik seridir:

$$50 + 50\delta + 50\delta^2 + \ldots = 50 / (1 - \delta)$$

Bu toplam şöyle bulunur. Seri S olsun. $\delta S = 50\delta + 50\delta^2 + \ldots$ olduğu için $S - \delta S = 50$'dir; buradan $S = 50/(1 - \delta)$ çıkar. Sapma yolunda firma bugün 100 alır, sonra tetik stratejisi devreye girer ve sonsuza dek 0 alır. Bugünkü değeri 100'dür. İşbirliğinin sürmesi için işbirliğinin değeri sapmanınkinden küçük olmamalıdır:

$$50/(1 - \delta) \geq 100 \iff 50 \geq 100(1 - \delta) \iff 1 - \delta \leq 1/2 \iff \delta \geq 1/2$$

İkinci adımda iki taraf $(1 - \delta)$ ile çarpılmıştır; $\delta < 1$ olduğu için bu sayı pozitiftir ve eşitsizliğin yönü değişmez. Sonuç: firmalar geleceğe yeterince değer veriyorsa ($\delta \geq 1/2$), kimse fiyat kırmak istemez ve yüksek fiyat sürer. Bu yapı tekrarlanan Mahkûmun İkilemi ile aynıdır.

Bu kuramsal sonucun algoritmalar için ne anlama geldiğini Calvano vd. (2020) bir benzetimle inceledi. Pekiştirmeli öğrenen (her dönem kâr sinyaline göre fiyat kuralını güncelleyen) fiyatlandırma ajanları, birbirleriyle hiç mesajlaşmadan rekabetçi düzeyin üstünde fiyatlar öğrenebildi. Dahası, bir ajan fiyat kırdığında öteki geçici bir fiyat savaşıyla karşılık verdi; yani ajanlar tetik stratejisine benzeyen bir cezalandırma davranışını kendiliğinden öğrendi.

Bu bulgu bir düzenleme sorusu doğurur. Rekabet hukuku geleneksel olarak *insanlar arası anlaşma* kavramı üzerine kuruludur: rakiplerin bir araya gelip fiyat konuşması yasaktır. Açıkça iletişim kurmadan yüksek fiyatı öğrenen algoritmalar bu çerçeveyi zorlar. Anlaşma yoksa danışıklılık var mıdır? Sorumlu kimdir: algoritmayı yazan mı, kullanan mı? Bu sorular oyun kuramının sonuçlarına dayanır: tekrarlanan etkileşim ve gelecek değeri, açık bir anlaşma olmadan da işbirliğini sürdürebilir.

14.2.5 Kod izleme: karsilastir()

Tetik koşulunu sayılarla görmek için iki yolun birikimli değeri dönem dönem hesaplanabilir. Aşağıdaki fonksiyon, her dönem 50 ile işbirliği yapan yolu ve s döneminde fiyat kırıp o dönem 100 alan, sonra 0 alan sapma yolunu T dönem boyunca karşılaştırır.

```
def karsilastir(delta, s, T):
    ib, sp = 0.0, 0.0            # işbirliği / sapma yolu
    for t in range(T):
        ib += delta**t * 50
        if t < s:
            sp += delta**t * 50
        elif t == s:
            sp += delta**t * 100   # A fiyat kırar
        # t > s: tetik devrede, kâr 0
    return round(ib, 2), round(sp, 2)
print(karsilastir(0.6, 2, 6))
print(karsilastir(0.4, 2, 6))
```

Kod satır satır şöyle çalışır:

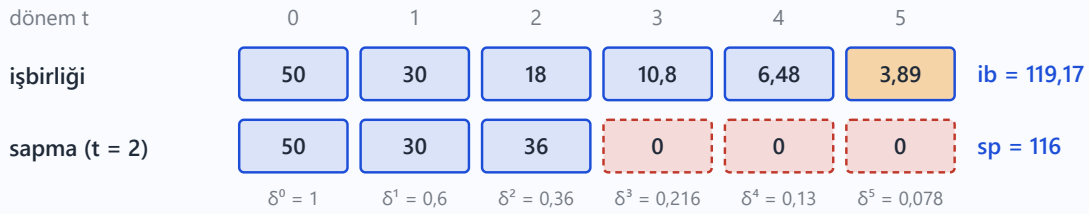
- `ib` işbirliği yolunun, `sp` sapma yolunun birikimli bugünkü değeridir; ikisi de 0 ile başlar.
- Döngü `t = 0, 1, …, T − 1` dönemleri üzerinde gezinir. Her dönemin kârı `delta**t` ile çarpılır; bu, *t* dönem sonraki kârın bugünkü değeridir.
- İşbirliği yolu her dönem 50 ekler.
- Sapma yolu, sapma anından önce (`t < s`) işbirliği yapar ve 50 ekler. Sapma anında (`t == s`) bütün pazarı alır ve 100 ekler. Sonra (`t > s`) tetik devrededir ve hiçbir şey eklenmez.
- Fonksiyon iki değeri iki basamağa yuvarlayarak döndürür.

İlk çağrı $\delta = 0{,}6$, $s = 2$, $T = 6$ içindir. İzleme Tablo 14.4'tedir.

t	δ^t	işbirliği ekler	ib	sapma ekler	sp
0	1	50	50	50	50
1	0,6	30	80	30	80
2	0,36	18	98	36 (sapma)	116
3	0,216	10,8	108,8	0	116
4	0,1296	6,48	115,28	0	116
5	0,0778	3,89	119,17	0	116

Tablo 14.4: `karsilastir(0.6, 2, 6)` izlemesi.

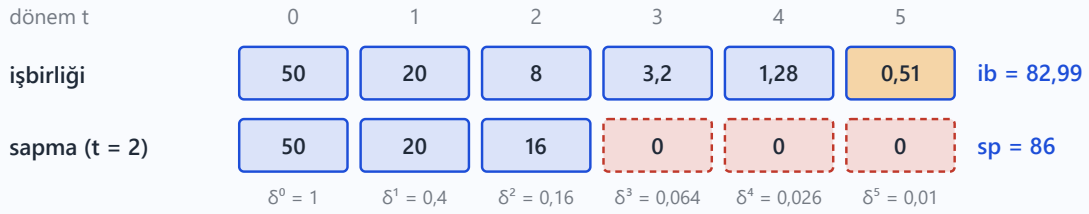
İlk iki dönemde iki yol aynıdır. *t = 2*'de sapan firma pazarın tümünü alır: $0{,}36 \cdot 100 = 36$; işbirliği yolu yalnızca $0{,}36 \cdot 50 = 18$ ekler ve sapma 18 öne geçer. Sonra tetik devreye girer; sapma yolu artık bir şey eklemez, işbirliği yolu ise her dönem eklemeye devam eder. *t = 4*'te fark 0,72'ye iner, *t = 5*'te işbirliği öne geçer: 119,17 > 116. Kısa vadeli 18'lik kazanç, kaybedilen gelecek kârlarla geri ödenmiştir.



Şekil 14.5: *δ = 0,6* için iki yolun dönem kârlarının bugünkü değerleri. Sapma *t = 2*'de 36 kazandırır, sonra üç dönem 0 gelir (kesikli kırmızı kutular). Altı dönemin toplamı işbirliği için 119,17, sapma için 116'dır.

Şekil 14.5 altı dönemlik pencereyi gösterir. Sonsuz ufukta fark daha da büyüktür: işbirliğinin değeri *50/(1 − 0,6) = 125*'tir, sapmanınki ise 116'da kalır (sapma anından sonra hiçbir şey eklenmez). Yani *δ = 0,6*'da sapma kârlı değildir; bu, *δ ≥ 1/2* koşuluyla tutarlıdır.

İkinci çağrı *δ = 0,4* içindir. Bu kez gelecek daha hızlı değer kaybeder.



Şekil 14.6: *δ = 0,4* için aynı karşılaştırma. Sapmanın *t = 2*'deki kazancı 16'dır ve sonraki kayıplar bunu karşılamaz: işbirliği 82,99, sapma 86.

Şekil 14.6'da işbirliği yolu 50, 20, 8, 3,2, 1,28 ve 0,51 ekleyerek 82,99'a ulaşır. Sapma yolu 50, 20 ve 16 ekleyip 86'da kalır. Sonsuz ufukta işbirliğinin değeri *50/0,6 ≈ 83,3*'tür; bu da 86'dan küçüktür. Sapma kârlıdır ve danışıklılık çöker. Eşik *δ = 1/2*'dir: bu değerde iki yol sonsuzda eşit olur (*50/0,5 = 100*).

Sık yapılan hata

Sonlu bir pencereye bakıp karar vermek. Kod 6 dönemi toplar; bu yalnızca bir gösterimdir. Tetik koşulu sonsuz ufukta tanımlanır. *δ = 0,6*'da 6 dönem sonunda fark 3,17'dir, sonsuzda ise *125 − 116 = 9*'dur. *δ* eşiğe çok yakınsa kısa bir pencere yanlış sonuç verebilir.

14.2.6 Danışıklılığı kolaylaştıran ve zorlaştıran koşullar

Tetik koşulu, danışıklılığın hangi piyasalarda daha olası olduğunu da söyler. İlk etken firma sayısıdır. *n* firma tekel kârını eşit paylaşıyorsa her biri dönem başına *100/n* alır. Sapan firma yine bir dönem bütün pazarı (100) alır, sonra 0. Koşul şöyle olur:

$$(100/n) / (1 - \delta) \geq 100 \iff 1/n \geq 1 - \delta \iff \delta \geq 1 - 1/n$$

Firma n	Pay 100/n	Gereken δ
2	50	0,50
3	33,3	0,67
4	25	0,75
5	20	0,80
10	10	0,90

Tablo 14.5: Firma sayısına göre işbirliği payı ve gereken en küçük δ.

Firma çoğaldıkça işbirliği payı küçülür, sapmanın getirisi ise aynı kalır. Bu yüzden kalabalık piyasada danışıklılığı sürdürmek için çok daha fazla sabır gerekir. İki firmada yeten $\delta = 0{,}6$, beş firmalı bir piyasada yetmez.

Örnek 14.3

Üç firma tekel kârını (100) paylaşıyor ve $\delta = 0{,}7$. Danışıklılık sürer mi?

1. İşbirliğinde her firmanın payı $100/3 \approx 33{,}3$'tür; değeri $33{,}3/(1 - 0{,}7) = 33{,}3/0{,}3 \approx 111{,}1$.
2. Sapmanın değeri 100'dür.
3. $111{,}1 \geq 100$ olduğu için sapma kârlı değildir; danışıklılık sürer. Formülle de: $\delta = 0{,}7 \geq 1 - 1/3 \approx 0{,}67$ ✓.
4. Dördüncü bir firma girerse eşik 0,75'e çıkar. $\delta = 0{,}7$ artık yetmez: $25/0{,}3 \approx 83{,}3 < 100$. Piyasaya yeni bir rakibin girmesi danışıklılığı bozar.

Firma sayısı dışında üç etken daha önemlidir. Bunlar algoritmik fiyatlandırmanın danışıklılığı neden kolaylaştırabileceğini açıklar.

Sık fiyat güncelleme
: δ bir dönemin uzunluğuna bağlıdır. Fiyatlar yılda bir değişiyorsa bir dönem sonraki kâr bugüne göre epey değer kaybeder. Fiyat dakikada bir güncelleniyorsa bir dakika sonraki kâr neredeyse bugünkü kadar değerlidir; dönem başına $\delta \approx 1$ olur. Ayrıca sapanın kazancı, rakip tepki verene kadar geçen birkaç dakikayla sınırlı kalır. İki etki de danışıklılığı kolaylaştırır.

Şeffaf fiyatlar
: Rakip fiyatı anında görülebiliyorsa sapma hemen fark edilir ve ceza gecikmez. Sapma ancak birkaç dönem sonra fark edilebilseydi sapan o dönemlerin hepsinde 100 alırdı ve eşik yükselirdi.

Fiyat eşleme garantisi
: "Rakip daha ucuzsa farkı öderiz" garantisi ilk bakışta tüketici dostu görünür. Oysa bu garanti, fiyat kırmanın kazancını sıfırlar: rakip fiyat kırarsa müşteri yine de garantiyi veren firmadan alır. Fiyat kırmak kimseye müşteri kazandırmadığı için kimse kırmaz ve danışıklılık korunur.

Düzenleme önerileri

Bir fiyatlandırma algoritmasını piyasaya sürmeden önce benzetim ortamında başka algoritmalarla karşılaştırıp sınamak; algoritmanın karar kayıtlarını denetime açmak. Bu öneriler, danışıklılığın bir anlaşmadan değil bir dinamikten doğabileceği gözlemine dayanır.

Sık yapılan hata

Firma sayısını unutmak. İki firmada yeten sabır kalabalık piyasada yetmez. Eşik $\delta \geq 1 - 1/n$'dir; $n = 5$ iken 0,80 olur. Koşulu her zaman $\delta \geq 1/2$ diye ezberlemek yanlıştır.

14.2.7 Turnuvaya dönüş

Giriş turnuvasının ilk oyunu Mahkûmun İkilemi'dir. Kazançlar karşılıklı işbirliğinde (3, 3), işbirliği yapan ile ihanet eden arasında (0, 5) ya da (5, 0), karşılıklı ihanette (1, 1)'dir. Her maç 60 turdur. Rastgele ajan her turda yarı yarıya karar verdiği için onun maçları beklenen değerle hesaplanır. En İyi Tepki ajanı ilk turda işbirliği yapar. Sonra rakibin en sık hamlesine en iyi tepki verir. Mahkûmun İkilemi'nde ihanet baskın strateji olduğu için rakip ne yaparsa yapsın en iyi tepki ihanettir. Yani En İyi Tepki ajanı ilk turdan sonra hep ihanet eder.

Örnek 14.4

Her ajanın üç maçının beklenen toplamını tek tek hesaplayalım.

1. **Hep İhanet.** Hep İşbirliği'ne karşı her tur 5: $60 \cdot 5 = 300$. Rastgele'ye karşı her tur yarı yarıya 5 ya da 1, beklenen 3: $60 \cdot 3 = 180$. En İyi Tepki'ye karşı ilk tur 5 (rakip işbirliği yapar), sonraki 59 tur 1: $5 + 59 = 64$. Toplam $300 + 180 + 64 = 544$.
2. **En İyi Tepki.** Hep İşbirliği'ne karşı ilk tur 3, sonra 59 tur 5: $3 + 295 = 298$. Hep İhanet'e karşı ilk tur 0, sonra 59 tur 1: 59. Rastgele'ye karşı ilk tur beklenen $(3 + 0)/2 = 1{,}5$, sonra her tur $(5 + 1)/2 = 3$: $1{,}5 + 177 = 178{,}5$. Toplam $535{,}5$.
3. **Rastgele.** Hep İşbirliği'ne karşı her tur $(3 + 5)/2 = 4$: 240. Hep İhanet'e karşı her tur $(0 + 1)/2 = 0{,}5$: 30. En İyi Tepki'ye karşı ilk tur 4, sonra 59 tur 0,5: $4 + 29{,}5 = 33{,}5$. Toplam $303{,}5$.
4. **Hep İşbirliği.** Hep İhanet'e karşı 0. Rastgele'ye karşı her tur $(3 + 0)/2 = 1{,}5$: 90. En İyi Tepki'ye karşı ilk tur 3, sonra 0: 3. Toplam 93.

Ajan	Beklenen toplam
Hep İhanet	544
En İyi Tepki	535,5
Rastgele	303,5
Hep İşbirliği	93

Tablo 14.6: Mahkûmun İkilemi'nde (Seviye 1, maç başına 60 tur) bek-

lenen toplamlar.

İlk sorunun cevabı Hep İhanet'tir. En İyi Tepki çok yakından ikinci olur; ikisinin farkı yalnızca ilk turdaki işbirliğinden doğar. Bu turnuvada karşılıklı işbirliği neredeyse hiç görülmez, çünkü hiçbir ajan geçmişe göre ödül ya da ceza vermez. İşbirliğini sürdürecek bir tehdit olmadığından baskın strateji kazanır.

İkinci soru için oyun değiştirilir. **Yazı-Tura Eşleştirme**'de bir oyuncu iki seçim aynıysa, öteki farklıysa kazanır; sıfır toplamlıdır ve saf dengesi yoktur. Burada sıralama değişir. Hep aynı hamleyi yapan iki sabit ajan kolayca sömürülür: En İyi Tepki ajanı onların hamlesini öğrenir ve ilk turdan sonra hep kazanır. Bu sayede yaklaşık +118 ile birinci olur. Rastgele ajan ise hiçbir rakibe karşı sömürülemez; her maçta beklenen kazancı 0'dır. Bu, Yazı-Tura'nın karışık dengesinin (yarı yarıya oynamak) güvence değeridir.

Gerçek turnuvalar da aynı dersi verir. Axelrod'un (1980) tekrarlanan Mahkûmun İkilemi turnuvasını **Tit-for-Tat** (ilk turda işbirliği yapan, sonra rakibin son hamlesini tekrarlayan strateji) kazandı. Karşılıklılık ve ceza tehdidi işbirliğini korudu. Buradaki dört ajanda geçmişe göre cezalandırma yoktur; bu yüzden işbirliği doğmaz. Calvano vd. (2020) çalışmasındaki fiyat ajanları ise önceki dönemi hatırlar ve cezalandırmayı öğrenip yüksek fiyatı sürdürebilir. Fark hafızadadır.

Turnuvanın genel dersi şudur: **evrensel en iyi strateji yoktur**. Bir stratejinin başarısı oynanan oyuna, rakip kitlesine, gürültüye ve hafızaya bağlıdır. Bir ajanı tek bir ortamda sınayıp her yerde iyi olacağını varsaymak yanlıştır. Bu gözlem, ajanları değerlendirmek için birden çok oyunda ve birden çok rakiple yapılan, tekrarlanabilir deneylerin neden gerekli olduğunu da açıklar (14.2.8).

14.2.8 Tekrarlanabilir deney protokolü

Dil modeli ajanlarının hamleleri rastgelelik içerir: aynı eşleşme iki kez farklı skor verebilir. Bir sonuç, başka biri aynı deneyi tekrarlayıp aynı sonucu alabiliyorsa güvenilirdir. Buna **tekrarlanabilirlik** (*reproducibility*) denir. Bir ajan turnuvası deneyinde şunlar kayda geçirilir:

- sabit rastgele tohum (*random seed*);
- istemin (*prompt*) tam metni;
- kullanılan modelin sürümü ve sıcaklık (*temperature*) değeri;
- rakip sırası ve tur sayısı: sabit tutulur ve raporda yazılır.

Raporda ise şunlar verilir:

- her eşleşme en az N kez tekrarlanır (örneğin $N = 30$);
- ortalama ile birlikte yayılım: standart sapma ya da %95 güven aralığı;
- kod ve oyun kayıtları açık bir depoda paylaşılır.

Güven aralığı (*confidence interval*), tek bir ortalama yerine, gerçek ortalamanın büyük olasılıkla içinde bulunduğu bir aralık verir. N tekrarın ortalaması m ve standart sapması s ise yaklaşık %95 güven aralığı $m \pm 1{,}96 \cdot s / \sqrt{N}$'dir. Burada $s/\sqrt{N}$, ortalamanın kendi belirsizliğidir; tekrar sayısı arttıkça kü-

çülür.

Örnek 14.5

30 tekrarda ortalama skor 2,6, standart sapma 0,4.

1. $\sqrt{30} \approx 5{,}48$, $0{,}4 / 5{,}48 \approx 0{,}073$.
2. Yarı genişlik: $1{,}96 \cdot 0{,}073 \approx 0{,}14$.
3. Aralık: $2{,}6 \pm 0{,}14$, yani [2,46; 2,74].

Başka bir ajanın ortalaması bu aralığın içinde kalıyorsa, iki ajan arasındaki farkın gürültüden kaynaklanabileceği akılda tutulmalıdır.

14.3 Öğrenen mekanizmalar

Mekanizma tasarımında kuralı, istenen sonucu verecek biçimde tasarımcı seçer. Klasik yolda bu seçim bir ispatla yapılır. Bu kısım ikinci bir yolu inceler: kuralı bir optimizasyon probleminin çözümü olarak ya da doğrudan veriden öğrenmek. Öğrenmenin getirdiği iki yeni sorun da ele alınır: az veride kendi başarısını abartma ve tek tek kişilerin verisinin mahremiyeti.

14.3.1 Otomatik mekanizma tasarımı

Mekanizma tasarımının klasik yolu şöyledir. Bir insan bir kural önerir: ikinci fiyat (Vickrey) açık artırması, VCG mekanizması ya da Myerson'un gelir optimal açık artırması gibi. Sonra kuralın teşvik uyumlu olduğu ve belirli bir amaç için optimal olduğu elle ispatlanır. Bu yol, tek nesne satışı gibi iyi yapılandırılmış problemlerde çok başarılıdır. Ama çok nesneli ve çok alıcılı durumlarda gelir optimal mekanizma çoğu zaman bilinmez; elle ispat edilecek bir aday bile yoktur.

Otomatik yolda mekanizma bir **optimizasyon probleminin çözümüdür**. Problemin bir amacı (gelir ya da sosyal refah) ve iki tür kısıtı vardır.

Tanım 14.5

Teşvik uyumluluğu (IC, *incentive compatibility*): Her oyuncu için gerçek tipini (değerini) bildirmek, başka herhangi bir tip bildirmekten daha kötü değildir. Kısaca, doğruyu söylemek en iyisidir.

Bireysel rasyonellik (IR, *individual rationality*): Her oyuncu için mekanizmaya katılmak, katılmamaktan daha kötü değildir. Kısaca, katılmak zarar ettirmez: beklenen faydası sıfırın altına düşmez.

Otomatik tasarımın iki temel biçimi vardır.

- **Doğrusal programlama.** Oyuncuların tipleri sonlu bir kümeden geliyorsa, her tip için satış olasılığı ve ödeme birer değişkendir. Amaç ve kısıtlar bu değişkenlerde doğrusal olur. Problem doğrusal programlamayla çözülür (Conitzer ve Sandholm, 2002). 14.3.2'deki örnek bu türdendir.
- **Sinir ağı ile açık artırma.** Tipler sürekli ve çok boyutluysa, mekanizma bir sinir ağıyla temsil edilir: ağ teklifleri alır, dağıtımı ve ödemeleri verir. IC kısıtı doğrudan yazılamaz; onun yerine bir **pişmanlık** cezası eklenir (Dütting vd., 2019). Burada pişmanlık, bir teklifçinin yalan söyleyerek elde edebileceği en büyük ek faydadır. Eğitim, geliri artırırken pişmanlığı sıfıra yaklaştırmaya çalışır.

Otomatik yolun üç sınırı vardır:

Yaklaşık IC
: Öğrenilen kural tam teşvik uyumlu olmayabilir; eğitimden sonra küçük bir pişmanlık kalır. Bu duruma **yaklaşık teşvik uyumluluğu** denir. Küçük bir yalan kazancı, çok sayıda katılımcının hesap yapmaya değer bulmayacağı kadar küçükse pratikte sorun olmayabilir; ama ispatlanmış bir garanti değildir.

Genelleme
: Kural, eğitim verisindeki değer dağılımına göre öğrenilir. Gerçek dağılım değişirse (örneğin bir ekonomik şokla) kural etkisizleşebilir.

Açıklanabilirlik
: Bir katılımcı “Neden bu fiyatı ödedim?” diye sorduğunda, sinir ağının ağırlıkları anlaşılır bir cevap vermez. Vickrey kuralı ise tek cümleyle açıklanabilir: “kazanan, ikinci en yüksek teklifi öder”.

14.3.2 Kod izleme: gelir()

Otomatik tasarımın en küçük örneği tek alıcı ve tek nesneyle kurulur. Alıcının iki olası tipi vardır: L tipi nesneye 1 değer verir ve olasılığı q_L'dir; H tipi 3 değer verir ve olasılığı $1 - q_L$'dir. Satıcı alıcının tipini bilmez; alıcıdan tipini bildirmesini ister. Mekanizma her bildirilen tip için bir satış olasılığı ve bir ödeme belirler. H tipine her zaman satılır. L tipine satış olasılığı x_L'dir. Ödemeler p_L ve p_H'dir.

Gelir maksimize edilirken iki kısıt sıkı (eşitlikle) sağlanır. Bunlar şöyle türetilir.

1. **L'nin katılım kısıtı (IR).** L tipinin faydası $1 \cdot x_L - p_L \geq 0$ olmalıdır. Satıcı geliri artırmak için p_L'yi bu sınıra kadar çıkarır: $p_L = x_L$. L'ye fiyat, L'nin ödemeye razı olduğu en yüksek değerdir.
2. **H'nin teşvik uyumluluğu kısıtı (IC).** H tipi “L'yim” derse nesneyi x_L olasılıkla alır ve p_L öder: faydası $3x_L - p_L$. Doğruyu söylerse faydası $3 - p_H$'dir. Doğruyu söylemek en az yalan kadar iyi olmalıdır: $3 - p_H \geq 3x_L - p_L$. Satıcı p_H'yi bu sınıra kadar çıkarır: $p_H = p_L + 3(1 - x_L)$.
3. **Beklenen gelir.** $G = q_L\, p_L + (1 - q_L)\, p_H$.

Ödemeler yerine konunca gelir yalnız x_L'ye bağlı kalır:

$$G(x_L) = q_L\, x_L + (1 - q_L)(3 - 2x_L) = 3(1 - q_L) + x_L(3q_L - 2)$$

Bu ifade x_L'de doğrusaldır. L'ye satmanın bir bedeli vardır: L'ye ucuza satmak, H'ye de bir yalan seçeneği açar ve H'nin fiyatını düşürür. Aşağıdaki kod geliri üç x_L değeri ve iki q_L için hesaplar.

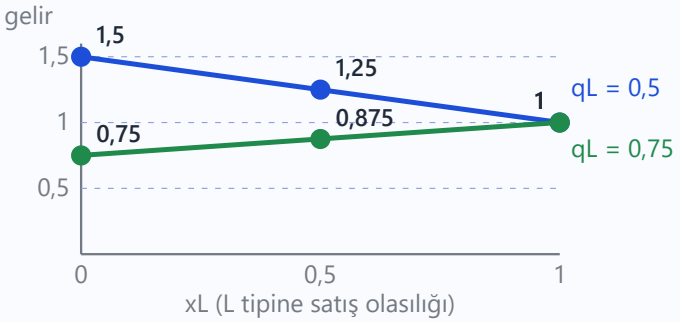
```
def gelir(xL, qL, vL=1, vH=3):
    pL = xL * vL                # L'nin IR kısıtı sıkı
    pH = pL + vH * (1 - xL)     # H'nin IC kısıtı sıkı
    return qL * pL + (1 - qL) * pH
for qL in [0.5, 0.75]:
    G = [gelir(x, qL) for x in [0, 0.5, 1]]
    print(qL, G)
```

Kod satır satır şöyle çalışır. `pL`, L'nin katılım kısıtını sıkı yapan ödemedir. `pH`, H'nin teşvik uyumluluğu kısıtını sıkı yapan ödemedir. Fonksiyon iki tipin olasılıklarıyla ağırlıklı ortalamayı döndürür. Döngü $q_L = 0{,}5$ ve $0{,}75$ için $x_L \in \{0; 0{,}5; 1\}$ değerlerini dener.

q_L	x_L	p_L	p_H	Gelir
0,5	0	0	3	$0{,}5 \cdot 3 = 1{,}5$
0,5	0,5	0,5	2	$0{,}25 + 1 = 1{,}25$
0,5	1	1	1	1
0,75	0	0	3	$0{,}25 \cdot 3 = 0{,}75$
0,75	0,5	0,5	2	$0{,}375 + 0{,}5 = 0{,}875$
0,75	1	1	1	1

Tablo 14.7: `gelir()` izlemesi.

$x_L = 0{,}5$ satırı ödünleşimi açıkça gösterir. L'ye yarı olasılıkla 0,5'ten satmak, H'ye "L desem $3 \cdot 0{,}5 - 0{,}5 = 1$ kazanırım" seçeneğini açar. H'yi doğruyu söylemeye razı etmek için onun fiyatı 3'ten 2'ye inmelidir. $q_L = 0{,}5$'te bu kayıp, L'den gelen ek gelirden büyüktür.



Şekil 14.7: İki tipli alıcıda gelirin x_L'ye göre değişimi. $q_L = 0{,}5$'te (mavi) gelir azalır ve en iyisi $x_L = 0$'dır; $q_L = 0{,}75$'te (yeşil) gelir artar ve en iyisi $x_L = 1$'dir.

Şekil 14.7'de iki doğru görülür. $q_L = 0{,}5$'te en iyisi $x_L = 0$'dır: yalnız H'ye 3'ten satılır, gelir 1,5. $q_L = 0{,}75$'te L tipi çoğunluktadır; yalnız H'ye satmak artık az kazandırır (0,75) ve en iyisi herkese 1'den satmaktır, gelir 1. Gelir formülündeki x_L katsayısı $3q_L - 2$'dir. Bu katsayı $q_L = 0{,}5$'te −0,5, $q_L = 0{,}75$'te +0,25'tir.

Önemli bir gözlem vardır: gelir x_L'de doğrusal olduğu için optimum her zaman bir uçtadır. Bu, doğrusal programlamanın genel bir özelliğidir: optimum, olurlu bölgenin bir köşesindedir. İki çözüm de aslında **ilan edilmiş bir fiyattır**: ya "fiyat 3" ya da "fiyat 1". Yani bu küçük örnekte otomatik tasarım, Myerson kuramının tek nesne için verdiği sonucu yeniden bulur: gelir optimal mekanizma, doğru seçilmiş bir rezerv fiyatıdır. Otomatik tasarımın asıl değeri, böyle bir kuramın bulunmadığı çok nesneli durumlarda ortaya çıkar.

14.3.3 Veriden rezerv fiyatı öğrenmek

Önceki örnekte satıcı tip dağılımını biliyordu. Gerçekte dağılım bilinmez; elde yalnızca geçmişte gözlenen değerler vardır. Bu durumda doğal bir yol, geçmiş veride en çok geliri getirecek fiyatı seçmektir.

Tanım 14.6

n örnek değer $v_1, \ldots, v_n$ verilsin. Fiyat p'de değeri p'ye eşit ya da büyük olan alıcılar satın alır. **Deneysel gelir** (*empirical revenue*), alıcı başına ortalama gelirdir:

$$g(p) = p \cdot (\text{değeri} \geq p \text{ olan örnek sayısı}) / n$$

Deneysel gelir maksimizasyonu, bu fonksiyonu en büyük yapan fiyatı seçmektir.

Hangi fiyatların denenmesi gerektiği şöyle bulunur. İki ardışık örnek değerin arasında bir fiyat seçilirse, alıcı sayısı değişmeden fiyat yükseltilebilir; bu da geliri artırır. Bu yüzden en iyi fiyat her zaman örnek değerlerden biridir. Değerler büyükten küçüğe sıralanır. k'inci en büyük değer fiyat olursa tam k alıcı satın alır (eşitlik yoksa). Bu yüzden yalnız n aday fiyatı denemek yeter.

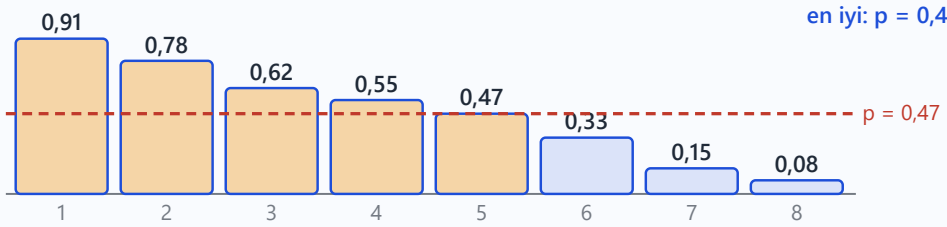
```
def en_iyi_fiyat(orn):
  v = sorted(orn, reverse=True)
  n = len(v)
  fiyat, gelir = 0, 0
  for k in range(1, n + 1):
    p = v[k - 1]        # k. en yüksek
    g = p * k / n       # k alıcı alır
    if g > gelir:
      fiyat, gelir = p, g
  return fiyat, gelir
```

Kod satır satır şöyle çalışır. `v`, örneklerin büyükten küçüğe sıralı hâlidir. `fiyat` ve `gelir`, şimdiye kadarki en iyi fiyatı ve onun deneysel gelirini tutar. Döngü *k = 1, ..., n* için *k*'inci en büyük değeri fiyat olarak dener. Bu fiyatta *k* alıcı satın alır ve deneysel gelir *p · k / n* olur. Gelir öncekinden büyükse en iyi fiyat güncellenir.

Örnekte geçmişte gözlenen 8 alıcı değeri 0,62; 0,15; 0,91; 0,47; 0,33; 0,78; 0,55; 0,08'dir. Sıralanınca 0,91; 0,78; 0,62; 0,55; 0,47; 0,33; 0,15; 0,08 olur.

k	p	g = p·k/8	En iyi (fiyat, gelir)
1	0,91	0,114	(0,91; 0,114) yeni
2	0,78	0,195	(0,78; 0,195) yeni
3	0,62	0,2325	(0,62; 0,2325) yeni
4	0,55	0,275	(0,55; 0,275) yeni
5	0,47	0,294	(0,47; 0,294) yeni
6	0,33	0,2475	değişmez
7	0,15	0,131	değişmez
8	0,08	0,08	değişmez

Tablo 14.8: `en_iyi_fiyat()` izlemesi, *n = 8*.



Şekil 14.8: Sıralı 8 örnek değer ve öğrenilen fiyat. *p = 0,47*'de değeri bu çizginin üstünde kalan 5 alıcı (turuncu) satın alır; deneysel gelir *0,47 · 5/8 ≈ 0,294*.

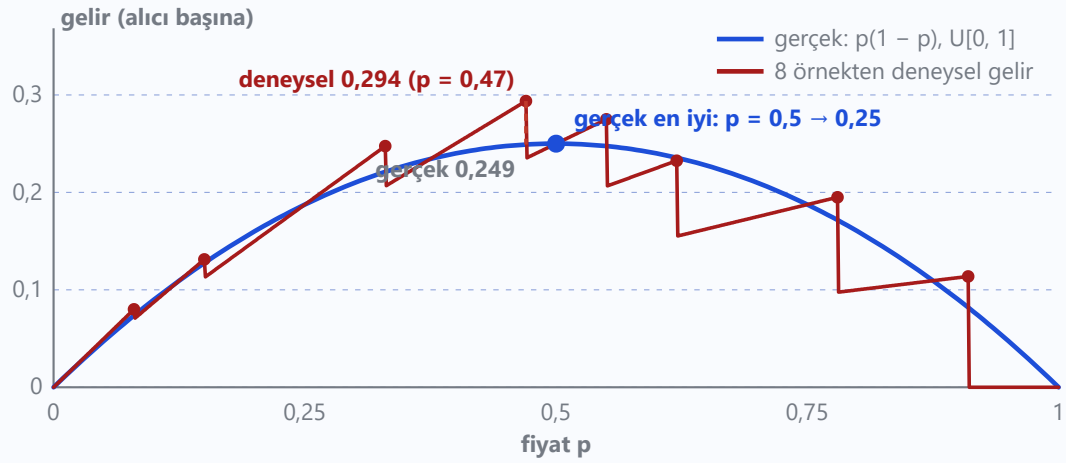
Şekil 14.8 öğrenilen fiyatı gösterir. Fiyat düştükçe alıcı sayısı artar ama her alıcıdan alınan para azalır. İlk beş adımda alıcı artışının etkisi ağır basar; altıncı adımda (0,33) fiyat düşüşünün etkisi ağır basmaya başlar. Öğrenilen fiyat 0,47, deneysel gelir yaklaşık 0,294'tür.

14.3.4 Deneysel ve gerçek gelir

Öğrenilen fiyatın gelecekte gerçekten ne kazandıracağı, değerlerin geldiği gerçek dağılıma bağlıdır. Örneğin değerlerin *U[0, 1]* dağılımından geldiğini varsayalım. Bu, 0 ile 1 arasındaki her değerin eşit olasılıklı olması demektir. Bu dağılımda bir alıcının değerinin *p*'den büyük olma olasılığı *1 − p*'dir. Bu yüzden fiyat *p*'nin alıcı başına **gerçek geliri** şudur:

$$G(p) = p \cdot (1 - p)$$

Bu fonksiyon bir paraboldür. En büyük değeri ortadaki noktadadır: *p = 0,5*'te *G = 0,5 · 0,5 = 0,25*. Öğrenilen fiyat için gerçek gelir *0,47 · 0,53 ≈ 0,249*'dur. Yani öğrenilen fiyat neredeyse en iyisi kadar iyidir. Ama deneysel gelir (0,294), gerçek gelirden (0,249) belirgin biçimde yüksektir.



Şekil 14.9: Deneysel gelir eğrisi (kırmızı, basamaklı) ve gerçek gelir eğrisi *p(1 − p)* (mavi). Deneysel eğri örnek değerlerde sıçrar ve en yüksek noktası bir örnek değerdedir (0,47). Gerçek en iyi *p = 0,5*'te 0,25'tir.

Şekil 14.9 iki eğriyi karşılaştırır. Deneysel eğri bir testere dişi gibidir: iki örnek değer arasında fiyatla doğrusal artar, bir örnek değeri geçince o alıcı kaybedildiği için aşağı düşer. Gerçek eğri ise düzgün bir paraboldür. Deneysel eğri bazı fiyatlarda gerçeğin üstünde, bazılarında altındadır.

Seçilen fiyatın deneysel başarısının abartılı olması bir rastlantı değildir. Fiyat, deneysel eğrinin *en yüksek* noktası olarak seçilmiştir. Deneysel eğrinin gerçeğin üstüne çıktığı yerler, örneklerin şans eseri bol olduğu yerlerdir. En yüksek nokta seçilince bu şanslı dalgalanmalardan biri de seçilmiş olur. Ölçülen ve seçilen aynı veri olduğu için bir **iyimser yanlılık** oluşur. Gerçek en iyi gelirle (0,25) seçilen fiyatın gerçek geliri (0,249) arasındaki fark küçüktür; ama örnek azaldıkça hem bu fark hem de iyimser yanlılık büyür.

Sık yapılan hata

Deneysel geliri gerçek sanmak. Aynı veride seçilip ölçülen fiyatın başarısı abartılıdır. Bu örnekte 0,294 ölçülür ama gerçek 0,249'dur; fark %18'dir. Bir fiyatın, açık artırma kuralının ya da herhangi bir öğrenilmiş modelin başarısını ölçmek için, seçimde kullanılmamış ayrı bir veri gerekir.

14.3.5 Kaç örnek yeter? Örneklem karmaşıklığı

Örnek sayısı arttıkça deneysel eğri gerçeğe yaklaşır. Bunu görmek için benzetim yapılabilir: *U[0, 1]*'den *n* değer çekilir, deneysel gelir maksimize edilir, seçilen fiyatın deneysel ve gerçek geliri kaydedilir. Bu işlem her *n* için yüzlerce kez tekrarlanıp ortalaması alınır.

Örnek n	Deneysel gelir	Gerçek gelir	0,25 – gerçek
10	0,311	0,224	0,0262
100	0,263	0,245	0,005
1.000	0,253	0,249	0,0011
10.000	0,25	0,25	0,0002

Tablo 14.9: Örnek sayısına göre deneysel ve gerçek gelir (değerler *U[0, 1]*, yüzlerce tekrarın ortalaması). En iyi gelir 0,25'tir ($p = 0{,}5$).

Tablo iki şey gösterir. Birincisi, **aşırı uyum** (*overfitting*): az örnekte deneysel gelir gerçeğinden çok yüksek görünür ($n = 10$'da 0,311'e karşı 0,224). Öğrenilen fiyat, gerçek dağılımın değil o belirli örneklerin özelliklerine uymuştur; bir anlamda örneklere "ezberlenmiştir". İkincisi, örnek arttıkça iki sütun birbirine ve 0,25'e yaklaşır. Gerçek gelirdeki kayıp her on kat örnekte kabaca beşte birine iner.

Tanım 14.7

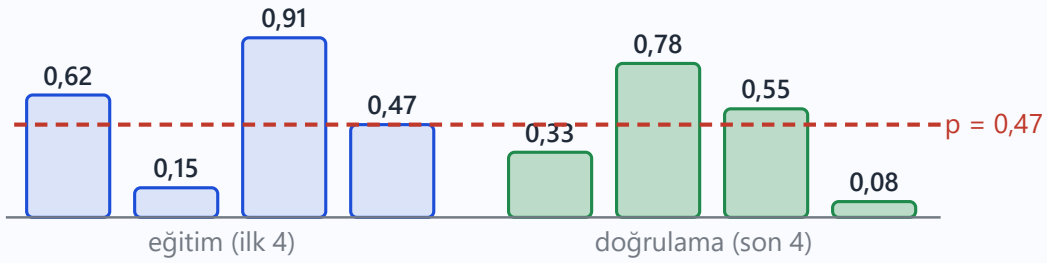
Örneklem karmaşıklığı (*sample complexity*): En iyi gelire istenen yakınlıkta (örneğin en fazla ε eksik) bir fiyatı, yüksek olasılıkla (örneğin %95) bulmak için gereken en az örnek sayısıdır.

Cole ve Roughgarden (2014) bu soruyu açık artırmalar için inceledi. Sonuç kabaca şöyledir: değer dağılımı çok uç değerler üretmiyorsa, iyi bir mekanizma için gereken örnek sayısı teklifçi sayısı ve istenen doğrulukla **ölçülü** (polinom) biçimde artar. Yani "biraz daha doğruluk" için örnek sayısı patlamaz. Uç değerli dağılımlarda ise böyle bir garanti verilemeyebilir. Bunun nedeni sezgiyle anlaşılır: çok nadir ama çok büyük değerler gelirin önemli bir kısmını oluşturuyorsa, bu değerleri yeterince görmek için çok fazla örnek gerekir.

Bu alt başlığın dersi makine öğrenmesinden tanıdıktır: veriden öğrenilen bir mekanizma, az veride kendi başarısını abartır. Eğitim ve değerlendirme verisini ayırmak, makine öğrenmesinde olduğu kadar mekanizma tasarımında da gereklidir.

14.3.6 Seçmek ve ölçmek için ayrı veri

İyimser yanlılığı önlemenin en basit yolu veriyi ikiye ayırmaktır. **Eğitim verisi** fiyatı seçmek için, **doğrulama verisi** seçilen fiyatı ölçmek için kullanılır. Doğrulama verisi seçimde kullanılmadığı için onun üzerinde yapılan ölçüm yanlı değildir.



Şekil 14.10: 8 örneğin ilk 4'ü eğitim, son 4'ü doğrulama için ayrılmıştır. Eğitimde seçilen fiyat 0,47'dir; doğrulamada bu fiyatın üstünde kalan yalnız 0,78 ve 0,55'tir.

Örnek 14.6

Şekil 14.10'daki bölme kullanılır: eğitim = 0,62; 0,15; 0,91; 0,47 ve doğrulama = 0,33; 0,78; 0,55; 0,08.

1. **Eğitimde fiyat seçimi.** Sıralı eğitim değerleri 0,91; 0,62; 0,47; 0,15. Deneysel gelirler: $0{,}91 \cdot 1/4 \approx 0{,}228$, $0{,}62 \cdot 2/4 = 0{,}31$, $0{,}47 \cdot 3/4 \approx 0{,}353$, $0{,}15 \cdot 4/4 = 0{,}15$. En iyisi $p = 0{,}47$, eğitim geliri 0,353.
2. **Doğrulamada ölçüm.** Doğrulama değerlerinden 0,78 ve 0,55 fiyatın üstündedir: $0{,}47 \cdot 2/4 = 0{,}235$.
3. **Karşılaştırma.** Gerçek gelir $0{,}47 \cdot 0{,}53 \approx 0{,}249$'dur. Eğitim ölçümü (0,353) çok iyimserdir; doğrulama ölçümü (0,235) gerçeğe yakındır.
4. **Roller değişince.** Eğitim = son 4 olursa sıralı değerler 0,78; 0,55; 0,33; 0,08 ve gelirler 0,195; 0,275; 0,2475; 0,08'dir. Seçilen fiyat 0,55'tir. İlk 4'te bu fiyatın üstünde 0,62 ve 0,91 vardır: $0{,}55 \cdot 2/4 = 0{,}275$.
5. **Çapraz doğrulama.** İki doğrulama tahmini ortalanır: $(0{,}235 + 0{,}275)/2 = 0{,}255$.

Ölçüm	Hesap	Gelir
Eğitimde (seçildiği veri)	$0{,}47 \cdot 3/4$	0,353
Doğrulamada	$0{,}47 \cdot 2/4$	0,235
Gerçek, $U[0, 1]$	$0{,}47 \cdot 0{,}53$	0,249

Tablo 14.10: $p = 0{,}47$ fiyatının üç ayrı ölçümü.

Bu yöntemin bir bedeli vardır: fiyat artık 8 değil yalnız 4 örnekle seçildiği için daha gürültülüdür. Çapraz doğrulama, bütün veriyi hem seçim hem ölçüm için sırayla kullanarak bu bedeli azaltır.

Mekanizmalara özgü bir risk daha vardır. Makine öğrenmesinde veri genellikle doğadan gelir; veriyi üretenler modelin ne yapacağını umursamaz. Açık artırmada ise veri, teklif veren alıcılardan gelir. Alıcılar geçmiş tekliflerinin gelecekteki fiyatı belirlediğini bilirse, bugün bilerek düşük teklif verebilir:

biraz kaybeder ama gelecekte daha düşük bir rezerv fiyatıyla karşılaşır. Yani **veri de stratejik olabilir**. Bu durumda öğrenme algoritmasının kendisi de bir mekanizmadır ve onun da teşvik uyumlu olması gerekir.

14.3.7 Diferansiyel mahremiyet ve üstel mekanizma

Veriden öğrenmenin bir başka sorunu mahremiyettir. Öğrenilen fiyat ilan edilince, bu fiyat geçmiş alıcıların değerleri hakkında bir şeyler sızdırır. Örneğin yalnız bir alıcının verisi eklenince fiyat çok değişiyorsa, fiyattan o alıcının değeri çıkarılabilir.

Tanım 14.8

Rastgele çıktı veren bir mekanizma M verilsin. Yalnız bir kişinin verisinde farklı olan her D ve D' veri kümesi ve her o sonucu için

$$P[M(D) = o] \le e^{\varepsilon} \cdot P[M(D') = o]$$

sağlanıyorsa M, **ε-diferansiyel mahremiyetlidir** (ε-*differentially private*).

Sade dille: tek bir kişinin verisi eklense ya da çıkarılsa da her sonucun olasılığı ancak biraz değişir. ε mahremiyetin ölçüsüdür. Küçük ε daha çok mahremiyet demektir: olasılıklar en fazla e^{ε} kat değişebilir. $\varepsilon = 0$ tam mahremiyettir ama sonuç veriye hiç bağlı olamaz; ε büyüdükçe sonuç veriye daha çok uyar ama mahremiyet azalır. Tanım rastgele bir mekanizma gerektirir. Kesin (rastgele olmayan) bir kural, bir kişinin verisiyle bir sonuçtan ötekine atlayabilir; o sonucun olasılığı 1'den 0'a düşer ve oran sonsuz olur.

McSherry ve Talwar (2007) bu tanımı sağlayan genel bir araç verdi.

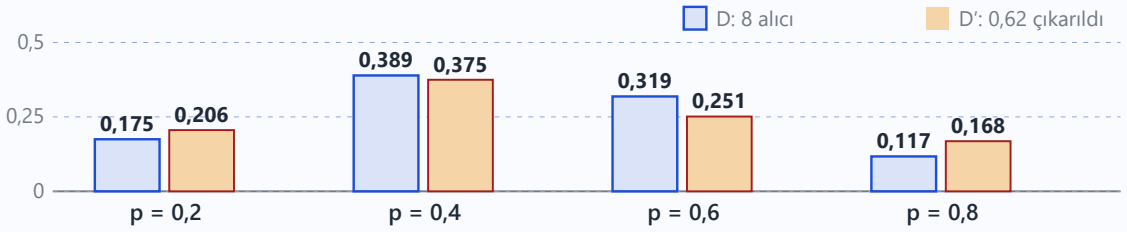
Tanım 14.9

Üstel mekanizma (*exponential mechanism*): Her aday sonuç p'nin bir puanı $R(p)$ vardır (örneğin fiyatın geliri). Δ, tek bir kişi çıkarılınca bir puanın en çok ne kadar değişebileceğidir. Mekanizma her adayı ağırlığıyla orantılı olasılıkla seçer:

$$P[p \text{ seçilir}] = \exp(\varepsilon \cdot R(p) / 2\Delta) \,/\, \Sigma_{p'} \exp(\varepsilon \cdot R(p') / 2\Delta)$$

Mekanizma en iyi fiyatı kesin olarak seçmez, bir kura çeker. İyi fiyatlar üstel olarak daha yüksek olasılık alır; kötü fiyatlar küçük ama sıfır olmayan bir olasılık alır. Formüldeki 2'nin nedeni şudur: bir kişi çıkarılınca hem payın (seçilen adayın ağırlığı) hem de paydanın (bütün ağırlıkların toplamı) değişebilir. Her biri en çok $e^{\varepsilon/2}$ kat değişir; ikisi birlikte en çok e^{ε} kat değişim üretir. Bu da tanımdaki sınırdır.

Bu mekanizmanın oyun kuramıyla güçlü bir bağı vardır. Tek bir kişinin raporu sonucu az etkilediği için, o kişinin yalan söyleyerek elde edebileceği kazanç da azdır. Yani diferansiyel mahremiyet **yaklaşık teşvik uyumluluğu** sağlar; üstelik mahremiyetle birlikte. Bu, mahremiyetin ve teşviklerin aynı matematiksel araçla ele alınabildiği şaşırtıcı bir sonuçtur.



Şekil 14.11: Rezerv fiyatı örneğinde üstel mekanizmanın seçim olasılıkları, $\varepsilon = 2$. Mavi: 8 alıcının verisi (D); turuncu: 0,62 değerli alıcı çıkarılınca (D′). İki dağılım birbirine yakındır; en büyük oran 1,44'tür.

Şekil 14.11 tanımın ne söylediğini görselleştirir. Bir alıcı veriden çıkarılınca seçim olasılıkları değişir ama çok az değişir. Bu örnekte en büyük değişim 1,44 kattır; tanımın izin verdiği sınır ise $e^2 \approx 7{,}39$'dur. Olasılıkların adım adım hesabı 14.3.8'dedir.

Sık yapılan hata

Üstel mekanizmada doğrusal orantı kurmak. Olasılık gelirle değil $\exp(\varepsilon \cdot R / 2\Delta)$ ile orantılıdır. Gelir farkı olasılık oranına üstel yansır: iki adayın puan farkı 1 ise ve $\varepsilon = 2$, $\Delta = 1$ ise olasılık oranı $e^1 \approx 2{,}72$'dir, puan oranı değildir.

14.3.8 Kod izleme: ustel()

Aşağıdaki kod, veriden fiyat öğrenme örneğindeki 8 alıcı değeriyle üstel mekanizmayı uygular. Aday rezerv fiyatları 0,2; 0,4; 0,6; 0,8'dir. Puan, fiyat ile o fiyattan satın alacak alıcı sayısının çarpımıdır (toplam gelir). Fiyatlar 1'den küçük olduğu için bir alıcı çıkarılınca bir fiyatın puanı en çok 1 değişir; bu yüzden $\Delta = 1$'dir. $\varepsilon = 2$ seçilince ağırlık formülü $\exp(2R/2) = e^R$'ye sadeleşir.

```
import math
def ustel(v, fiyatlar, eps):
    R = [p * sum(x >= p for x in v) for p in fiyatlar]
    w = [math.exp(eps * r / 2) for r in R]   # Δ = 1
    W = sum(w)
    return [x / W for x in w]
v = [0.62, 0.15, 0.91, 0.47, 0.33, 0.78, 0.55, 0.08]
P = [0.2, 0.4, 0.6, 0.8]
q = ustel(v, P, 2)
print([round(x, 3) for x in q])
q = ustel(v[1:], P, 2)          # 0.62 çıkarıldı
print([round(x, 3) for x in q])
```

Kod satır satır şöyle çalışır. `R`, her aday fiyatın puanını hesaplar; `sum(x >= p for x in v)`, değeri fiyata eşit ya da büyük olan alıcıları sayar. `w`, ağırlıklardır. `W`, ağırlıkların toplamıdır. Fonksiyon her ağırlığı toplama bölerek olasılıkları döndürür. Kod önce bütün veriyle, sonra ilk alıcı (0,62) çıkarılmış veriyle çalıştırılır.

p	R (D)	w = e^{R}	q (D)	R (D′)	w′	q (D′)	Oran
0,2	*0,2 · 6 = 1,20*	3,32	0,175	*0,2 · 5 = 1,00*	2,72	0,206	1,18
0,4	*0,4 · 5 = 2,00*	7,39	0,389	*0,4 · 4 = 1,60*	4,95	0,375	1,04
0,6	*0,6 · 3 = 1,80*	6,05	0,319	*0,6 · 2 = 1,20*	3,32	0,251	1,27
0,8	*0,8 · 1 = 0,80*	2,23	0,117	*0,8 · 1 = 0,80*	2,23	0,168	1,44
Toplam		W = 18,98	1		W′ = 13,22	1	

Tablo 14.11: `ustel()` izlemesi, $\varepsilon = 2$, $\Delta = 1$. D: 8 alıcı; D′: 0,62 çıkarıldı.

Bütün veriyle en olası fiyat *p = 0,4*'tür (%38,9); ama tek başına değildir, kötü fiyat 0,8'in bile %11,7 şansı vardır. Ağırlıklar arasındaki farklar üsteldir: puan 0,8'den 2'ye çıkınca ağırlık 2,23'ten 7,39'a çıkar. 0,62 değerli alıcı çıkarılınca *p = 0,2*, 0,4 ve 0,6'nın puanları düşer; en çok *p = 0,6*'nınki düşer (0,6). *p = 0,8*'in puanı değişmez, çünkü 0,62 zaten 0,8'in altındaydı. Toplam ağırlık küçüldüğü için *p = 0,8*'in olasılığı artar.

Son sütun iki dağılımın oranıdır (büyüğün küçüğe oranı). En büyük oran 1,44'tür ve $e^{2} \approx 7{,}39$ sınırının çok altındadır. Tek kişinin verisi sonucu sınırlı etkiler; mekanizma ε-diferansiyel mahremiyetlidir. Sınır en kötü durum içindir; tipik bir veri kümesinde gerçek değişim çoğu zaman çok daha küçüktür.

14.4 Blokzincir mekanizmaları

Blokzincirler, kuralları herkese açık kodla uygulayan sistemlerdir. Bu kısım onları bir teknoloji olarak değil, bir mekanizma tasarımı ortamı olarak inceler. Üç mekanizma ele alınır: kıt blok alanını dağıtan işlem ücreti açık artırmaları, fiyatı bir formülle belirleyen otomatik piyasa yapıcı ve işlemlerin sırasını

belirleme gücünden doğan değer (MEV). Bu alanda kullanılan somut kurallar ve araçlar sık değişir; not, değişmeyen ilkeleri öne çıkarır.

14.4.1 Blokzincir ve akıllı sözleşmeler

Tanım 14.10

Akıllı sözleşme (*smart contract*): Kodu herkese açık olan ve bir blokzincirde, belirlenen koşullar oluşunca otomatik yürütülen programdır. Mekanizma tasarımı açısından kendi kendini uygulayan bir mekanizmadır.

Bu tanımın oyun kuramı için önemi şudur. Mekanizma tasarımı, tasarımcının ilan ettiği kurallara sonradan bağlı kalacağını, yani bir **taahhüt** (*commitment*) olduğunu varsayar. Örneğin ikinci fiyat açık artırmasında satıcı, teklifleri gördükten sonra "aslında birinci fiyatı ödeyeceksiniz" diyemez. Gerçek hayatta bu varsayım bir güvene ya da hukuka dayanır. Akıllı sözleşmede ise taahhüt kodla sağlanır: kural yayımlandıktan sonra kimse onu değiştiremez.

Ama bu ortam yeni bir sorun da getirir. Bekleyen işlemler, bloğa girmeden önce herkese görünür. Bir işlemin ne yapacağını önceden görmek, ondan kâr etmenin yolunu açar. Ayrıca işlemleri bloklara yerleştiren bir **blok üreticisi** vardır. Üretici hangi işlemin bloğa gireceğine ve hangi sırayla gireceğine karar verir. Blok alanı sınırlıdır; yani kıt bir kaynaktır ve kıt kaynak doğal olarak açık artırmayla dağıtılır.



Şekil 14.12: Bekleme havuzundaki işlemler blok üreticisine gider; üretici kapasite kadar işlemi seçer ve sıralar. Herkes bekleyen işlemleri görebildiği için sıra değerlidir.

Şekil 14.12 süreci gösterir. Kullanıcılar işlemlerini **bekleme havuzuna** (*mempool*) gönderir. Blok üreticisi havuzdan işlem seçip sıralar. Bu yapıda üç mekanizma tasarımı sorusu doğar:

- **İşlem ücreti açık artırması:** Kısıtlı blok alanı, işlem ücreti teklif eden kullanıcılara nasıl dağıtılmalı?
- **Merkeziyetsiz borsa:** Alım satımın karşı tarafı bir insan yerine bir formül olabilir mi? Bu, **otomatik piyasa yapıcı** (AMM, *automated market maker*) fikridir.

- **Zincir üstü yönetişim:** Bir sistemin kurallarını değiştirmek için jeton (*token*) ağırlıklı oylama yapılır. Oylama kurallarının bilinen sorunları (stratejik oy, Condorcet döngüleri, büyük oy sahiplerinin gücü) burada da geçerlidir.

14.4.2 İşlem ücreti açık artırmaları

Bir bloğa sığabilecek işlem sayısı sınırlıdır. Daha fazla işlem bekliyorsa hangilerinin gireceğine bir kuralla karar verilir. Bu kurala **işlem ücreti mekanizması** (*transaction fee mechanism*) denir. İki tasarımı karşılaştıralım.

Birinci fiyat tasarımı. İlk blokzincirlerin kullandığı tasarımda her işlem bir ücret teklif eder. Üretici en yüksek teklifleri alır ve herkes kendi teklifini öder. Bu, bildiğimiz birinci fiyat açık artırmasıdır. Bekleyen beş işlemin değerleri 9, 7, 6, 4, 2, blok kapasitesi 3 olsun. Herkes dürüst teklif verirse en yüksek üç işlem girer ve üreticiye $9 + 7 + 6 = 22$ ödenir.

Ama birinci fiyat açık artırmasında dürüst teklif en iyi strateji değildir. Değeri 9 olan kullanıcı 9 teklif ederse fayda elde etmez; girmeye yetecek en küçük teklifi vermek ister. Bu yüzden herkes teklifini kırpar. Kırpma miktarı, ötekilerin ne teklif edeceğine dair tahmine bağlıdır. Sonuç olarak her kullanıcı "ne kadar vermeliyim?" sorusuyla uğraşır. Bu yorucudur ve tahmin yapamayan kullanıcılar ya fazla öder ya da dışarıda kalır.

Taban ücret + bahşiş tasarımı. Literatürde EIP-1559 (Ethereum iyileştirme önerisi, *Ethereum Improvement Proposal*) adıyla bilinen tasarımda protokol, her blok için bir **taban ücret** (*base fee*) r belirler. Taban ücreti ödemeye razı olan her işlem girer. Taban ücret **yakılır**: kimseye ödenmez ve dolaşımdan çıkar. Üreticiye yalnızca kullanıcının eklediği küçük bir **bahşiş** (*tip*) gider.

Örnek 14.7

Aynı beş işlem (değerler 9, 7, 6, 4, 2), kapasite 3, taban ücret $r = 5$, bahşiş 1.

1. Bir işlemin girmesi için değeri en az $r + \text{bahşiş} = 6$ olmalıdır. 9, 7 ve 6 bu eşiği geçer; 4 ve 2 geçmez.
2. 3 işlem girer; bu kapasiteyi aşmaz, açık artırmaya gerek kalmaz.
3. Yakılan: $3 \cdot 5 = 15$. Üreticiye giden bahşiş: $3 \cdot 1 = 3$.
4. Kullanıcıların net faydaları: $9 - 6 = 3$, $7 - 6 = 1$, $6 - 6 = 0$.

Kullanıcı tarafında hiçbir tahmin gerekmez: her kullanıcı yalnızca "değerim 6'dan büyük mü?" sorusunu cevaplar.



Şekil 14.13: Beş işlemin değerleri ve giriş eşiği *r + bahşiş = 6*. Eşiğin üstündeki üç işlem girer. Birinci fiyatta dürüst teklifle 22 ödenir; taban ücret tasarımında 15 yakılır, 3 bahşiş olarak üreticiye gider.

Şekil 14.13 iki tasarımı aynı veride karşılaştırır. Taban ücret tasarımının asıl üstünlüğü basitliktir: talep kapasiteyi aşmıyorsa en iyi strateji "taban ücret + küçük bahşiş" ödemektir ve fiyat tahmini gerekmez. Bu, ilan edilmiş bir fiyattan satın almaya benzer. Talep kapasiteyi aşınca ise tasarım yeniden bir açık artırmaya döner: kullanıcılar bahşiş üzerinden yarışır.

Roughgarden (2021), iyi bir işlem ücreti mekanizmasının hedeflerini biçimsel olarak tanımladı: kullanıcı için basit teklif, üretici için dürüst davranış ve danışıklılığa direnç. Bu hedeflerin hepsini tam olarak sağlayan bir kural bulmak zordur. Bu hedefler 14.4.4'te ayrıntılı ele alınır.

14.4.3 Taban ücret talebe göre ayarlanır

Taban ücret sabit değildir; talebi izler. Genel ilke basittir: blok hedeflenen doluluğun üstündeyse ücret artar, altındaysa düşer. Tasarımın ilk parametreleriyle güncelleme kuralı şöyledir:

$$r_{t+1} = r_t \cdot (1 + \tfrac{1}{8} \cdot (g_t - g_{hedef}) / g_{hedef})$$

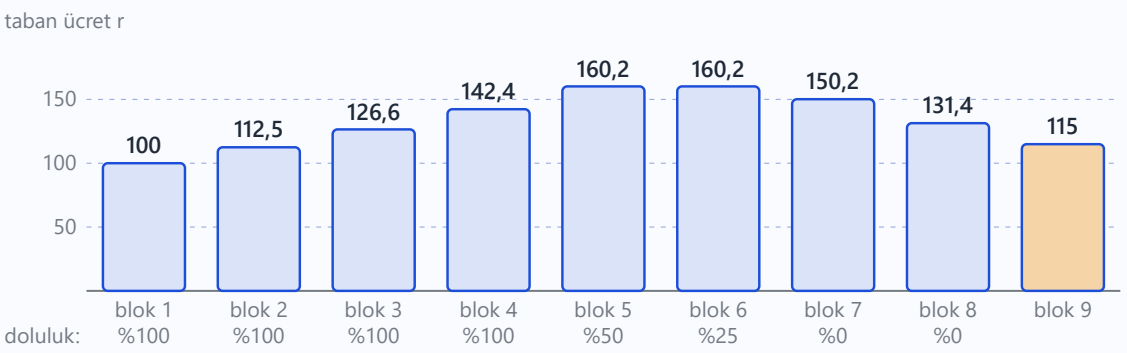
Burada g_t, t bloğunda kullanılan alan; g_{hedef} ise blok kapasitesinin yarısıdır. Formülün parçaları şöyle okunur. $(g_t - g_{hedef})/g_{hedef}$, doluluğun hedeften göreli sapmasıdır. Blok tam doluysa $g_t = 2g_{hedef}$ ve sapma +1'dir. Blok boşsa sapma −1'dir. Hedefteyse sapma 0'dır. Sapma ⅛ ile çarpıldığı için ücret bir blokta en çok %12,5 artar ya da azalır.

Örnek 14.8

Başlangıçta $r = 100$. Sekiz bloğun dolulukları %100, %100, %100, %100, %50, %25, %0, %0'dır.

1. Blok 1 %100 dolu: sapma +1, değişim +%12,5. $r = 100 \cdot 1{,}125 = 112{,}5$.
2. Blok 2 %100 dolu: $r = 112{,}5 \cdot 1{,}125 \approx 126{,}6$.
3. Blok 3 %100 dolu: $r \approx 126{,}6 \cdot 1{,}125 \approx 142{,}4$.
4. Blok 4 %100 dolu: $r \approx 160{,}2$. Talep patlamasında ücret her blokta en çok %12,5 artar.
5. Blok 5 %50 dolu: sapma 0, ücret sabit kalır: $r \approx 160{,}2$.
6. Blok 6 %25 dolu: sapma $(0{,}25 - 0{,}5)/0{,}5 = -0{,}5$, değişim $-0{,}5/8 = -\%6{,}25$. $r \approx 160{,}2 \cdot 0{,}9375 \approx 150{,}2$.

7. Blok 7 %0 dolu: sapma −1, değişim −%12,5. $r \approx 150{,}2 \cdot 0{,}875 \approx 131{,}4$.
8. Blok 8 %0 dolu: $r \approx 131{,}4 \cdot 0{,}875 \approx 115$. Boş bloklar ücreti düşürür; talep azalınca fiyat geri gelir.



Şekil 14.14: Her bloğun başındaki taban ücret ve o bloğun doluluğu. Dört dolu blok ücreti 100'den 160,2'ye çıkarır; hedefteki blok sabit bırakır; az dolu ve boş bloklar ücreti 115'e indirir.

Şekil 14.14 ücretin talebi nasıl izlediğini gösterir. Taban ücret, talebi izleyen bir **ilan edilmiş fiyat** gibi çalışır. Talep yükselince fiyat da yükselir ve bloklar yeniden hedef doluluğa iner. Ama ayarlama zaman alır: ani bir talep patlamasında ücret yalnızca %12,5'lik adımlarla yükselebilir. Bu arada bloklar doludur ve sistem geçici olarak yeniden bahşişe dayalı bir açık artırmaya döner.

Değişim **çarpımsaldır**. Dört dolu bloktan sonra ücret $100 \cdot 1{,}125^4 \approx 160{,}2$'dir, $100 + 4 \cdot 12{,}5 = 150$ değil. Ayrıca bir artışı aynı oranda bir azalış tam geri almaz: $1{,}125 \cdot 0{,}875 \approx 0{,}984$ olduğu için bir dolu ve bir boş blok ücreti %1,6 düşürür.

Sık yapılan hata

Taban ücreti toplamsal güncellemek. Değişim çarpımsaldır: iki dolu blok $100 \to 112{,}5 \to 126{,}6$ verir, 125 değil. İkinci bir yanlış, taban ücretin üreticiye gittiğini sanmaktır. Taban ücret yakılır; üreticiye yalnız bahşiş gider.

14.4.4 Neden yakılır?

Taban ücretin üreticiye verilmek yerine yakılması ilk bakışta israf gibi görünür. Bu tercihin nedeni, iyi bir ücret mekanizmasının üç özelliğiyle açıklanır.

Kullanıcı teşvik uyumu
: Talep kapasiteyi aşmıyorsa kullanıcı için en iyi teklif taban ücret ile küçük bir bahşiştir. Kullanıcı “teklifimi ne kadar kırpmalıyım?” diye düşünmez.

Üretici teşvik uyumu

Üretici kurala uyarak en çok kazanır. Sahte işlem eklemek ya da gerçek işlemleri saklamak ona kazandırmaz.

Danışıklılığa dayanıklılık

Üretici ve bazı kullanıcılar zincir dışında anlaşıp ortak kazançlarını artıramaz. Örneğin "bahşişi zincir dışında öde, taban ücreti düşük tutalım" gibi bir yan anlaşma kazandırmamalıdır.



Şekil 14.15: Gerçek talep blokları %50 dolduruyorsa ücret değişmez. Üretici 5 sahte işlem ekleyip bloğu %100 doldurursa sonraki taban ücret %12,5 artar.

Yakmanın gerekçesi Şekil 14.15'te görülür. Taban ücret üreticiye gitseydi, üreticinin bloğa kendi sahte işlemlerini eklemesi bedava olurdu: ödediği ücret yine kendisine dönerdi. Bu sahte işlemler bloğu doldurur ve taban ücretin bir sonraki blokta yükselmesine yol açar. Üretici böylece ücreti istediği gibi şişirebilirdi. Ücret yakılınca her sahte işlem r kadar gerçek bir kayıptır: 5 sahte işlem $5r$ yakar. Sahte talep yaratmak artık pahalıdır.

Yakmak bütün sorunları çözmez. Blok doluyken kullanıcılar yine bahşiş üzerinden yarışır ve birinci fiyat davranışı geri gelir. Bu, kuramsal bir sınırın yansımasıdır. Daha sonraki çalışmalar, blok alanı kıtken Roughgarden'ın (2021) biçimselleştirdiği üç özelliğin aynı anda tam olarak sağlanamayacağını gösterdi (Chung ve Shi, 2023; Chung, Roughgarden ve Shi, 2024). İki çalışma, danışıklılığa dayanıklılığın farklı tanımlarını kullanır; sonuç iki tanımda da bir imkânsızlıktır. Yani her işlem ücreti mekanizması bu üç hedef arasında bir ödünleşim yapmak zorundadır.

14.4.5 Otomatik piyasa yapıcı: sabit çarpım kuralı

Klasik bir borsada alım satım, alıcı ve satıcı emirlerinin eşleşmesiyle olur. Merkeziyetsiz borsaların çoğunda ise karşı taraf bir insan değil bir **havuzdur**. Havuzda iki varlık bulunur ve fiyat bir formülle belirlenir.

Tanım 14.11

Sabit çarpımlı piyasa yapıcı: Havuzda x birim A ve y birim B vardır. Her işlemden sonra $x \cdot y = k$ çarpımı korunur. Δx birim A satan kullanıcı, çarpımı koruyan miktarda B alır:

$$(x + \Delta x) \cdot (y - \Delta y) = x \cdot y \Longrightarrow \Delta y = y \cdot \Delta x \,/\, (x + \Delta x)$$

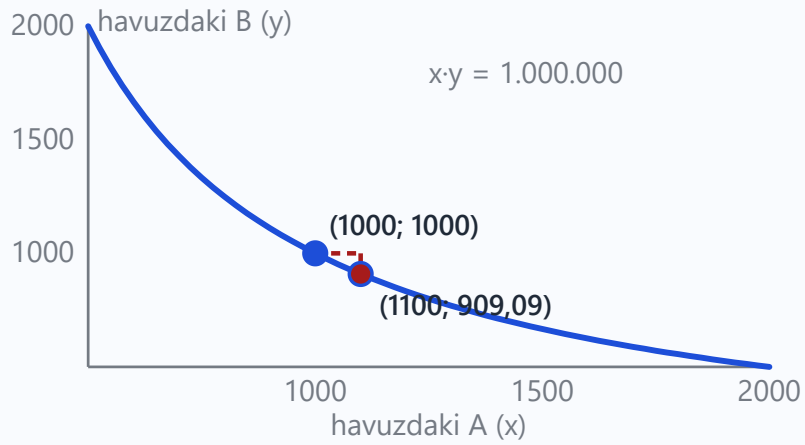
Formül şöyle türetilir. İşlemden sonra havuzda $x + \Delta x$ A ve $y - \Delta y$ B vardır ve çarpım değişmez. İki taraf $x + \Delta x$'e bölünür: $y - \Delta y = xy/(x + \Delta x)$. Buradan $\Delta y = y - xy/(x + \Delta x) = y(x + \Delta x - x)/(x + \Delta x) = y\Delta x/(x + \Delta x)$.

Havuzun anlık fiyatı y/x'tir (bir A kaç B eder). Havuz (1000; 1000) ise başlangıç fiyatı 1 A = 1 B'dir. Ama büyük bir işlem bu fiyattan yapılamaz; çünkü işlem sürdükçe havuzdaki B azalır ve B pahalanır. Kullanıcının başlangıç fiyatına göre eksik aldığı orana **kayma** (*slippage*) denir.

Δx (A)	Hesap	Alınan B	Kayma
10	*1000 · 10 / 1010*	9,90	%1,0
100	*1000 · 100 / 1100*	90,91	%9,1
250	*1000 · 250 / 1250*	200,00	%20,0
500	*1000 · 500 / 1500*	333,33	%33,3

Tablo 14.12: Havuz (1000; 1000) için satılan A miktarına göre alınan B ve kayma.

Tablodan genel bir kural çıkar: kayma, $\Delta x/(x + \Delta x)$'e eşittir; yani işlemin havuza göre büyüklüğüne bağlıdır. 10 A için 10 B yerine 9,90 B alınır, kayma %1'dir. Havuzun yarısı kadar (500 A) satan kullanıcı ise üçte bir kayıpla işlem yapar. Gerçek havuzlar ayrıca küçük bir işlem ücreti alır. %0,3 ücretle 100 A'nın yalnızca *99,7*'si havuza işler: *1000 · 99,7 / 1099,7 ≈ 90,66* B alınır. Ücret havuzda kalır ve havuza varlık koyan **likidite sağlayıcılarına** gider.



Şekil 14.16: *x · y = 1.000.000* eğrisi. 100 A satılınca havuz (1000; 1000) noktasından (1100; 909,09) noktasına kayar; eğri boyunca ilerledikçe B pahalanır.

Şekil 14.16 işlemin havuzu eğri boyunca nasıl hareket ettirdiğini gösterir. 100 A'lık işlemden sonra yeni fiyat *1100 / 909,09 ≈ 1,21* A'dır: 1 B artık 1,21 A eder. Büyük işlem fiyatı kendi aleyhine kaydırır; buna **fiyat etkisi** denir. Bu yapının üç sonucu vardır:

- Havuz fiyatı ile dış piyasa fiyatı arasında fark açılınca, ucuzdan alıp pahalıdan satan **arbitrajcılar** havuzu yeniden dengeler. Havuz fiyatı böylece dış fiyatı izler.

- Fiyat etkisi formülle, önceden ve kesin olarak hesaplanabilir. Bekleyen bir işlemin havuzu nereye götüreceği herkesçe bilinir.
- Bu yüzden bekleyen büyük bir işlemin **önüne geçmek** kârlıdır (14.4.6).

14.4.6 MEV ve sandviç saldırısı

Bekleyen işlemlerin görünür olması ile fiyat etkisinin önceden hesaplanabilmesi birleşince, işlemlerin sırasını belirleyebilen biri kâr edebilir. En bilinen yol **sandviç saldırısıdır**. Örnek, sabit çarpımlı havuz (1000; 1000) ile kurulur; $x \cdot y = 1.000.000$. Bir kullanıcı 100 A satmak istiyor. Tek başına işlem yapsa 90,91 B alırdı.

Örnek 14.9

1. **Görme.** Kullanıcının işlemi bloğa girmeden önce bekleme havuzunda herkese görünür.
2. **Öne geçme** (*front-running*). Sırayı belirleyebilen saldırgan kendi işlemini kullanıcınınkinin önüne koyar ve 50 A satar. Aldığı B: $1000 \cdot 50 / 1050 \approx 47{,}62$. Havuz (1050; 952,38) olur. Saldırgan B'yi ucuzken almış ve fiyatı yukarı itmiştir.
3. **Kullanıcının işlemi.** Kullanıcı 100 A satar: $952{,}38 \cdot 100 / 1150 \approx 82{,}82$ B alır. Havuz (1150; 869,57) olur. Kullanıcı tek başına 90,91 B alacaktı; $90{,}91 - 82{,}82 = 8{,}09$ B kaybetmiştir. Kullanıcının işlemi B'nin fiyatını daha da yukarı iter.
4. **Arkadan gelme** (*back-running*). Saldırgan elindeki 47,62 B'yi havuza geri satar. Havuzda artık B pahalı, A boldur: $1150 \cdot 47{,}62 / (869{,}57 + 47{,}62) \approx 59{,}71$ A alır. Havuz (1090,29; 917,18) olur.
5. **Sonuç.** Saldırgan 50 A ile başladı, 59,71 A ile bitirdi: kâr 9,71 A (işlem ücretleri hariç). Kullanıcının kaybı ise 8,09 B'dir.

Sıra	İşlem	Havuz (A, B)	Sonuç
0	Başlangıç	(1000; 1000)	
1	Saldırgan 50 A satar (öne geçme)	(1050; 952,38)	47,62 B alır
2	Kullanıcı 100 A satar	(1150; 869,57)	82,82 B alır (tek başına 90,91)
3	Saldırgan B'leri geri satar (arkadan gelme)	(1090,29; 917,18)	59,71 A alır

Tablo 14.13: Sandviç saldırısının adımları, havuz $x \cdot y = 1.000.000$.

Saldırının iki yarısı bir “sandviç” oluşturur: kullanıcının işlemi, saldırganın iki işleminin arasında kalır. Saldırganın kârı, kullanıcıya kötü fiyattan işlem yaptırmaktan gelir. Saldırgan B'nin fiyatını kullanıcıdan önce yukarı iter, sonra kullanıcının daha da yükselttiği fiyattan satar.

Tanım 14.12

MEV (maksimum elde edilebilir değer, *maximal extractable value*): Bir bloktaki işlemleri sıralama, bloğa ekleme ve bloktan çıkarma gücünden elde edilebilecek değerdir (Daian vd., 2020).

Sandviç saldırısı MEV'nin bir örneğidir. Arbitraj (havuzlar arası fiyat farkını ilk kapatan olmak) ve tasfiye (teminatı yetersiz kalan bir borcu ilk kapatan olmak) gibi başka MEV kaynakları da vardır. Ortak noktaları, sıranın değer taşımasıdır. Mekanizma tasarımı açısından temel soru şudur: **sıralama gücü kimde ve bu gücün bedelini kim ödüyor?** Sandviç örneğinde bedeli kullanıcı öder.

14.4.7 Kod izleme: sandvic()

Kullanıcının elinde basit bir koruma vardır: **kayma toleransı**. Kullanıcı, en az ne kadar B alacağını bir sınır olarak belirler. Fiyat bu sınırın altına inerse işlem iptal edilir. Aşağıdaki kod, sandviç saldırısını bu sınırla birlikte uygular.

```
cikti = lambda x, y, dx: y - x * y / (x + dx)  # xy=k
def sandvic(x, y, a, u, min_out):
    b1 = cikti(x, y, a)          # öne geç: a A sat
    x, y = x + a, y - b1
    b2 = cikti(x, y, u)          # kullanıcı: u A sat
    if b2 < min_out:
        return None              # işlem iptal
    x, y = x + u, y - b2
    a2 = cikti(y, x, b1)         # arkadan: b1 B sat
    return round(a2 - a, 2)
for a in [20, 50, 80]:
    print(a, sandvic(1000, 1000, a, 100, 82))
```

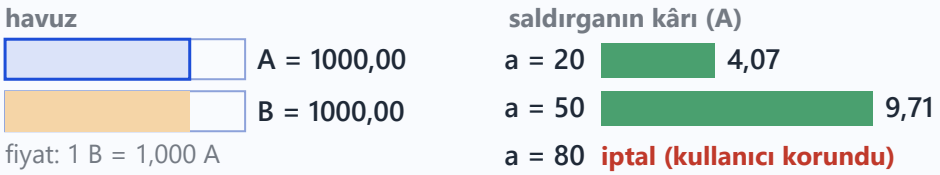
Kod satır satır şöyle çalışır:

- `cikti(x, y, dx)`, havuza `dx` birim ilk varlık satınca alınan ikinci varlık miktarıdır: $y - xy/(x + dx)$. Bu, Tanım 14.11'deki formülün aynısıdır.
- `b1` : saldırgan `a` A satıp `b1` B alır. Havuz güncellenir.
- `b2` : kullanıcı `u` A satıp `b2` B alır. `b2` , kullanıcının sınırı `min_out` 'tan küçükse işlem iptal edilir ve fonksiyon `None` döndürür.
- `a2` : saldırgan `b1` B'yi geri satar. Bu kez havuza B girer ve A çıkar; bu yüzden fonksiyon argümanların yeri değiştirilerek çağrılır: `cikti(y, x, b1)` .
- Fonksiyon saldırganın kârını, yani `a2 - a` 'yı döndürür.

Kullanıcı 100 A satacak ve en az 82 B istiyor. Saldırgan öne geçme miktarı a için üç değer dener.

a	b1 (saldırgan B)	b2 (kullanıcı B)	b2 ≥ 82?	a2 (geri A)	Kâr
20	19,61	87,54	evet	24,07	4,07
50	47,62	82,82	evet	59,71	9,71
80	74,07	78,47	hayır	—	None (iptal)

Tablo 14.14: `sandvic(1000, 1000, a, 100, 82)` izlemesi.



Şekil 14.17: Öne geçme miktarına göre saldırganın kârı. *a = 20*'de 4,07 A, *a = 50*'de 9,71 A; *a = 80*'de kullanıcının işlemi iptal olur ve kullanıcı korunur.

Şekil 14.17 sonucu özetler. Daha büyük bir öne geçme, fiyatı daha çok iter ve daha çok kâr getirir. Ama kullanıcının kayma sınırı bunu keser: *a = 80*'de kullanıcı yalnızca 78,47 B alacağı için işlem iptal olur ve saldırgan hiçbir şey kazanamaz. Bu yüzden saldırganın en iyi hamlesi, kullanıcının sınırını tam zorlayan öne geçme miktarıdır. Sıkı bir tolerans, saldırganın alanını daraltır ve MEV'ye karşı basit bir korumadır.

Sık yapılan hata

Gevşek kayma toleransını güvenli sanmak. Gevşek tolerans işlemin geçme şansını artırır, ama saldırgana daha büyük bir öne geçme alanı bırakır. Kullanıcının sınırı 75 B olsaydı *a = 80* de geçerdi ve saldırgan daha çok kazanırdı. Öte yandan çok sıkı bir tolerans, normal fiyat oynamalarında bile işlemin iptal edilmesine yol açar. Tolerans bir ödünleşimdir.

14.4.8 MEV'ye karşı önlemler

MEV'ye karşı önerilen yöntemler beş fikirde toplanabilir. Her birinin bir bedeli vardır.

Yöntem	Fikir	Bedeli
Sıkı kayma toleransı	En az alınacak miktar belirlenir; büyük öne geçme işlemi iptal ettirir.	İşlem daha sık geri çevrilir.
Taahhüt–açıklama	Önce işlemin özeti (*hash*) yayımlanır; içerik, sıra kesinleşince açılır.	İki adım ve gecikme; açılmayan işlemler.
Şifreli bekleme havuzu	İşlem, sırası belirlendikten sonra çözülür.	Anahtarı tutanlara güven gerekir.
Toplu açık artırma	Bir zaman dilimindeki bütün emirler aynı fiyattan eşleşir; sıra önemsizleşir.	Dilim kadar bekleme.

Yöntem	Fikir	Bedeli
Sıralama hakkını açık artırmak	Blok kurma hakkı açıkça satılır; MEV geliri gizli değil, rekabetçi biçimde dağılır.	MEV yok olmaz; sıralama az sayıda büyük blok kurucusunda toplanır.

Tablo 14.15: MEV'ye karşı önlemler.

İlk üç yöntem saldırganın ya bilgisini ya da alanını kısıtlar. Taahhüt–açıklama ve şifreli havuz, işlemin içeriğini sıra belirlenene kadar gizler; saldırgan neyin önüne geçeceğini bilemez. Bu fikir, kapalı zarf açık artırmasının mantığına benzer: teklifler, herkes teklifini verdikten sonra açılır.

Toplu açık artırma sandviçi tamamen etkisiz kılar. Bir zaman dilimindeki bütün alış ve satış emirleri tek bir fiyattan eşleşir. Saldırganın "önden alış" ve "arkadan satış" emirleri kullanıcınınkiyle aynı partide, aynı fiyattan gerçekleşir. Alış ile satış fiyatı aynı olduğu için kâr 0'dır. Sıra artık değer taşımaz.

Son yöntem farklı bir yaklaşımdır: MEV'yi ortadan kaldırmaya çalışmak yerine onu açıkça fiyatlar. Sıralama hakkı bir açık artırmayla satılınca, MEV'den elde edilecek değer rekabet yoluyla bu hakkın fiyatına yansır. Ama bu yöntem gücün az sayıda büyük aktörde toplanmasına yol açabilir.

Bu önlemlerin uygulamadaki biçimleri sık değişir. Kalıcı fikir şudur: **sıralama gücü değer taşır** ve bu değer ya azaltılmalı (gizleme, toplu eşleştirme) ya da açıkça fiyatlanmalıdır (açık artırma). Bu, mekanizma tasarımının genel bir dersidir: bir kaynak (burada sıra) değerliyse ve bedelsiz dağıtılıyorsa, onu ele geçirmek için gizli bir rekabet doğar.

14.5 Ölçek ve açık sorular

Önceki kısımlardaki örnekler birkaç oyuncu ve birkaç seçenekle çalıştı. Gerçek uygulamalarda oyuncu, nesne ve strateji sayıları çok büyüktür. Bu kısım ölçeğin getirdiği hesaplama sorunlarını, bunlara karşı geliştirilen yaklaşık yöntemleri ve alanın açık sorularını ele alır.

14.5.1 Hesaplama ve ölçek

Oyun kuramının kavramları hesaplanabilir olmalıdır. Bir dengenin var olduğunu bilmek, onu bulabileceğimiz anlamına gelmez. Bu konuda dört gözlem öne çıkar.

- **Denge bulmak zor olabilir.** Genel oyunlarda Nash dengesi bulmak PPAD-tam bir problemdir. Yani büyük oyunlar için verimli (polinom zamanlı) bir algoritma beklenmez. Pratikte kesin denge yerine **yaklaşık denge** ve **öğrenme dinamikleri** kullanılır: oyuncular tekrar tekrar oynayıp pişmanlıklarını azaltacak biçimde stratejilerini günceller.
- **Öğrenme ve arama birlikte güçlüdür.** Çok oyunculu pokerde insan üstü sonuçlar, pişmanlığı en aza indiren algoritmalar ile aramanın birleştirilmesiyle elde edildi (Brown ve Sandholm, 2019). Poker eksik bilgili, çok oyunculu ve devasa bir oyundur; kesin denge hesaplanamaz ama iyi bir yaklaşık strateji bulunabilir.

- **Kombinatoryal açık artırma zordur.** Birden çok nesnenin paketler hâlinde satıldığı açık artırmalarda kazananı belirlemek NP-zordur. Bu yüzden teklif dili (teklifçilerin hangi paketlere nasıl değer bildirebileceği) dikkatle tasarlanır ve yaklaşık algoritmalar kullanılır.
- **Kuram dışı maliyetler de önemlidir.** Kuramsal bir garanti kadar gecikme, veri erişimi ve enerji maliyeti de bir mekanizmanın uygulanabilirliğini belirler. Bir reklam açık artırması milisaniyeler içinde bitmelidir; daha iyi ama yavaş bir kural işe yaramaz.

Ölçek sorununun kaynağı, olasılıkların üstel büyümesidir. m nesnelik bir açık artırmada her nesne bir pakette ya vardır ya yoktur. Bu yüzden alt küme sayısı 2^m'dir; boş paket çıkarılınca $2^m - 1$ paket kalır.

Nesne m	Paket sayısı $2^m - 1$
5	31
10	1.023
20	1.048.575
30	1.073.741.823

Tablo 14.16: Nesne sayısına göre boş olmayan paket sayısı.

Her teklifçi her pakete bir değer bildirirse, girdi nesne sayısıyla üstel büyür. 30 nesnede bir teklifçinin bildireceği değer sayısı bir milyarı aşar. Aynı patlama oyunlarda da görülür: iki eylemli n oyunculu bir oyunda strateji profili sayısı 2^n'dir; $n = 20$ için 1.048.576 profil vardır. Bir kazanç matrisini tablo olarak yazmak bile mümkün olmayabilir. Bu yüzden büyük oyunlar, tıkanıklık oyunlarında olduğu gibi, kısa bir yapıyla (örneğin yol başına bir gecikme fonksiyonuyla) tanımlanır.

14.5.2 Kombinatoryal açık artırmada kazanan belirleme

Kombinatoryal açık artırmada teklifçiler tek tek nesnelere değil, nesne paketlerine teklif verir. Örneğin bir telekomünikasyon firması iki komşu bölgenin frekans lisansını birlikte istiyor olabilir; tek başına bir bölge ona az değer ifade eder. Açık artırmacı hangi tekliflerin kazanacağına karar vermelidir.

Tanım 14.13

Kazanan belirleme problemi (*winner determination problem*): Her teklif bir paket ve bir değerden oluşur: (S_i, v_i). Problem, ortak nesnesi olmayan (birbiriyle çakışmayan) teklifler arasından toplam değeri maksimize eden kümeyi seçmektir.

Bu problem, bilgisayar bilimindeki **ağırlıklı küme paketleme** (*weighted set packing*) problemiyle aynıdır. NP-zordur ve genel durumda iyi yaklaştırılamaz: verimli bir algoritmanın her girdide optimuma belirli bir oranda yakın kalacağı garanti edilemez. Kaba kuvvet yöntemi n teklif için 2^n alt kümeyi dener. Pratikte bu aramayı akıllıca budayan (umut vermeyen dalları erken kesen) tam sayılı programlama çözücüleri kullanılır.

Kazanan belirlemenin zorluğu VCG mekanizmasını da etkiler. VCG'de her kazanan, varlığıyla ötekilere verdiği zararı öder. Bunu hesaplamak için problem bir kez bütün tekliflerle, sonra her teklifçi bir kez çıkarılarak çözülür; toplam *n + 1* çözüm gerekir. Problem yaklaşık çözülürse VCG'nin dürüstlük garantisi bozulabilir. Çünkü bir teklifçi, yalan söyleyerek yaklaşık algoritmanın hatasından yararlanabilir ve kendi lehine daha iyi bir dağıtım elde edebilir.

Örnek 14.10

Üç nesne (a, b, c) ve dört teklif vardır.

Teklif	Paket	Değer
t1	{a, b, c}	10
t2	{a}	4
t3	{b}	4
t4	{c}	3

Tablo 14.17: Dört paket teklifi.

1. t1 bütün nesneleri içerdiği için diğer üç teklifle çakışır. t1 seçilirse başka teklif seçilemez: toplam 10.
2. t2, t3 ve t4 birbirleriyle çakışmaz. Hepsi birlikte seçilebilir: *4 + 4 + 3 = 11*.
3. Ara seçimler daha azdır: örneğin {t2, t3} = 8.
4. Optimum {t2, t3, t4} = 11'dir. Tek bir büyük teklif, küçük tekliflerin toplamından az değerlidir.

14.5.3 Kod izleme: acgozlu()

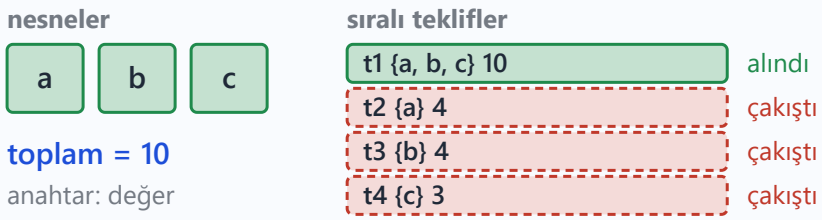
NP-zor problemlerde sık kullanılan bir yaklaşım **açgözlü** (*greedy*) seçimdir: teklifler bir anahtara göre sıralanır ve sırayla, öncekilerle çakışmayan her teklif alınır. Sonucu belirleyen, hangi anahtarın kullanıldığıdır.

```
def acgozlu(teklifler, anahtar):
    sirali = sorted(teklifler, key=anahtar, reverse=True)
    toplam, kullanilan = 0, set()
    for S, v in sirali:
        if not (S & kullanilan):  # çakışma yok
            toplam += v
            kullanilan |= S
    return toplam
T = [({'a','b','c'}, 10), ({'a'}, 4), ({'b'}, 4), ({'c'}, 3)]
print(acgozlu(T, lambda t: t[1]))
print(acgozlu(T, lambda t: t[1] / len(t[0])))
```

Kod satır satır şöyle çalışır. `sirali`, teklifleri verilen anahtara göre büyükten küçüğe sıralar. `toplam` seçilenlerin değerini, `kullanilan` seçilen paketlerdeki nesneleri tutar. Döngü her teklif için paketin kullanılmış nesnelerle kesişimine bakar (`S & kullanilan`). Kesişim boşsa teklif alınır: değeri toplama eklenir, nesneleri kullanılmış kümeye katılır (`|=`). Kod iki anahtarla çalıştırılır: önce teklifin değeri, sonra nesne başına değer.

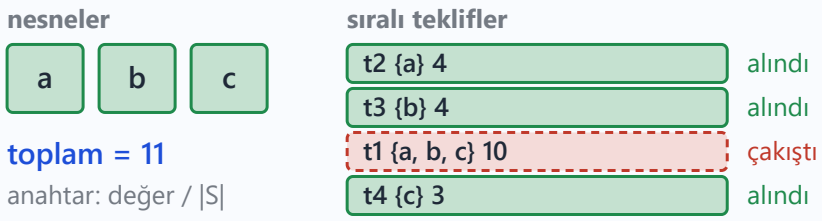
Adım	Değere göre	Toplam	Nesne başına değere göre	Toplam
1	t1 {a, b, c} (10): alınır	10	t2 {a} (4): alınır	4
2	t2 {a} (4): a dolu, atlanır	10	t3 {b} (4): alınır	8
3	t3 {b} (4): b dolu, atlanır	10	t1 {a, b, c} (10/3 ≈ 3,33): a ve b dolu, atlanır	8
4	t4 {c} (3): c dolu, atlanır	10	t4 {c} (3): alınır	11

Tablo 14.18: `acgozlu()` izlemesi; sol: değere göre, sağ: nesne başına değere göre.



Şekil 14.18: Değere göre açgözlü seçim. t1 ilk sırada alınır ve üç nesneyi de doldurur; diğer üç teklif çakışır. Toplam 10.

Şekil 14.18 ilk anahtarın sonucunu gösterir. Değere göre sıralama, büyük paketi erken alır ve küçük tekliflerin hepsini dışarıda bırakır. Sonuç 10'dur; optimum ise 11'dir. Değere göre açgözlü seçim büyük paketlere aldanır, çünkü bir paketin değerini kullandığı kaynakla karşılaştırmaz.



Şekil 14.19: Nesne başına değere göre açgözlü seçim. t2 ve t3 önce alınır; t1 atlanır; t4 boştaki c'yi alır. Toplam 11 = optimum.

Şekil 14.19'da ikinci anahtar kullanılır. Nesne başına değerler t2 ve t3 için 4, t1 için *10/3 ≈ 3,33*, t4 için 3'tür. t2 ile t3 eşittir; Python'un `sorted` fonksiyonu eşitlikte ilk sırayı korur. Bu sıralama, değeri kaynağa göre ölçtüğü için optimumu bulur.

Ama bu anahtar da yanılabilir. ({a, b}, 10) ve ({a}, 6) tekliflerini düşünelim. Nesne başına değerler 5 ve 6'dır; açgözlü seçim önce 6'yı alır, sonra 10'u çakıştığı için atlar ve 6'da kalır. Oysa 10'luk teklif tek başına daha iyidir. Her iki anahtarın da kötü örnekleri vardır. Lehmann, O'Callaghan ve Shoham (2002), değeri paket büyüklüğünün kareköküne bölen bir anahtar önerdi: $v / \sqrt{|S|}$. Bu anahtarla açgözlü seçim, en kötü durumda bile optimumun en az $1/\sqrt{m}$'i kadar değer garanti eder (m nesne sayısı). Ayrıca bu açgözlü algoritma, uygun bir ödeme kuralıyla birleşince, her teklifçinin yalnızca tek bir pakete değer verdiği (*single-minded*) durumda teşvik uyumludur.

Sık yapılan hata

Bir örnekte optimumu bulan açgözlü kuralı her zaman doğru sanmak. Nesne başına değer, Tablo 14.17'deki örnekte optimumu bulur ama ({a, b}, 10) ile ({a}, 6) örneğinde bulamaz. NP-zor bir problemde basit bir kural, ancak bir yaklaşıklık garantisiyle birlikte anlamlıdır.

14.5.4 Diğer araştırma yönleri

Bölümün ana konuları dışında dört araştırma yönü daha öne çıkar.

Sosyal ağlarda etki maksimizasyonu. Bir firma k kişiye ücretsiz ürün verecek ve bu kişilerin ürünü arkadaşlarına yaymasını umuyor. Yayılmayı en büyük yapacak k "tohum" kişi kimdir (Kempe, Kleinberg ve Tardos, 2003)? Yayılma bir modelle tanımlanır.

Tanım 14.14

Bağımsız çağlayan modeli (*independent cascade model*): Başlangıçta tohum kişiler etkilenmiştir. Yeni etkilenen her kişi, her komşusunu yalnızca bir kez, p olasılıkla etkilemeye çalışır. Etkilenen komşular bir sonraki adımda aynı şeyi yapar. Kimse yeni etkilenmeyince süreç durur.

En iyi tohum kümesini bulmak NP-zordur. Ama problemin bir yapısı vardır: beklenen yayılım **alt modülerdir** (*submodular*), yani azalan marjinal getirilidir. Bir kişiyi tohum yapmanın ek katkısı, tohum kümesi büyüdükçe azalır; çünkü onun ulaşacağı kişilere başka tohumlar zaten ulaşmış olabilir. Bu yapı sayesinde basit bir açgözlü algoritma iyi çalışır: her adımda beklenen yayılıma en çok katkı yapan kişi tohum kümesine eklenir. Bu algoritma, en iyi yayılımın en az $1 - 1/e \approx \%63$'ünü garanti eder. Kazanan belirleme probleminin aksine burada yapı, açgözlü seçime güçlü bir garanti verir.

Çok ajanlı pekiştirmeli öğrenme
: Her ajanın ortamı, öğrenen diğer ajanları da içerir. Bir ajan öğrenirken ötekiler de değiştiği için ortam durağan değildir. Tek ajanlı öğrenmedeki yakınsama garantileri, ortamın sabit olduğunu varsayar; bu yüzden burada geçerli değildir. Algoritmik danışıklılık bu sorunun bir örneğidir.

Sürdürülebilir sistemler

Karbon emisyon izni piyasaları ve elektrikte talep yanıtı gibi mekanizmalar, dışsallığı fiyatlar. Bir oyuncunun başkalarına yüklediği maliyet ona bir vergiyle ödetilir (Pigou vergisi). Bu, tıkanıklık ücretlendirmesindeki fikrin aynısıdır.

Dayanıklılık
: Saldırgan, hatalı ya da sınırlı rasyonel oyunculara rağmen iyi çalışan mekanizmalar. Klasik kuram herkesin rasyonel olduğunu varsayar; gerçek sistemlerde bazı katılımcılar bu varsayıma uymaz.

Bunlara ek olarak şu yönler de etkindir: **bilgi tasarımı** ya da Bayesçi ikna (bilgiyi açıklama kuralını seçerek alıcının kararını yönlendirmek); yapay zekâ hizalamasında insan tercihlerini toplamak (birçok kişinin tercihlerinden tek bir hedef üretmek bir sosyal seçim problemidir); **tahmin piyasaları** ve doğrulanamayan raporları başkalarının raporlarıyla karşılaştırarak ödüllendiren **eşler arası tahmin** yöntemleri.

14.5.5 Açık araştırma soruları

Alanın bugün cevabı aranan soruları altı başlıkta toplanabilir. Bu sorular belirli bir teknolojiye değil, bölüm boyunca görülen kalıcı gerilimlere dayanır.

İnsan + yapay zekâ pazarları
: İnsanlar ve ajanlar aynı pazarda etkileştiğinde kurallar nasıl değişmeli? Ajanlar hızlı ve yorulmazdır; insanlar yavaş ve sınırlı dikkatlidir. Aynı kural iki gruba farklı etki edebilir.

Ödünleşimleri ölçmek
: Verimlilik, mahremiyet ve adalet arasındaki ödünleşimler nasıl ölçülmeli? Üstel mekanizmada ε küçüldükçe mahremiyet artar ama gelir düşer; bu tür ödünleşimlerin ortak bir dili yoktur.

Uyarlanabilir dayanıklılık
: Stratejiye dayanıklı mekanizmalar, değişen davranışlara nasıl uyarlanır? Bir mekanizma öğrenerek uyarlanırsa, öğrenme sürecinin kendisi manipüle edilebilir.

Kuram ve veri
: Modellerin varsayımları gerçek platform verileriyle nasıl sınanır? Kuram belirli değer dağılımları ve rasyonellik varsayar; veriler bu varsayımları doğrulamayabilir.

Öğrenen ajanların dengesi
: Birbirine karşı öğrenen algoritmalar hangi sonuca yakınsar ve danışıklılık nasıl önlenir?

Doğrulanabilir mekanizma
: Öğrenilmiş bir kuralın teşvik uyumlu olduğu nasıl kanıtlanır ya da denetlenir? Elle tasarlanan kurallar için ispat vardır; sinir ağı kuralları için yalnızca ölçülmüş pişmanlık vardır.

14.6 Değerlendirme: sık hatalar ve örnek problemler

Bu kısım bölümün sayısal sonuçlarını bir tabloda toplar, en sık hataları bir arada listeler ve sunumdaki örnek problemleri çözümleriyle verir.

Konu	Açıklama	Bu bölümdeki sayı
Ajanlar arası pazarlık	Geriye doğru çıkarım; sabır ve son söz güçtür	$T = 4$: A 18,1; sonsuz ufuk 52,6
Algoritmik danışıklılık	Tetik stratejisi, δ yeterince büyükse işbirliğini sürdürür	$\delta \geq 1/2$
Veriden fiyat	Deneysel gelir maksimizasyonu; az örnekte aşırı uyum	$p = 0{,}47$: deneysel 0,294, gerçek 0,249
Üstel mekanizma	Olasılık $\propto \exp(\varepsilon \cdot \text{gelir} / 2\Delta)$; mahremiyet + yaklaşık IC	$\varepsilon = 2$: oran $\leq e^2 \approx 7{,}39$ (örnekte 1,44)
Taban ücret	Dolu blokta +%12,5, boş blokta −%12,5; ücret yakılır	100 → 160,2 → 115
MEV	Sıralama gücü değer taşır; sandviç saldırısı	Kullanıcı −8,09 B, saldırgan +9,71 A
Dersin haritası	Denge → tasarım → uygulama → öğrenme → gelecek	Hazırlıktan gelecek yönelimlere tek kavram haritası

Tablo 14.19: Bölümün konuları ve temel sayıları.

Sık yapılan hata

- **Firma sayısını unutmak:** İki firmada yeten sabır kalabalık piyasada yetmez: eşik $\delta \geq 1 - 1/n$, $n = 5$ iken 0,80.
- **Sonlu ve sonsuz ufku karıştırmak:** Sonlu ufukta son öneren güçlüdür ($T = 4$'te A 18,1); sonsuz ufukta ilk öneren biraz öndedir (52,6).
- **Deneysel geliri gerçek sanmak:** Aynı veride seçilip ölçülen fiyatın başarısı abartılıdır; ayrı doğrulama verisi gerekir.
- **Üstel mekanizmada doğrusal orantı:** Olasılık gelirle değil $\exp(\varepsilon \cdot R / 2\Delta)$ ile orantılıdır; gelir farkı olasılık oranına üstel yansır.
- **Taban ücreti toplamsal güncellemek:** Değişim çarpımsaldır: iki dolu blok 100 → 112,5 → 126,6 (125 değil). Taban ücret üreticiye değil, yakılmaya gider.
- **Gevşek kayma toleransını güvenli sanmak:** İşlemin geçme şansını artırır, ama saldırgana daha büyük öne geçme alanı bırakır.

Aşağıdaki altı örnek problem, bölümün hesaplarını kısa biçimde tekrar eder.

Örnek 14.11

1. **İki turluk pazarlık.** 100 TL, A sonra B önerir, $\delta = 0{,}9$. Tur 2 son turdur ve B orada 100 alır. Tur 1'de A, B'ye bekleme değeri kadar önerir: $0{,}9 \cdot 100 = 90$. Paylaşım A 10, B 90. Sonsuz ufukta A'nın payı $100/1{,}9 \approx 52{,}6$.

2. **Üç firmalı tetik.** Tekel kârı 100 eşit paylaşılıyor; sapan bir dönem 100 alıyor, sonra 0. Koşul $(100/3)/(1 - \delta) \geq 100 \iff 1 - \delta \leq 1/3 \iff \delta \geq 2/3$. Oyuncu sayısı arttıkça danışıklılık zorlaşır.

3. **Deneysel fiyat.** Örnek değerler 0,9; 0,6; 0,5; 0,2. Gelirler $0{,}9 \cdot 1/4 = 0{,}225$, $0{,}6 \cdot 2/4 = 0{,}3$, $0{,}5 \cdot 3/4 = 0{,}375$, $0{,}2 \cdot 4/4 = 0{,}2$. En iyisi $p = 0{,}5$, gelir 0,375.
4. **Taban ücret.** $r = 100$, üç dolu ve bir boş blok: $100 \cdot 1{,}125^3 \cdot 0{,}875 \approx 124{,}58$.
5. **Harita sorusu.** Çevrim İçi Platformlar düğümündeki GSP–VCG karşılaştırmasında VCG, Mekanizma Tasarımı konusundan gelir. GSP'nin dayandığı ikinci fiyat fikri ise Açık Artırmalar konusundaki ikinci fiyat (Vickrey) açık artırmasından gelir.
6. **Üstel mekanizma.** İki aday fiyatın gelirleri 2 ve 1, $\varepsilon = 2$, $\Delta = 1$. Ağırlıklar e^2 ve e^1'dir. Olasılıklar $e^2/(e^2 + e) = e/(e + 1) \approx 0{,}731$ ve $1/(e + 1) \approx 0{,}269$.

Bölümün konuları iki etkileşimli oyunla da çalışılabilir. Yapay Zekâ Ajan Turnuvası, sabit, uyarlanabilir ve öğrenen ajanları Mahkûmun İkilemi, Yazı-Tura Eşleştirme, Koordinasyon Oyunu ve Taş-Kağıt-Makas'ta yarıştırır; 14.2.7'deki sonuçlar burada farklı oyunlarda denenebilir. Age of Çotanak bir bölge kontrol oyunudur; yapay zekâ rakibi çekişmeli arama (adversarial search) kullanır. Kendi başına denemek için basit bir örnek de pizza oylamasıdır: üç arkadaş tek pizzada anlaşmalıdır, her biri malzeme önerir ve sonra oylarlar. Küçük bir konsol uygulaması bile oylama kuralını ve stratejik oyu gösterir (14.6.1).

14.6.1 Örnek: pizza oylaması

Küçük bir örnek, bir durumu modellemeyi, çözmeyi ve stratejik davranışı görmeyi bir arada gösterebilir. Üç arkadaş tek bir pizzada anlaşmalıdır. Tercihleri şöyledir:

Kişi	1.	2.	3.
Ali	mantar	sucuk	zeytin
Banu	sucuk	mantar	zeytin
Cem	zeytin	sucuk	mantar

Tablo 14.20: Pizza oylamasında gerçek tercihler.

Örnek 14.12

1. **Model.** Oyuncular üç kişidir. Strateji, bildirilen sıralamadır. Kazanç, kazanan malzemenin kişinin gerçek sıralamasındaki yerine göre 2, 1 ya da 0'dır.
2. **Çoğunluk.** Her aday birer birincilik alır: 1–1–1 beraberlik. Çoğunluk kuralı karar veremez.
3. **Borda (2, 1, 0).** Mantar: $2 + 1 + 0 = 3$; sucuk: $1 + 2 + 1 = 4$; zeytin: $0 + 0 + 2 = 2$. Sucuk kazanır.
4. **Condorcet.** Sucuk–mantar: Banu ve Cem sucuğu tercih eder, 2–1. Sucuk–zeytin: Ali ve Banu sucuğu tercih eder, 2–1. Sucuk ikili karşılaştırmaların ikisini de kazanır: Condorcet kazananıdır.

5. **Stratejik oy.** Ali, sucuğu sona atıp "mantar, zeytin, sucuk" bildirirse Borda puanları mantar *2 + 1 + 0 = 3*, zeytin *1 + 0 + 2 = 3*, sucuk *0 + 2 + 1 = 3* olur. Beraberlik alfabetik çözülürse mantar kazanır. Ali yalan söyleyerek en sevdiğini aldırmıştır: kazancı 1'den 2'ye çıkar.

Bu bulgu, Borda sayımında **gömme** stratejisinin (güçlü rakibi sıralamanın sonuna atmak) bir örneğidir ve Gibbard–Satterthwaite teoreminin küçük bir yansımasıdır. Örnek kodla da denenebilir. Bir konsol sürümü sıralamaları alır ve çoğunluk, iki turlu, Borda, Condorcet ve onay oylaması kurallarını yan yana sayar. Bir web sürümünde kullanıcı bir oyuncunun bildirimini değiştirir ve kazananın nasıl kaydığını görür. Beraberlik kuralını değiştirmek de kazananı değiştirir; kuralın küçük bir ayrıntısı sonucu belirleyebilir.

14.7 Özet

- Yazılım ajanları artık oyuncudur: hedefi, bilgisi ve stratejisi vardır. Asil–vekil sorunu, çerçevelemeye duyarlılık, istem enjeksiyonu ve sorumluluk, ajanlarla birlikte yeniden önem kazanan oyun kuramı sorularıdır.
- Sonlu ufuklu sırayla teklif pazarlığı geriye doğru çıkarımla çözülür; son söz hakkı güçtür. 100 TL, $\delta = 0{,}9$ ve dört turda ilk öneren yalnızca 18,1 alır.
- Sonsuz ufukta Rubinstein çözümü ilk önerene $(1 - \delta_B)/(1 - \delta_A\delta_B)$ payını verir; simetrik sabırda $1/(1 + \delta)$. Sabır pazarlık gücüdür.
- Kinci tetik stratejisi, $\delta \geq 1 - 1/n$ ise danışıklılığı sürdürür. Sık fiyat güncelleme, şeffaf fiyatlar ve fiyat eşleme garantisi danışıklılığı kolaylaştırır; öğrenen fiyat algoritmaları açık anlaşma olmadan yüksek fiyat öğrenebilir.
- Evrensel en iyi strateji yoktur: bir ajanın başarısı oyuna, rakiplere, gürültüye ve hafızaya bağlıdır.
- Otomatik mekanizma tasarımı, kuralı IC ve IR kısıtlı bir optimizasyonun çözümü olarak bulur. İki tipli örnekte optimum doğrusal programın bir köşesidir ve ilan edilmiş bir fiyata karşılık gelir.
- Deneysel gelir maksimizasyonu, aynı veride seçilip ölçüldüğü için başarısını abartır (0,294'e karşı gerçek 0,249). Örneklem karmaşıklığı, iyi bir fiyat için kaç örnek gerektiğini sorar; seçim ve ölçüm için ayrı veri gerekir.
- Üstel mekanizma, adayları $\exp(\varepsilon \cdot R / 2\Delta)$ ile orantılı olasılıkla seçer; ε-diferansiyel mahremiyet ve yaklaşık teşvik uyumluluğu sağlar.
- Akıllı sözleşme taahhüdü kodla sağlar ama bekleyen işlemleri görünür kılar. Taban ücret + bahşiş kuralı, kullanıcı için teklifi basitleştirir; taban ücret talebe göre blok başına en çok %12,5 değişir ve sahte talebi pahalı kılmak için yakılır.
- Kullanıcı teşvik uyumu, üretici teşvik uyumu ve danışıklılığa dayanıklılık hedeflerinin üçü birlikte, blok alanı kıtken tam sağlanamaz.
- Sabit çarpımlı piyasa yapıcıda fiyat etkisi önceden hesaplanabilir; bu, sandviç saldırısını mümkün kılar. MEV, sıralama gücünün değeridir; önlemler bu değeri ya azaltır ya da açıkça fiyatlar. Sıkı kayma toleransı en basit korumadır.

- Kombinatoryal açık artırmada kazanan belirleme NP-zordur; açgözlü kurallar yaklaşıktır ve anahtar seçimi sonucu belirler. Etki maksimizasyonunda ise alt modülerlik, açgözlü seçime *1 − 1/e* garantisi verir.

14.8 Kavram listesi

asil–vekil sorunu *principal–agent problem* · çerçeveleme etkisi *framing effect* · istem enjeksiyonu *prompt injection* · geriye doğru çıkarım *backward induction* · alt oyun mükemmel denge *subgame perfect equilibrium* · iskonto faktörü *discount factor* · Rubinstein pazarlık modeli *Rubinstein bargaining model* · durağan strateji *stationary strategy* · danışıklılık *collusion* · algoritmik danışıklılık *algorithmic collusion* · kinci tetik stratejisi *grim trigger strategy* · pekiştirmeli öğrenme *reinforcement learning* · otomatik mekanizma tasarımı *automated mechanism design* · yaklaşık teşvik uyumluluğu *approximate incentive compatibility* · deneysel gelir *empirical revenue* · aşırı uyum *overfitting* · örneklem karmaşıklığı *sample complexity* · çapraz doğrulama *cross-validation* · diferansiyel mahremiyet *differential privacy* · üstel mekanizma *exponential mechanism* · akıllı sözleşme *smart contract* · taahhüt *commitment* · bekleme havuzu *mempool* · işlem ücreti mekanizması *transaction fee mechanism* · taban ücret *base fee* · bahşiş *tip* · ücret yakma *fee burning* · otomatik piyasa yapıcı *automated market maker* · kayma toleransı *slippage tolerance* · fiyat etkisi *price impact* · MEV (maksimum elde edilebilir değer) *maximal extractable value* · sandviç saldırısı *sandwich attack* · öne geçme *front-running* · arkadan gelme *back-running* · toplu açık artırma *batch auction* · kombinatoryal açık artırma *combinatorial auction* · kazanan belirleme problemi *winner determination problem* · NP-zor *NP-hard* · PPAD-tam *PPAD-complete* · etki maksimizasyonu *influence maximization* · bağımsız çağlayan modeli *independent cascade model* · alt modülerlik *submodularity* · çok ajanlı pekiştirmeli öğrenme *multi-agent reinforcement learning* · Pigou vergisi *Pigouvian tax* · bilgi tasarımı (Bayesçi ikna) *information design* · tahmin piyasası *prediction market* · eşler arası tahmin *peer prediction* · tekrarlanabilirlik *reproducibility* · güven aralığı *confidence interval* · Borda sayımı *Borda count* · Condorcet kazananı *Condorcet winner*

14.9 Kendini Dene

1. İki ajan 100 TL'yi üç turda paylaşacak: A 1. ve 3. turlarda, B 2. turda önerir; $\delta = 0{,}8$. Geriye doğru çıkarımla bulunan dengedeki paylaşım nedir? Sonsuz ufukta, A'nın sabrı $\delta_A = 0{,}8$ ve B'ninki $\delta_B = 0{,}9$ olsaydı A'nın payı ne olurdu?

   Cevap

   Tur 3'te son söz A'dadır: A 100. Tur 2'de B, A'ya bekleme değeri $0{,}8 \cdot 100 = 80$'i önerir: A 80, B 20. Tur 1'de A, B'ye $0{,}8 \cdot 20 = 16$ önerir: A 84, B 16 ve öneri hemen kabul edilir. Sonsuz ufukta $x^* = (1 - 0{,}9)/(1 - 0{,}72) = 0{,}1/0{,}28 \approx 0{,}357$: A 35,7 alır. Daha sabırsız olan A, ilk öneren olduğu hâlde küçük payı alır.

2. İki firma tekel fiyatında her dönem 50'şer kâr ediyor. Fiyat kıran firma her dönem 100 alıyor; ancak rakip fiyat kırıldığını iki dönem sonra fark ediyor, yani sapan iki dönem boyunca 100 alıyor, sonra sonsuza dek 0. Kinci tetik stratejisi hangi δ için işbirliğini sürdürür? Anında fark edilen duruma göre eşik nasıl değişir?

   Cevap

   İşbirliği $50/(1 - \delta)$, sapma $100 + 100\delta$ değerindedir. Koşul $50/(1 - \delta) \geq 100(1 + \delta) \iff 50 \geq 100(1 - \delta)(1 + \delta) = 100(1 - \delta^2) \iff \delta^2 \geq 1/2 \iff \delta \geq 0{,}707$. Anında fark edilince eşik 0,5 idi. Gecikmeli fark etme sapmayı daha kârlı kılar ve gereken sabrı artırır; şeffaf fiyatların danışıklılığı kolaylaştırmasının nedeni budur.

3. Geçmişte gözlenen 5 alıcı değeri 0,8; 0,7; 0,3; 0,25; 0,1'dir. Deneysel geliri maksimize eden fiyat ve gelir nedir? Değerler gerçekte $U[0, 1]$'den geliyorsa bu fiyatın gerçek geliri kaçtır?

   Cevap

   Gelirler $0{,}8 \cdot 1/5 = 0{,}16$; $0{,}7 \cdot 2/5 = 0{,}28$; $0{,}3 \cdot 3/5 = 0{,}18$; $0{,}25 \cdot 4/5 = 0{,}2$; $0{,}1 \cdot 5/5 = 0{,}1$. En iyisi $p = 0{,}7$, deneysel gelir 0,28. Gerçek gelir $0{,}7 \cdot 0{,}3 = 0{,}21$'dir; en iyi 0,25'in altında kalır. Beş örnek, fiyatı gerçek en iyiden (0,5) uzağa götürmüştür ve deneysel gelir yine iyimserdir.

4. Üstel mekanizmada üç aday fiyatın puanları 0, 1 ve 2'dir; $\varepsilon = 1$, $\Delta = 1$. Seçim olasılıkları nedir? En iyi adayın olasılığı en kötününkünün kaç katıdır?

   Cevap

   Ağırlıklar $\exp(R/2)$: $e^0 = 1$, $e^{0{,}5} \approx 1{,}649$, $e^1 \approx 2{,}718$; toplam $\approx 5{,}367$. Olasılıklar $\approx 0{,}186$; 0,307; 0,506. En iyi aday, en kötünün $e^1 \approx 2{,}72$ katı olasılık alır; puan oranı (2/0) ile ilgisi yoktur.

5. Taban ücret 200'dür. Sıradaki dört blok %100, %100, %0, %0 doludur. Dördüncü bloktan sonra

taban ücret kaç olur? Ücret neden 200'e geri dönmez?

Cevap

200 → *200 · 1,125 = 225* → *253,125* → *253,125 · 0,875 ≈ 221,48* → *≈ 193,80*. Değişim çarpımsaldır ve *1,125 · 0,875 ≈ 0,984 < 1*'dir; eşit sayıda dolu ve boş blok ücreti biraz düşürür.

6. Sabit çarpımlı havuz (2000; 2000)'dir. Bir kullanıcı 200 A satıyor. Kaç B alır ve kayma yüzde kaçtır? Bu sonuç, (1000; 1000) havuzunda 100 A satmakla nasıl karşılaştırılır?

Cevap

Δy = 2000 · 200 / 2200 ≈ 181,82 B; kayma *1 − 181,82/200 ≈ %9,1*. (1000; 1000) havuzunda 100 A satmak da %9,1 kayma verir. Kayma *Δx/(x + Δx)*'e, yani işlemin havuza göre büyüklüğüne bağlıdır; derin havuz aynı işlemi daha az kaymayla taşır.

7. Dört paket teklifi: t1 = ({a, b}, 6), t2 = ({b, c}, 6), t3 = ({a}, 4), t4 = ({c}, 4). Değere göre ve nesne başına değere göre açgözlü seçim hangi toplamları verir? Optimum nedir?

Cevap

Değere göre sıra t1, t2, t3, t4: t1 alınır (a, b); t2 b yüzünden, t3 a yüzünden atlanır; t4 alınır. Toplam 10. Nesne başına değer t3 ve t4 için 4, t1 ve t2 için 3'tür: t3 ve t4 alınır, t1 (a) ve t2 (c) çakışır. Toplam 8. Optimum 10'dur ({t1, t4} ya da {t2, t3}). Bu kez nesne başına değer küçük paketlere aldanır.

8. Bir ajan turnuvasında bir eşleşme 25 kez tekrarlanıyor; ortalama skor 3,1, standart sapma 0,5. Yaklaşık %95 güven aralığı nedir? Tekrar sayısı 100'e çıkarılırsa aralığın genişliği nasıl değişir?

Cevap

Yarı genişlik *1,96 · 0,5 / √25 = 1,96 · 0,1 ≈ 0,196*; aralık yaklaşık [2,90; 3,30]. 100 tekrarda *√100 = 10* olur ve yarı genişlik ≈ 0,098'e iner: dört kat tekrar, aralığı yarıya daraltır.

14.10 İleri okuma

- T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory* (2016): mekanizma tasarımı, kombinatoryal açık artırmalar ve öğrenme dinamiklerini bilgisayar bilimi bakışıyla, kısa derslerle anlatır.
- T. Roughgarden, “Transaction Fee Mechanism Design” (2021): işlem ücreti mekanizmalarının hedeflerini (kullanıcı ve üretici teşvik uyumu, danışıklılığa dayanıklılık) biçimsel olarak tanımlar ve taban ücret kuralını bu hedeflerle çözümler.

- E. Calvano, G. Calzolari, V. Denicolò, S. Pastorello, "Artificial Intelligence, Algorithmic Pricing, and Collusion" (2020): öğrenen fiyat ajanlarının iletişim kurmadan yüksek fiyat öğrenip sapmaları cezalandırdığı benzetimleri ayrıntılı verir.
- D. Easley, J. Kleinberg, *Networks, Crowds, and Markets* (2010): ağlarda yayılma, pazarlık ve piyasaları az matematikle, çok örnekle anlatır.

14.11 Kaynaklar

1. R. Axelrod, "Effective choice in the Prisoner's Dilemma", *Journal of Conflict Resolution* 24(1), 1980.
2. N. Brown, T. Sandholm, "Superhuman AI for multiplayer poker", *Science*, 2019.
3. E. Calvano, G. Calzolari, V. Denicolò, S. Pastorello, "Artificial Intelligence, Algorithmic Pricing, and Collusion", *American Economic Review*, 2020.
4. H. Chung, E. Shi, "Foundations of transaction fee mechanism design", *ACM-SIAM SODA*, 2023.
5. H. Chung, T. Roughgarden, E. Shi, "Collusion-resilience in transaction fee mechanism design", *ACM EC*, 2024.
6. R. Cole, T. Roughgarden, "The Sample Complexity of Revenue Maximization", *ACM STOC*, 2014.
7. V. Conitzer, T. Sandholm, "Complexity of Mechanism Design", *UAI*, 2002.
8. P. Daian vd., "Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability", *IEEE S&P*, 2020.
9. P. Dütting, Z. Feng, H. Narasimhan, D. Parkes, S. Ravindranath, "Optimal Auctions through Deep Learning", *ICML*, 2019.
10. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010.
11. R. A. Hearn, E. D. Demaine, *Games, Puzzles, and Computation*, A K Peters, 2009.
12. A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017.
13. D. Kempe, J. Kleinberg, É. Tardos, "Maximizing the Spread of Influence through a Social Network", *KDD*, 2003.
14. D. Lehmann, L. I. O'Callaghan, Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions", *Journal of the ACM* 49(5), 2002.
15. F. McSherry, K. Talwar, "Mechanism Design via Differential Privacy", *IEEE FOCS*, 2007.
16. N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007.
17. T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press, 2016.
18. T. Roughgarden, "Transaction Fee Mechanism Design", *ACM EC*, 2021.
19. A. Rubinstein, "Perfect Equilibrium in a Bargaining Model", *Econometrica* 50(1), 1982.

EK A

Terim Sözlüğü

Kitapta geçen İngilizce terimler ve her biri için kullanılan tek Türkçe karşılık. Sağdaki sayı, terimin ayrıntılı anlatıldığı bölümdür. Sözlüğün aranabilir hâli ders sayfasındadır.

#

(agent) goal alignment → hedef uyumu 14
2-approval → 2-onay 7

A

accountability → hesap verebilirlik 11
accuracy → doğruluk 11
action → eylem 2
adaptation → adaptasyon 13
adaptive → uyarlanabilir 14
additivity → toplamsallık 10
adoption model → benimseme modeli 6
advertiser → reklamveren 6
affine transformation → afin dönüşüm 1
agenda (manipulation) → gündem (manipülasyonu) 7
agent → ajan 10
agent architecture → ajan mimarisi 10
AI agent → yapay zekâ ajanı 14
algorithmic audit → algoritmik denetim 11
algorithmic collusion → algoritmik danışıklılık 14
algorithmic fairness → algoritmik adalet 11
algorithmic game theory (AGT) → algoritmik oyun kuramı 2
all-pay auction → tümü öder açık artırması 5
Allais paradox → Allais paradoksu 1
allocation → tahsis 4
allocation rule → tahsis kuralı 5
alpha–beta pruning → alfa-beta budaması 2
alternating-offers bargaining → sırayla teklif pazarlığı 14
altruistic punishment → özgeci ceza 12
ambiguity aversion → belirsizlikten kaçınma 1
anonymity → anonimlik 7
anti-coordination → anti-koordinasyon 2
antisocial punishment → antisosyal ceza 12
approval voting → onay oylaması 7
approximate incentive compatibility → yaklaşık teşvik uyumluluğu 14
approximation algorithm → yaklaşım algoritması 9
approximation ratio → yaklaşım oranı 9
argmax → argmax 0
Arrow's impossibility theorem → Arrow'un imkânsızlık teoremi 7
ask (sell order, double auction) → satış isteği 5
asymmetric game → asimetrik oyun 2
atomic → atomik 8
auction → açık artırma 5
automated market maker (AMM) → otomatik piyasa yapıcı (AMM) 14
automated mechanism design → otomatik mekanizma tasarımı 14
autonomy → otonomi 10
average mechanism → ortalama mekanizması 4
Axelrod tournament → Axelrod turnuvası 13

B

back-running → arkadan gelme 14
backward induction → geriye doğru çıkarım 2
Balance algorithm → dengeleme (Balance) kuralı 6
balance for the positive/negative class → sınıf dengesi 11
balanced game → dengeli oyun 10
ballot → oy pusulası 7
bandwidth → bant genişliği 8
Banzhaf index → Banzhaf endeksi 10
bargaining → pazarlık 14
base fee → taban ücret 14
base rate → taban oranı 11
base-rate neglect → taban oranı ihmali 1
batch auction → toplu açık artırma 14
Battle of the Sexes → Cinsiyetler Savaşı 2
Bayes' rule → Bayes kuralı 0
Bayesian average → Bayesçi ortalama 6
Bayesian decision rule → Bayesçi karar kuralı 1
Bayesian game → Bayesçi oyun 4
Bayesian incentive compatible (BIC) → Bayesçi teşvik uyumlu (BIC) 4
Bayes–Nash equilibrium (BNE) → Bayes–Nash dengesi 4
Beckmann potential → Beckmann potansiyeli 8
belief–desire–intention (BDI) → inanç–arzu–niyet (BDI) 10

Bell number → Bell sayısı 10
Bellman–Ford algorithm → Bellman–Ford algoritması 8
Bertrand competition → Bertrand rekabeti 3
Bertrand paradox → Bertrand paradoksu 3
best response → en iyi tepki 3
best-response correspondence → en iyi tepki eşlemesi 3
best-response curve → en iyi tepki eğrisi 3
best-response dynamics → en iyi tepki dinamiği 13
best-response marking → en iyi tepki işaretleme 2
BGP (Border Gateway Protocol) → BGP (ağlar arası yönlendirme protokolü) 8
bias (societal / data) → önyargı 11
bid → teklif 5
bid (buy order, double auction) → alış teklifi 5
bid shading → teklif kırpma 5
bidder → teklifçi 5
big-O and Θ notation → büyük-O ve Θ gösterimi 0
bilateral trade → iki taraflı ticaret 4
binding agreement → bağlayıcı anlaşma 2
bisection → ikiye bölme 11
block producer → blok üreticisi 14
blockchain → blokzincir 14
blocking pair → engelleyici çift 4
bluffing → blöf 1
Bondareva–Shapley theorem → Bondareva–Shapley teoremi 10
Borda count → Borda sayımı 7
Boston mechanism (immediate acceptance) → Boston mekanizması 4
bounded rationality → sınırlı rasyonellik 12
BPR latency function → BPR gecikme fonksiyonu 8
Braess's paradox → Braess paradoksu 8
branch → dal 2
branching factor → dallanma etkeni 2
budget balance → bütçe dengesi 4
budget surplus → bütçe fazlası 4
bundle → demet 11
burying → gömme 7
buyer / seller surplus → alıcı / satıcı artığı 5

C

cake cutting → pasta kesme 11
calibration (within groups) → kalibrasyon 11
capacity → kapasite 8
capacity augmentation → kapasite artırımı 9
cardinal (von Neumann–Morgenstern) utility → kardinal fayda 1
cartel → kartel 3
Cartesian product → Kartezyen çarpım 0
case analysis → durum analizi 2
centipede game → kırkayak oyunu 2
certainty effect → kesinlik etkisi 1
certainty equivalent → kesinlik eşdeğeri (KE) 1
chain rule → zincir kuralı 0
chance node → şans düğümü 1
characteristic function → karakteristik fonksiyon 10
chicken-and-egg problem → tavuk-yumurta sorunu 6
Chouldechova's identity → Chouldechova özdeşliği 11
Clarke pivot rule → Clarke pivot kuralı 4
click-through rate (CTR) → tıklanma oranı 6
coalition → koalisyon 10
coalition formation → koalisyon oluşumu 10
coalition structure → koalisyon yapısı 10
coarse correlated equilibrium (CCE) → kaba korelasyonlu denge 13
coefficient of optimism → iyimserlik katsayısı 1
cold start → soğuk başlangıç 6
collective decision → kolektif karar 12
collusion → danışıklılık 4
combination (binomial coefficient) → kombinasyon C(n, k) 0
combinatorial auction → kombinatoryal açık artırma 14
combinatorial game → kombinatoryal oyun 2
commitment → taahhüt 14
common knowledge → ortak bilgi 2
common prior → ortak ön olasılık 4
common value → ortak değer 5
competitive ratio → rekabet oranı 9
complement (of an event) → tümleyen 0
complementary goods → tamamlayıcı ürünler 5
complete graph → tam çizge 8
complete information → tam bilgi 2
completeness / continuity / independence axioms → tamlık / süreklilik / bağımsızlık aksiyomları 1
completion time → bitiş süresi 9
computational social choice → hesaplamalı sosyal seçim 7
concave → içbükey 1
conditional cooperator → koşullu işbirlikçi 10
conditional probability → koşullu olasılık 0
Condorcet jury theorem → Condorcet jüri teoremi 12
Condorcet loser → Condorcet kaybedeni 7
Condorcet paradox / cycle → Condorcet paradoksu / döngüsü 7
Condorcet winner → Condorcet kazananı 7
confidence interval → güven aralığı 14
confusion matrix → karışıklık matrisi 11
congestion game → tıkanıklık oyunu 8
congestion pricing → tıkanıklık fiyatlandırması 9
congestion toll → tıkanıklık ücreti 9
consensus → uzlaşı 12
constant-product pool → sabit çarpımlı havuz 14

constant-sum game → sabit toplamlı oyun 2
Contract Net Protocol → sözleşme ağı protokolü 10
convention / social law → konvansiyon / sosyal yasa 10
convergence → yakınsama 13
convex → dışbükey 1
convex game → dışbükey oyun 10
cooperate (Prisoner's Dilemma action) → işbirliği 13
cooperative game → işbirlikçi oyun 2
coordination failure → koordinasyon başarısızlığı 3
coordination game → koordinasyon oyunu 3
coordination ratio → koordinasyon oranı 9
Copeland rule → Copeland kuralı 7
core → çekirdek 10
correlated equilibrium (CE) → korelasyonlu denge 3
counterfactual regret minimization (CFR) → karşı-olgusal pişmanlık minimizasyonu (CFR) 13
Cournot duopoly → Cournot duopolü 3
critical (swing) member → kritik üye 10
critical bid → kritik teklif 4
critical mass → kritik kitle 6
cross-side (indirect) network effect → çapraz ağ etkisi 6
cross-validation → çapraz doğrulama 14
cut query → kesim sorgusu 11
cut-and-choose → kes-ve-seç 11
cutoff test → kesme testi 2

D

D'Hondt method → D'Hondt yöntemi 7
data drift → veri kayması 11
decentralized exchange (DEX) → merkeziyetsiz borsa 14
decision making → karar alma 1
decision node → karar düğümü 2
decision tree → karar ağacı 1
decision under risk → risk altında karar 12
decision under uncertainty → belirsizlik altında karar 1
deep Q-network (DQN) → DQN 13
defect (Prisoner's Dilemma action) → ihanet 2
degenerate game → dejenere oyun 3
deliberative agent → planlayan ajan 10
demographic parity → demografik eşitlik 11
depth-limited search → derinlik sınırlı arama 2
derivative → türev 0
dictator game → diktatör oyunu 12
differential privacy → diferansiyel mahremiyet 14
Digital Markets Act (DMA) → Dijital Piyasalar Yasası 6
Dijkstra's algorithm → Dijkstra algoritması 8
diminishing marginal utility → azalan marjinal fayda 1
direct mechanism → doğrudan mekanizma 4
direct network effect → doğrudan ağ etkisi 6
discount factor → iskonto faktörü 0
discriminant → diskriminant 0
discriminatory (pay-as-bid) auction → ayrımcı fiyatlı açık artırma 5
diversity prediction theorem → çeşitlilik tahmin teoremi 12
doctrinal paradox (discursive dilemma) → doktrin paradoksu 7
Dodgson rule → Dodgson kuralı 7
dominance by a mixed strategy → karışık stratejiyle baskılama 3
dominant strategy → baskın strateji 2
dominant-strategy equilibrium → baskın strateji dengesi 3
dominant-strategy incentive compatible (DSIC) → baskın stratejide teşvik uyumlu (DSIC) 4
dominated strategy → baskılanan strateji 3
double auction → çift taraflı açık artırma 5
dummy (null) player → kukla oyuncu 10
Dutch (descending) auction → Hollanda açık artırması 5
Duverger's law → Duverger yasası 7

E

ECMP (equal-cost multi-path) → eşit maliyetli çok yol (ECMP) 8
edge → kenar 8
efficiency (allocative / social) → verimlilik 4
efficiency axiom → verimlilik aksiyomu 10
efficiency loss → verimlilik kaybı 9
efficient → verimli 4
egalitarian (Rawlsian) welfare → eşitlikçi refah 11
egalitarian criterion → eşitlikçi ölçüt 12
EIP-1559 → EIP-1559 14
Ellsberg paradox → Ellsberg paradoksu 1
emergent behavior → ortaya çıkan davranış 10
empirical revenue → deneysel gelir 14
endowment effect → bağış etkisi 12
English (ascending) auction → İngiliz açık artırması 5
entry (deterrence) game → piyasaya giriş oyunu 2
envy-free equilibrium (GSP) → kıskançsız denge 6
envy-free up to one good (EF1) → bir mala kadar kıskançsız (EF1) 11
envy-freeness → kıskançsızlık 11
episode (RL) → epizot 13
equal opportunity → fırsat eşitliği 11
equal split → eşit bölüşüm 10
equalized odds → eşitlenmiş olasılıklar 11
equilibrium selection → denge seçimi 3
evaluation function → değerlendirme fonksiyonu 2
evidence → kanıt 1
evolutionarily stable strategy (ESS) → evrimsel kararlı strateji (ESS) 13
EXP3 (adversarial bandit) → EXP3 (çekişmeli haydut) 13
expected loss → beklenen kayıp 1

expected payoff → beklenen kazanç 2
expected utility → beklenen fayda 1
expected value → beklenen değer 0
expected value of perfect information (EVPI) → mükemmel bilginin beklenen değeri (EVPI) 1
expected value of sample information (EVSI) → örnek bilginin beklenen değeri (EVSI) 1
expert (online learning) → uzman 13
explainability → açıklanabilirlik 11
exploration–exploitation → keşif–sömürü 13
exponential mechanism → üstel mekanizma 14
extensive form → genişletilmiş form 2
external regret → dışsal pişmanlık 13
externality → dışsallık 4

F

facility location → tesis yerleşimi 4
factorial → faktöriyel 0
fair cost sharing → adil maliyet paylaşımı 9
fair division → adil bölüşüm 11
fairness → adalet 11
false negative rate (FNR) → yanlış negatif oranı 11
false positive rate (FPR) → yanlış pozitif oranı 11
false-name (sybil) bidding → sahte kimlik 4
federated learning → federe öğrenme 10
fee burning → ücret yakma 14
feedback loop → geri besleme döngüsü 6
fictitious play → kurgusal oyun 13
finite / infinite horizon → sonlu / sonsuz ufuk 2
finite game → sonlu oyun 2
finite improvement property → sonlu iyileşme özelliği 8
first-order condition → birinci mertebe koşulu 5
first-price sealed-bid auction → birinci fiyat kapalı zarf açık artırması 5
fitness → uygunluk 13
fixed-point theorem (Brouwer, Kakutani) → sabit nokta teoremi 3
FIXP → FIXP (sabit nokta bulma problemleri sınıfı; çözümü irrasyonel olabilen problemler) 3
focal point (Schelling) → odak noktası 3
folk theorem → halk teoremi 13
follow the leader → lideri izle 13
four-fifths rule (disparate impact) → beşte dört kuralı 11
framing effect → çerçeveleme etkisi 12
Free Money game → Bedava Para oyunu 2
free rider → beleşçi 10
free riding → beleşçilik 10
front-running → öne geçme 14

G

Gale–Shapley algorithm (deferred acceptance) → Gale–Shapley (ertelenmiş kabul) algoritması 4
game → oyun 2
game of Chicken → Tavuk oyunu 2
game theory → oyun kuramı 2
game tree → oyun ağacı 2
gaming (a metric) → oyunlama 11
gatekeeper (DMA) → geçit bekçisi 6
generalization → genelleme 14
Generalized Second Price (GSP) auction → genelleştirilmiş ikinci fiyat (GSP) açık artırması 6
geometric series → geometrik seri 0
Gibbard–Satterthwaite theorem → Gibbard–Satterthwaite teoremi 7
glove game → eldiven oyunu 10
Goodhart's law → Goodhart yasası 11
grand coalition → büyük koalisyon 10
graph → çizge 8
greedy (algorithm) → açgözlü 10
greedy list scheduling → açgözlü liste algoritması 9
grim trigger strategy → kinci tetik stratejisi 2
group fairness → grup adaleti 11
guess 2/3 of the average → 2/3 ortalaması oyunu 2

H

harmonic number H_n → harmonik sayı H_n 9
Harsanyi transformation → Harsanyi dönüşümü 4
Hawk–Dove game → Şahin–Güvercin oyunu 13
heuristics → sezgisel kurallar 12
horizon effect → ufuk etkisi 2
housing market (house exchange) → ev takası 4
Hungarian algorithm → Macar algoritması 10
Hurwicz criterion → Hurwicz ölçütü 1
hybrid architecture → hibrit mimari 10

I

impartial game → tarafsız oyun 2
imperfect information → mükemmel olmayan bilgi 2
implementation theory → uygulama kuramı 4
impossibility (incompatibility) theorems → uyumsuzluk teoremleri 11
imputation → paylaşım 10
incentive → teşvik 4
incentive compatibility (IC) → teşvik uyumluluğu 4
incentive compatible → teşvik uyumlu 4
incomplete information → eksik bilgi 4
incremental mean → artımlı ortalama 13
independence of irrelevant alternatives (IIA) → ilgisiz alternatiflerden bağımsızlık (IIA) 7
independent cascade model → bağımsız çağlayan modeli 14

independent private values (IPV) → bağımsız özel değerler 5

indifference principle → kayıtsızlık ilkesi 3

indirect mechanism → dolaylı mekanizma 4

individual fairness → bireysel adalet 11

individual rationality (IR) → bireysel rasyonellik 4

indivisible goods → bölünemez mallar 11

inequity aversion → eşitsizlikten kaçınma 12

influence maximization → etki maksimizasyonu 14

information asymmetry → bilgi asimetrisi 6

information cascade → bilgi şelalesi 12

information design (Bayesian persuasion) → bilgi tasarımı (Bayesçi ikna) 14

information set → bilgi kümesi 2

information structure → bilgi yapısı 10

inspection game → denetim oyunu 3

instant-runoff voting (IRV) → anında ikinci tur 7

integral → integral 0

interchangeability (of equilibria) → değiştirilebilirlik 3

interdependent values → birbirine bağlı değerler 5

internal (swap) regret → iç pişmanlık 13

Internet service provider (ISP) → internet servis sağlayıcısı (İSS) 8

iterated elimination of strictly dominated strategies (IESDS) → yinelemeli (baskın strateji) eleme 3

iterative deepening → yinelemeli derinleştirme 2

J

Jensen's inequality → Jensen eşitsizliği 1

judgment aggregation → yargı birleştirme 7

K

k-approval → k-onay 7

Kemeny rule → Kemeny kuralı 7

kidney exchange → böbrek takası 4

L

Laplace criterion → Laplace ölçütü 1

large language model (LLM) → büyük dil modeli (LLM) 14

largest remainder (Hare) method → en büyük kalan (Hare) yöntemi 7

last diminisher → son azalan 11

last iterate → son yineleme 13

latency (delay) function → gecikme fonksiyonu 8

leaf (terminal node) → yaprak 2

learning architecture → öğrenen mimari 10

learning rate → öğrenme oranı 13

Lemke–Howson algorithm → Lemke–Howson algoritması 3

level-k reasoning → seviye-k düşünme 2

likelihood → olabilirlik 1

linear programming (LP) → doğrusal programlama 3

link → bağlantı 8

Lipschitz condition → Lipschitz koşulu 11

load balancing → yük dengeleme 9

locally envy-free equilibrium (GSP) → yerel kıskançsız denge 6

logit choice (logit response) → logit seçim 12

logit function → logit fonksiyonu 0

loss aversion → kayıptan kaçınma 12

loss function → kayıp fonksiyonu 11

lottery → piyango 1

M

M/M/1 queue → M/M/1 kuyruğu (tek sunuculu kuyruk) 8

majority game → çoğunluk oyunu 10

majority rule → mutlak çoğunluk 7

makespan → en büyük yük 9

manager / contractor (contract net) → yönetici / yüklenici 10

manipulation-resistant → manipülasyona dayanıklı 5

marginal contribution → marjinal katkı 10

marginal cost → marjinal maliyet 8

marginal per-capita return (MPCR) → kişi başı marjinal getiri (MPCR) 10

marginal-cost toll → marjinal maliyet ücreti 9

market → piyasa 5

market design → piyasa tasarımı 5

market for lemons → limonlar piyasası 6

market maker → piyasa yapıcı 5

marketplace → pazar yeri 6

markup → kâr payı (fiyat ile marjinal maliyet farkı) 6

matcher / mismatcher (Matching Pennies roles) → Eşleştiren / Ayrıştıran 2

matching market → eşleştirme piyasası 4

Matching Pennies → Yazı-Tura Eşleştirme 2

max-min fairness → maksimin adalet 8

maximal extractable value (MEV) → MEV (maksimum elde edilebilir değer) 14

maximax → maksimaks 1

maximin → maksimin 3

maximin (Simpson) rule → maksimin (Simpson) kuralı 7

maximize → maksimize etmek 2

May's theorem → May teoremi 7

McAfee double auction → McAfee kuralı 5

mechanism → mekanizma 4

mechanism design → mekanizma tasarımı 4

mechanism design without money → parasız mekanizma tasarımı 4

median mechanism → medyan mekanizması 4

median voter theorem → medyan seçmen teoremi 7

mempool → bekleme havuzu 14

Metcalfe's law → Metcalfe yasası 8

midpoint rule → orta nokta kuralı 5

minimax → minimaks 3
minimax regret (Savage) → minimaks pişmanlık 1
minimax theorem (von Neumann) → minimaks teoremi 3
minimize → minimize etmek (en aza indirmek) 8
mixed (strategy) equilibrium → karışık denge 3
mixed Nash equilibrium → karışık Nash dengesi 3
mixed strategy → karışık strateji 3
mixed-motive → karma güdü 10
model card → model kartı 11
monotone allocation rule → monoton tahsis kuralı 4
monotonicity → monotonluk 7
Monte Carlo tree search (MCTS) → Monte Carlo ağaç araması 2
move ordering → hamle sıralaması 2
moving knife (Dubins–Spanier) → hareketli bıçak (Dubins–Spanier) 11
multi-agent reinforcement learning → çok ajanlı pekiştirmeli öğrenme 14
multi-agent system (MAS) → çoklu ajan sistemi 10
multi-armed bandit → çok kollu haydut 13
multi-homing → çoklu kullanım 6
multi-unit auction → çok birimli açık artırma 5
multiple equilibria → çoklu denge 3
multiplicative weights (Hedge) → çarpımsal ağırlıklar (Hedge) 13
Myerson's lemma → Myerson lemması 4
Myerson–Satterthwaite theorem → Myerson–Satterthwaite teoremi 4

N

N-position → N-konumu 2
Nash equilibrium (NE) → Nash dengesi 3
Nash social welfare → Nash sosyal refahı 11
Nash's existence theorem → Nash'in varlık teoremi 3
natural frequencies → doğal sıklıklar 1
natural logarithm → doğal logaritma 0
Nature (chance player) → Doğa 4
negamax → negamaks 2
negative predictive value (NPV) → negatif kestirim değeri (NPV) 11
neighboring datasets → komşu veri kümeleri 14
network → ağ 8
network design → ağ tasarımı 8
network effect → ağ etkisi 6
network formation game → ağ oluşturma oyunu 8
neutrality → tarafsızlık 7
nim-sum → nim toplamı 2
no-regret → pişmanlıksız 13
node → düğüm 8
noise (trembles) → gürültü 13
non-cooperative game → işbirlikçi olmayan oyun 2
non-credible threat → inandırıcı olmayan tehdit 2
non-dictatorship → diktatör yok (diktatörlük olmaması) 7
non-stationary → durağan olmayan 10
nonatomic (flow) → bölünebilir (atomik olmayan) 8
normal form (strategic form) → normal form (stratejik form) 2
normal play convention → normal oyun kuralı 2
NP-hard → NP-zor 3
nucleolus → çekirdekçik 10

O

obedience constraints → uyma kısıtları 3
oddness theorem (Wilson) → tek sayılılık teoremi 3
on-chain governance → zincir üstü yönetişim 14
one-shot game → tek seferlik oyun 3
online allocation → çevrim içi tahsis 6
online platform → çevrim içi platform 6
optimal auction (Myerson) → en iyi (optimal) açık artırma 5
optimistic bias (in-sample overestimation) → iyimser yanlılık 14
optimization → optimizasyon 14
order book → emir defteri 5
ordinal utility → sıralı fayda 2
OSPF (Open Shortest Path First) → OSPF (bağlantı durumu yönlendirme protokolü) 8
outcome → sonuç 2
overconfidence → aşırı güven 12
overfitting → aşırı uyum 14

P

P-position → P-konumu 2
pairwise majority graph → ikili çoğunluk çizgesi 7
pairwise stability → ikili kararlılık 8
Pareto (unanimity) condition → Pareto (oybirliği) koşulu 7
Pareto dominance → Pareto baskınlık 3
Pareto efficiency → Pareto verimliliği 11
Pareto optimal → Pareto optimal 2
Pareto optimality → Pareto optimallik 2
partial derivative → kısmi türev 0
participation equilibrium → katılım dengesi 6
payment rule → ödeme kuralı 5
payoff → kazanç 2
payoff function → kazanç fonksiyonu 2
payoff matrix → kazanç matrisi 2
peer prediction → eşler arası tahmin 14
peering → internet eşlemesi 8
perfect information → mükemmel bilgi 2
Pigou network → Pigou ağı 9
Pigouvian tax → Pigou vergisi 12
pivotal player → kilit oyuncu 10

player → oyuncu 2
PLS-complete → PLS-tam 8
plurality rule → çoğulluk 7
ply → yarım hamle 2
policy (agent / RL) → politika 13
positional scoring rule → pozisyonel puanlama kuralı 7
positive predictive value (PPV), precision → kesinlik (PPV) 11
positive responsiveness → pozitif duyarlılık 7
posted price → ilan edilmiş fiyat 14
posterior → son olasılık 0
potential function → potansiyel fonksiyonu 8
potential game → potansiyel oyun 8
power index → güç endeksi 10
PPAD-complete → PPAD-tam 3
pre-processing / post-processing → ön işleme / son işleme 11
prediction market → tahmin piyasası 14
predictive parity → tahmin eşitliği 11
preference elicitation → tercih çıkarımı 7
preference profile → tercih profili 7
price discovery → fiyat keşfi 5
price impact → fiyat etkisi 14
Price of Anarchy (PoA) → Anarşinin Bedeli 9
price of fairness → adaletin bedeli 11
Price of Stability (PoS) → İstikrarın Bedeli 9
price structure → fiyat yapısı 6
price-matching guarantee → fiyat eşleme garantisi 14
pricing → fiyatlandırma 6
principal–agent problem → asil–vekil sorunu 14
prior → ön olasılık 0
priority fee (tip) → bahşiş 14
Prisoner's Dilemma → Mahkûmun İkilemi 2
privacy → mahremiyet 11
private information → özel bilgi 4
private value → özel değer 5
pro-activeness → proaktivite 10
probability density → olasılık yoğunluğu 0
probability distribution → olasılık dağılımı 0
probability vector → olasılık vektörü 0
probability weighting → olasılık ağırlığı 12
product rule (counting) → çarpım kuralı 0
profitable deviation → kârlı sapma 6
prompt → istem 14
prompt injection → istem enjeksiyonu 14
proof of stake → hisse ispatı 14
proportional fairness → orantılı adalet 8
proportional representation → nispi temsil 7
proportionality → orantılılık 11
proposing side → teklif veren taraf 4
prospect theory → beklenti kuramı 12
protected attribute → korunan özellik 11
proxy variable → vekil değişken 11
public good → kamu malı 10
public goods game → kamu malı oyunu 10
pure coordination game → saf koordinasyon oyunu 2
pure strategy → saf strateji 2
pure(-strategy) Nash equilibrium → saf Nash dengesi 3

Q

Q-learning → Q-öğrenme 13
qualified majority → nitelikli çoğunluk 7
quality score → kalite puanı 6
quantal response equilibrium → kuantal tepki dengesi 12
quasi-linear utility → yarı doğrusal fayda 4
quiescence search → sakinlik araması 2
quota → kota 10

R

random seed → rastgele tohum 14
random serial dictatorship (RSD) → rastgele seri diktatörlük 4
random variable → rastgele değişken 0
rank aggregation → sıralama birleştirme 7
rational player → rasyonel oyuncu 2
rationality → rasyonellik 2
re-auction (re-tendering) → yeniden ihale 10
reactive architecture → reaktif mimari 10
reactivity → reaktivite 10
reciprocity → karşılıklılık 12
reference point → referans noktası 12
regret → pişmanlık 13
regret matching → pişmanlık eşleştirme 13
regret minimization → pişmanlığı en aza indirme 13
regular distribution → düzenli dağılım 4
reinforcement learning → pekiştirmeli öğrenme 13
related machines → ilişkili (farklı hızlı) makineler 9
repeated game → tekrarlanan oyun 13
replicator dynamics → replikatör dinamiği 13
report → bildirim 4
reproducibility → tekrarlanabilirlik 14
reputation system → itibar sistemi 6
reserve price → rezerv fiyat 5
resource allocation → kaynak tahsisi 10
revelation principle → açıklama ilkesi 4
revenue → gelir 5
revenue equivalence theorem → gelir eşdeğerliği teoremi 5
revenue maximization → gelir maksimizasyonu 5
reverse auction → ters açık artırma 10
RIP (Routing Information Protocol) → RIP (uzaklık vektörü protokolü) 8

risk attitude → risk tutumu 1
risk dominance → risk baskınlık 3
risk premium → risk primi 1
risk-averse → riskten kaçınan 1
risk-loving / risk-seeking → risk seven 1
risk-neutral → risk nötr 1
Robertson–Webb query model → Robertson–Webb modeli 11
robust Price of Anarchy → sağlam PoA 9
Rock–Paper–Scissors (RPS) → Taş-Kağıt-Makas (TKM) 2
root (of a game tree) → kök (oyun ağacında) 2
Rosenthal potential → Rosenthal potansiyeli 8
round-robin → sırayla seçim 11
router → yönlendirici 8
routing → yönlendirme 8
row / column player → satır / sütun oyuncusu 2
Rubinstein bargaining model → Rubinstein pazarlık modeli 14

S

saddle point → eyer noktası 3
sample complexity → örneklem karmaşıklığı 14
sandwich attack → sandviç saldırısı 14
SARSA → SARSA 13
satisficing → yeterince iyiyi arama 12
school choice → okul seçimi 4
sealed bid → kapalı zarf 5
second derivative → ikinci türev 0
second-order free riding → ikinci derece beleşçilik 12
second-price (Vickrey) auction → ikinci fiyat (Vickrey) açık artırması 5
security game → güvenlik oyunu 3
seeding → tohumlama 6
selection rate → seçim oranı 11
self-fulfilling expectation → kendini doğrulayan beklenti 6
self-interest → kendi çıkarı 2
self-play → kendine karşı oyun 13
self-preferencing → kendi ürününü kayırma 6
selfish routing → bencil yönlendirme 8
sensitivity (true positive rate) → duyarlılık 1
sequential game → sıralı oyun 2
sequential pairwise voting → ikili elemeli oylama 7
sequential updating → ardışık güncelleme 1
serial dictatorship → seri diktatörlük 4
series-parallel network → seri-paralel ağ 8
shadow of the future → geleceğin gölgesi 13
SHAP (SHapley Additive exPlanations) → SHAP 10
Shapley value → Shapley değeri 10
Shapley–Shubik index → Shapley–Shubik endeksi 10
simple majority → basit çoğunluk 7
simultaneous-move game → eşzamanlı oyun 2
single point of failure → tek nokta hatası 8
single-parameter environment → tek parametreli ortam 4
single-peaked preferences → tek tepeli tercihler 7
slashing → teminattan kesme 14
slippage tolerance → kayma toleransı 14
slot (ad position) → yuva 6
smart contract → akıllı sözleşme 14
smoothness, (λ, μ)-smooth game → pürüzsüzlük, (λ, μ)-pürüzsüz oyun 9
social ability → sosyal yetenek 10
social choice → sosyal seçim 7
social choice function (SCF) → sosyal seçim fonksiyonu 4
social cost → sosyal maliyet 8
social optimum → sosyal optimum 9
social welfare → sosyal refah 4
social welfare function (SWF) → sosyal refah fonksiyonu 7
softmax choice → softmax seçimi 13
software-defined networking (SDN) → yazılım tanımlı ağ (SDN) 8
spectrum auction → spektrum açık artırması 5
sponsored search auction → sponsorlu arama açık artırması 6
Sprague–Grundy theorem → Sprague–Grundy teoremi 2
St. Petersburg paradox → St. Petersburg paradoksu 1
stability → kararlılık 4
stable matching → kararlı eşleştirme 4
Stackelberg routing → Stackelberg yönlendirme 8
Stag Hunt → Geyik Avı 2
star / ring / tree / mesh topology → yıldız / halka / ağaç / tam örgü 8
state–action value → durum-eylem değeri 13
stationary strategy → durağan strateji 14
statistical bias → yanlılık 14
status quo bias → statüko yanlılığı 12
strategic classification → stratejik sınıflandırma 11
strategic interaction → stratejik etkileşim 2
strategic voting → stratejik oy 7
strategy → strateji 2
strategy profile → strateji profili 3
strategy set → strateji kümesi 3
strategy-proof → stratejiye dayanıklı 4
strategy-stealing argument → strateji çalma argümanı 2
strict dominance → kesin baskılama 3
strict equilibrium → kesin denge 2
strong (exact) budget balance → tam bütçe dengesi 4
subgame → alt oyun 2
subgame perfect equilibrium (SPE) → alt oyun mükemmel denge 2
submodularity → alt modülerlik 14

subsidy → sübvansiyon 6
superadditive → süperadditif 10
support (of a mixed strategy) → destek 3
support enumeration → destek sayımı 3
surjectivity (onto) → örtenlik 7
surplus (gains from trade) → artık 5
suspicious / generous Tit-for-Tat → şüpheci / cömert TFT 13
switching cost → geçiş maliyeti 6
symmetric game → simetrik oyun 2
symmetry axiom → simetri aksiyomu 10
system optimum (Wardrop's 2nd principle) → sistem optimumu 8

T

tactical voting (plurality) → taktik oy 7
tail probability → kuyruk olasılığı 0
take-it-or-leave-it offer → al ya da bırak teklifi 4
task allocation → görev tahsisi 10
task announcement / bid / award → duyuru / teklif / ödül 10
temperature (sampling) → sıcaklık 14
Thompson sampling → Thompson örneklemesi 13
threshold public good → eşikli kamu malı (eşikli proje) 12
tie-breaking rule → eşitlik bozma kuralı 7
tight bound → sıkı sınır 9
time average → zaman ortalaması 13
tipping → devrilme 6
Tit-for-Tat (TFT) → Tit-for-Tat (TFT) 13
to block (a coalition blocks) → engellemek 10
token → jeton 14
Top Trading Cycles (TTC) → En İyi Takas Döngüleri (TTC) 4
tournament → turnuva 7
trade-off → ödünleşim 4
tragedy of the commons → ortak mallar trajedisi 12
transaction fee mechanism → işlem ücreti mekanizması 14
transferable-utility (TU) game → aktarılabilir faydalı (TU) oyun 10
transitivity → geçişlilik 7
transparency → şeffaflık 11
transposition table → geçiş tablosu 2
trigger strategy → tetik stratejisi 2
true positive rate (TPR) → doğru pozitif oranı 11
trust game → güven oyunu 12
truthful bid(ding) → dürüst teklif 5
truthful reporting → dürüst bildirim 4
truthfulness → dürüstlük 5
two-round (runoff) system → iki turlu sistem 7
two-sided market → iki taraflı pazar 6
type (private information of a player) → tip 4
type profile → tip profili 4

U

Ultimatum game → Últimatom oyunu 2
uniform distribution → düzgün dağılım 0
uniform-price auction → tek tip fiyatlı açık artırma 5
unilateral deviation → tek taraflı sapma 3
unrestricted domain → evrensel alan 7
unstable equilibrium → kararsız denge 6
upper confidence bound (UCB) → üst güven sınırı (UCB) 13
user equilibrium (Wardrop's 1st principle) → kullanıcı dengesi 8
utilitarian criterion → faydacı ölçüt 12
utilitarian welfare → faydacı refah 11
utility → fayda 2
utility function → fayda fonksiyonu 2

V

validator → doğrulayıcı 14
valuation → değerleme 5
valuation function (cake cutting) → değer fonksiyonu 11
value → değer 5
value function (prospect theory) → değer fonksiyonu 12
value of the game → oyunun değeri 2
veil of ignorance → cehalet peçesi 12
veto rule → veto kuralı 7
Vickrey–Clarke–Groves (VCG) mechanism → VCG mekanizması 4
virtual value → sanal değer 5
virtual welfare → sanal refah 5
von Neumann–Morgenstern axioms → von Neumann–Morgenstern aksiyomları 1
vote splitting → oy bölünmesi 7

W

Wardrop equilibrium → Wardrop dengesi 8
water-filling → su doldurma 9
weak budget balance → zayıf bütçe dengesi 4
weak dominance → zayıf baskılama 3
weak equilibrium → zayıf denge 2
weak notion of agency → zayıf ajan tanımı 10
weakly dominant strategy → zayıf baskın strateji 5
weighted set packing → ağırlıklı küme paketleme 14
weighted voting game → ağırlıklı oylama oyunu 10
Wheatstone network → Wheatstone yapısı 8
Win-Stay Lose-Shift (WSLS, Pavlov) → kazan-kal, kaybet-değiş (WSLS) 13
winner determination problem → kazanan belirleme problemi 14
winner's curse → kazananın laneti 5
winner-takes-all → kazanan her şeyi alır 6
wisdom of crowds → kalabalığın bilgeliği 12
worst-case equilibrium → en kötü denge 9

Z

Zermelo's theorem → Zermelo teoremi 2

zero-sum game → sıfır toplamlı oyun 2

E

ε-approximate equilibrium → ε-yaklaşık denge 3

ε-core → ε-çekirdek 10

ε-greedy → ε-açgözlü 13

EK B

Algoritmik Oyun Kuramı İçin Ön Bilgiler

Algoritmik Oyun Kuramı · Ön Bilgiler

Bu sayfa 16 sunumun dayandığı temel bilgileri toplar. Bölüm 0 · Hazırlık sunumu en gerekli çekirdeği anlatır: kümeler, sayma, toplamlar ve geometrik seri, argmax ve türev, olasılık, Bayes kuralı, U[0, 1], beklenen değer, x^TAy, en iyi tepki, kayıtsızlık ve geriye doğru çıkarım. 1–14. bölümler aşağıdaki konuları kullanır. Her konunun kartında ne bilinmesi gerektiği, hangi bölümlerde kullanıldığı ve iki kısa sınama sorusu yer alır. Ayrıntılar **Ayrıntılar ve sınama** düğmesiyle açılır. Sorunun üstüne tıklanınca cevabı görünür.

En gerekli çekirdek. Kümeler ve mantık, beklenen değer, koşullu olasılık, kazanç matrisi okuma, argmax, Büyük-O gösterimi ve temel Python. Diğer konular sunumlarda anlatılır. Yine de önceden bilinirse ders daha hızlı ilerler.

Terimler. TERİM satırı slaytlardaki Türkçe terimleri ve İngilizce karşılıklarını verir. Tam liste ders sözlüğündedir.

Matematik

Kümeler, ispatlar, sayma, toplamlar ve logaritma, türev, dışbükeylik, matrisler ve sabit noktalar.

Kümeler, fonksiyonlar ve mantık

Kartezyen çarpım · profiller · niceleyiciler

TERİM ***strateji kümesi*** (*strategy set*); ***strateji profili*** (*strategy profile*); ***tek taraflı sapma*** (*unilateral deviation*)

Her oyun kümelerden ve fonksiyonlardan oluşan bir çokludur: oyuncular N, strateji kümeleri S_i, kazançlar $u_i : S \to \mathbb{R}$. Denge tanımları “her … için” ve “öyle bir … vardır ki” kalıplarıyla kurulur.

BÖLÜMLER **Bölüm 0** **Bölüm 2** **Bölüm 3** **Bölüm 4** **Bölüm 7**

- Strateji profillerinin kümesi Kartezyen çarpımdır: $S = S_1 \times S_2 \times \cdots \times S_n$. Eleman sayısı $|S_1| \cdot |S_2| \cdot \ldots \cdot |S_n|$ olur.

- s_{-i}, i oyuncusu dışındaki herkesin stratejisini gösterir. Böylece bir profil (s_i, s_{-i}) biçiminde yazılır.
- Niceleyicilerle Nash dengesi: her i oyuncusu ve her $s_i' \in S_i$ için $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}^*)$.
- Değillemede niceleyiciler yer değiştirir: $u_i(s_i', s_{-i}^*) > u_i(s_i^*, s_{-i}^*)$ koşulunu sağlayan bir i oyuncusu ve bir s_i' stratejisi vardır. Bu, kârlı bir sapmadır.
- Bağıntılar: tercihler ikili bağıntılardır ($\succ$, $\succsim$). Oylama kuramı geçişlilik ve tamlık özelliklerine dayanır.

KENDİNİ SINAMA

▶ **Üç oyuncunun sırasıyla 2, 3 ve 4 saf stratejisi var. Kaç saf strateji profili vardır?**

▶ **"Her oyuncu için hiçbir sapma kârlı değildir" önermesinin değili nedir?**

İspat teknikleri

Doğrudan ispat · çelişki · tümevarım · karşı örnek

TERİM ***geriye doğru çıkarım*** (*backward induction*); ***strateji çalma argümanı*** (*strategy-stealing argument*)

Dersteki sonuçların çoğu her sapma, her mekanizma ya da her tercih profili hakkında bir iddiadır. Bu yüzden kısa ispatları okuyup yazabilmek gerekir.

BÖLÜMLER **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 3** **Bölüm 4** **Bölüm 7** **Bölüm 9**

- Doğrudan ispat: hiçbir tek taraflı sapmanın hiçbir oyuncunun kazancını artırmadığı tek tek gösterilir.
- Tümevarım: geriye doğru çıkarım ve Zermelo teoremi sonlu oyun ağacının yapraklarından köke doğru ilerler. Nim stratejileri konum üzerinde tümevarımla kanıtlanır.
- Çelişki: strateji çalma argümanı, Hex oyununda ilk oyuncunun kaybedemeyeceğini gösterir.
- Karşı örnek: genel bir iddiayı çürütmek için küçük bir oyun yeter. Örneğin monotonluğu bozan bir oy profili.
- İmkânsızlık teoremleri (Arrow, Gibbard–Satterthwaite, Myerson–Satterthwaite) bir özellik listesini aynı anda sağlayan hiçbir kural olmadığını söyler.

KENDİNİ SINAMA

▶ **"Her sonlu oyunun saf stratejili bir Nash dengesi vardır" iddiası nasıl çürütülür?**

▶ **Geriye doğru çıkarım hangi ispat tekniğine dayanır?**

n!

Kombinatorik ve permütasyonlar

Çarpma kuralı · n! · alt kümeler · bölüntüler

TERİM ***Shapley değeri*** (*Shapley value*); ***marjinal katkı*** (*marginal contribution*); ***Bell sayısı*** (*Bell number*)

Strateji uzayları, sıralamalar, oyuncu sıraları ve koalisyon yapıları çok hızlı büyür. Sayma, kaba kuvvetle aramanın ne zaman umutsuz olduğunu gösterir.

BÖLÜMLER **Bölüm 0** **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 3** **Bölüm 7** **Bölüm 10**

- n nesnenin n! sıralaması (permütasyonu) vardır. n elemanlı bir kümenin 2^n alt kümesi vardır. $C(n, k) = n! / (k!(n - k)!)$.
- Shapley değeri, i oyuncusunun n! geliş sırası üzerinden marjinal katkılarının ortalamasıdır. Eşdeğer biçim: $\varphi_i = \Sigma\ (S \subseteq N\backslash\{i\})\ |S|!(n - |S| - 1)!/n! \cdot [v(S \cup \{i\}) - v(S)]$. Bu toplam 2^{n-1} koalisyon üzerinden alınır.
- m aday için m! kesin sıralama vardır. Kemeny kuralının çok büyük bir uzayda arama yapması bundandır.
- n oyuncunun koalisyon yapılarının (küme bölüntülerinin) sayısı Bell sayısı B_n'dir: $B_3 = 5$, $B_4 = 15$, $B_{10} = 115.975$.

KENDİNİ SINAMA

▶ **4 oyunculu bir oyunda Shapley değerini tam hesaplamak için kaç geliş sırasının ortalaması alınır?**

▶ **n = 3 iken formül |S| = 1 olan bir S koalisyonuna hangi ağırlığı verir?**

ln

Toplamlar, logaritma ve üstel fonksiyon

Geometrik seri · H_n · ln · e^x

TERİM ***iskonto faktörü*** (*discount factor*); ***harmonik sayı H_n*** (*harmonic number*); ***öğrenme oranı*** (*learning rate*)

İskontolu tekrarlanan oyunlar, İstikrarın Bedeli, çarpımsal ağırlıklar ve diferansiyel mahremiyet geometrik toplamlar, logaritma ve üstel fonksiyonla yazılır.

BÖLÜMLER **Bölüm 0** **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 9** **Bölüm 13** **Bölüm 14**

- Geometrik seri: $0 \le \delta < 1$ için $\Sigma_{t\ge 0}\ \delta^t = 1/(1-\delta)$. Her turda c kazancının toplam değeri bu yüzden $c/(1-\delta)$ olur.
- Harmonik sayı $H_n = 1 + 1/2 + \ldots + 1/n \approx \ln n + 0{,}577$. Adil maliyet paylaşımı oyunlarında İstikrarın Bedeli (PoS) bu sayıyla sınırlıdır: $\text{PoS} \le H_n$.
- Logaritma kuralları: $\ln(ab) = \ln a + \ln b$, $\ln(a^k) = k \ln a$. Logaritmik büyüme çok yavaştır ($\ln 10^6 \approx 13{,}8$).
- Üstel fonksiyon: her x için $e^x \ge 1 + x$; küçük ε için $e^\varepsilon \approx 1 + \varepsilon$. Diferansiyel mahremiyet güvenceleri e^ε çarpanlarıyla ifade edilir.
- Çarpımsal ağırlıklar güncellemesi $w_i \leftarrow w_i \cdot e^{\eta \cdot g_i}$ biçimindedir. Üsteki kazançları toplamak ağırlıkları çarpmak demektir.

KENDİNİ SINAMA

▶ **R = 3, T = 5, P = 1 olan Mahkûmun İkilemi'nde kinci tetik stratejisiyle işbirliği hangi δ değerlerinde sürer?**

▶ **H_4 kaçtır?**

f' Türev ve birinci mertebe koşulları

Türev · maksimum · integral

TERİM ***maksimize etmek*** (*maximize*); ***minimize etmek (en aza indirmek)*** (*minimize*); ***birinci mertebe koşulu*** (*first-order condition*)

Sürekli stratejiler (miktar, teklif, fiyat, akış) türevi sıfıra eşitleyerek optimize edilir. Ardından bulunan noktanın gerçekten maksimum olduğu kontrol edilir.

BÖLÜMLER **Bölüm 0** **Bölüm 1** **Bölüm 3** **Bölüm 5** **Bölüm 6** **Bölüm 8**

- Birinci mertebe koşulu: türevlenebilir bir f'nin iç maksimumunda $f'(x) = 0$ olur. İkinci türeve ($f'' < 0$) ya da uç noktalara da bakılır, çünkü $f' = 0$ bir minimumu da gösterebilir.
- Çarpım kuralı $(fg)' = f'g + fg'$ ve kuvvet kuralı $(x^k)' = k\,x^{k-1}$ dersteki türetmeler için yeterlidir.

- Cournot: her firma $q_i \cdot P(q_1 + q_2)$ − maliyet kârını maksimize eder. Türev alınınca en iyi tepki fonksiyonu çıkar.
- Birikmiş maliyet olarak integral: Beckmann potansiyeli $\Phi(f) = \Sigma_e \int_0^{f_e} \ell_e(t)\,dt$. Bu potansiyelin f_e'ye göre türevi $\ell_e(f_e)$ gecikmesidir.

Örnek (Bölüm 5). n teklifçi var ve rakipler $\beta \cdot v_j$ teklif ediyor, $v_j \sim U[0, 1]$. Bu durumda b teklifi $(b/\beta)^{n-1}$ olasılıkla kazanır. $(v - b)(b/\beta)^{n-1}$ maksimize edilirken türev $(n - 1)(v - b) = b$ olduğunda sıfırlanır. Sonuç $b = (n - 1)v/n$ olur.

KENDİNİ SINAMA

▶ **$q \geq 0$ için $f(q) = q(10 - q)$ fonksiyonunun maksimumu nedir?**

▶ **$\int_0^x t\,dt$ nedir ve türevi nedir?**

Dışbükey ve içbükey fonksiyonlar

Dışbükey kümeler · Jensen · risk tutumları

TERİM ***dışbükey*** (*convex*); ***içbükey*** (*concave*); ***Jensen eşitsizliği*** (*Jensen's inequality*)

Dışbükeylik optimizasyonu kolaylaştırır: yerel optimum aynı zamanda genel optimumdur. Riskten kaçınma da Jensen eşitsizliğiyle açıklanır.

BÖLÜMLER **Bölüm 1** **Bölüm 3** **Bölüm 8** **Bölüm 12**

- Her $\lambda \in [0, 1]$ için $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$ ise f dışbükeydir. İki kez türevlenebilir f için bu koşul $f'' \geq 0$ demektir. $-f$ dışbükeyse f içbükeydir.
- Dışbükey bir fonksiyonun yerel minimumu genel minimumdur. Dışbükey Beckmann potansiyelinin Wardrop dengesini iyi tanımlaması bu yüzdendir.
- Bir küme, herhangi iki noktası arasındaki doğru parçasını içeriyorsa dışbükeydir. Karışık stratejiler kümesi (simpleks) dışbükeydir. Kakutani teoremi bu özelliğe dayanır.
- Jensen eşitsizliği: f dışbükeyse $f(E[X]) \leq E[f(X)]$; u içbükeyse $E[u(X)] \leq u(E[X])$. Bu yüzden içbükey fayda riskten kaçınma anlamına gelir.

KENDİNİ SINAMA

▶ **$u(x) = \sqrt{x}$. Bir piyango 1/2'şer olasılıkla 0 ya da 900 ödüyor. E[u] ve kesinlik eşdeğeri kaçtır?**

▶ **$f(x) = x^2$ dışbükey mi, içbükey mi, hiçbiri mi?**

A Vektörler ve kazanç matrisleri

Matris indisleri · olasılık vektörleri · x^TAy

TERİM ***kazanç matrisi*** (*payoff matrix*); ***karışık strateji*** (*mixed strategy*); ***destek*** (*support*)

İki oyunculu sonlu oyunlar bir matris çiftidir. Karışık stratejiler olasılık vektörleridir. Beklenen kazançlar matris–vektör çarpımıyla bulunur.

BÖLÜMLER **Bölüm 0** **Bölüm 2** **Bölüm 3** **Bölüm 13**

- İki matrisli oyun (A, B): i satırı j sütunuyla karşılaşınca satır oyuncusu A_{ij}, sütun oyuncusu B_{ij} kazanır. Sıfır toplamlı oyunda $B = -A$ olur.
- Karışık strateji $\Delta = \{ x : x_i \geq 0, \Sigma x_i = 1 \}$ simpleksindeki bir vektördür. Desteği, $x_i > 0$ olan eylemlerin kümesidir.
- Beklenen kazançlar: $(Ay)_i$, satır oyuncusunun y'ye karşı saf i eyleminden aldığı kazançtır. x^TAy ise x'in y'ye karşı kazancıdır.
- Destek sayımı küçük doğrusal denklem sistemleri çözer: destekteki kazançlar eşittir ve olasılıkların toplamı 1'dir.

KENDİNİ SINAMA

▶ **A = [[3, 0], [5, 1]] (satırlar: işbirliği, ihanet). x = (1/2, 1/2) ve y = (1, 0) için x^TAy kaçtır?**

▶ **Aynı matriste D satırı C satırını baskılar mı?**

x* Sabit noktalar (Brouwer ve Kakutani)

x = f(x) · varlık · eşlemeler

TERİM ***sabit nokta teoremi*** (*fixed-point theorem*); ***Nash'in varlık teoremi*** (*Nash's existence theorem*); ***en iyi tepki eşlemesi*** (*best-response correspondence*)

Nash'in varlık teoremi bir sabit nokta teoremidir. Denge, kendisine en iyi tepki olan bir profildir.

BÖLÜMLER **Bölüm 3** **Bölüm 6** **Bölüm 13**

- f'nin sabit noktası $f(x) = x$ koşulunu sağlayan noktadır. Küme değerli bir F eşlemesi için bu koşul $x \in F(x)$ olur.
- Brouwer: boş olmayan, kompakt (kapalı ve sınırlı) ve dışbükey bir $K \subset \mathbb{R}^n$ kümesini kendine götüren her sürekli f'nin bir sabit noktası vardır.

- Kakutani: F(x) değerleri boş olmayan ve dışbükey, grafiği kapalı olan bir F eşlemesi için de aynı sonuç geçerlidir. Nash bu teoremi karışık profiller üzerindeki en iyi tepki eşlemesine uygular: $\sigma^* \in BR(\sigma^*)$.
- Bu teoremler varlığı kanıtlar ama verimli bir algoritma vermez. PPAD sınıfı tam olarak bu boşluğu ölçer.
- Bölüm 6'daki benimseme modeli ya da Bölüm 13'teki öğrenme kuralları gibi dinamikler, güncelleme dönüşümlerinin sabit noktalarında durur.

KENDİNİ SINAMA

▶ **[0, 1] üzerinde f(x) = 1 − x fonksiyonunun sabit noktası nedir?**

▶ **Açık (0, 1) aralığında f(x) = x/2 fonksiyonunun sabit noktası neden yoktur? Brouwer teoremi buna engel değil mi?**

Olasılık ve istatistik

Dağılımlar, beklenen değer, Bayes kuralı, sıra istatistikleri, örnekleme ve Markov modelleri.

P

Olasılık ve dağılımlar

Olaylar · bağımsızlık · dağılım fonksiyonu · düzgün dağılım

TERİM ***düzgün dağılım*** (*uniform distribution*); ***ortak ön olasılık*** (*common prior*); ***bağımsız özel değerler*** (*independent private values*, IPV)

Karışık stratejiler, özel tipler, teklifçi değerleri ve model hataları rastgele değişkenlerdir. Dağılımları okumak ve basit olasılıkları hesaplamak gerekir.

BÖLÜMLER **Bölüm 0** **Bölüm 1** **Bölüm 3** **Bölüm 4** **Bölüm 5** **Bölüm 6** **Bölüm 12**

- Kesikli dağılım: $p_i \geq 0$ olasılıkları ve $\Sigma p_i = 1$. Sürekli dağılım: $f \geq 0$ yoğunluğu ve $F(x) = P(X \leq x)$ dağılım fonksiyonu.
- Düzgün dağılım U[0, 1]: [0, 1] üzerinde $F(x) = x$ ve $f(x) = 1$. $E[X] = 1/2$, $Var(X) = 1/12$.
- Bağımsızlık: $P(A \cap B) = P(A) \cdot P(B)$. Bağımsız özel değerler (IPV), her teklifçinin değerinin bağımsız çekildiği anlamına gelir.
- Açık artırmalardaki temel adım: k bağımsız U[0, 1] değer için $P(\text{hepsi} \leq b) = b^k$. Bu, değerini teklif eden k rakibi geçme olasılığıdır.

KENDİNİ SINAMA

▶ **X ~ U[0, 1]. P(X ≤ 0,3) kaçtır?**

▶ **İki bağımsız U[0, 1] değerin ikisinin de en çok 0,5 olma olasılığı nedir?**

E

Beklenen değer ve varyans

Doğrusallık · E[f(X)] · yayılım

TERİM ***beklenen değer*** (*expected value*); ***beklenen kazanç*** (*expected payoff*); ***piyango*** (*lottery*)

Karışık strateji kazançları, beklenen gelir, haydut ödülleri ve karar ağaçları birer beklenen değerdir.

BÖLÜMLER **Bölüm 0** **Bölüm 1** **Bölüm 2** **Bölüm 3** **Bölüm 5** **Bölüm 12** **Bölüm 13**

- $E[X] = \Sigma\, x \cdot P(X = x)$ (kesikli) ya da $\int x\, f(x)\, dx$ (sürekli).
- Doğrusallık: $E[aX + bY] = aE[X] + bE[Y]$. X ile Y bağımlı olsa bile geçerlidir.
- f doğrusal değilse genelde $E[f(X)] \neq f(E[X])$ olur (dışbükeylik kartındaki Jensen eşitsizliği).
- $\mathrm{Var}(X) = E[X^2] - (E[X])^2$ yayılımı ölçer. Ortalaması aynı iki piyangonun riski çok farklı olabilir.
- Beklenen kazanç en olası kazanç değildir: 0,3 olasılıkla 100 ödeyen bir piyango hiçbir zaman 30 ödemez.

KENDİNİ SINAMA

▶ **Bir piyango 0,3 olasılıkla 100, aksi hâlde 0 ödüyor. E[X] ve Var(X) kaçtır?**

▶ **E[X + Y] = E[X] + E[Y] eşitliği bağımsızlık gerektirir mi?**

P|

Koşullu olasılık ve Bayes kuralı

Ön olasılık · olabilirlik · son olasılık · taban oranları

TERİM ***Bayes kuralı*** (*Bayes' rule*); ***ön olasılık*** (*prior*); ***son olasılık*** (*posterior*); ***taban oranı*** (*base rate*)

Bayesçi oyunlar, itibar sistemleri, adalet ölçütleri ve karar analizi inançları koşullu olasılıklarla günceller.

BÖLÜMLER **Bölüm 0** **Bölüm 1** **Bölüm 3** **Bölüm 4** **Bölüm 6** **Bölüm 11**

- $P(B) > 0$ için $P(A \mid B) = P(A \cap B) / P(B)$.
- Bayes kuralı: $P(H \mid E) = P(E \mid H) \cdot P(H) / P(E)$. Payda toplam olasılık kuralıyla bulunur: $P(E) = P(E \mid H)P(H) + P(E \mid \neg H)P(\neg H)$.
- Adalet oranları koşullu olasılıklardır: doğru pozitif oranı $TPR = P(\hat{Y} = 1 \mid Y = 1)$, yanlış pozitif oranı $FPR = P(\hat{Y} = 1 \mid Y = 0)$, kesinlik $PPV = P(Y = 1 \mid \hat{Y} = 1)$.
- Taban oranını göz ardı etmek: $P(H)$ hesaba katılmazsa doğru bir testin pozitif sonucu olduğundan çok daha kesin görünür.
- Ardışık güncelleme: bugünün son olasılığı yarının ön olasılığıdır.

KENDİNİ SINAMA

▶ **Yaygınlık %1, duyarlılık %90, yanlış pozitif oranı %5. P(hasta | pozitif) kaçtır?**

▶ **Bağımsız ikinci bir test de pozitif çıkıyor. Güncellenmiş olasılık kaçtır?**

$v_{(1)}$

Açık artırmalar için sıra istatistikleri

En yüksek · ikinci en yüksek · gelir

TERİM ***gelir eşdeğerliği teoremi*** (*revenue equivalence theorem*); ***kazananın laneti*** (*winner's curse*); ***sanal değer*** (*virtual value*)

Açık artırma geliri en yüksek ya da ikinci en yüksek teklifin beklenen değeridir. Bu yüzden n rastgele değerin k'nci en büyüğünün beklenen değeri Bölüm 5'in her yerinde karşımıza çıkar.

BÖLÜMLER **Bölüm 5** **Bölüm 14**

- n bağımsız U[0, 1] değer için k'nci en yüksek değerin beklentisi $(n + 1 - k)/(n + 1)$ olur. En yüksek değerin beklentisi $n/(n + 1)$, ikinci en yüksek değerinki $(n - 1)/(n + 1)$'dir.
- İkinci fiyat açık artırmasının geliri = E[ikinci en yüksek değer] = $(n - 1)/(n + 1)$.
- $b = (n - 1)v/n$ teklifiyle birinci fiyat açık artırmasının geliri = $(n - 1)/n \cdot n/(n + 1) = (n - 1)/(n + 1)$. Gelir eşdeğerliğinin öngördüğü gibi sonuç aynıdır.
- Kazananın laneti: tahmin hataları $U[-a, a]$ dağılımındaysa n hatanın en büyüğünün beklentisi $a(n - 1)/(n + 1)$'dir. Kazanan, değeri en çok abartan teklifçidir.
- Sanal değer $\varphi(v) = v - (1 - F(v))/f(v)$. $U[0, 1]$ için $\varphi(v) = 2v - 1$ olur ve en iyi rezerv fiyat 1/2 çıkar.

KENDİNİ SINAMA

▶ **Değerleri U[0, 1] olan iki teklifçide ikinci fiyat açık artırmasının beklenen geliri nedir?**

▶ **4 teklifçide kazananın beklenen değeri nedir?**

$\bar{x}$ Örnekleme, oranlar ve simülasyon

Örneklem ortalaması · hata · karışıklık matrisi

TERİM ***karışıklık matrisi*** (*confusion matrix*); ***doğru pozitif oranı*** (*true positive rate*, TPR); ***örneklem karmaşıklığı*** (*sample complexity*)

Simülasyonlar, haydut tahminleri, adalet denetimleri ve öğrenilen mekanizmalar gürültülü örneklerden büyüklük tahmin eder.

BÖLÜMLER **Bölüm 5** **Bölüm 11** **Bölüm 13** **Bölüm 14**

- Büyük sayılar yasası: N bağımsız çekilişin örneklem ortalaması gerçek ortalamaya yaklaşır. Standart hatası $\sigma/\sqrt{N}$'dir.
- Karışıklık matrisi oranları: TPR = TP/(TP + FN), FPR = FP/(FP + TN), PPV = TP/(TP + FP), seçim oranı = (TP + FP)/toplam.
- Bir simülasyonda oyunların görülme sıklıkları karışık stratejileri tahmin eder. Küçük örneklerde bu sıklıklar çok dalgalanır.
- Örneklem karmaşıklığı, ε doğruluğunda tahmin için kaç örnek gerektiğini sorar. Örnek sayısı azsa aşırı uyum ortaya çıkar.

KENDİNİ SINAMA

▶ **Standart hatayı yarıya indirmek için kaç kat örnek gerekir?**

▶ **TP = 40, FN = 10, FP = 10, TN = 40. TPR ve FPR kaçtır?**

γ Markov zincirleri ve Markov karar süreçleri

Durumlar · geçişler · ödüller · iskonto

TERİM ***politika*** (*policy*); ***durum-eylem değeri*** (*state–action value*); ***iskonto faktörü***

Q-öğrenme ve tekrarlanan oyunlar uzun ödül dizilerini değerlendirir. Bunların arkasındaki model Markov karar sürecidir.

BÖLÜMLER **Bölüm 2** **Bölüm 13**

- Markov özelliği: sonraki durum geçmişin tamamına değil, yalnız şimdiki duruma (ve eyleme) bağlıdır.
- Markov karar süreci (*Markov decision process*, MDP) şu öğelerden oluşur: durumlar S, eylemler A, geçiş olasılıkları $P(s' \mid s, a)$, ödüller r ve iskonto $\gamma \in [0, 1)$. Amaç $E[\Sigma\, \gamma^t\, r_t]$ değerini maksimize etmektir.
- Politika durumları eylemlere eşler. Açgözlü politika $\text{argmax}_a\, Q(s, a)$ eylemini seçer.
- Q-öğrenme güncellemesi: $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a)]$. Burada α öğrenme oranıdır.

KENDİNİ SINAMA

▶ **$Q(s, a) = 2$; $r = 1$; $\gamma = 0{,}9$; $\max Q(s', \cdot) = 4$; $\alpha = 0{,}5$. Yeni $Q(s, a)$ kaçtır?**

▶ **Her adımda sonsuza dek 1 ödül alınıyor ve $\gamma = 0{,}9$. İskontolu toplam getiri kaçtır?**

Optimizasyon

argmax, doğrusal programlama ve dualite. En iyi tepkilerin, refahın ve minimaksın dili budur.

arg

Optimizasyonun temelleri ve argmax

Amaç · kısıtlar · max ve argmax

TERİM ***optimizasyon*** (*optimization*); ***sosyal refah*** (*social welfare*); ***sosyal optimum*** (*social optimum*)

En iyi tepkiler, VCG tahsisleri ve sosyal optimumlar aynı işi yapar: uygun seçenekler arasından amacı en büyük olanı seçer.

BÖLÜMLER **Bölüm 0** **Bölüm 3** **Bölüm 4** **Bölüm 5** **Bölüm 9** **Bölüm 10** **Bölüm 12**

- Bir optimizasyon probleminde karar değişkenleri, maksimize ya da minimize edilecek bir amaç ve uygun kümeyi tanımlayan kısıtlar bulunur.
- max f bir değerdir. argmax f ise maksimum noktalarının kümesidir. Eşitlik durumunda argmax birden çok nokta içerebilir. Mekanizmaların bu yüzden bir eşitlik bozma kuralına ihtiyacı vardır.
- En iyi tepki: $BR_i(s_{-i}) = \text{argmax}$ (s_i üzerinden) $u_i(s_i, s_{-i})$. VCG, $x^* = \text{argmax}_x\, \Sigma_i\, b_i(x)$ tahsisini seçer.

- OPT en iyi amaç değerini gösterir. Anarşinin Bedeli bir dengenin maliyetini OPT ile karşılaştırır.

KENDİNİ SINAMA

▶ **[0, 5] üzerinde $f(x) = -(x - 2)^2$ için max f ve argmax f nedir?**

▶ **Kazançlar a: 3, b: 5, c: 5. argmax nedir?**

LP

Doğrusal programlama ve dualite

Uygun çokyüzlü · dual · minimaks

TERİM ***doğrusal programlama*** (*linear programming*, LP); ***minimaks teoremi*** (*minimax theorem*); ***oyunun değeri*** (*value of the game*); ***eyer noktası*** (*saddle point*)

Sıfır toplamlı oyunlar, korelasyonlu dengeler ve bazı mekanizma tasarımı problemleri doğrusal programla çözülür. Maksimin değerinin minimaks değerine eşit olmasının nedeni LP dualitesidir.

BÖLÜMLER **Bölüm 3** **Bölüm 13** **Bölüm 14**

- Bir LP, doğrusal eşitsizlik kısıtları altında doğrusal bir amacı maksimize ya da minimize eder. Uygun küme dışbükey bir çokyüzlüdür. Optimum varsa bir köşede elde edilir.
- LP'ler polinom zamanda çözülür (elipsoit ve iç nokta yöntemleri). Simpleks yöntemi pratikte hızlıdır ama en kötü durumda üsteldir.
- Dualite: her maksimizasyon LP'sinin bir minimizasyon duali vardır. Zayıf dualite: birincil değer ≤ dual değer. Güçlü dualite: optimumda ikisi eşittir.
- Sıfır toplamlı oyunda satır oyuncusunun LP'si: her j sütunu için $\Sigma_i\, x_i A_{ij} \geq v$, $\Sigma x_i = 1$ ve $x \geq 0$ kısıtları altında v maksimize edilir. Sütun oyuncusunun LP'si bunun dualidir. Dolayısıyla max min = min max olur (von Neumann'ın minimaks teoremi).
- Korelasyonlu ve kaba korelasyonlu dengeler de LP uygunluk problemleridir. Bu yüzden polinom zamanda hesaplanırlar.

KENDİNİ SINAMA

▶ **A = [[3, 1], [4, 2]] (satır oyuncusu maksimize eder). Maksimin, minimaks ve eyer noktası nedir?**

▶ **Yazı-Tura Eşleştirme oyununda (kazançlar ±1) oyunun değeri ve en iyi strateji nedir?**

Algoritmalar ve karmaşıklık

Çalışma süresi, algoritma tasarımı, XOR, çizgeler ve P, NP, PPAD, PLS sınıfları.

O Büyük-O ve çalışma süresi

Büyüme hızları · polinom ve üstel

TERİM ***açgözlü*** (*greedy*); ***alfa-beta budaması*** (*alpha–beta pruning*)

Algoritmik oyun kuramı bir sonucun var olup olmadığıyla yetinmez. O sonucun hızlı hesaplanıp hesaplanamayacağını da sorar.

BÖLÜMLER **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 4** **Bölüm 8** **Bölüm 9** **Bölüm 10**

- Yaygın büyüme hızları: $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$. O bir üst sınırdır, Θ ise sıkı sınırdır.
- Polinom zaman ($O(n^k)$), "verimli" sözcüğünün çalışma tanımıdır.
- Dallanma etkeni b ve derinliği d olan bir oyun ağacının yaklaşık b^d yaprağı vardır. Hamleler en iyi sırayla incelenirse alfa-beta budaması yaklaşık $b^{d/2}$ yaprağa bakar.
- Dersteki örnekler: Gale–Shapley $O(n^2)$, Macar algoritması $O(n^3)$, ikili yığınla Dijkstra $O((V + E) \log V)$.

KENDİNİ SINAMA

▶ **10 robotu 10 göreve atamak: kaba kuvvet ile Macar algoritması nasıl karşılaştırılır?**

▶ **O(n log n) polinom mudur?**

⟳ Algoritma tasarımı ve adım adım izleme

Özyineleme · açgözlü · dinamik programlama

TERİM ***geriye doğru çıkarım***; ***Gale–Shapley (ertelenmiş kabul) algoritması*** (*deferred acceptance*); ***açgözlü***

Birçok sunumda adım adım kod izlemeleri vardır. Bir algoritmayı izleyebilmek, girdisini ve çıktısını söyleyebilmek, durduğunu ve doğru çalıştığını gerekçelendirebilmek gerekir.

BÖLÜMLER **Bölüm 0** **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 4** **Bölüm 10**

- Özyineleme: minimaks bir düğümü çocuklarının değerlerinden hesaplar. Geriye doğru çıkarım da oyun ağacı üzerinde aynı fikirdir.
- Dinamik programlama çözülmüş alt problemleri saklar. Örneğin bir çıkarma oyununun kazanan ve kaybeden konumlarını.
- Açgözlü algoritmalar her adımda yerel olarak en iyi seçimi yapar. Hızlıdır ama her zaman optimum vermez. Macar algoritması ise optimum atamayı bulur.
- Gale–Shapley (ertelenmiş kabul): teklif edenler tercih sırasıyla teklif eder, alıcılar en iyi teklifi bekletir. Algoritma, teklif eden taraf için en iyi olan kararlı bir eşleştirmeyle biter.

KENDİNİ SINAMA

▶ **Gale–Shapley neden sonlanır?**

▶ **Bir MAX düğümünün altındaki minimaks yaprakları 3, 7 ve 2. Düğümün değeri kaçtır?**

b^d

Ağaçlar, özyineleme ve üstel büyüme

Kök · yaprak · derinlik · dallanma etkeni · $\log_b$

TERİM ***oyun ağacı*** (*game tree*); ***dallanma etkeni*** (*branching factor*); ***derinlik öncelikli arama*** (*depth-first search*); ***yinelemeli derinleştirme*** (*iterative deepening*)

Minimaks ve alfa-beta, oyun ağacını özyinelemeli olarak, derinlik öncelikli sırayla gezer. Ağacın boyu dallanma etkeni ve derinlikle üstel büyür. Belirli bir sürede ne kadar derine inilebileceği logaritmayla hesaplanır.

BÖLÜMLER **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

- Ağaç terimleri: kök başlangıç konumudur, yapraklar oyun sonları ya da derinlik sınırındaki konumlardır. Bir düğümün derinliği kökten uzaklığıdır. Oyun ağacında katmanlar sırayla MAX ve MIN oyuncusuna aittir.
- Her iç düğümün b çocuğu varsa d derinlikte b^d yaprak, toplamda $1 + b + \dots + b^d = (b^{d+1} - 1)/(b - 1)$ düğüm vardır. Düğümlerin büyük çoğunluğu son katmandadır: b = 10 için yaklaşık %90'ı.
- Özyinelemeli gezinmede bir düğümün değeri çocuklarının değerlerinden hesaplanır. Bellekte yalnızca kökten o anki düğüme giden yol durur. Bu yüzden süre $O(b^d)$ iken bellek $O(b \cdot d)$ kadardır.

- Logaritma: N yapraklık bir bütçeyle inilebilecek derinlik $d = \log_b N = \ln N / \ln b$ olur. $b^{d/2} = (\sqrt{b})^d$ olduğundan en iyi hamle sırasında alfa-beta aynı bütçeyle yaklaşık iki kat derine iner.
- Yinelemeli derinleştirme aramayı $d = 1, 2, 3, \ldots$ için tekrarlar. Önceki turların toplam maliyeti son turun yalnızca yaklaşık $1/(b - 1)$ katıdır.

KENDİNİ SINAMA

▶ **$b = 3$ ve $d = 2$ olan düzgün bir ağaçta kaç düğüm ve kaç yaprak vardır?**

▶ **Satrançta $b \approx 35$ ve program saniyede 10^6 yaprak değerlendiriyor. 10 saniyede kaç katman derine inilebilir?**

⊕ İkili sayılar ve XOR

Bit işlemleri · Nim

TERİM ***nim toplamı*** (*nim-sum*); ***tarafsız oyun*** (*impartial game*); ***Sprague–Grundy teoremi*** (*Sprague–Grundy theorem*)

Nim, yığın boyutlarının bit düzeyinde XOR'u ile tam olarak çözülür. Sprague–Grundy kuramı bu sonucu diğer tarafsız oyunlara genişletir.

BÖLÜMLER **Bölüm 2** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

- XOR ($\oplus$) ikili basamakları eldesiz toplar: sonucun bir biti, o konumda 1 olan girdi sayısı tek ise 1 olur.
- $x \oplus x = 0$ ve $x \oplus 0 = x$. XOR değişmeli ve birleşmelidir.
- Nim (normal oyun): yığınların nim toplamı 0 ise hamle sırası gelen oyuncu, iki taraf da en iyi oynadığında kaybeder.

KENDİNİ SINAMA

▶ **$3 \oplus 4 \oplus 5$ kaçtır?**

▶ **Yığınlar 1, 2 ve 3. En iyi oyunda kim kazanır?**

NP P, NP, PPAD ve PLS

İndirgemeler · zorluk · toplam arama problemleri

TERİM ***NP-zor*** (*NP-hard*); ***PPAD-tam*** (*PPAD-complete*); ***PLS-tam*** (*PLS-complete*); ***ε-yaklaşık denge*** (*ε-approximate equilibrium*)

Karmaşıklık sınıfları, oyun kuramındaki hangi nesnelerin kolay hesaplandığını (sıfır toplamlı oyunların dengeleri, korelasyonlu dengeler), hangilerinin zor sayıldığını (genel Nash dengeleri, optimum tahsisler) açıklar.

BÖLÜMLER **Bölüm 3** **Bölüm 4** **Bölüm 7** **Bölüm 8** **Bölüm 14**

- P: polinom zamanda çözülebilen problemler. NP: önerilen bir çözümün polinom zamanda doğrulanabildiği problemler.
- NP-zor: her NP problemi polinom zamanda buna indirgenir. NP-tam = hem NP-zor hem NP'de. Dersteki örnekler: kombinatoryal açık artırmalarda VCG'nin argmax adımı, Kemeny sıralaması, bazı oylama kurallarını manipüle etmek.
- Nash dengesi bulmak neden NP-tam değildir? Denge her zaman vardır, bu yüzden evet/hayır sorusu önemsizdir. Zorluk dengeyi bulmaktadır.
- PPAD: çözümü bir eşlik (yolun sonu) argümanıyla güvence altında olan arama problemleri. Brouwer, Sperner ve Nash bu sınıftadır. İki oyunculu Nash dengesi PPAD-tamdır (Chen–Deng, 2006; dergi sürümü Chen–Deng–Teng, 2009).
- PLS: yerel arama problemleri. Bir tıkanıklık oyununun saf dengesini bulmak PLS-tamdır.

KENDİNİ SINAMA

▶ **Korelasyonlu denge polinom zamanda hesaplanabilir mi?**

▶ **"Bu oyunun karışık bir Nash dengesi var mı?" sorusu neden ilginç bir NP sorusu değildir?**

G Çizgeler, en kısa yollar ve akışlar

Yönlü çizgeler · döngüler · akışlar · yük

TERİM ***çizge*** (*graph*); ***düğüm*** (*node*); ***kenar*** (*edge*); ***gecikme fonksiyonu*** (*latency function*)

Yönlendirme oyunları, ağ oluşturma, eşleştirme piyasaları, oylama turnuvaları ve böbrek takası birer çizge problemidir.

BÖLÜMLER **Bölüm 4** **Bölüm 7** **Bölüm 8** **Bölüm 9** **Bölüm 10**

- Çizgeler düğümlerden ve (yönlü ya da yönsüz) kenarlardan oluşur. Komşuluk listesi ya da komşuluk matrisiyle saklanır. Temel kavramlar: yol, döngü, bağlantılılık, iki parçalı çizge.
- En kısa yollar: Dijkstra negatif olmayan ağırlıklar ister. Bellman–Ford negatif ağırlıklara izin verir (negatif döngü olmamak koşuluyla).

- Akışlar: s'den t'ye r birimlik akış yollara bölünür. Bir kenarın yükü, o kenarı kullanan yolların akışlarının toplamıdır. Her iç düğümde akış korunur.
- Yönlendirme oyunlarında kenar maliyetleri sabit ağırlık değil, yüke bağlı $\ell_e(x)$ gecikme fonksiyonlarıdır. Tıkanıklığı ve Braess paradoksunu doğuran budur.
- Döngüler başka yerlerde de önemlidir: En İyi Takas Döngüleri (TTC) bir işaret çizgesinin döngülerini çıkarır. Döngü içeren bir çoğunluk çizgesi Condorcet paradoksudur.

KENDİNİ SINAMA

▶ **Pigou ağı: bir birim akış, kenar maliyetleri 1 ve $\ell(x) = x$. Denge maliyeti, en iyi maliyet ve PoA kaçtır?**

▶ **Dijkstra negatif kenar ağırlıklarıyla çalışır mı?**

Stratejik akıl yürütme ve ekonomi

Beklenen fayda, en iyi tepkiler, refah terimleri ve pişmanlık.

E[u]

Beklenen fayda ve karışık stratejiler

Piyangolar · kayıtsızlık · risk tutumu

TERİM ***beklenen fayda*** (*expected utility*); ***kayıtsızlık ilkesi*** (*indifference principle*); ***risk nötr*** (*risk-neutral*); ***kesinlik eşdeğeri (KE)*** (*certainty equivalent*)

Rastlantı varken oyuncu piyangoları $\Sigma\, p(x)\cdot u(x)$ beklenen faydasıyla karşılaştırır. Karışık dengeler bu yolla hesaplanır, risk tutumları da bu yolla modellenir.

BÖLÜMLER **Bölüm 0** **Bölüm 1** **Bölüm 3** **Bölüm 5** **Bölüm 12**

- Bir piyangonun beklenen faydası $\Sigma\, p(x)\cdot u(x)$'tir. Risk nötr oyuncuda $u(x) = x$ olur. Riskten kaçınma içbükey u demektir.
- Kayıtsızlık ilkesi: karışık Nash dengesinde bir oyuncunun desteğindeki her eylem aynı beklenen kazancı getirir. Destek dışındaki eylemler daha fazlasını getirmez.
- Bu yüzden her oyuncunun karıştırma olasılıkları *öteki* oyuncuyu kayıtsız bırakacak biçimde seçilir. Bir oyuncunun kendi kazançları rakibin karışımını belirler.
- Açık artırma sonuçlarının çoğu teklifçilerin risk nötr olduğunu varsayar (Bölüm 5). Risk tutumları ise Bölüm 1'in konusudur.

KENDİNİ SINAMA

▶ **Sütun oyuncusu Sol'u q olasılıkla oynuyor. Satır oyuncusunun kazançları: Üst = 2q, Alt = 1 − q. Satır oyuncusu hangi q değerinde karıştırmaya razıdır?**

▶ **Karışık bir dengede destek dışındaki bir eylem, destekteki eylemlerden kesin olarak fazla kazandırabilir mi?**

BR

En iyi tepkiler ve baskınlık

En iyi tepki · baskınlık · sapma

TERİM ***en iyi tepki*** (*best response*); ***baskın strateji*** (*dominant strategy*); ***baskılanan strateji*** (*dominated strategy*); ***Nash dengesi*** (*Nash equilibrium*, NE)

Dersin temel becerisi şudur: diğerlerinin seçimleri belliyken her oyuncunun en iyi tepkisi nedir ve sapmak isteyen biri var mıdır?

BÖLÜMLER **Bölüm 0** **Bölüm 2** **Bölüm 3** **Bölüm 8** **Bölüm 12**

- En iyi tepki: diğerlerinin eylemleri sabitken oyuncunun kazancını maksimize eden eylem ya da eylemler.
- Kesin baskınlık: s_i her durumda s_i''den kesin olarak iyidir. Kesin baskılanan bir strateji hiçbir Nash dengesinde oynanmaz, pozitif olasılıkla bile.
- Zayıf baskınlık: hiçbir zaman daha kötü değil, bazen kesin olarak daha iyi. Zayıf baskılanan stratejiler yine de dengelerde görülebilir. Bunları elemenin sonucu eleme sırasına bağlı olabilir.
- Her oyuncu diğerlerine en iyi tepkiyi veriyorsa profil bir Nash dengesidir. Bireysel kararlılık sosyal optimumu garanti etmez (Mahkûmun İkilemi).

KENDİNİ SINAMA

▶ **Mahkûmun İkilemi'nde (İşbirliği, İşbirliği) bir Nash dengesi midir?**

▶ **Kesin baskılanan bir strateji karışık bir dengenin desteğinde yer alabilir mi?**

€

Ekonomi terimleri

Değerleme · fayda · refah · gelir

TERİM ***değerleme*** (*valuation*); ***yarı doğrusal fayda*** (*quasi-linear utility*); ***sosyal refah***; ***gelir*** (*revenue*); ***artık*** (*surplus*)

Mekanizma tasarımı ve açık artırmalar küçük bir ekonomi terimleri kümesini tekrar tekrar kullanır. Bu terimleri, özellikle refah ile geliri karıştırmak sık yapılan bir hatadır.

BÖLÜMLER **Bölüm 4** **Bölüm 5** **Bölüm 6** **Bölüm 9**

- Değerleme v_i: bir sonucun i ajanı için değeri. Yarı doğrusal fayda: $u_i = v_i(x) - p_i$, yani değer eksi ödeme.
- Sosyal refah: $\Sigma_i\, v_i(x)$. Ödemeler ajanlarla satıcı arasındaki transferlerdir. Bu yüzden refahta birbirini götürür.
- Gelir: $\Sigma_i\, p_i$, satıcının topladığı tutar. Refah = ajanların toplam faydası + gelir.
- Bir piyasada artık (ticaretten kazanç), gerçekleşen alışverişlerde alıcı değeri ile satıcı maliyetinin farkıdır.
- Dışsallık, bir ajanın eyleminin diğerlerinin kazancına etkisidir. Ajan bu etkinin bedelini ödemez (tıkanıklık, ağ etkileri).

KENDİNİ SINAMA

▶ **Değeri 10 olan bir teklifçi kazanıyor ve 7 ödüyor. Fayda, gelir ve refah kaçtır?**

▶ **Bir mekanizma refahı düşürürken geliri artırabilir mi?**

R

Pişmanlık ve çevrim içi öğrenme

Dışsal pişmanlık · Hedge · haydutlar

TERİM ***pişmanlık*** (*regret*); ***pişmanlıksız*** (*no-regret*); ***çarpımsal ağırlıklar (Hedge)*** (*multiplicative weights*); ***çok kollu haydut*** (*multi-armed bandit*)

Oyunlarda öğrenme, bir algoritmayı pişmanlığıyla ölçer: geriye bakıldığında en iyi tek eyleme göre ne kadar kötü sonuç aldığı.

BÖLÜMLER **Bölüm 1** **Bölüm 9** **Bölüm 13**

- T tur sonunda dışsal pişmanlık: $R_T = \max_a \Sigma_t\, g_t(a) - \Sigma_t\, g_t(a_t)$. $R_T / T \to 0$ ise algoritma pişmanlıksızdır.
- n eylem ve [0, 1] aralığındaki kazançlarla çarpımsal ağırlıklar (Hedge): $R_T \le \ln n/\eta + \eta T$. $\eta = \sqrt{(\ln n / T)}$ seçilirse $R_T \le 2\sqrt{(T \ln n)}$ olur.
- Lideri izle algoritması sürekli yer değiştirmeye zorlanabilir ve doğrusal pişmanlığa uğrar. Rastgeleleştirme bu sorunu giderir.

- Bütün oyuncular pişmanlıksız öğrenme kullanırsa oyunun zaman ortalaması kaba korelasyonlu dengeye yakınsar. Pürüzsüz oyunlardaki PoA sınırları da geçerli kalır.
- Haydut problemlerinde yalnız seçilen eylemin ödülü görülür. UCB, $\sqrt{(2 \ln t / N_a)}$ büyüklüğünde bir keşif payı ekler.
- Bölüm 1'deki minimaks pişmanlık (Savage) aynı sözcüğü taşır ama tek bir karar için kullanılan farklı bir ölçüttür.

KENDİNİ SINAMA

▶ **n = 2 eylem ve T = 100 turda Hedge: toplam ve ortalama pişmanlığın sınırı nedir?**

▶ **Ortalama pişmanlığın 0'a gitmesi, öğrenenin her turda en iyi oynadığı anlamına neden gelmez?**

Programlama

Her sunumda kısa Python programları ve kod izlemeleri bulunur.

py

Python ile programlama

Döngüler · fonksiyonlar · listeler ve sözlükler · random

Her sunumda kısa Python programları ve kod izlemeleri vardır: kazanç tabloları, açık artırmalar, Shapley değerleri, haydutlar, Q-öğrenme. Bu programları okuyup çalıştırabilmek ve değiştirebilmek gerekir.

BÖLÜMLER **Bölüm 1** **Bölüm 2** **Bölüm 3** **Bölüm 4** **Bölüm 5** **Bölüm 6** **Bölüm 7** **Bölüm 8** **Bölüm 9** **Bölüm 10** **Bölüm 11** **Bölüm 12** **Bölüm 13** **Bölüm 14**

- Dilin çekirdeği: değişkenler, döngüler, fonksiyonlar, özyineleme, listeler, demetler, sözlükler ve liste üreteçleri.
- İç içe listelerle kazanç matrisleri (A[i][j]), itertools.product ile strateji profilleri, itertools.permutations ile oyuncu sıraları.
- Rastgelelik: random.random(), random.choice ve tekrarlanabilir simülasyonlar için tohumlu üreteçler (random.Random(3)).
- Matematik fonksiyonları: math.exp, math.log, math.sqrt (çarpımsal ağırlıklar, UCB).

- Deneyler: bir simülasyon çok kez tekrarlanır, sonuçların ortalaması alınır ve kuramsal öngörüyle karşılaştırılır.

KENDİNİ SINAMA

▶ **itertools.product(range(2), repeat=3) kaç demet üretir?**

▶ **itertools.permutations(range(3)) kaç sıra üretir?**

Arama ya da filtrelerle eşleşen konu yok.

▶ Bölüm bölüm ön bilgiler

© Sercan Külcü · Algoritmik Oyun Kuramı

EK C

Algoritmik Oyun Kuramında Önemli İsimler

Algoritmik Oyun Kuramı · İsimler

Derste kullanılan sonuçların arkasındaki matematikçi, iktisatçı ve bilgisayar bilimcilerinin 62 kartta toplanan kısa profilleri. Liste Borda ve Condorcet'den Anarşinin Bedeli ve PPAD'ye uzanır. Her kart, kişinin çalışmalarının geçtiği ders bölümlerine bağlanır; ayrıntılar kartın altındaki düğmeyle açılır.

JB

Jean-Charles de Borda

1733–1799 · Fransız · deniz subayı ve matematikçi, Académie des sciences

Borda sayımı · 1770

Borda sayımını (*Borda count*) önerdi. Bu pozisyonel oylama kuralında her sıralamadaki seçeneklere m−1, m−2, …, 0 puan verilir.

pozisyonel puanlama | sıralı oy pusulası | Condorcet ölçütü

DERS BÖLÜMLERİ **Bölüm 7 · Sosyal Seçim ve Oylama**

Önemi: Borda sayımı klasik puanlama kuralıdır. Condorcet kaybedenini hiçbir zaman seçmez ama Condorcet kazananını seçemeyebilir.

Borda yöntemini 1770'te Akademiye sundu; yöntem 1784'te yayımlandı. Güçlü rakipler listenin sonuna gömülerek kural kolayca manipüle edilebilir. Bu zayıflık daha sonra Gibbard–Satterthwaite teoremiyle kesin bir biçim kazandı.

MC

Marquis de Condorcet

1743–1794 · Fransız · filozof ve matematikçi

Condorcet paradoksu · jüri teoremi · 1785

1785 tarihli *Essai* adlı eserinde ikili çoğunluk oylamasının döngüye girebileceğini gösterdi: A, B'yi yener; B, C'yi yener; C de A'yı yener.

ikili çoğunluk | Condorcet kazananı | jüri teoremi

DERS BÖLÜMLERİ **Bölüm 7 · Sosyal Seçim ve Oylama**

Önemi: Condorcet kazananı ve oylama paradoksu, sosyal seçim kuramının ve Arrow'un imkânsızlık teoreminin çıkış noktasıdır.

Jüri teoremine göre her seçmen birbirinden bağımsız olarak $p > 1/2$ olasılıkla doğru karar veriyorsa seçmen sayısı arttıkça çoğunluk kararının doğru olma olasılığı 1'e yaklaşır. Condorcet, Fransız Devrimi sırasında hapiste öldü.

CB

Charles L. Bouton

1869–1922 · Amerikalı · Harvard University

Nim'in tam çözümü · 1901–1902

"Nim, a game with a complete mathematical theory" (*Annals of Mathematics*, 1901–1902) makalesinde Nim'i tamamen çözdü. Hamle sırası gelen oyuncu, yığın boyutlarının ikili XOR toplamı (nim toplamı) 0 ise en iyi oyunda kaybeder.

Nim · nim toplamı · P ve N konumları

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Önemi: Bouton'un kuralı, minimaksın bütün oyun ağacını gezerek bulduğu sonucu tek bir XOR işlemiyle verir. Kombinatoryal oyun kuramının ilk tam sonuçlarındandır.

Kazandıran hamle, nim toplamını 0 yapan hamledir. Sprague (1935) ve Grundy (1939) bu fikri bütün tarafsız oyunlara genişletti.

ÉB

Émile Borel

1871–1956 · Fransız · Université de Paris (1925'te Deniz Kuvvetleri Bakanı)

Karışık stratejiler · 1921

1921–1927 arasında yayımladığı notlarda strateji oyunlarını karışık (rastgele) stratejilerle (*mixed strategies*) ele aldı.

karışık strateji · sıfır toplamlı oyun · rastgeleleştirme

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller**

Önemi: Borel, simetrik sıfır toplamlı oyunlar için minimaks sorusunu sordu. Genel minimaks teoremini 1928'de von Neumann kanıtladı.

Borel küçük durumları çözdü ama minimaks özelliğinin her oyunda geçerli olduğundan kuşkuluydu. Borel ile von Neumann arasındaki öncelik tartışması 1953'te Econometrica dergisinde yeniden ele alındı.

EZ

Ernst Zermelo

1871–1953 · Alman · Göttingen, Zürih, Freiburg

Satranç üzerine Zermelo teoremi · 1913

Oyun kuramının ilk biçimsel teoremi ona atfedilir (1913). Teoreme göre satrançta ya Beyaz kazanmayı zorlayabilir ya Siyah kazanmayı zorlayabilir ya da iki taraf da en az beraberliği garanti eder.

mükemmel bilgi · belirlenmişlik · geriye doğru çıkarım

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Önemi: Sonuç, geriye doğru çıkarımın (*backward induction*) ve mükemmel bilgili sonlu iki oyunculu oyunların belirlenmişliğinin temelidir.

Zermelo daha çok küme kuramı aksiyomlarıyla (1908) tanınır. Satranç makalesi, ders laboratuvarlarında XOX, Nim ve Dörtlü Bağla için kullanılan oyun ağacı aramasının atasıdır.

AP

Arthur Cecil Pigou

1877–1959 · İngiliz · University of Cambridge

Pigou örneği · dışsallıklar · 1912/1920

Biri tıkanabilen, diğeri sabit gecikmeli iki yol tanımladı. Bencil sürücüler tıkanabilen yolu gereğinden fazla kullanır (*Wealth and Welfare*, 1912; *The Economics of Welfare*, 1920).

dışsallık · Pigou vergisi · tıkanıklık

DERS BÖLÜMLERİ **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** **Bölüm 9 · Anarşinin Bedeli**

Önemi: Pigou örneği bencil yönlendirmenin standart örneğidir. Doğrusal gecikmelerde Anarşinin Bedeli (*price of anarchy*) 4/3'tür. Roughgarden ve Tardos bunun en kötü durum olduğunu kanıtladı.

Pigou, olumsuz dışsallık yaratan etkinliklerin vergilendirilmesini önerdi. Tıkanıklık fiyatlandırması (marjinal maliyet ücreti) bu fikrin ağdaki karşılığıdır ve Wardrop dengesini optimum yapar.

HS

Hugo Steinhaus

1887–1972 · Polonyalı · Lwów ve Wrocław üniversiteleri

Adil pasta kesme · 1948

Adil bölüşümün matematiksel kuramını "The Problem of Fair Division" (Econometrica, 1948) makalesiyle başlattı.

orantılılık · kes-ve-seç · pasta kesme

DERS BÖLÜMLERİ **Bölüm 11 · Algoritmik Adalet ve Etik**

Önemi: Üç oyuncu için orantılı bir yöntem verdi. n oyuncu için Banach–Knaster son azalan (*last diminisher*) protokolünü de duyurdu.

Adil bölüşüm iki tür paylaşım arar: orantılı (herkes kendi değerlemesine göre en az 1/n alır) ya da kıskançsız. Algoritmik kuramı daha sonra Brams, Taylor, Procaccia ve başkaları geliştirdi.

OM

Oskar Morgenstern

1902–1977 · Almanya doğumlu Avusturyalı-Amerikalı · Princeton University

Theory of Games and Economic Behavior · 1944

von Neumann ile birlikte *Theory of Games and Economic Behavior* (1944) kitabını yazdı. Bu kitap oyun kuramını iktisatta ayrı bir alan olarak kurdu.

beklenen fayda · stratejik form · rasyonel seçim

DERS BÖLÜMLERİ **Bölüm 1 · Karar Kuramı** **Bölüm 2 · Oyunlar ve Temsiller**

Önemi: von Neumann–Morgenstern beklenen fayda aksiyomları, oyuncuları beklenen kazancı maksimize eden karar vericiler olarak modellemeyi haklı çıkarır.

Fayda aksiyomları (tamlık, geçişlilik, süreklilik, bağımsızlık) 1947'deki ikinci baskıya eklendi. Allais, Kahneman ve Tversky'nin davranışsal çalışmaları bu aksiyomları daha sonra sınadı.

JvN

John von Neumann

1903–1957 · Macar-Amerikalı · Institute for Advanced Study, Princeton

Minimaks teoremi · 1928

İki oyunculu sıfır toplamlı oyunlar için minimaks teoremini kanıtladı (1928). Morgenstern ile *Theory of Games and Economic Behavior* kitabını yazdı (1944).

sıfır toplamlı oyun · karışık strateji · eyer noktası

DERS BÖLÜMLERİ **Bölüm 1 · Karar Kuramı** **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 3 · Nash Dengesi**

Önemi: Minimaks, stratejik optimizasyonu doğrusal programlamadaki dualiteye ve çekişmeli aramaya bağlar.

Her sonlu iki oyunculu sıfır toplamlı oyunda karışık stratejilerle max-min değeri min-max değerine eşittir. 1944 kitabı işbirlikçi oyunları (kararlı kümeler) tanıttı; 1947 baskısı beklenen fayda aksiyomlarını ekledi.

WV

William Vickrey

1914–1996 · Kanadalı-Amerikalı · Columbia University

İkinci fiyat açık artırması · 1961

"Counterspeculation, Auctions, and Competitive Sealed Tenders" (1961) makalesinde açık artırmaları inceledi. Kapalı zarf ikinci fiyat açık artırmasında kendi değerini teklif etmenin baskın strateji (*dominant strategy*) olduğunu gösterdi.

Ödüller: Nobel Ekonomi Ödülü 1996 (James Mirrlees ile); ödülün açıklanmasından üç gün sonra öldü

ikinci fiyat açık artırması · baskın strateji · gelir eşdeğerliği

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 5 · Açık Artırmalar** **Bölüm 6 · Çevrim İçi Platformlar**

Önemi: Vickrey açık artırması ilk VCG mekanizmasıdır ve dürüst mekanizma tasarımının şablonudur.

Aynı makale, standart açık artırma biçimlerinin gelir eşdeğerliğini önceden sezdi. Vickrey yollar ve toplu taşıma için tıkanıklık fiyatlandırmasını da savundu.

HS

Herbert A. Simon

1916–2001 · Amerikalı · Carnegie Mellon University

Sınırlı rasyonellik · yeterince iyiyi arama · 1955

Sınırlı rasyonellik (*bounded rationality*) kavramını getirdi. Gerçek karar vericilerin bilgisi ve hesaplama gücü sınırlıdır. Bu yüzden en iyiyi değil, yeterince iyi olanı ararlar (*satisficing*).

Ödüller: Turing Award 1975 (Allen Newell ile); Nobel Ekonomi Ödülü 1978

sınırlı rasyonellik · yeterince iyiyi arama · sezgisel kurallar

DERS BÖLÜMLERİ **Bölüm 3 · Nash Dengesi** **Bölüm 12 · Davranışsal ve Kolektif Karar**

Önemi: Sınırlı rasyonellik, ajanların hesaplama sınırlarını öne çıkarır. Bu konu algoritmik oyun kuramında ve çok ajanlı yapay zekâda sık sık karşımıza çıkar.

"A Behavioral Model of Rational Choice" (1955) makalesi, arama maliyetleri yüzünden tam optimizasyonun gerçekçi olmadığını savundu. Simon yapay zekânın kurucuları arasındadır (Logic Theorist, 1956).

CS

Claude E. Shannon

1916–2001 · Amerikalı · Bell Labs, MIT

Satranç programlama · 1950

"Programming a Computer for Playing Chess" (*Philosophical Magazine*, 1950) makalesinde satranç oynayan bir programın iskeletini çizdi: derinlik sınırlı minimaks arama ve yapraklarda bir değerlendirme fonksiyonu.

Ödüller: Kyoto Prize 1985

değerlendirme fonksiyonu · derinlik sınırı · Shannon sayısı

DERS BÖLÜMLERİ **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Önemi: Satrançtaki olası oyun sayısı için verdiği kaba tahmin (yaklaşık 10^{120}), kaba kuvvetle tam aramanın neden imkânsız olduğunu gösteren standart sayıdır ve Shannon sayısı olarak bilinir.

Makalede her hamleyi aynı derinlikte inceleyen tam genişlikli arama (A tipi) ile yalnızca umut verici hamleleri inceleyen seçici arama (B tipi) ayrımını yaptı. Bilgi kuramını kuran "A Mathematical Theory of Communication" (1948) makalesi de ona aittir.

GB

George W. Brown

1917–2005 · Amerikalı · RAND, UCLA, UC Irvine

Kurgusal oyun · 1951

Kurgusal oyunu (*fictitious play*) önerdi (1951). Bu dinamikte her oyuncu, rakibin geçmiş hamlelerinin gözlenen sıklıklarına en iyi tepkiyi verir.

kurgusal oyun · en iyi tepki · gözlenen sıklıklar

DERS BÖLÜMLERİ **Bölüm 13 · Öğrenme ve Adaptasyon**

Önemi: Kurgusal oyun, oyunlar için ilk öğrenme dinamiğidir. Julia Robinson 1951'de bu dinamiğin iki oyunculu sıfır toplamlı oyunlarda yakınsadığını kanıtladı.

Shapley (1964), kurgusal oyunun döngüye girdiği 3×3'lük bir oyun verdi. Böylece basit öğrenmenin genel oyunlarda Nash dengesine ulaşmak zorunda olmadığı görüldü.

LH

Leonid Hurwicz

1917–2008 · Polonyalı-Amerikalı (Moskova doğumlu) · University of Minnesota

Mekanizma tasarımı · teşvik uyumluluğu · 1972

Mekanizma tasarımı kuramını kurdu ve teşvik uyumluluğu (*incentive compatibility*) kavramını tanıttı (1972).

Ödüller: Nobel Ekonomi Ödülü 2007 (Eric Maskin ve Roger Myerson ile); 90 yaşıyla o güne kadarki en yaşlı ödül sahibi

teşvik uyumluluğu | bilginin merkezsizleşmesi | uygulama

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı**

Önemi: Mekanizma tasarımı oyun kuramını tersine çevirir: kurallar, kendi çıkarını gözeten davranış istenen sonucu üretecek biçimde seçilir.

Hurwicz, değişim ekonomilerinde verimli ve bireysel rasyonel hiçbir mekanizmanın baskın stratejide teşvik uyumlu olamayacağını gösterdi. Bu sonuç, alanın incelediği ödünleşimlerin çerçevesini çizdi.

BH

David Blackwell & James Hannan

1919–2010 · 1922–2010 · Amerikalı · UC Berkeley (Blackwell); Michigan State University (Hannan)

Yaklaşılabilirlik 1956 · Hannan tutarlılığı 1957

Pişmanlıksız (*no-regret*) öğrenmenin temellerini attılar. Hannan (1957), tekrarlanan oyunlarda dışsal pişmanlığı giderek sıfıra inen bir strateji kurdu. Blackwell (1956), vektör kazançlar için yaklaşılabilirlik (*approachability*) teoremini kanıtladı.

Ödüller: Blackwell: ABD Ulusal Bilimler Akademisine seçilen ilk siyahi üye (1965); von Neumann Theory Prize 1979

dışsal pişmanlık | yaklaşılabilirlik | tekrarlanan oyun

DERS BÖLÜMLERİ **Bölüm 13 · Öğrenme ve Adaptasyon**

Önemi: Hedge ve pişmanlık eşleştirme gibi pişmanlıksız algoritmalar bu sonuçlardan türer ve kaba korelasyonlu dengelere yakınsar.

Blackwell teoremi, bir oyuncunun ortalama vektör kazancını hangi dışbükey kümelere yaklaştırmaya zorlayabileceğini belirler. Hannan tutarlılığı ve kalibrasyon bu teoremin sonuçları olarak elde edilir.

JH

John Harsanyi

1920–2000 · Macar-Amerikalı · UC Berkeley

Bayesçi oyunlar · 1967–68

Eksik bilgili oyunlar için kuramsal çerçeveyi geliştirdi ("Games with Incomplete Information Played by Bayesian Players", 1967–68).

Ödüller: Nobel Ekonomi Ödülü 1994 (John Nash ve Reinhard Selten ile)

tip uzayları · ortak ön olasılık · Bayes–Nash dengesi

DERS BÖLÜMLERİ **Bölüm 3 · Nash Dengesi** **Bölüm 4 · Mekanizma Tasarımı**

Önemi: Bayesçi oyunlar (*Bayesian games*), ortak bir ön olasılıktan (*common prior*) çekilen özel tipleri (değerler, maliyetler) modeller. Açık artırma ve mekanizma kuramı bu modele dayanır.

Harsanyi dönüşümü, her oyuncunun tipini çeken bir şans hamlesi ("Doğa") ekler. Böylece eksik bilgi, mükemmel olmayan bilgiye dönüşür. Arındırma teoremi (1973) karışık stratejilere yeni bir yorum getirir.

JM

John Maynard Smith

1920–2004 · İngiliz · University of Sussex

Evrimsel kararlı strateji · 1973

George Price ile birlikte evrimsel kararlı stratejiyi (*evolutionarily stable strategy*, ESS) tanıttı ("The Logic of Animal Conflict", Nature, 1973).

Ödüller: Crafoord Prize 1999 (Ernst Mayr ve George C. Williams ile)

ESS · Şahin–Güvercin · replikatör dinamiği

DERS BÖLÜMLERİ **Bölüm 13 · Öğrenme ve Adaptasyon**

Önemi: ESS ve replikatör dinamiği (Taylor ve Jonker, 1978), oyun kuramını rasyonel oyuncu varsaymayan popülasyonlara taşır.

Bir popülasyonun benimsediği stratejiyi küçük bir mutant grubu istila edemiyorsa bu strateji evrimsel kararlıdır. Her ESS simetrik bir Nash dengesidir. Kitap: *Evolution and the Theory of Games* (1982).

DG

David Gale

1921–2008 · Amerikalı · UC Berkeley

Ertelenmiş kabul · 1962

Shapley ile birlikte ertelenmiş kabul (*deferred acceptance*) algoritmasını buldu ("College Admissions and the Stability of Marriage", 1962).

kararlı eşleştirme | engelleyici çift | ertelenmiş kabul

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 5 · Açık Artırmalar**

Önemi: Ertelenmiş kabul, $O(n^2)$ teklifle her zaman kararlı bir eşleştirme bulur. Bulunan eşleştirme teklif veren taraf için en iyisidir ve algoritma bu taraf için stratejiye dayanıklıdır (*strategy-proof*).

Gale 2008'de öldü; kararlı tahsis kuramı 2012'de Nobel Ekonomi Ödülü ile ödüllendirildi. Doğrusal programlamadaki dualiteye ve Hex oyununa da katkı yaptı. Hex'in berabere bitemeyeceğinin bilinen ispatlarından birini Gale verdi ve bu sonucu Brouwer sabit nokta teoremine bağladı.

KA

Kenneth Arrow

1921–2017 · Amerikalı · Stanford University

Arrow'un imkânsızlık teoremi · 1951

Social Choice and Individual Values (1951) kitabında, birkaç doğal aksiyomu birlikte sağlayan bir sıralama birleştirme kuralı olmadığını kanıtladı.

Ödüller: Nobel Ekonomi Ödülü 1972 (John Hicks ile)

tercih birleştirme | IIA | diktatör yok

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 7 · Sosyal Seçim ve Oylama**

Önemi: Üç ya da daha fazla seçenek varsa oybirliğini (Pareto) ve ilgisiz alternatiflerden bağımsızlığı (*independence of irrelevant alternatives*, IIA) sağlayan her kural bir diktatörlüktür.

Arrow teoremi aksiyomatik sosyal seçimi başlattı. Arrow, rekabetçi dengenin varlığını da Debreu ile birlikte kanıtladı (Arrow–Debreu, 1954). Bu ispat, Nash'inkine benzer sabit nokta argümanlarına dayanır.

TS

Thomas Schelling

1921–2016 · Amerikalı · Harvard University, University of Maryland

Odak noktaları · taahhüt · ayrışma modeli

The Strategy of Conflict (1960) kitabında çatışma ve koordinasyonu inceledi. Odak (Schelling) noktalarını (*focal points*) ve inandırıcı taahhüt kavramını tanıttı.

Ödüller: Nobel Ekonomi Ödülü 2005 (Robert Aumann ile)

odak noktası · taahhüt · bireysel güdüler ve toplu davranış

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 3 · Nash Dengesi** **Bölüm 10 · Çoklu Ajan Sistemleri**

Önemi: Ayrışma modeli (1969–71), hafif bireysel tercihlerin güçlü bir toplu ayrışma doğurabildiğini gösterdi. Bu model ajan tabanlı simülasyonun ilk örneklerindendir.

Schelling, kişinin kendi seçeneklerini kısıtlamasının (köprüleri yakmanın) stratejik değer taşıyabileceğini gösterdi. Bu fikir caydırma ve pazarlık kuramında sık geçer.

JW

John Wardrop

1922–1989 · İngiliz · Road Research Laboratory, University College London

Wardrop dengesi · 1952

Trafik ataması için iki ilke önerdi (1952): kullanıcı dengesi ve sistem optimumu. Bu ilkeler bugün onun adıyla anılır.

kullanıcı dengesi · sistem optimumu · trafik ataması

DERS BÖLÜMLERİ **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** **Bölüm 9 · Anarşinin Bedeli**

Önemi: Wardrop dengesinde kullanılan her yolun gecikmesi eşit ve en küçüktür. Anarşinin Bedeli bu dengeyi sistem optimumuyla karşılaştırır.

Beckmann, McGuire ve Winsten (1956), Wardrop dengelerinin bir potansiyel fonksiyonunu minimize ettiğini gösterdi. Bu yüzden dengeler dışbükey optimizasyonla hesaplanabilir.

LS

Lloyd Shapley

1923–2016 · Amerikalı · RAND Corporation, UCLA

Shapley değeri · stokastik oyunlar · eşleştirme

Shapley değerini (1953) ve stokastik oyunları (1953) tanıttı. Gale ile ertelenmiş kabul algoritmasını buldu (1962).

Ödüller: Nobel Ekonomi Ödülü 2012 (Alvin Roth ile)

Shapley değeri · koalisyon · çekirdek

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 5 · Açık Artırmalar** **Bölüm 10 · Çoklu Ajan Sistemleri**

Önemi: Shapley değeri; verimlilik, simetri, toplamsallık ve kukla oyuncu aksiyomlarını birlikte sağlayan tek paylaşım kuralıdır.

Bu değer, bir oyuncunun tüm geliş sıralarındaki marjinal katkısının ortalamasıdır. Bugün maliyet paylaşımında ve makine öğrenmesi tahminlerini açıklamada (SHAP) kullanılır. Bondareva–Shapley teoremi, çekirdeği boş olmayan oyunları belirler.

HK

Harold W. Kuhn

1925–2014 · Amerikalı · Princeton University

Kuhn teoremi · 1953 · Macar yöntemi · 1955

"Extensive games and the problem of information" (1953) makalesinde genişletilmiş form oyunlarını bugünkü bilgi kümeleriyle tanımladı. Mükemmel bilgili sonlu oyunlarda saf stratejilerin yettiğini ve geriye doğru çıkarımın denge verdiğini gösterdi.

Ödüller: John von Neumann Theory Prize 1980 (David Gale ve Albert W. Tucker ile)

genişletilmiş biçim · bilgi kümesi · Macar yöntemi

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 10 · Çoklu Ajan Sistemleri**

Önemi: Zermelo'nun satranç sonucu Kuhn'un çerçevesinde genel bir teoreme dönüşür. Minimaks ağacındaki değer, mükemmel bilgili oyunun saf stratejili değeridir.

Atama problemini çözen Macar yöntemini 1955'te yayımladı. Karush–Kuhn–Tucker koşulları da onun adını taşır. Nash'in yakın arkadaşı ve meslektaşıydı.

JN

John Nash

1928–2015 · Amerikalı · Princeton University, MIT

Nash dengesi · 1950–51

Nash dengesini tanımladı ve her sonlu oyunun karışık stratejilerde bir dengesi olduğunu kanıtladı (PNAS 1950; Annals of Mathematics 1951).

Ödüller: Nobel Ekonomi Ödülü 1994 (John Harsanyi ve Reinhard Selten ile); Abel Prize 2015 (Louis Nirenberg ile)

en iyi tepki · sabit nokta · pazarlık çözümü

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 3 · Nash Dengesi**

Önemi: Nash dengesi, işbirlikçi olmayan oyun kuramının temel çözüm kavramıdır. Varlık ispatı bir sabit nokta teoremine dayanır.

1950 ispatı Kakutani'nin, 1951 ispatı Brouwer'in sabit nokta teoremini kullanır. İspat yapıcı değildir; dengeyi hesaplamanın PPAD-tam olduğu çok sonra gösterildi. Nash ayrıca Nash pazarlık çözümünü tanıttı (1950).

RS

Reinhard Selten

1930–2016 · Alman · Universität Bonn

Alt oyun mükemmel denge · 1965

Alt oyun mükemmel dengeyi (*subgame-perfect equilibrium*, 1965) ve titreyen el mükemmel dengesini (*trembling-hand perfect equilibrium*, 1975) tanıttı.

Ödüller: Nobel Ekonomi Ödülü 1994 (John Nash ve John Harsanyi ile)

genişletilmiş form · geriye doğru çıkarım · titreyen el

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 3 · Nash Dengesi**

Önemi: Bu iyileştirmeler, inandırıcı olmayan tehditlere dayanan Nash dengelerini eler.

Alt oyun mükemmel denge her alt oyunda bir Nash dengesi oluşturur. Sonlu mükemmel bilgili oyunlarda geriye doğru çıkarımla bulunur. Selten zincir mağaza paradoksunu da (1978) ortaya koydu; deneysel iktisadın öncülerindendir.

RA

Robert Aumann

d. 1930 · İsrailli-Amerikalı · Hebrew University of Jerusalem

Korelasyonlu denge · tekrarlanan oyunlar

Korelasyonlu dengeyi (*correlated equilibrium*) tanıttı (1974). Ortak bilgiyi biçimsel olarak tanımladı (“Agreeing to Disagree”, 1976) ve tekrarlanan oyunlar kuramını geliştirdi.

Ödüller: Nobel Ekonomi Ödülü 2005 (Thomas Schelling ile)

korelasyonlu denge · ortak bilgi · halk teoremi

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 3 · Nash Dengesi**

Önemi: Korelasyonlu dengeler, doğrusal programlamayla hesaplanabilen bir çokyüzlü oluşturur. İç pişmanlığı sıfıra indiren öğrenme bu dengelere yakınsar.

Korelasyonlu dengede bir aracı bir eylem profili çeker ve her oyuncuya kendi eylemini gizlice önerir. Öneriye uymamak hiçbir oyuncuya kazanç sağlamaz. Halk teoremi, sonsuz kez tekrarlanan oyunlarda işbirliğini açıklar.

KT

Daniel Kahneman & Amos Tversky

1934–2024 · 1937–1996 · İsrailli-Amerikalı · Princeton (Kahneman); Stanford (Tversky)

Beklenti kuramı · 1979

Yargıdaki sistematik yanlılıkları belgelediler (sezgisel kurallar ve yanlılıklar, 1974). Beklenti kuramını (*prospect theory*) önerdiler (Econometrica, 1979).

Ödüller: Kahneman: Nobel Ekonomi Ödülü 2002 (Vernon Smith ile); Tversky ödülden önce, 1996'da öldü

kayıptan kaçınma · çerçeveleme · olasılık ağırlığı

DERS BÖLÜMLERİ **Bölüm 12 · Davranışsal ve Kolektif Karar**

Önemi: Beklenti kuramı, beklenen faydanın yerine referans noktasına bağlılığı, kayıptan kaçınmayı ve olasılık ağırlıklandırmasını koyar. Davranışsal oyun kuramı bu temele dayanır.

Çerçeveleme etkileri, rasyonel seçim modellerinin varsaydığı değişmezliği bozar. Kahneman'ın *Thinking, Fast and Slow* (2011) kitabı ikili süreç görüşünü geniş kitlelere tanıttı.

RW

Robert Wilson

d. 1937 · Amerikalı · Stanford University

Ortak değerli açık artırmalar · açık artırma biçimleri

Ortak değerli açık artırmaların ve kazananın lanetinin (*winner's curse*) çözümlemesine öncülük etti (1969, 1977). Eşzamanlı çok turlu açık artırmanın tasarımcılarından biridir.

Ödüller: Nobel Ekonomi Ödülü 2020 (Paul Milgrom ile)

ortak değer · kazananın laneti · ardışık denge

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 5 · Açık Artırmalar**

Önemi: Kreps ile ardışık dengeyi (*sequential equilibrium*) tanımladı (1982). Milgrom ile, ABD Federal İletişim Komisyonunun (FCC) 1994'ten beri kullandığı spektrum açık artırmalarını tasarladı.

Wilson, Milgrom'un doktora danışmanıdır; Roth ve Rosenthal'ın da danışmanlığını yaptı. *Nonlinear Pricing* (1993) kitabı elektrik piyasalarının tasarımını etkiledi.

DB

Dietrich Braess

d. 1938 · Alman · Ruhr-Universität Bochum

Braess paradoksu · 1968

Bir ağa yeni bir yol eklemenin, dengede her sürücünün yolculuk süresini artırabileceğini 1968'de gösterdi.

ağ paradoksu | Wardrop dengesi | bencil yönlendirme

DERS BÖLÜMLERİ **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** **Bölüm 9 · Anarşinin Bedeli**

Önemi: Braess paradoksu, ek kapasitenin bencil yönlendirmeye zarar verebileceğinin standart örneğidir. Doğrusal gecikmelerde bu örneğin Anarşinin Bedeli 4/3'tür.

Makale ("Über ein Paradoxon aus der Verkehrsplanung", Unternehmensforschung, 1968) 2005'te İngilizceye çevrildi. Braess'in asıl alanı sayısal analizdir (sonlu elemanlar).

DK

Donald E. Knuth

d. 1938 · Amerikalı · Stanford University

Alfa-beta budamasının çözümlemesi · 1975

Ronald W. Moore ile yazdığı "An analysis of alpha-beta pruning" (*Artificial Intelligence*, 1975) makalesinde alfa-beta budamasının ne kadar iş kazandırdığını ölçtü.

Ödüller: Turing Award 1974

alfa-beta budaması | hamle sıralaması | etkin dallanma etkeni

DERS BÖLÜMLERİ **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Önemi: Hamleler en iyi sırayla incelendiğinde alfa-beta, b dallanmalı ve d derinlikli düzgün ağaçta $b^{\lceil d/2 \rceil} + b^{\lfloor d/2 \rfloor} - 1$ yaprağa bakar. Aynı sürede yaklaşık iki kat derine inilebilmesi bu sonuçtan gelir.

Alfa-beta fikri 1950'ler ve 1960'larda birkaç araştırmacı tarafından birbirinden bağımsız olarak bulunmuştu. Knuth ve Moore yöntemi kesin biçimde yazıp çözümledi. Knuth daha çok *The Art of Computer Programming* dizisi ve TeX ile tanınır.

CG

Edward H. Clarke & Theodore Groves

Clarke 1939–2013 · Amerikalı · Clarke: ABD federal hükümetinde iktisatçı; Groves: UC San Diego

Clarke pivot kuralı 1971 · Groves mekanizmaları 1973

Vickrey'nin fikrini VCG mekanizmasına genellediler. Clarke (Public Choice, 1971) pivot ödemesini, Groves (Econometrica, 1973) genel mekanizma sınıfını tanıttı.

VCG · dışsallık fiyatlandırması · kamu malı

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı**

Önemi: VCG (Vickrey 1961, Clarke 1971, Groves 1973), sosyal refahı (*social welfare*) maksimize eden sonucu seçer. Her ajan, başkalarına yüklediği dışsallık kadar öder. Böylece dürüst bildirim baskın strateji olur.

Green ve Laffont (1977), kısıtsız yarı doğrusal tercihler altında baskın stratejide verimli mekanizmaların yalnızca Groves mekanizmaları olduğunu gösterdi. VCG bütçe açığı verebilir ve danışıklılığa karşı savunmasızdır.

AG

Allan Gibbard

d. 1942 · Amerikalı · University of Michigan

Gibbard–Satterthwaite teoremi · 1973

En az üç olası sonucu olan ve diktatörlük olmayan her oylama kuralının manipüle edilebileceğini kanıtladı (Econometrica, 1973).

stratejiye dayanıklılık · manipülasyon · diktatörlük

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 7 · Sosyal Seçim ve Oylama**

Önemi: Gibbard ve Satterthwaite teoremi birbirinden bağımsız kanıtladı. Teorem, stratejiye dayanıklılığı mekanizma tasarımında temel bir kısıt hâline getirir.

Çıkış yolları şunlardır: kısıtlı tercih alanları (tek tepeli tercihler ve medyan kuralı), para (VCG), rastgeleleştirme ve manipülasyonun hesaplama açısından zor olması.

RA

Robert Axelrod

d. 1943 · Amerikalı · University of Michigan

Tekrarlanan Mahkûmun İkilemi turnuvaları · 1980

Tekrarlanan Mahkûmun İkilemi için bilgisayar turnuvaları düzenledi (1980). Turnuvaları Anatol Rapoport'un Tit for Tat stratejisi kazandı.

Ödüller: MacArthur Fellowship 1987

tekrarlanan Mahkûmun İkilemi · Tit for Tat · karşılıklılık

DERS BÖLÜMLERİ **Bölüm 10 · Çoklu Ajan Sistemleri** **Bölüm 13 · Öğrenme ve Adaptasyon**

Önemi: "The Evolution of Cooperation" (W. D. Hamilton ile, Science 1981; kitap 1984), karşılıklılığın işbirliğini nasıl sürdürdüğünü açıkladı.

Başarılı stratejiler iyi niyetli, misilleme yapan, affedici ve anlaşılırdı. Sonuçlar ajan tabanlı modellemeye ve evrimsel dinamiklere ilham verdi.

MS

Mark Satterthwaite

d. 1945 · Amerikalı · Kellogg School, Northwestern University

Gibbard–Satterthwaite 1975 · Myerson–Satterthwaite 1983

Manipüle edilebilirlik teoremini Gibbard'dan bağımsız olarak kanıtladı (Journal of Economic Theory, 1975).

stratejiye dayanıklılık · iki taraflı ticaret · imkânsızlık

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 5 · Açık Artırmalar** **Bölüm 7 · Sosyal Seçim ve Oylama**

Önemi: Myerson–Satterthwaite teoremi (1983), iki taraflı ticarette aynı anda verimli, teşvik uyumlu, bireysel rasyonel ve bütçe dengeli bir mekanizma olmadığını gösterir.

Bu iki imkânsızlık sonucu, gerçek piyasaların ve seçimlerin neden bazı istenen özelliklerden vazgeçmek zorunda olduğunu açıklar.

RR

Robert W. Rosenthal

1945–2002 · Amerikalı · Boston University

Tıkanıklık oyunları · potansiyel · 1973

Tıkanıklık oyunlarını (*congestion games*) tanıttı (1973). Bir potansiyel fonksiyonu yardımıyla bu oyunların her zaman saf Nash dengesi olduğunu kanıtladı.

tıkanıklık oyunu · potansiyel fonksiyonu · saf denge

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** **Bölüm 9 · Anarşinin Bedeli**

Önemi: Rosenthal potansiyeli, en iyi tepki dinamiğinin sonlanacağını garanti eder. Yönlendirme oyunlarının ve İstikrarın Bedeli sınırlarının temel aracıdır.

Monderer ve Shapley (1996), her tam potansiyel oyunun bir tıkanıklık oyununa izomorf olduğunu gösterdi. Rosenthal kırkayak oyununu da tanıttı (1981).

HV

Hal Varian

d. 1947 · Amerikalı · UC Berkeley, Google

Pozisyon açık artırmaları · 2007

Google'ın baş ekonomisti olarak sponsorlu aramadaki reklam açık artırmalarını inceledi ("Position Auctions", 2007).

sponsorlu arama | GSP açık artırması | tıklanma oranı

DERS BÖLÜMLERİ **Bölüm 6 · Çevrim İçi Platformlar**

Önemi: Genelleştirilmiş ikinci fiyat (*generalized second-price*, GSP) açık artırmasının simetrik Nash dengeleri, reklam fiyatlarını tıklanma oranlarına ve teklifçi değerlerine bağlar.

Varian 2002'de Google'a katıldı; şirketin reklam açık artırmasının ve rezerv fiyat politikalarının tasarımına katkı verdi. *Intermediate Microeconomics* ders kitabı yaygın olarak okutulur.

PM

Paul Milgrom

d. 1948 · Amerikalı · Stanford University

Açık artırma kuramı · piyasa tasarımı

İlişkili değerler (*affiliated values*) kuramını ve bağlantı ilkesini (*linkage principle*) geliştirdi (Weber ile, 1982). Spektrum açık artırmaları tasarladı.

Ödüller: Nobel Ekonomi Ödülü 2020 (Robert Wilson ile)

bağlantı ilkesi | artan fiyatlı açık artırma | spektrum açık artırması

DERS BÖLÜMLERİ **Bölüm 5 · Açık Artırmalar**

Önemi: FCC'nin eşzamanlı çok turlu açık artırmasının (1994) ve 2016–17 teşvik açık artırmasının tasarımcıları arasındadır.

Bağlantı ilkesi, değerler ilişkili olduğunda açık ve artan fiyatlı açık artırmaların neden daha çok gelir getirdiğini açıklar. Kitap: *Putting Auction Theory to Work* (2004).

CP

Christos Papadimitriou

d. 1949 · Yunan · UC Berkeley, Columbia University

PPAD 1994 · Anarşinin Bedeli 1999

PPAD karmaşıklık sınıfını tanımladı (1994). Koutsoupias ile en kötü dengenin maliyetini optimumla karşılaştıran oranı tanıttı (1999); bu oran sonradan Anarşinin Bedeli adını aldı.

Ödüller: Knuth Prize 2002; Gödel Prize 2012 (Anarşinin Bedeli çalışması için, Koutsoupias ile); IEEE John von Neumann Medal 2016

PPAD | Anarşinin Bedeli | toplam arama problemleri

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 3 · Nash Dengesi** **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** **Bölüm 9 · Anarşinin Bedeli**

Önemi: Daskalakis ve Goldberg ile Nash dengesi hesaplamanın PPAD-tam olduğunu kanıtladı (2006).

PPAD, çözümünün varlığı yönlü çizgeler üzerindeki bir eşlik argümanıyla (yolun sonu, *end-of-the-line*) garanti edilen toplam arama problemlerini kapsar. 2001'deki "Algorithms, Games, and the Internet" başlıklı STOC konuşması, algoritmik oyun kuramının doğuşuna katkı verdi.

SH

Sergiu Hart

d. 1949 · İsrailli · Hebrew University of Jerusalem

Pişmanlık eşleştirme · 2000

Andreu Mas-Colell ile pişmanlık eşleştirmeyi (*regret matching*) tanıttı (Econometrica, 2000). Bu basit uyarlanabilir yöntemde gözlenen oyun, korelasyonlu dengeler kümesine yakınsar.

pişmanlık eşleştirme | iç pişmanlık | korelasyonlu denge

DERS BÖLÜMLERİ **Bölüm 13 · Öğrenme ve Adaptasyon**

Önemi: İç pişmanlığı sıfıra indiren öğrenme korelasyonlu dengelere, dışsal pişmanlığı sıfıra indiren öğrenme kaba korelasyonlu dengelere götürür.

Hart ve Mas-Colell, "bağlantısız" (*uncoupled*) dinamiklerin her oyunda Nash dengesine yakınsamayı garanti edemeyeceğini de gösterdi (2003).

EM

Eric Maskin

d. 1950 · Amerikalı · Harvard University

Uygulama kuramı · Maskin monotonluğu · 1977

Hangi sosyal seçim kurallarının Nash dengesinde uygulanabileceğini belirledi (1977; yayımı 1999).

Ödüller: Nobel Ekonomi Ödülü 2007 (Leonid Hurwicz ve Roger Myerson ile)

Nash uygulaması · monotonluk · sosyal seçim kuralları

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı**

Önemi: Maskin monotonluğu Nash uygulaması için gereklidir. Üç ya da daha fazla ajan varsa ve hiçbir ajanın veto gücü yoksa yeterlidir de.

Uygulama kuramı (*implementation theory*), tasarlanan oyunun yalnızca bazı dengelerinin değil, tüm dengelerinin istenen sonucu verip vermediğini sorar.

AR

Alvin Roth

d. 1951 · Amerikalı · Harvard University, Stanford University

Piyasa tasarımı · böbrek takası

Eşleştirme kuramını uygulamaya taşıdı: ABD tıp ihtisas eşleştirmesinin yeniden tasarımı (Roth–Peranson, 1999), Boston ve New York'ta okul seçimi ve böbrek takası.

Ödüller: Nobel Ekonomi Ödülü 2012 (Lloyd Shapley ile)

ertelenmiş kabul · böbrek takası · okul seçimi

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı**

Önemi: Piyasa tasarımı (*market design*), piyasaları mühendislik ürünü sistemler olarak ele alır. İyi bir piyasa kalabalık olmalı, tıkanmamalı ve dürüst katılımı güvenli kılmalıdır.

Roth, iki taraf için birden stratejiye dayanıklı olan kararlı bir eşleştirme mekanizması olmadığını gösterdi (1982). Kitap: *Who Gets What — and Why* (2015).

RM

Roger Myerson

d. 1951 · Amerikalı · Northwestern University, University of Chicago

Açıklama ilkesi 1979 · optimal açık artırmalar 1981

Açıklama ilkesini (*revelation principle*) ortaya koydu (1979). Geliri maksimize eden açık artırmaları belirledi (“Optimal Auction Design”, 1981).

Ödüller: Nobel Ekonomi Ödülü 2007 (Leonid Hurwicz ve Eric Maskin ile)

açıklama ilkesi · sanal değer · Myerson lemması

DERS BÖLÜMLERİ **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 5 · Açık Artırmalar**

Önemi: Sanal değerler (*virtual values*), gelir maksimizasyonunu refah maksimizasyonuna indirger. Değerler bağımsız, aynı dağılımlı ve düzenliyse optimal açık artırma, rezerv fiyatlı ikinci fiyat açık artırmasıdır.

Myerson lemması: tek parametreli bir tahsis kuralı ancak ve ancak monotonsa uygulanabilir; bu durumda ödeme kuralı tek türlü belirlenir. Satterthwaite ile iki taraflı ticaret imkânsızlık teoremini kanıtladı (1983).

TR

Jean Tirole & Jean-Charles Rochet

Tirole d. 1953 · Fransız · Toulouse School of Economics

İki taraflı pazarlar · 2003

İki taraflı platformların iktisadını kurdular (“Platform Competition in Two-Sided Markets”, 2003).

Ödüller: Tirole: Nobel Ekonomi Ödülü 2014 (tek başına)

iki taraflı pazar · ağ etkisi · platform fiyatlandırması

DERS BÖLÜMLERİ **Bölüm 6 · Çevrim İçi Platformlar**

Önemi: Platformlar iki tarafa da fiyat koyar ve bir tarafı çekmek için diğer tarafı sübvanse edebilir. Uygulama mağazalarının, kart ağlarının ve pazar yerlerinin fiyatlandırması bu çerçeveyle açıklanır.

Tirole, Nobel'i piyasa gücü ve düzenleme üzerine çözümlemeleriyle aldı. Drew Fudenberg ile *Game Theory* (1991) ders kitabını yazdı.

YS

Yoav Shoham

İsrailli-Amerikalı · Stanford University

Çoklu ajan sistemleri

Yapay zekâ ile oyun kuramını birleştirdi. Kevin Leyton-Brown ile *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* (2009) kitabını yazdı.

çoklu ajan sistemi · ajan tasarımı · hesaplamalı denge

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 10 · Çoklu Ajan Sistemleri**

Önemi: Stratejik akıl yürütmeyi, müzakereyi ve öğrenmeyi otonom ajanların hesaplamalı modellerine taşıdı.

AI Index raporunun ve AI21 Labs şirketinin kurucuları arasındadır.

ÉT

Éva Tardos

d. 1957 · Macar-Amerikalı · Cornell University

Bencil yönlendirme · 2000–02

Roughgarden ile bencil yönlendirmede (*selfish routing*) Anarşinin Bedeline sınır getirdi ("How Bad Is Selfish Routing?", FOCS 2000; JACM 2002).

Ödüller: Fulkerson Prize 1988; Gödel Prize 2012; IEEE John von Neumann Medal 2019

ağ akışları · Anarşinin Bedeli · oyunlarda öğrenme

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** **Bölüm 9 · Anarşinin Bedeli**

Önemi: Doğrusal gecikmelerde Wardrop dengesinin maliyeti, ağ ne olursa olsun optimumun en çok 4/3 katıdır.

Algorithmic Game Theory (2007) kitabının editörlerindendir. Kleinberg ile *Algorithm Design* (2005) kitabını yazdı. Son çalışmaları, açık artırmalarda ve yönlendirmede pişmanlıksız öğrenen oyuncuları inceler.

VV

Vijay V. Vazirani

d. 1957 · Hint asıllı Amerikalı · Georgia Tech, UC Irvine

AdWords · çevrim içi eşleştirme · piyasa dengeleri

AdWords algoritmasını birlikte geliştirdi (Mehta, Saberi, U. Vazirani, V. Vazirani, 2005). Algoritma, bütçeli çevrim içi tahsiste 1−1/e rekabet oranına (*competitive ratio*) ulaşır.

çevrim içi eşleştirme · AdWords · piyasa dengesi

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 6 · Çevrim İçi Platformlar**

Önemi: Çevrim içi iki parçalı eşleştirme (Karp–Vazirani–Vazirani, 1990) ve onun AdWords genellemesi, reklam platformlarının sorguları reklamverenlere nasıl atadığını modeller.

Nisan, Roughgarden ve Tardos ile *Algorithmic Game Theory* (2007) kitabının editörlüğünü yaptı. Fisher piyasası dengeleri için kombinatoryal algoritmalar da tasarladı.

JS

Jonathan Schaeffer

d. 1957 · Kanadalı · University of Alberta

Chinook · damanın çözümü · 2007

Dama programı Chinook'u geliştirdi. Chinook 1994'te insan–makine dünya şampiyonu ilan edildi. Ekibi 2007'de damayı çözdü: iki taraf da hata yapmazsa oyun berabere biter (*Science*, 2007).

çözülmüş oyunlar · dama · son oyun veritabanı

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller**

Önemi: Zermelo teoremine göre damanın sonucu en iyi oyunda belirlidir. Dama, bu sonucun bilgisayarla gerçekten bulunabildiği en büyük oyunlardandır.

Çözüm zayıf bir çözümdür: başlangıç konumunun değerini ve bu değeri koruyan oyunu verir, her konum için en iyi hamleyi vermez. Arama, alfa-beta türevleri ile çok büyük son oyun veritabanlarını birleştirdi.

CD

Cynthia Dwork

d. 1958 · Amerikalı · Harvard University

Diferansiyel mahremiyet 2006 · farkındalık yoluyla adalet 2012

Diferansiyel mahremiyeti (*differential privacy*) birlikte buldu (Dwork, McSherry, Nissim, Smith, 2006). Bireysel adaleti (*individual fairness*) tanımladı (“Fairness Through Awareness”, 2012).

Ödüller: Gödel Prize 2017; Knuth Prize 2020

diferansiyel mahremiyet · bireysel adalet · algoritmik etik

DERS BÖLÜMLERİ **Bölüm 7 · Sosyal Seçim ve Oylama** **Bölüm 11 · Algoritmik Adalet ve Etik** **Bölüm 14 · Gelecek Yönelimler**

Önemi: Bireysel adalet, benzer bireylere benzer davranılmasını ister. Diferansiyel mahremiyetten yaklaşık dürüst mekanizmalar da elde edilir (McSherry–Talwar, 2007).

Çalışmaları mahremiyeti ve adaleti, doğrulukla ödünleşebilen matematiksel güvenceler olarak ele alır. Web için sıralama birleştirme yöntemleri üzerine çalışması (Kumar, Naor ve Sivakumar ile, 2001) oylama kuramıyla bağlantılıdır.

DB

Feng-hsiung Hsu, Murray Campbell & A. Joseph Hoane

Deep Blue ekibi · Carnegie Mellon University, IBM Research

Deep Blue · 1997

Hsu'nun Carnegie Mellon'daki doktora çalışmasıyla başlayan ChipTest ve Deep Thought programları IBM'de Deep Blue'ya dönüştü. Deep Blue 1997'de dünya şampiyonu Garri Kasparov'u altı oyunluk maçta 3½–2½ yendi.

Ödüller: Fredkin Prize 1997 (Deep Blue ekibi)

paralel alfa-beta · satranç yongası · değerlendirme fonksiyonu

DERS BÖLÜMLERİ **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Önemi: Deep Blue, derinlik sınırlı alfa-beta aramasının özel donanımla bir dünya şampiyonunu yenebileceğini gösterdi. 1997 sürümü saniyede yaklaşık 200 milyon konum değerlendiriyordu.

1996'daki ilk maçı Kasparov 4–2 kazanmıştı. Hsu (d. 1959) sistemin baş tasarımcısıdır ve süreci *Behind Deep Blue* (2002) kitabında anlattı. Üçlü sistemi "Deep Blue" (*Artificial Intelligence*, 2002) makalesinde tanıttı.

NN

Noam Nisan

d. 1961 · İsrailli · Hebrew University of Jerusalem

Algoritmik mekanizma tasarımı · 1999

Amir Ronen ile algoritmik mekanizma tasarımını kurdu ("Algorithmic Mechanism Design", STOC 1999; GEB 2001).

Ödüller: Gödel Prize 2012 (Amir Ronen ile); Knuth Prize 2016

dürüst yaklaşım algoritmaları · kombinatoryal açık artırma · iletişim karmaşıklığı

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 4 · Mekanizma Tasarımı** **Bölüm 6 · Çevrim İçi Platformlar**

Önemi: Mekanizmalar hem teşvik uyumlu hem de hesaplama açısından verimli olmalıdır. Yaklaşık bir algoritmayla çalıştırılan VCG genelde dürüst değildir.

Algorithmic Game Theory (2007) kitabının editörlerindendir. Kombinatoryal açık artırmalar üzerine çalışmaları, verimli tahsis için ne kadar iletişim gerektiğini ölçer.

MK

Michael Kearns

Amerikalı · University of Pennsylvania

Grafiksel oyunlar · etik algoritmalar

Grafiksel oyunları tanıttı (Littman ve Singh ile, 2001). Makine öğrenmesinde adaleti ve mahremiyeti inceledi.

grafiksel oyunlar · öğrenme kuramı · algoritmik adalet

DERS BÖLÜMLERİ **Bölüm 11 · Algoritmik Adalet ve Etik** **Bölüm 14 · Gelecek Yönelimler**

Önemi: Her oyuncunun kazancı yalnızca komşularına bağlıysa grafiksel oyunlar büyük oyunları kısa biçimde temsil eder.

Aaron Roth ile, algoritmalara adalet ve mahremiyet yerleştirmeyi anlatan *The Ethical Algorithm* (2019) kitabını yazdı.

EK

Elias Koutsoupias

Yunan · University of Oxford

En kötü durum dengeleri · 1999

Papadimitriou ile en kötü denge ile optimum arasındaki oranı tanıttı ("Worst-case Equilibria", STACS 1999). Bu orana sonradan Anarşinin Bedeli dendi.

Ödüller: Gödel Prize 2012 (Christos Papadimitriou ile)

Anarşinin Bedeli · yük dengeleme · en kötü durum çözümlemesi

DERS BÖLÜMLERİ **Bölüm 9 · Anarşinin Bedeli**

Önemi: Anarşinin Bedeli, bencil davranışın sistem başarımını en kötü durumda ne kadar düşürdüğüne bir güvence verir.

İlk model, paralel bağlantılar üzerinde bencil işlerin yük dengelemesiydi. Koutsoupias çevrim içi algoritmalar üzerine de çalıştı (iş fonksiyonu algoritması için k-sunucu sanısı).

PG

Paul W. Goldberg

İngiliz · University of Oxford

Nash dengesi PPAD-tamdır · 2006

Nash dengesinin PPAD-tamlığını Daskalakis ve Papadimitriou ile birlikte kanıtladı (2006).

PPAD · grafiksel oyunlar · indirgemeler

DERS BÖLÜMLERİ **Bölüm 3 · Nash Dengesi**

Önemi: İndirgeme, aritmetik devreleri oyunlar olarak kodlar. Böylece her denge bir yolun sonu örneğini çözer.

Goldberg yaklaşık dengelerin ve Lemke–Howson gibi yol izleyen yöntemlerin karmaşıklığını da inceler.

AA

James D. Allen & Victor Allis

Allis: Hollandalı, d. 1965 · Vrije Universiteit Amsterdam, Maastricht University

Dörtlü Bağla'nın çözümü · 1988

Dörtlü Bağla'yı (*Connect Four*) 1988'de birbirinden bağımsız olarak çözdüler: ilk oyuncu ortadaki sütundan başlarsa kazanmayı garanti eder. Allen'ın sonucu 1 Ekim'de, Allis'inki 16 Ekim'de duyuruldu.

çözülmüş oyunlar · bilgiye dayalı arama · kanıt sayısı araması

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Önemi: Oyun yaklaşık $4{,}5 \cdot 10^{12}$ yasal konum içerir. Çözüm kaba kuvvetle değil, oyuna özgü kurallar ve güçlü aramayla elde edildi.

Allis yüksek lisans tezinde dokuz kurala dayanan bilgiye dayalı bir yaklaşım kullandı. 1994'teki doktora tezinde kanıt sayısı aramasını (*proof-number search*) tanıttı.

JK

Jon Kleinberg

d. 1971 · Amerikalı · Cornell University

Ağlar · adalet ödünleşimleri · 2016

Mullainathan ve Raghavan ile, risk puanlarında kalibrasyonun ve dengeli hata oranlarının özel durumlar dışında birlikte sağlanamayacağını kanıtladı (2016).

Ödüller: Nevanlinna Prize 2006

adalet imkânsızlığı · ağlar · HITS

DERS BÖLÜMLERİ **Bölüm 6 · Çevrim İçi Platformlar** **Bölüm 11 · Algoritmik Adalet ve Etik**

Önemi: Bu imkânsızlık sonucu, adalet ölçütlerinin birbiriyle çatıştığını gösterir. Algoritmik adalet bu yüzden açık tercihler yapmayı gerektirir.

HITS algoritması web sayfalarını merkezler (*hubs*) ve otoriteler (*authorities*) olarak sıralar. Easley ile yazdığı *Networks, Crowds, and Markets* (2010) kitabı sponsorlu aramayı, eşleştirme piyasalarını ve Braess paradoksunu da kapsar.

TR

Tim Roughgarden

d. 1975 · Amerikalı · Stanford University, Columbia University

Bencil yönlendirme · pürüzsüzlük

Tardos ile bencil yönlendirmede Anarşinin Bedeli için sıkı sınırlar kanıtladı (2000–02). Daha sonra pürüzsüz oyunları (*smooth games*) tanıttı (2009).

Ödüller: Gödel Prize 2012 (Éva Tardos ile)

Anarşinin Bedeli · pürüzsüzlük · bencil yönlendirme

DERS BÖLÜMLERİ: **Bölüm 2 · Oyunlar ve Temsiller** · **Bölüm 3 · Nash Dengesi** · **Bölüm 4 · Mekanizma Tasarımı** · **Bölüm 5 · Açık Artırmalar** · **Bölüm 8 · Ağ Tasarımı ve Yönlendirme** · **Bölüm 9 · Anarşinin Bedeli** · **Bölüm 13 · Öğrenme ve Adaptasyon** · **Bölüm 14 · Gelecek Yönelimler**

Önemi: Pürüzsüzlük argümanları, Anarşinin Bedeli sınırlarını karışık, korelasyonlu ve pişmanlıksız sonuçlara genişletir.

Twenty Lectures on Algorithmic Game Theory (2016) kitabı yaygın bir ders kaynağıdır. Son çalışmaları blokzincir işlem ücreti mekanizmalarını inceler.

JH

Jason Hartline

Amerikalı · Northwestern University

Mekanizma tasarımı ve yaklaşım

Optimal geliri yaklaşık olarak sağlayan basit ve ön olasılıktan bağımsız (*prior-independent*) mekanizmalar geliştirdi.

ön olasılıktan bağımsız mekanizmalar · yaklaşım · gelir

DERS BÖLÜMLERİ: **Bölüm 4 · Mekanizma Tasarımı** · **Bölüm 5 · Açık Artırmalar**

Önemi: Basit açık artırmaların (örneğin anonim rezerv fiyatlı ikinci fiyat açık artırması) ne zaman yaklaşık optimal olduğunu gösterir. Böylece Myerson kuramıyla uygulama arasında köprü kurar.

Mechanism Design and Approximation kitabı, Bayesçi mekanizma tasarımını algoritmik bir bakışla geliştirir.

DS

David Silver & Demis Hassabis

d. 1976 · d. 1976 · Google DeepMind

AlphaGo · 2016

Silver'ın yönettiği AlphaGo projesi, Monte Carlo ağaç aramasını derin sinir ağlarıyla birleştirdi. AlphaGo, Mart 2016'da Lee Sedol'u 4–1 yendi (*Nature*, 2016). Hassabis, DeepMind'ı 2010'da kuranlardandır.

Ödüller: Silver: ACM Prize in Computing 2019 · Hassabis: Nobel Kimya Ödülü 2024 (John M. Jumper ile, protein yapısı tahmini için)

Monte Carlo ağaç araması · sinir ağıyla değerlendirme · kendi kendine oynama

DERS BÖLÜMLERİ **Bölüm 2 · Minimaks ve Alfa-Beta Budaması** **Bölüm 13 · Öğrenme ve Adaptasyon**

Önemi: Go'da dallanma etkeni yaklaşık 250'dir ve iyi bir değerlendirme fonksiyonu elle yazılamaz. AlphaGo değerlendirmeyi ve hamle seçimini öğrenilmiş ağlara bıraktı.

Ardından gelen AlphaZero, insan oyunu kullanmadan, yalnızca kendi kendine oynayarak Go, satranç ve şogi öğrendi. Arama fikri yine minimaksla aynıdır: rakibin en iyi cevabını varsayarak ileriye bakmak.

AP

Ariel Procaccia

İsrailli-Amerikalı · Carnegie Mellon University, Harvard University

Hesaplamalı sosyal seçim · adil bölüşüm

Algoritmaları oylamaya ve adil bölüşüme uyguladı. Kira, ortak işteki katkı payı ve görevlerin kanıtlanabilir biçimde adil bölüşümü için kurulan Spliddit sitesinin yaratıcılarındandır.

Ödüller: IJCAI Computers and Thought Award 2015

kıskançsızlık · hesaplamalı sosyal seçim · yurttaş meclisleri

DERS BÖLÜMLERİ **Bölüm 7 · Sosyal Seçim ve Oylama** **Bölüm 11 · Algoritmik Adalet ve Etik**

Önemi: Bir mala kadar kıskançsızlık (*envy-free up to one good*, EF1), en büyük Nash sosyal refahı ve leksimin tahsisleri, adalet güvencelerini uygulanabilir kılar.

Kurayla temsilî yurttaş meclisleri seçmek için de algoritmalar tasarladı.

CD

Constantinos Daskalakis

d. 1981 · Yunan · MIT

Nash dengesi PPAD-tamdır · 2006

Goldberg ve Papadimitriou ile Nash dengesini hesaplamanın PPAD-tam olduğunu kanıtladı (STOC 2006; SIAM J. Computing 2009).

Ödüller: Nevanlinna Prize 2018

PPAD-tamlık · denge hesaplama · min-max optimizasyonu

DERS BÖLÜMLERİ **Bölüm 2 · Oyunlar ve Temsiller** **Bölüm 3 · Nash Dengesi** **Bölüm 14 · Gelecek Yönelimler**

Önemi: PPAD kolay değilse genel oyunlarda Nash dengesini bulan verimli bir algoritma yoktur. Bu sonuç, dengenin bir öngörü olarak kullanılmasını sorgulatır.

DGP ispatı üç ve daha fazla oyunculu oyunları kapsar. Chen ve Deng (2006) zorluğu iki oyunculu oyunlara genişletti. Son çalışmaları min-max optimizasyonunda öğrenme dinamiklerini ve çok ajanlı öğrenmeyi inceler.

EOS

Benjamin Edelman, Michael Ostrovsky & Michael Schwarz

Harvard, Stanford ve Yahoo! Research (2007'deki kurumları)

Genelleştirilmiş ikinci fiyat açık artırması · 2007

Arama reklamlarında kullanılan genelleştirilmiş ikinci fiyat (GSP) açık artırmasını incelediler (American Economic Review, 2007).

GSP · sponsorlu arama · kıskançsız denge

DERS BÖLÜMLERİ **Bölüm 6 · Çevrim İçi Platformlar**

Önemi: GSP dürüst değildir. Yine de en düşük gelirli yerel kıskançsız dengesi, VCG ile aynı yerleşimleri ve ödemeleri verir.

Varian'dan bağımsız olarak, dürüst teklif baskın strateji olmadığı hâlde GSP fiyatlarının uygulamada neden kararlı kaldığını açıkladılar.

MC

Rémi Coulom, Levente Kocsis & Csaba Szepesvári

Coulom: Fransız · Kocsis ve Szepesvári: MTA SZTAKI, Budapeşte (2006'daki kurumları)

Monte Carlo ağaç araması ve UCT · 2006

Coulom 2006'da oyun ağacı aramasını rastgele oynanan oyunlarla birleştirdi ve yönteme Monte Carlo ağaç araması (MCTS) adını verdi. Kocsis ve Szepesvári aynı yıl haydut algoritması UCB1'i ağaca uygulayan UCT'yi önerdi (ECML 2006).

MCTS · UCT · keşif ve sömürü

DERS BÖLÜMLERİ **Bölüm 2 · Minimaks ve Alfa-Beta Budaması**

Bölüm 13 · Öğrenme ve Adaptasyon

Önemi: MCTS, iyi bir değerlendirme fonksiyonu yazılamayan oyunlarda konumları rastgele oyunların sonuçlarıyla değerlendirir. Güçlü Go programlarının ve AlphaGo'nun arama iskeleti bu yöntemdir.

Coulom'un Crazy Stone programı 2006 Bilgisayar Olimpiyatı'nda 9×9 Go turnuvasını kazandı. UCT, her düğümde hamle seçimini çok kollu haydut problemi gibi ele alır.

Ödül yılları, ödülün verildiği yılı gösterir. “Nobel Ekonomi Ödülü” ifadesi, Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel için kullanılır. Sonuçları birlikte anlatılan kişiler ortak kartta toplanmıştır. Doğum yılı gösterilmeyen kartlarda sıralama, ana çalışmanın dönemine göre yapılır.