Bölüm 12: Davranışsal ve Kolektif Karar

Algoritmik Oyun Kuramı

Beklenti kuramı | Kayıptan kaçınma | Kuantal tepki | Sosyal tercihler | Ortak mallar ve Ostrom | Kamu malı ve ceza | Eşikli projeler | Bilgi şelaleleri | Condorcet jüri teoremi

Sercan KÜLCÜ · Giresun Üniversitesi · Bilgisayar Mühendisliği

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/agt/slides/Bolum_12_Davranissal_ve_Kolektif_Karar.html

DAVRANIŞSAL KARAR

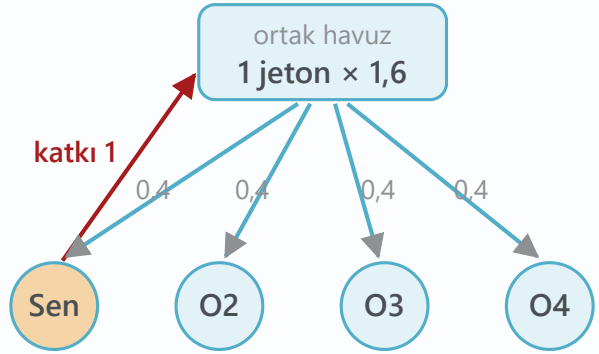
Oyunla Başlangıç: Kamu Malı Oyunu

Oyunun kuralı

Dört oyuncu her tur 10 jeton alır ve dilediği kadarını ortak havuza katar; havuz 1,6 ile çarpılıp dördüne eşit bölünür, katılmayan jeton oyuncuda kalır.

Merak sorusu

- Sizce dört kişilik bir grupta ilk turda ortalama kaç jeton katılır, on tur sonra kaç?
- Katkısı 3 jetondan az olana 2 jeton ceza kesilirse (oyundaki Cezalı katkı senaryosu) katkılar nasıl değişir?



Sana net −1 + 0,4 = −0,6; diğer üçüne toplam +1,2

Kamu Malı Oyunu →

DAVRANIŞSAL KARAR

Öğrenme Çıktıları

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.

DAVRANIŞSAL KARAR

İnsanlar Her Zaman Rasyonel mi? Sınırlı Rasyonellik

- **Sınırlı rasyonellik** (bounded rationality; Simon): bilgi, zaman ve hesaplama kapasitesi sınırlıdır; insanlar en iyiyi değil "yeterince iyi"yi arar.
- Sapmalar rastgele değil **sistematiktir**: aynı hatalar tekrar tekrar gözlenir, bu yüzden modellenebilir.
- Nedenler: bilgi eksikliği, zaman baskısı, duygular, kısa yollar (sezgisel kurallar).

Aşırı güven
Kendi tahminlerine gereğinden çok güvenmek: yatırımcıların aşırı işlem yapması.

Çerçeveleme etkisi
Aynı seçenek farklı sunulunca kararın değişmesi.

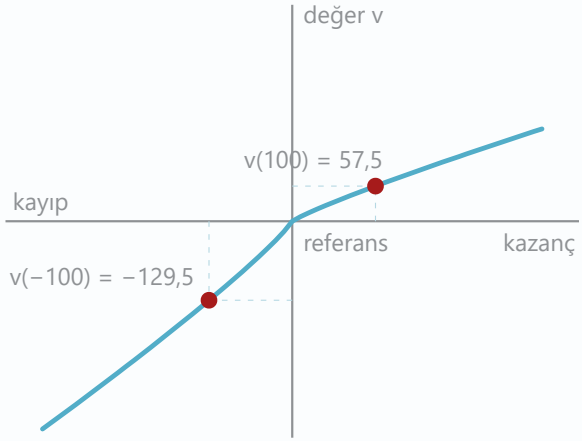
Kayıptan kaçınma
Eşit büyüklükteki kaybın kazançtan daha güçlü hissedilmesi: panik satışı.

Taban oranı ihmali
Pozitif bir test sonucunu yorumlarken hastalığın yaygınlığını (ön olasılığı) yok saymak.

DAVRANIŞSAL KARAR

Beklenti Kuramı (Prospect Theory)

- Kahneman ve Tversky (1979): sonuçlar toplam servete göre değil, bir **referans noktasına** göre kazanç/kayıp olarak değerlendirilir.
- Değer fonksiyonu kazançta içbükey, kayıpta dışbükey, kayıpta **daha diktir**:
  $v(x) = x^{0,88}$ $(x \geq 0)$; $v(x) = -2,25 \cdot (-x)^{0,88}$ $(x < 0)$
- $x^{0,88}$, x'ten yavaş büyür: $v(200) \approx 105,9$, $v(100)$'ün 2 değil 1,84 katı. İçbükey: eğri sağa doğru yatıklaşır.
- $\lambda = 2,25$ **kayıptan kaçınma** (loss aversion) katsayısı (Tversky ve Kahneman, 1992 tahmini).
- Sonuç: kazançlarda riskten kaçınan, kayıplarda risk seven davranış.



Referans noktası örneği: 10.000 TL ikramiye bekleyen biri 9.000 TL alınca bunu 1.000 TL'lik bir **kayıp** gibi yaşar; hiç beklemeyen biri için aynı tutar büyük bir kazançtır.

DAVRANIŞSAL KARAR

Kayıptan Kaçınma: Yazı-Tura Bahsi

Bahis
Yazı gelirse +100 TL, tura gelirse −100 TL. Beklenen değer 0.

Beklenti değeri
½·v(100) + ½·v(−100) = ½·57,5 − ½·129,5 = **−36,0** < 0 ⇒ reddedilir.

Kabul eşiği
$\frac{1}{2} \cdot G^{0,88} = \frac{1}{2} \cdot 2,25 \cdot 100^{0,88} \Rightarrow (G/100)^{0,88} = 2,25 \Rightarrow G = 100 \cdot 2,25^{1/0,88} \approx$ **251 TL**.

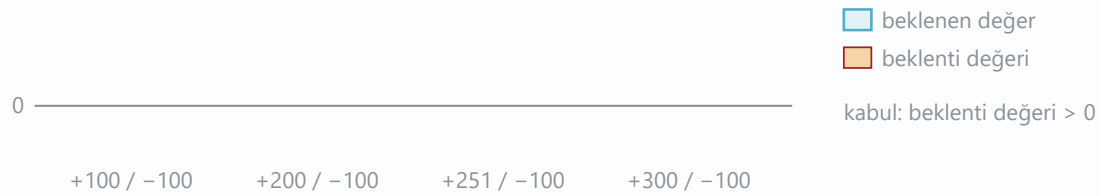
100 TL kaybetme riskini göze almak için yaklaşık **251 TL** kazanç ihtimali istenir. Deneylerde de insanlar tipik olarak kaybın iki katından fazlasını ister.

Olasılık ağırlığı w(p)	**p = 0,01**	**p = 0,50**	**p = 0,99**
Algılanan ağırlık	0,055	0,421	0,912

Olasılık ağırlığı w(p), bir olasılığın nasıl hissedildiğini gösterir (Tversky ve Kahneman, 1992 tahmini): %1 yaklaşık %5,5, %50 yaklaşık %42, %99 yaklaşık %91 gibi algılanır. Küçük olasılıklar abartılır, büyükler küçümsenir: aynı kişi hem piyango bileti hem sigorta alabilir.

DAVRANIŞSAL KARAR

Kod İzleme: Bahis Ne Zaman Kabul Edilir?



Dört yazı-tura bahsi: ½ olasılıkla +G kazanç, ½ olasılıkla 100 TL kayıp. Beklenen değer (G − 100)/2; beklenti değeri v ile hesaplanır.

Adım 1 / 6

```
1 def v(x):
2     if x >= 0: return x ** 0.88
3     return -2.25 * (-x) ** 0.88
4 def beklenti(G, K):      # ½: +G, ½: −K
5     return 0.5 * v(G) + 0.5 * v(-K)
▶ for G in [100, 200, 251, 300]:
7     print(G, round(beklenti(G, 100), 1))
```

DEĞİŞKENLER

`G = –` `K = 100`

ÇIKTI

DAVRANIŞSAL KARAR

Kod İzleme: Bahis Ne Zaman Kabul Edilir?



G = 100: v(100) = 57,5 ve v(−100) = −129,5 ⇒ ½·57,5 − ½·129,5 = **−36,0**. Beklenen değer 0 olan bahis reddedilir.

Adım 2 / 6

```
1 def v(x):
2     if x >= 0: return x ** 0.88
3     return -2.25 * (-x) ** 0.88
4 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))
```

DEĞİŞKENLER

`G = 100` `v(G) = 57.5` `v(−K) = −129.5` `deger = −36.0`

ÇIKTI

```
100 -36.0
```

DAVRANIŞSAL KARAR

Kod İzleme: Bahis Ne Zaman Kabul Edilir?



G = 200: v(200) = 105,9 ⇒ ½·105,9 − ½·129,5 = **−11,8**. Beklenen değer +50 olduğu hâlde hâlâ reddedilir.

Adım 3 / 6

```
1 def v(x):
2     if x >= 0: return x ** 0.88
3     return -2.25 * (-x) ** 0.88
4 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))
```

DEĞİŞKENLER

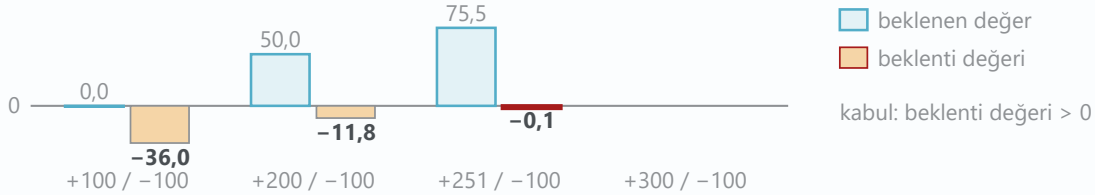
`G = 200` `v(G) = 105.9` `v(−K) = −129.5` `deger = −11.8`

ÇIKTI

```
100 -36.0
200 -11.8
```

DAVRANIŞSAL KARAR

Kod İzleme: Bahis Ne Zaman Kabul Edilir?



G = 251: v(251) = 129,3 ⇒ ½·129,3 − ½·129,5 ≈ **−0,1**: kabul eşiği (≈ 251 TL) tam buradadır.

Adım 4 / 6

```
1 def v(x):
2     if x >= 0: return x ** 0.88
3     return -2.25 * (-x) ** 0.88
4 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))
```

DEĞİŞKENLER

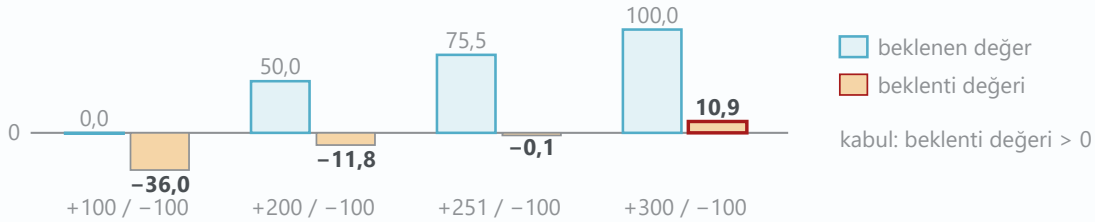
`G = 251` `v(G) = 129.3` `v(−K) = −129.5` `deger = −0.1`

ÇIKTI

```
100 -36.0
200 -11.8
251 -0.1
```

DAVRANIŞSAL KARAR

Kod İzleme: Bahis Ne Zaman Kabul Edilir?



G = 300: v(300) = 151,3 ⇒ ½·151,3 − ½·129,5 = **+10,9** > 0: bahis artık kabul edilir.

Adım 5 / 6

```
1 def v(x):
2     if x >= 0: return x ** 0.88
3     return -2.25 * (-x) ** 0.88
4 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))
```

DEĞİŞKENLER

`G = 300` `v(G) = 151.3` `v(−K) = −129.5` `deger = 10.9`

ÇIKTI

```
100 -36.0
200 -11.8
251 -0.1
300 10.9
```

DAVRANIŞSAL KARAR

Kod İzleme: Bahis Ne Zaman Kabul Edilir?



Risk nötr biri +200 / −100 bahsini (beklenen değer +50) hemen kabul eder; λ = 2,25 ile kayıp o kadar ağır gelir ki kazanç 251 TL'yi aşmadan bahis kabul edilmez.

Adım 6 / 6

```
1 def v(x):
2     if x >= 0: return x ** 0.88
3     return -2.25 * (-x) ** 0.88
4 def beklenti(G, K):      # ½: +G, ½: −K
5     return 0.5 * v(G) + 0.5 * v(-K)
6 for G in [100, 200, 251, 300]:
▶     print(G, round(beklenti(G, 100), 1))
```

DEĞİŞKENLER

`G = 300`

ÇIKTI

```
100 -36.0
200 -11.8
251 -0.1
300 10.9
```

DAVRANIŞSAL KARAR

Kayıptan Kaçınmanın İzi: Bağış Etkisi

- **Deney** (Kahneman, Knetsch ve Thaler, 1990): öğrencilerin yarısına rastgele bir kupa verildi. Kupa sahiplerinin istediği en düşük satış fiyatı, almak isteyenlerin ödemeye razı olduğu en yüksek fiyatın yaklaşık **iki katıydı**; beklenenden çok daha az takas oldu.
- **Açıklama:** kupayı vermek referans noktasından bir **kayıp**, kupayı almak bir **kazançtır**; kayıp λ kat ağır gelir.
- **Statüko yanlılığı:** varsayılandan ayrılmak da kayıp gibi algılanır; emeklilik planına otomatik katılım bu yüzden katılımı artırır.

Örnek

Kupanın kullanım değeri 20 TL, değer fonksiyonu doğrusal, $\lambda = 2{,}25$.

- Sahip: kupayı vermek $-2{,}25 \cdot 20 = -45$ hissettirir $\Rightarrow$ en az **45 TL** ister.
- Alıcı: kupayı almak $+20 \Rightarrow$ en çok **20 TL** öder.
- Oran $45/20 = 2{,}25 = \lambda$: aradaki fiyatlarda takas olmaz.

Sık hata

"Sahibi kupayı daha çok sever." Hayır: nesne aynıdır, değişen yalnızca referans noktasıdır.

DAVRANIŞSAL KARAR

Çerçeveleme Etkisi: Aynı Seçenek, Farklı Karar

Tversky ve Kahneman (1981): 600 kişiyi etkilemesi beklenen bir salgın için iki program.

Kazanç çerçevesi

+ A: 200 kişi **kurtulur**.
+ B: 1/3 olasılıkla 600 kişi kurtulur, 2/3 olasılıkla kimse kurtulmaz.
+ Katılımcıların **%72**'si A'yı (kesin) seçti.

Kayıp çerçevesi

– C: 400 kişi **ölür**.
– D: 1/3 olasılıkla kimse ölmez, 2/3 olasılıkla 600 kişi ölür.
– Katılımcıların **%78**'i D'yi (riskli) seçti.

A ≡ C ve B ≡ D (beklenen kurtulan hepsinde 200). Beklenen fayda kuramına göre tercih değişmemeli; beklenti kuramı ise bunu kazançta riskten kaçınan, kayıpta risk seven davranışla açıklar.

DAVRANIŞSAL KARAR

Sık Yapılan Hatalar: Beklenti Kuramı

"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.

"w(p) bir olasılıktır."

Hayır, bir karar ağırlığıdır: w(0,01) + w(0,99) = 0,055 + 0,912 = 0,967 ≠ 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.

DAVRANIŞSAL KARAR

Hatalı ama Akıllı: Kuantal Tepki

- İnsanlar daha iyi seçeneği daha sık seçer ama her zaman değil; fark büyüdükçe hata azalır.
- **Logit seçim:** $P(a) = e^{\lambda \cdot u(a)} / \Sigma_b\, e^{\lambda \cdot u(b)}$ ($e \approx 2{,}718$; e^x, x büyüdükçe hızla büyür). $\lambda = 0$ iken seçim rastgele, $\lambda \to \infty$ iken en iyi tepki.
- Buradaki λ, kayıptan kaçınmadaki $\lambda = 2{,}25$ ile ilgisizdir; seçimin ne kadar "keskin" olduğunu ölçer.
- İki seçenekte formül sadeleşir: $P(\text{iyi}) = 1 / (1 + e^{-\lambda \cdot \text{fark}})$. Fark 2 ve $\lambda = 1$ iken $P \approx 0{,}88$.
- **Kuantal tepki dengesi** (quantal response equilibrium, QRE; McKelvey ve Palfrey, 1995): her oyuncu, diğerlerinin bu hatalı seçimlerine karşı logit tepki verir; $\lambda \to \infty$ iken Nash dengesine gider.

λ	P(kazanç 10)
0	0,500
0,5	0,731
1	0,881
2	0,982
∞	1

İki seçenek, kazançlar 10 ve 8 (fark 2): $P = 1/(1 + e^{-2\lambda})$

Ayrı bir model, seviye-k düşünme: hatayı değil akıl yürütme derinliğini sınırlar; seviye-k oyuncu diğerlerini seviye $k-1$ sayar (Nagel, 1995).

DAVRANIŞSAL KARAR

Kuantal Tepki: Asimetrik Yazı-Tura

	Sol	Sağ
Üst	320, 40	40, 80
Alt	40, 80	80, 40

- Sütunu kayıtsız bırakan p: $40p + 80(1 - p) = 80p + 40(1 - p) \Rightarrow$ **p = 1/2**.
- Satırı kayıtsız bırakan q: $320q + 40(1 - q) = 40q + 80(1 - q) \Rightarrow$ **q = 1/8**.
- Nash: satırın kendi 320'si ne olursa olsun Üst olasılığı 1/2 kalır.

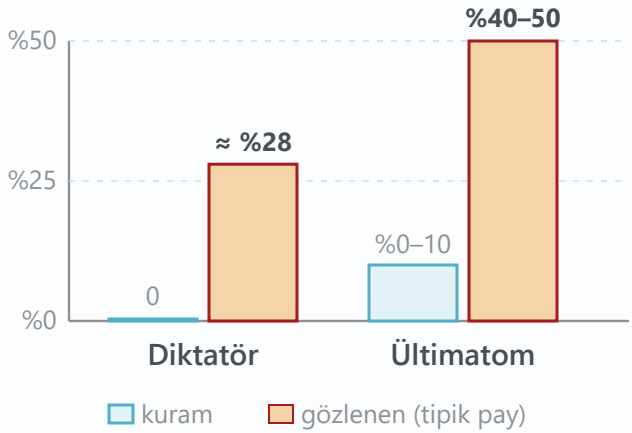
λ	p = P(Üst)	q = P(Sol)
0	0,50	0,50
0,02	0,83	0,37
0,1	0,71	0,15
1	0,52	0,125
∞ (Nash)	0,50	0,125

Değerler, iki logit denkleminin birlikte (bilgisayarla) çözülmesiyle bulunur. λ küçükken satır 320'ye kapılıp Üst'ü sık oynar (0,83). λ büyüdükçe sütunun Sağ'a kaçtığını hesaba katar ve 1/2'ye döner.

Deneylerde satır oyuncuları büyük kazanca (320) doğru açıkça kaydı (Goeree ve Holt, 2001). QRE bu **kendi kazancı etkisini** orta λ değerleriyle yakalar. Sık hata: karışık Nash'te bir oyuncunun olasılığı *rakibinin* kazançlarından belirlenir; insanlar ise kendi kazancına da tepki verir.

Sosyal Tercihler: Yalnızca Para mı?

- **Diktatör oyunu:** bir kişi 10 jetonu kendisiyle karşısındaki arasında böler; karşıdaki hiçbir şey yapamaz. Yalnızca parayı önemseyen biri 0 verir.
- Yüzü aşkın çalışmanın meta-analizinde verilen pay ortalama **yaklaşık %28**; azımsanmayacak bir kesim de hiç vermez (Engel, 2011).
- **Ültimatom oyunu:** karşı tarafa ret hakkı verilince teklifler %40–50'ye çıkar, düşük teklifler reddedilir.



Diktatörde verilen pay adalet ya da özgecilikten gelir; ültimatomdaki ek pay ise ret korkusundan (stratejik). **Sosyal tercih** modelleri bu güdüleri fayda fonksiyonunun içine yazar.

SOSYAL TERCİHLER

Eşitsizlikten Kaçınma: Fehr–Schmidt Modeli

İki kişi için: $u_i = x_i - \alpha \cdot \max(x_j - x_i, 0) - \beta \cdot \max(x_i - x_j, 0)$, $0 \le \beta \le \alpha$, $\beta < 1$ (Fehr ve Schmidt, 1999)

- **α**: geride kalmanın acısı (kıskançlık); **β**: önde olmanın rahatsızlığı (suçluluk). İnsanlar geride kalmaktan daha çok rahatsız olur: $\alpha \ge \beta$.
- $\alpha = \beta = 0$ ise klasik, yalnızca parayı önemseyen oyuncu elde edilir.

Ültimatom

10'dan s pay alan yanıtlayan için kabul: $s - \alpha(10 - 2s) \ge 0 \Leftrightarrow$ **$s \ge 10\alpha/(1 + 2\alpha)$**. $\alpha = 1 \Rightarrow s \ge 3{,}33$.

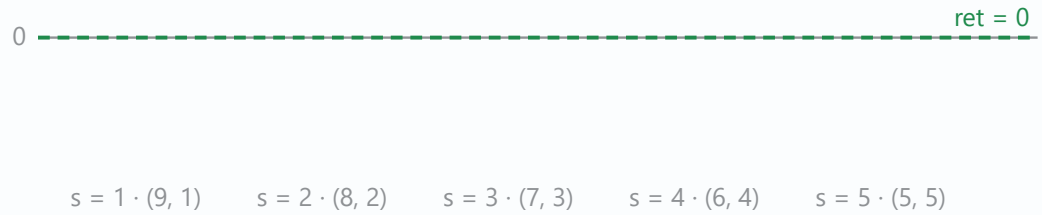
Diktatör

$s \le 5$ verirse önde kalır, yalnız β terimi çalışır: $u = (10 - s) - \beta \cdot (10 - 2s) = 10 - 10\beta + s(2\beta - 1)$. **$\beta > 1/2$** ise eşit böler, değilse hiç vermez.

Kamu malı

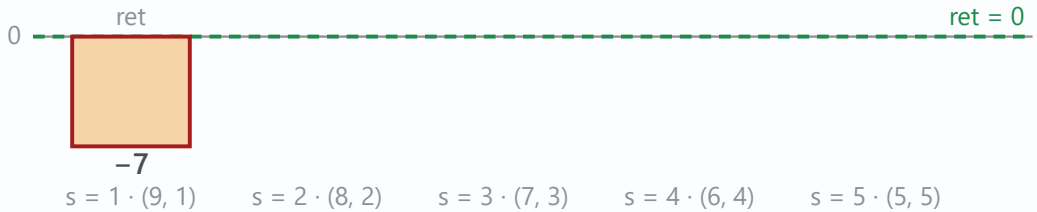
Yeterince eşitsizlikten kaçınan oyuncu, diğerleri katıyorsa katar, katmıyorsa katmaz: koşullu işbirlikçi.

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)



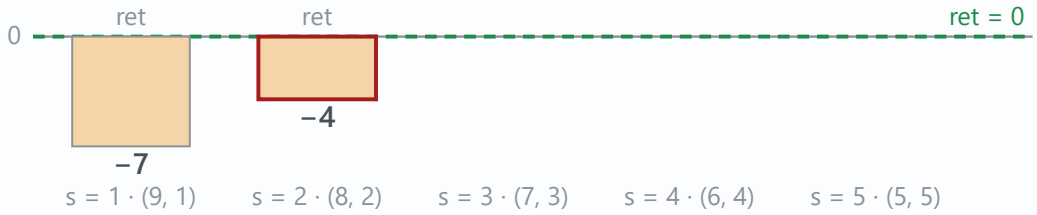
Yanıtlayan α = 1 ile kabulde u = s − 1·(10 − 2s) alır, retle 0. Teklif eden (10 − s, s) önerir.

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)



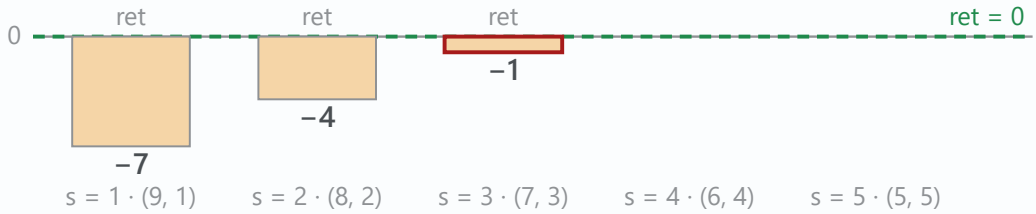
s = 1: u = 1 − 8 = **−7** < 0 ⇒ ret. 1 jeton, 8 jetonluk farkın acısını karşılamaz.

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)



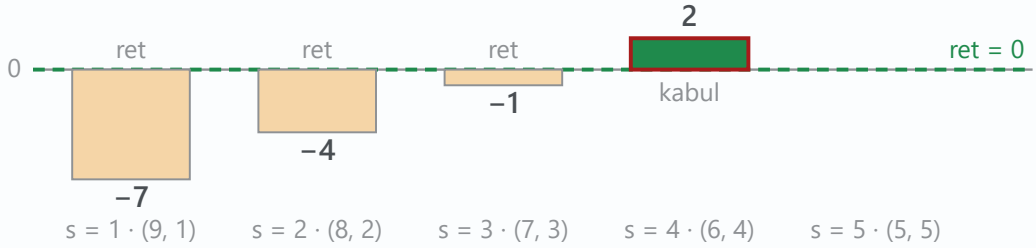
s = 2: u = 2 − 6 = **−4** ⇒ ret.

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)



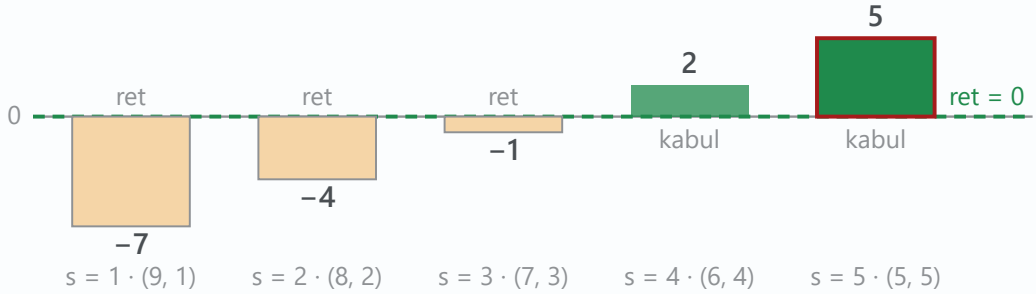
s = 3: u = 3 − 4 = **−1** ⇒ hâlâ ret (kıl payı).

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)



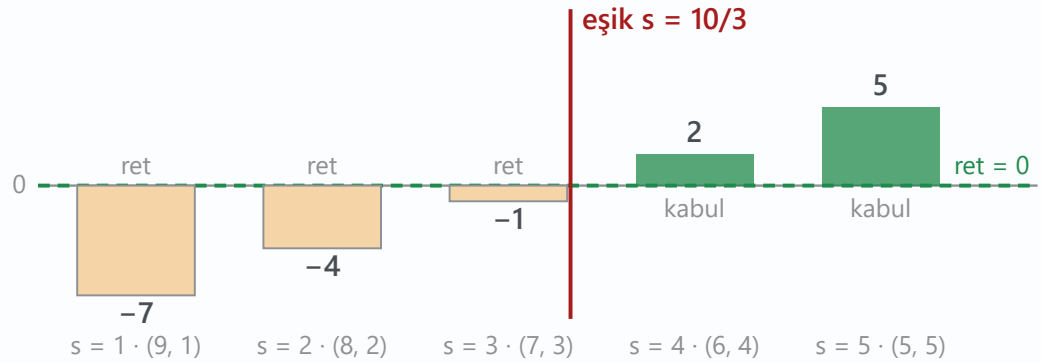
s = 4: u = 4 − 2 = **+2** > 0 ⇒ kabul.

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)



s = 5: u = **5**, fark yok ⇒ kabul.

Adım Adım: Ültimatomda Kabul Eşiği (α = 1)

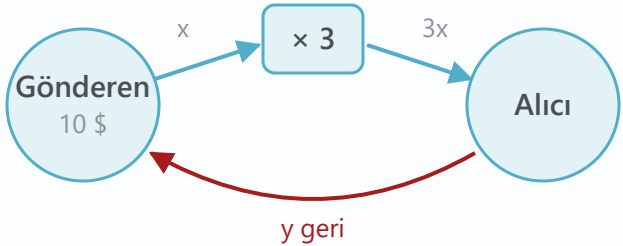


Eşik $10\alpha/(1 + 2\alpha) = 10/3 \approx 3{,}33$: en küçük kabul edilen tam sayı teklif 4. Teklifi bilen teklif eden 4 önerip 6 alır: düşük teklifin reddi, adalet duygusunun sonucudur.

SOSYAL TERCİHLER

Karşılıklılık: Güven Oyunu

- **Güven oyunu** (Berg, Dickhaut ve McCabe, 1995): gönderen 10 doların istediği kadarını gönderir, miktar üçe katlanır, alıcı istediği kadarını geri yollar. Tek atışta ve tam gizlilikte oynandı.
- **Kuram** (yalnızca para): alıcı hiçbir şey geri vermez; bunu bilen gönderen 0 gönderir.
- **Gerçekte** ortalama 5,16 $ gönderildi, 4,66 $ geri döndü.
- **Karşılıklılık:** niyete yanıt vermek; iyiliğe iyilik, kötülüğe ceza (Rabin, 1993). Eşitsizlikten kaçınma sonuçların dağılımına, karşılıklılık niyete bakar.



ortalama: x = 5,16 · 3x = 15,48 · y = 4,66

Sık hata: "Güvenmek kârlıydı." Ortalamada geri dönen (4,66) gönderilenden (5,16) azdı; buna rağmen çoğu kişi gönderdi.

KOLEKTİF KARARLAR

Kolektif Karar Alma: Bireysel ve Toplumsal Çıkar

- Grup kararlarında **sosyal etki**, uyma baskısı ve ortak çıkarlar rol oynar.
- Topluluk kararları çoğunluk, uzlaşı (konsensüs) ya da oy sistemleriyle alınır.
- Bireysel çıkar kendi faydasını, kolektif çıkar grubun faydasını maksimize eder; ikisi çoğu zaman **çelişir**.

Trafik

Herkes kendine en kısa yolu seçer → toplam gecikme gereğinden büyük olur (bencil yönlendirme; Pigou örneği, anarşinin bedeli).

İş grupları

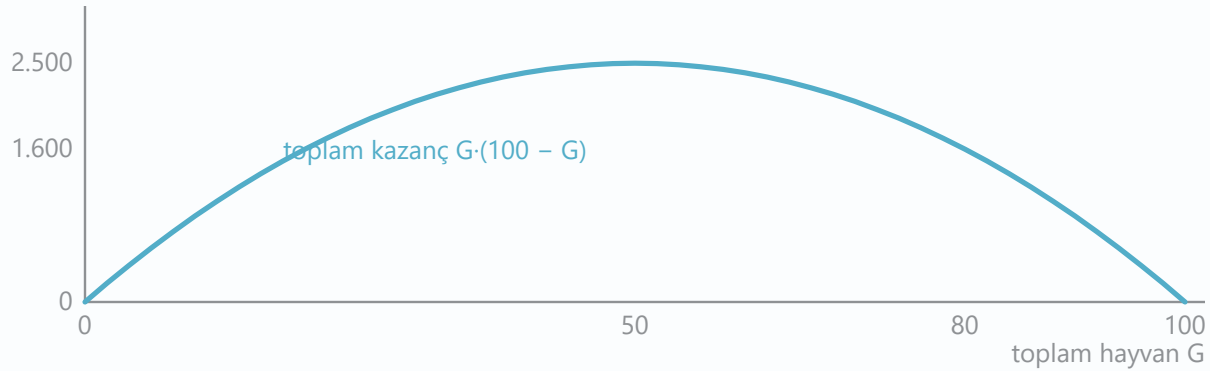
Herkes kendi işine odaklanır → ortak proje aksar (kamu malı).

Ortak kaynak

Balıkçılık, mera, yeraltı suyu, atmosfer → aşırı kullanım.

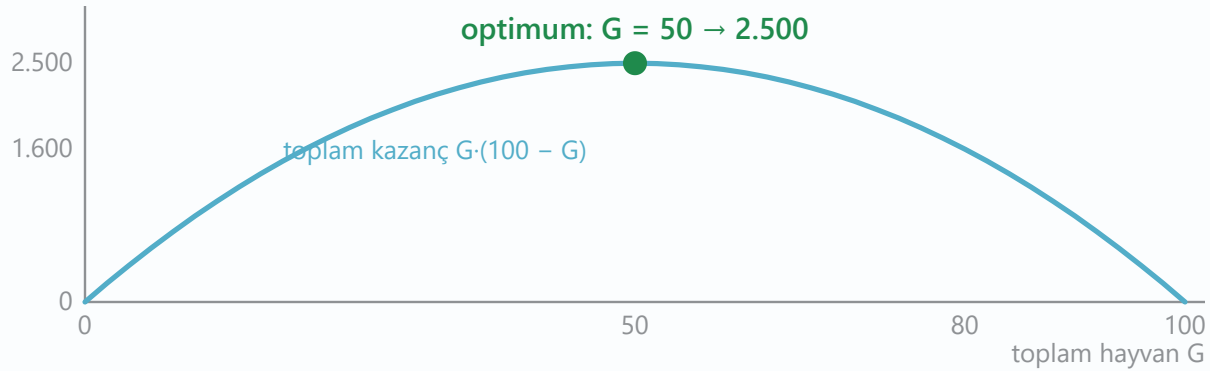
Son örneğin klasik adı: **ortak mallar trajedisi** (tragedy of the commons, Hardin 1968).

Ortak Mallar Trajedisi: Sayısal Model



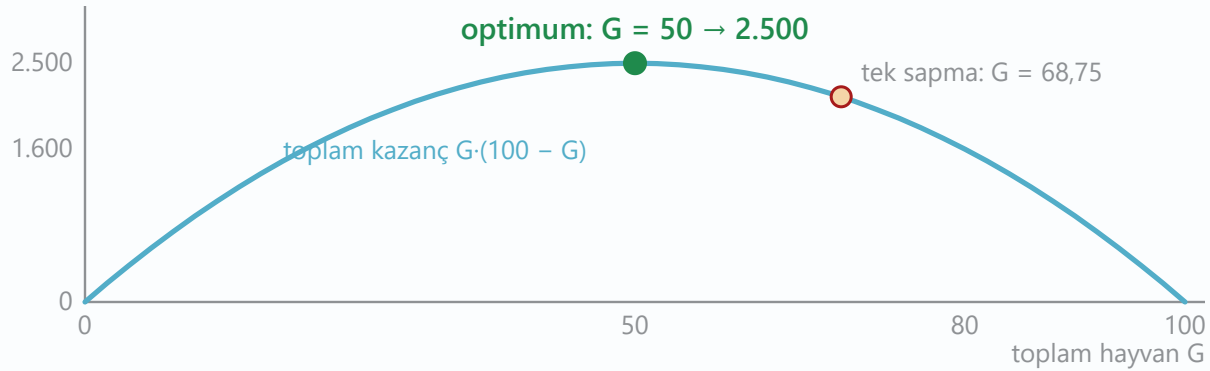
4 çoban ortak meraya hayvan salar. Hayvan başı değer 100 − G (G = toplam hayvan): mera kalabalıklaştıkça her hayvan daha az besi alır. Çoban i'nin kazancı $u_i = g_i(100 - G)$.

Ortak Mallar Trajedisi: Sayısal Model



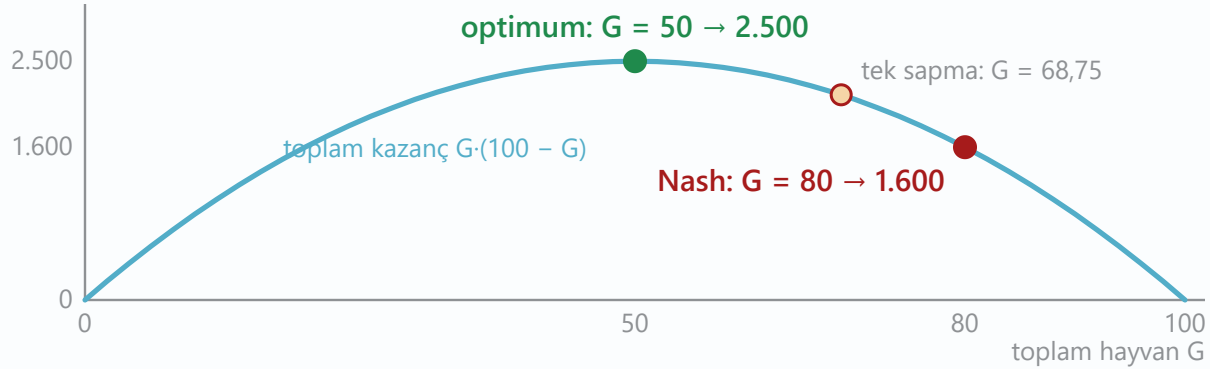
Sosyal optimum: G(100 − G) en büyük ⇒ G = 50, toplam 2.500. Eşit paylaşımda her çoban 12,5 hayvan, kazanç 625.

Ortak Mallar Trajedisi: Sayısal Model



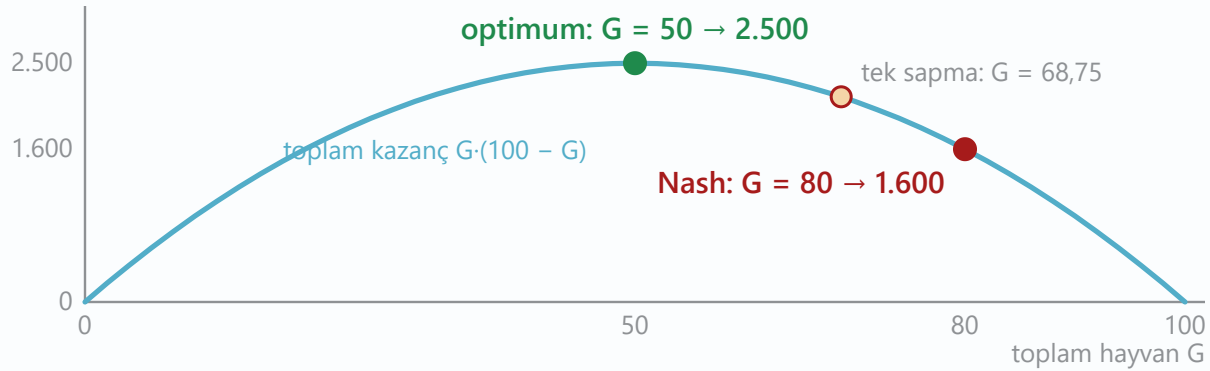
Ama diğer üç çoban toplam 37,5 hayvan (her biri 12,5) salarken çoban 1'in kazancı g·(100 − 37,5 − g) olur. Türevi 62,5 − 2g = 0 ⇒ g = 31,25: kazancı 976,6 > 625. Diğerlerinin kazancı 390,6'ya düşer.

Ortak Mallar Trajedisi: Sayısal Model



Herkes en iyi tepki verince: g = (100 − 3g)/2 ⇒ **g* = 20**, G = 80. Kontrol: diğerleri 60 iken en iyi tepki (100 − 60)/2 = 20. Kişi başı 400, toplam 1.600. Çobanlar sırayla en iyi tepki verse de bu dengeye yakınsarlar.

Ortak Mallar Trajedisi: Sayısal Model



Trajedi: dengede toplam kazanç 1.600, optimumun %64'i. Her çoban kendi hayvanının meraya verdiği zararın yalnızca 1/4'ünü hisseder (dışsallık).

Çoban Sayısı Arttıkça ve Çözümler

n	g* (kişi başı)	G*	toplam	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

Dengede $g^* = 100/(n + 1)$; $n \to \infty$ iken $G^* \to 100$, toplam $\to 0$

Pigou vergisi

Vergiyle çobanın kazancı $g \cdot (100 - G - t)$ olur; aynı hesap simetrik dengede $g = (100 - t)/5$ verir. $t = 37{,}5 \Rightarrow g = 12{,}5$, $G = 50$.

Kota / izin

Toplam 50 hayvan hakkı paylaştırılır; ticareti yapılabilir.

Ostrom (2009 Nobel)

Yerel topluluklar izleme, kademeli yaptırım ve ortak kurallarla kaynağı sürdürebilir.

KOLEKTİF KARARLAR

Devlet mi, Piyasa mı? Ostrom'un Üçüncü Yolu

Hardin'e göre çözüm ya merkezî denetim ya özelleştirmedir. Ostrom (1990), yüzyıllardır süren yerel düzenlerde ortak ilkeler buldu (2009 Nobel).

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.

5. Kademeli yaptırım
İlk ihlale hafif, tekrarına ağır ceza.

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.

Ostrom'un örneklerinden biri Alanya'nın kıyı balıkçılarıdır: avlak yerleri sezon başında kurayla dağıtılır, tekneler balığın göçüyle sırayla yer değiştirir.

Kolektif Kararı Değerlendirmek: Cehalet Peçesi

Üç refah ölçütü

Faydacı ölçüt (SW = Σu_i), eşitlikçi ölçüt (min u_i) ve Nash sosyal refahı (Πu_i) aynı seçenekleri farklı sıralayabilir. Ortak meradaki optimum (G = 50, toplam 2.500) faydacı ölçütün en iyisidir.

Cehalet peçesi: politika, kimin hangi konumda olacağı bilinmeden seçilir. Her konuma eşit olasılık verip beklenen faydayı maksimize ederseniz (Laplace kuralı) **faydacı** ölçüte (Harsanyi, 1955), maksimin uygularsanız **eşitlikçi** ölçüte (Rawls, 1971) varırsınız.

Örnek

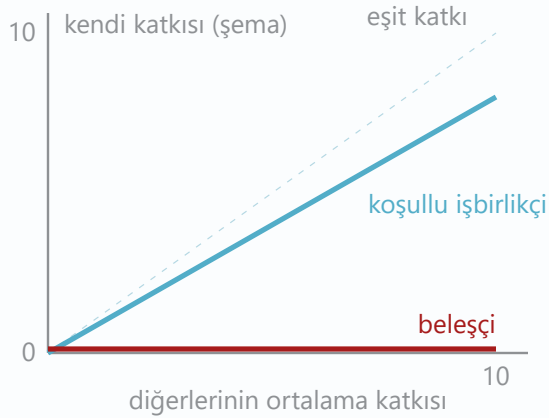
Üç konum: P = (10, 6, 2), R = (5, 5, 5). Laplace: P'nin ortalaması 6 > 5 ⇒ P. Maksimin: en kötü 2 < 5 ⇒ R.

Belirsizlik altında karar

Peçe, toplumsal seçimi bir karar kuramı problemine çevirir: aynı tablo, hangi belirsizlik ölçütü seçildiğine göre farklı politika önerir.

Laboratuvarda Kamu Malı: Ne Gözleniyor?

- Katılan her jeton katana 0,4, gruba 1,6 döndürür; katmamak baskındır, kuram 0 katkı öngörür.
- **İlk turlar:** bütçenin yaklaşık %40–60'ı katılır (Ledyard, 1995).
- **Tekrarla düşüş:** katkılar tur tur azalır, son turlarda beleşçilik düzeyine yaklaşır.
- **Kimler?** Deneklerin yaklaşık yarısı koşullu işbirlikçidir (diğerleri ne kadar katarsa o kadar katmak ister), yaklaşık üçte biri beleşçidir (Fischbacher, Gächter ve Fehr, 2001).
- **Düşüşün mekanizması:** koşullu işbirlikçiler ortalamanın biraz altında katar, beleşçiler ortalamayı aşağı çeker: katkı her tur biraz daha erir.



Kod İzleme: `kosullu_katki(c, tur)`



tur 0: katkılar

[0, 10, 10, 10]

kazanç u = 10 − k + 0,4·Σk

O1: 22 O2: 12

Model: O1 hep 0 verir; O2–O4 **koşullu işbirlikçi**: bu tur, diğerlerinin önceki turdaki ortalaması kadar verir.

Adım 1 / 12

```
1  def kosullu_katki(c, tur):
▶    gecmis = [c]
3    for t in range(tur):
4      yeni = [0]                  # O1 hep beleşçi
5      for i in range(1, len(c)):
6        digerleri = sum(c) - c[i]
7        yeni.append(digerleri / (len(c) - 1))
8      c = yeni
9      gecmis.append(c)
10   return gecmis
```

DEĞİŞKENLER

`c = [0, 10, 10, 10]`

`gecmis = [c]`

Kod İzleme: `kosullu_katki(c, tur)`



tur 0: katkılar

[0, 10, 10, 10]

kazanç u = 10 − k + 0,4·Σk

O1: 22 O2: 12

t = 0: O2 için diğerlerinin toplamı 20 ⇒ ortalama 20 / 3 = **6,7** (O3, O4 de aynı).

Adım 2 / 12

```
1  def kosullu_katki(c, tur):
2      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))
8          c = yeni
9          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 0` `c = [0, 10, 10, 10]` `digerleri = 20` `yeni = [0, 6.7, 6.7, 6.7]`

Kod İzleme: `kosullu_katki(c, tur)`



tur 1: katkılar

[0, 6.7, 6.7, 6.7]

kazanç u = 10 − k + 0,4·Σk

O1: 18 O2: 11,3

Tur 1: katkılar [0, 6.7, 6.7, 6.7]. Beleşçi O1, herkesten fazla kazanır.

Adım 3 / 12

```
1  def kosullu_katki(c, tur):
2      gecmis = [c]
3      for t in range(tur):
4          yeni = [0]                  # O1 hep beleşçi
5          for i in range(1, len(c)):
6              digerleri = sum(c) - c[i]
7              yeni.append(digerleri / (len(c) - 1))
▶          c = yeni
▶          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 0` `c = [0, 6.7, 6.7, 6.7]` `gecmis = 2 tur`

Kod İzleme: `kosullu_katki(c, tur)`



tur 1: katkılar

[0, 6.7, 6.7, 6.7]

kazanç u = 10 − k + 0,4·Σk

O1: 18 O2: 11,3

t = 1: O2 için diğerlerinin toplamı 13,3 ⇒ ortalama 13,3 / 3 = **4,4** (O3, O4 de aynı).

Adım 4 / 12

```
1  def kosullu_katki(c, tur):
2      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))
8          c = yeni
9          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 1` `c = [0, 6.7, 6.7, 6.7]` `digerleri = 13.3` `yeni = [0, 4.4, 4.4, 4.4]`

Kod İzleme: `kosullu_katki(c, tur)`



tur 2: katkılar
[0, 4.4, 4.4, 4.4]

kazanç u = 10 − k + 0,4·Σk
O1: 15,3 O2: 10,9

Tur 2: katkılar [0, 4.4, 4.4, 4.4]. Her tur katkı bir önceki değerin üçte ikisine iner (üçte bir oranında azalır).

Adım 5 / 12

```
1  def kosullu_katki(c, tur):
2      gecmis = [c]
3      for t in range(tur):
4          yeni = [0]                  # O1 hep beleşçi
5          for i in range(1, len(c)):
6              digerleri = sum(c) - c[i]
7              yeni.append(digerleri / (len(c) - 1))
▶          c = yeni
▶          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 1` `c = [0, 4.4, 4.4, 4.4]` `gecmis = 3 tur`

Kod İzleme: `kosullu_katki(c, tur)`



tur 2: katkılar

[0, 4.4, 4.4, 4.4]

kazanç u = 10 − k + 0,4·Σk

O1: 15,3 O2: 10,9

t = 2: O2 için diğerlerinin toplamı 8,9 ⇒ ortalama 8,9 / 3 = **3,0** (O3, O4 de aynı).

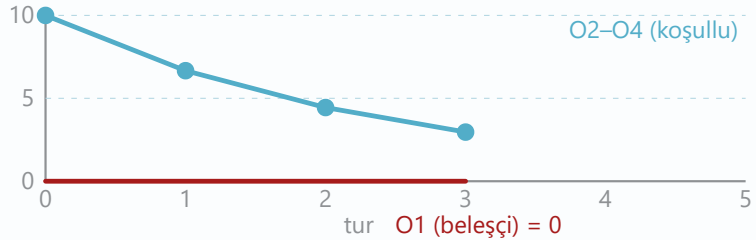
Adım 6 / 12

```
1  def kosullu_katki(c, tur):
2      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))
8          c = yeni
9          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 2` `c = [0, 4.4, 4.4, 4.4]` `digerleri = 8.9` `yeni = [0, 3.0, 3.0, 3.0]`

Kod İzleme: `kosullu_katki(c, tur)`



tur 3: katkılar

[0, 3.0, 3.0, 3.0]

kazanç u = 10 − k + 0,4·Σk

O1: 13,6 O2: 10,6

Tur 3: katkılar [0, 3.0, 3.0, 3.0].

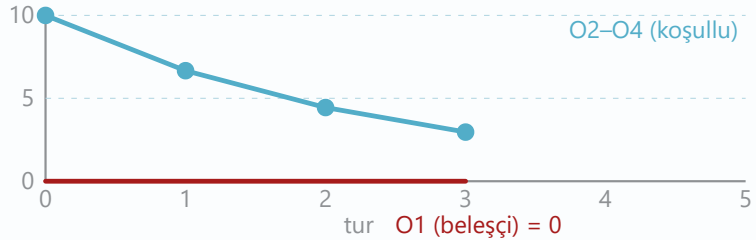
Adım 7 / 12

```
1  def kosullu_katki(c, tur):
2    gecmis = [c]
3    for t in range(tur):
4      yeni = [0]                    # O1 hep beleşçi
5      for i in range(1, len(c)):
6        digerleri = sum(c) - c[i]
7        yeni.append(digerleri / (len(c) - 1))
▶      c = yeni
▶      gecmis.append(c)
10   return gecmis
```

DEĞİŞKENLER

`t = 2` `c = [0, 3.0, 3.0, 3.0]` `gecmis = 4 tur`

Kod İzleme: `kosullu_katki(c, tur)`



tur 3: katkılar

[0, 3.0, 3.0, 3.0]

kazanç u = 10 − k + 0,4·Σk

O1: 13,6 O2: 10,6

t = 3: O2 için diğerlerinin toplamı 5,9 ⇒ ortalama 5,9 / 3 = **2,0** (O3, O4 de aynı).

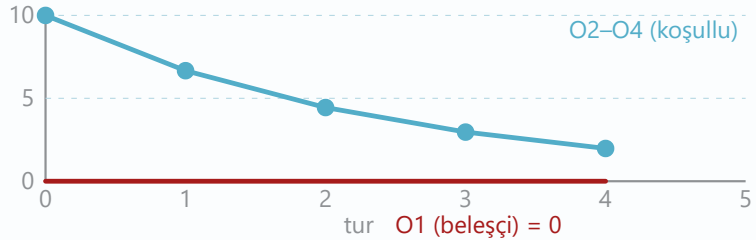
Adım 8 / 12

```
1  def kosullu_katki(c, tur):
2      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))
8          c = yeni
9          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 3` `c = [0, 3.0, 3.0, 3.0]` `digerleri = 5.9` `yeni = [0, 2.0, 2.0, 2.0]`

Kod İzleme: `kosullu_katki(c, tur)`



tur 4: katkılar
[0, 2.0, 2.0, 2.0]

kazanç u = 10 − k + 0,4·Σk
O1: 12,4 O2: 10,4

Tur 4: katkılar [0, 2.0, 2.0, 2.0].

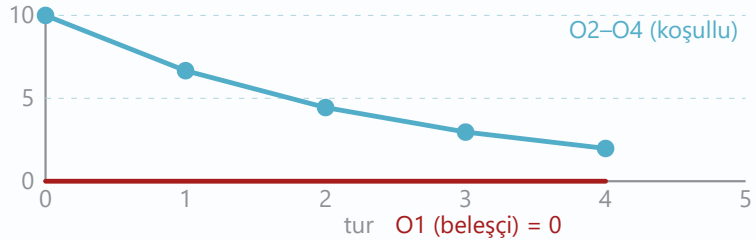
Adım 9 / 12

```
1  def kosullu_katki(c, tur):
2      gecmis = [c]
3      for t in range(tur):
4          yeni = [0]                  # O1 hep beleşçi
5          for i in range(1, len(c)):
6              digerleri = sum(c) - c[i]
7              yeni.append(digerleri / (len(c) - 1))
▶          c = yeni
▶          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 3` `c = [0, 2.0, 2.0, 2.0]` `gecmis = 5 tur`

Kod İzleme: `kosullu_katki(c, tur)`



tur 4: katkılar

[0, 2.0, 2.0, 2.0]

kazanç u = 10 − k + 0,4·Σk

O1: 12,4 O2: 10,4

t = 4: O2 için diğerlerinin toplamı 4,0 ⇒ ortalama 4,0 / 3 = **1,3** (O3, O4 de aynı).

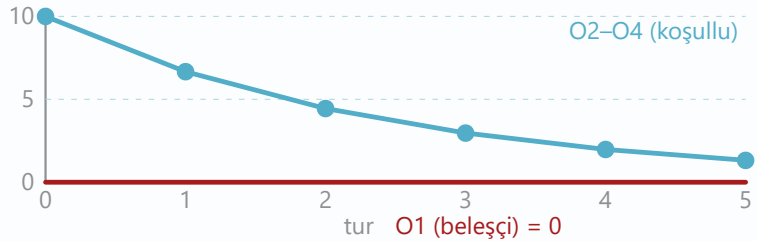
Adım 10 / 12

```
1  def kosullu_katki(c, tur):
2      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))
8          c = yeni
9          gecmis.append(c)
10     return gecmis
```

DEĞİŞKENLER

`t = 4` `c = [0, 2.0, 2.0, 2.0]` `digerleri = 4.0` `yeni = [0, 1.3, 1.3, 1.3]`

Kod İzleme: `kosullu_katki(c, tur)`



tur 5: katkılar
[0, 1.3, 1.3, 1.3]

kazanç u = 10 − k + 0,4·Σk
O1: 11,6 O2: 10,3

Tur 5: katkılar [0, 1.3, 1.3, 1.3].

Adım 11 / 12

```
1  def kosullu_katki(c, tur):
2    gecmis = [c]
3    for t in range(tur):
4      yeni = [0]                  # O1 hep beleşçi
5      for i in range(1, len(c)):
6        digerleri = sum(c) - c[i]
7        yeni.append(digerleri / (len(c) - 1))
▶      c = yeni
▶      gecmis.append(c)
10   return gecmis
```

DEĞİŞKENLER

`t = 4` `c = [0, 1.3, 1.3, 1.3]` `gecmis = 6 tur`

Kod İzleme: `kosullu_katki(c, tur)`



tur 5: katkılar

[0, 1.3, 1.3, 1.3]

kazanç u = 10 − k + 0,4·Σk

O1: 11,6 O2: 10,3

5 turda katkı 10 → 1,3: tek beleşçi, çoğunluğun işbirliğini çözer (laboratuvardaki düşüşün basit modeli).

Adım 12 / 12

```
1  def kosullu_katki(c, tur):
2    gecmis = [c]
3    for t in range(tur):
4      yeni = [0]                    # O1 hep beleşçi
5      for i in range(1, len(c)):
6        digerleri = sum(c) - c[i]
7        yeni.append(digerleri / (len(c) - 1))
8      c = yeni
9      gecmis.append(c)
▶    return gecmis
```

DEĞİŞKENLER

`gecmis` = **6 tur**

KAMU MALI VE CEZA

Cezalı ve Cezasız: Fehr ve Gächter (2000)

Tasarım

4 kişilik gruplar; kamu malına katılan her jeton katana 0,4 jeton geri döner. Katkılar açıklandıktan sonra her denek, kendi parasından ödeyerek diğerlerine ceza puanı verebilir. Cezalı ve cezasız 10'ar tur; sabit gruplar ve her tur karışan gruplar.

Cezasız

Katkılar ilk turlarda yüksek, sonra sıfıra doğru düşer.

Cezalı

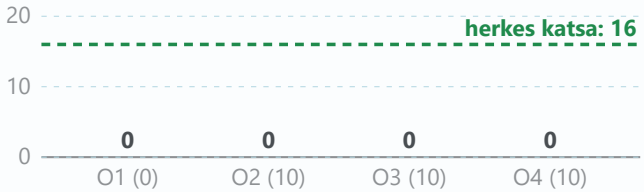
Katkılar yükselir; sabit gruplarda tam işbirliğine yaklaşır. Grup ortalamasından en çok aşağıda katan en çok cezayı alır.

Her tur karışan gruplarda da ceza verildi: aynı kişiyle bir daha karşılaşma olasılığı yokken ceza vermek ceza verene hiçbir şey kazandırmaz. Bu davranışa **özgeci ceza** denir (Fehr ve Gächter, 2002).

Kuramın öngörüsü: yalnızca parayı önemseyen kimse ceza vermez; bunu bilen herkes yine 0 katar. Deneyde hem ceza verildi hem de katkılar yüksek kaldı: kuramın iki öngörüsü de tutmadı.

KAMU MALI VE CEZA

Kod İzleme: Ceza Beleşçiliği Kârsız Kılar mı?



katkılar c = [0, 10, 10, 10]

ceza puanı: 1 öder, 3 kaybettirir

henüz hesap yok

Çağrı `ceza(c, p)` : O1 beleşçi, O2–O4 tam katkı. Her işbirlikçi O1'e 2 ceza puanı verir.

Adım 1 / 6

```
▶ def ceza(c, p):          # p[i][j]: i'nin j'ye puanı
2     n = len(c)
3     pay = 1.6 * sum(c) / n
4     u = [10 - c[i] + pay for i in range(n)]
5     for i in range(n):
6         for j in range(n):
7             u[i] -= p[i][j]       # puan başı 1 öder
8             u[j] -= 3 * p[i][j]   # cezalı 3 kaybeder
9     return u
```

DEĞİŞKENLER

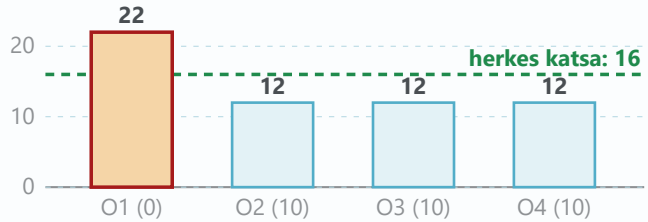
`c = [0, 10, 10, 10]`

`p = p[1][0]=p[2][0]=p[3][0]=2`

ÇAĞRI YIĞINI

`ceza(c, p)`

Kod İzleme: Ceza Beleşçiliği Kârsız Kılar mı?



katkılar c = [0, 10, 10, 10]

ceza puanı: 1 öder, 3 kaybettirir

ceza öncesi

pay = 1,6 · 30 / 4 = 12. Ceza öncesi u = [22, 12, 12, 12]: beleşçi, katkı yapanlardan 10 fazla alır.

Adım 2 / 6

```
1 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)]
5       for i in range(n):
6           for j in range(n):
7               u[i] -= p[i][j]        # puan başı 1 öder
8               u[j] -= 3 * p[i][j]    # cezalı 3 kaybeder
9       return u
```

DEĞİŞKENLER

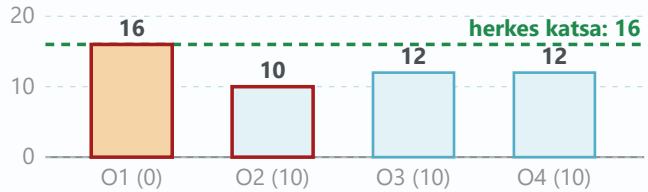
`n = 4` `pay = 12.0`

`u = [22, 12, 12, 12]`

ÇAĞRI YIĞINI

`ceza(c, p)`

Kod İzleme: Ceza Beleşçiliği Kârsız Kılar mı?



katkılar c = [0, 10, 10, 10]

ceza puanı: 1 öder, 3 kaybettirir

O2 → O1: 2 puan

i = 0 (O1) kimseyi cezalandırmaz. i = 1: O2, 2 puan için 2 öder (12 → 10); O1 2 · 3 = 6 kaybeder (22 → 16).

Adım 3 / 6

```
1 def ceza(c, p):          # p[i][j]: i'nin j'ye puanı
2     n = len(c)
3     pay = 1.6 * sum(c) / n
4     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
9     return u
```

DEĞİŞKENLER

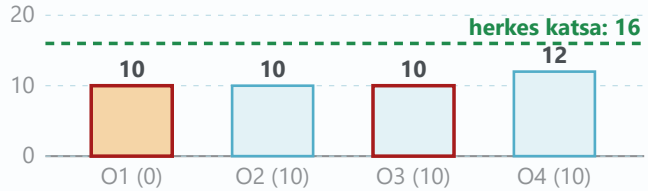
`i = 1` `j = 0` `p[i][j] = 2`

`u = [16, 10, 12, 12]`

ÇAĞRI YIĞINI

`ceza(c, p)`

Kod İzleme: Ceza Beleşçiliği Kârsız Kılar mı?



katkılar c = [0, 10, 10, 10]

ceza puanı: 1 öder, 3 kaybettirir

O3 → O1: 2 puan

i = 2: O3 de 2 öder (12 → 10); O1 16 → 10. Beleşçinin üstünlüğü bitti.

Adım 4 / 6

```
1 def ceza(c, p):          # p[i][j]: i'nin j'ye puanı
2     n = len(c)
3     pay = 1.6 * sum(c) / n
4     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
9     return u
```

DEĞİŞKENLER

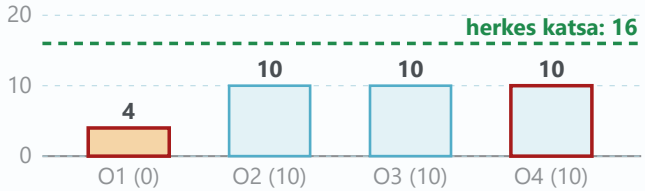
`i = 2` `j = 0` `p[i][j] = 2`

`u = [10, 10, 10, 12]`

ÇAĞRI YIĞINI

`ceza(c, p)`

Kod İzleme: Ceza Beleşçiliği Kârsız Kılar mı?



katkılar c = [0, 10, 10, 10]

ceza puanı: 1 öder, 3 kaybettirir

O4 → O1: 2 puan

i = 3: O4 de 2 öder (12 → 10); O1 10 → 4.

Adım 5 / 6

```
1 def ceza(c, p):          # p[i][j]: i'nin j'ye puanı
2     n = len(c)
3     pay = 1.6 * sum(c) / n
4     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
9     return u
```

DEĞİŞKENLER

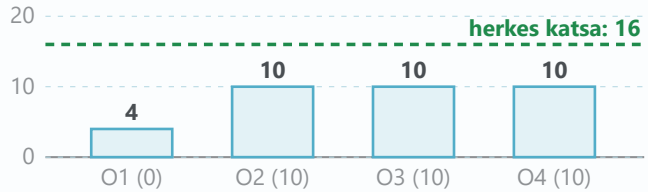
`i = 3` `j = 0` `p[i][j] = 2`

`u = [4, 10, 10, 10]`

ÇAĞRI YIĞINI

`ceza(c, p)`

Kod İzleme: Ceza Beleşçiliği Kârsız Kılar mı?



katkılar c = [0, 10, 10, 10]

ceza puanı: 1 öder, 3 kaybettirir

return u

return u: beleşçi 4 alır; katkı yapsaydı (cezasız, herkes 10) 16 alırdı. Ceza tehdidi inandırıcıysa katmak kârlıdır, ama ceza verenler de 2'şer kaybetti.

Adım 6 / 6

```
1 def ceza(c, p):          # p[i][j]: i'nin j'ye puanı
2     n = len(c)
3     pay = 1.6 * sum(c) / n
4     u = [10 - c[i] + pay for i in range(n)]
5     for i in range(n):
6         for j in range(n):
7             u[i] -= p[i][j]       # puan başı 1 öder
8             u[j] -= 3 * p[i][j]   # cezalı 3 kaybeder
▶     return u
```

DEĞİŞKENLER

`u = [4, 10, 10, 10]`

ÇAĞRI YIĞINI

`ceza(c, p)`

Cezayı Kim Verecek? İkinci Derece Beleşçilik

Ceza verenler (2'şer puan)	O1	O2	O3	O4
Kimse	22	12	12	12
O2, O3, O4	4	10	10	10
O2, O3 (O4 vermez)	10	10	10	12

Katkılar [0, 10, 10, 10]; ceza puanı 1 öder, 3 kaybettirir

- Ceza da bir kamu malıdır: maliyeti cezalandırana, faydası (daha yüksek katkı) herkese. Ceza vermeyen O4 en çok kazanandır.
- Geriye doğru çıkarım: son turda ceza vermek yalnızca maliyettir ⇒ kimse vermez ⇒ tehdit inandırıcı değildir.

Antisosyal ceza

İstanbul dahil 16 şehirde yapılan kamu malı deneylerinde, bazı yerlerde yüksek katkı yapanlar da cezalandırıldı; bu, cezanın işbirliğini artırıcı etkisini silebildi. Yurttaşlık normlarının ve hukukun üstünlüğünün zayıf olduğu yerlerde antisosyal ceza daha sıktı (Herrmann, Thöni ve Gächter, 2008).

Ders: ceza bir kurumdur ve işleyişi, onu çevreleyen normlara bağlıdır.

Oyuna Dönüş: Kamu Malı Oyunu

Kuramın cevabı: 0 jeton

Katılan her jeton size yalnızca 1,6/4 = 0,4 döndürür: diğerleri ne yaparsa yapsın, katmamak jeton başına 0,6 daha iyidir. Katmamak baskın stratejidir; herkes 0 katarsa herkes 10, herkes 10 katarsa herkes 16 alır.

Senaryo	denge katkısı	kişi başı
Temel	0	10
Cezalı (3'ten az katana 2 ceza)	3	11,8
Optimum (herkes 10)	10	16

Cezalı senaryoda 3 jeton katmak 3 · 0,6 = 1,8 kaybettirir ama 2 jetonluk cezadan kurtarır: 3 katmak baskın olur.

Gerçekte

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).

Ceza kuralı, ortak meradaki vergi gibi teşvikleri değiştiren bir **kurumdur**: dengeyi 0'dan 3'e taşır, ama optimuma (10) götürmez.

Eşikli Proje: Katkı Bir Koordinasyon Oyunu Olur

Kamu Malı Oyunu'nun eşikli proje senaryosu: 4 oyuncu, 10'ar jeton; toplam katkı 22'ye ulaşırsa herkes 12 ek jeton alır, ulaşmazsa katkılar yanar. $u_i = 10 - k_i + 12 \cdot [\Sigma k \geq 22]$.

Denge 1: kimse katmaz

Tek başına en çok 10 katılabilir, eşik 22'ye ulaşılamaz: katan yalnızca kaybeder. Herkes 10, toplam 40.

Denge 2: toplam tam 22

Her katkıcı belirleyicidir: 1 jeton çekse eşik kaçar ve 12 kaybeder; fazlası boşa gider. (6, 6, 5, 5) ⇒ (16, 16, 17, 17), toplam 66.

Hangi denge?

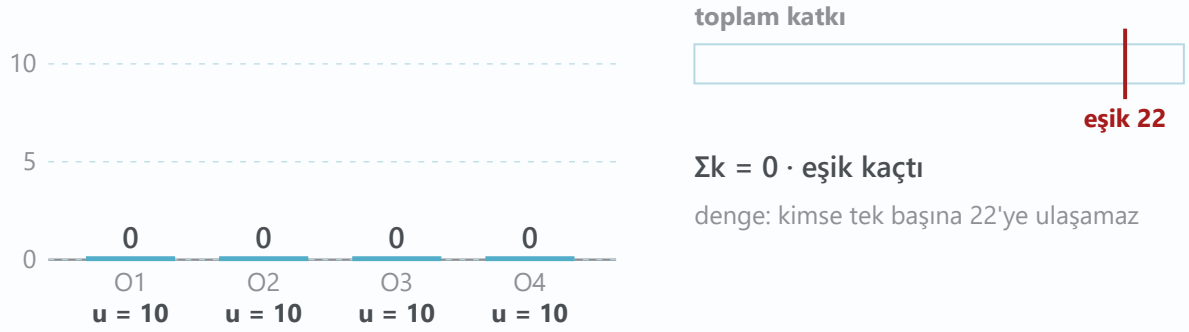
Katan, diğerleri katmazsa katkısını yakar: 6 katan 4'e düşer. Güven yoksa sıfır dengesi seçilebilir.

Kim ne kadar?

(10, 10, 2, 0) katkı profili de dengedir; kazançlar (12, 12, 20, 22). Eşik dengeleri verimlidir ama eşit olmayabilir.

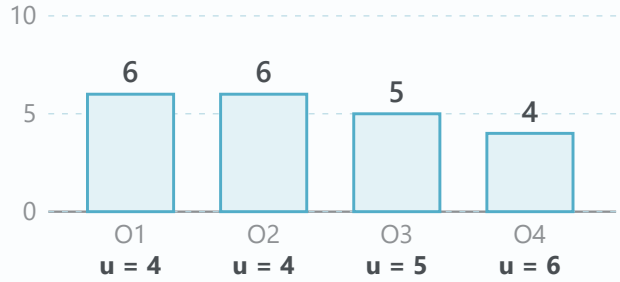
Eşik + iade (hedefe ulaşmayan kitlesel fonlamada paranın geri verilmesi) "katıp yakma" riskini kaldırır ve koordinasyonu kolaylaştırır.

Adım Adım: Hangi Katkı Profili Dengedir?



(0, 0, 0, 0): O1 tek başına 10 katsa toplam 10 < 22, 10 jetonu yanar. Kimse sapmak istemez: herkesin sıfır kattığı bu profil bir **Nash dengesi**dir; herkesin kazancı 10.

Adım Adım: Hangi Katkı Profili Dengedir?



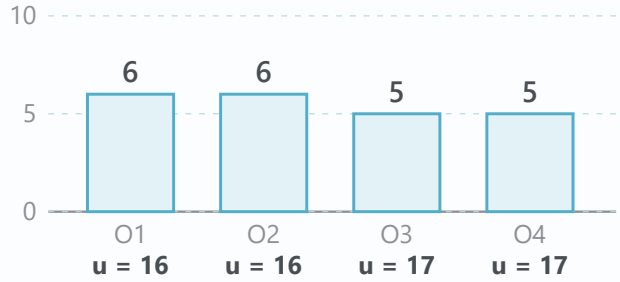


Σk = 21 · eşik kaçtı

denge değil: O4 bir jeton daha katar

(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 değil**.

Adım Adım: Hangi Katkı Profili Dengedir?



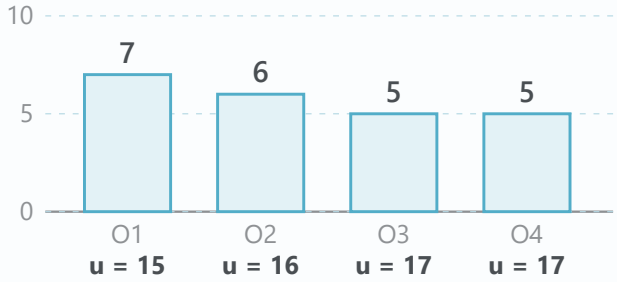


Σk = 22 · eşik aşıldı

denge: her katkıcı belirleyici

(6, 6, 5, 5): toplam 22 ⇒ kazançlar (16, 16, 17, 17). O1 5'e inerse toplam 21 olur, O1 5 alır < 16. Fazla katan boşa öder: **Nash dengesi**, toplam 66.

Adım Adım: Hangi Katkı Profili Dengedir?



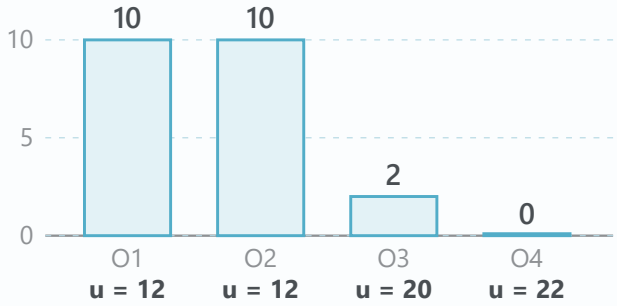


Σk = 23 · eşik aşıldı

denge değil: O1 bir jeton geri çeker

(7, 6, 5, 5): toplam 23. O1 6'ya inse eşik yine tutar ve 1 jeton kazanır: fazla katkı **denge değildir**.

Adım Adım: Hangi Katkı Profili Dengedir?





Σk = 22 · eşik aşıldı

denge ama eşitsiz: (12, 12, 20, 22)

(10, 10, 2, 0) katkı profili de dengedir (O3 çekilirse eşik kaçar). Verimli dengeler çoktur; kimin ne katacağı bir pazarlık ve koordinasyon sorunudur.

TOPLU BİLGİ

Kalabalığın Bilgeliği

- Galton (1907): bir hayvan fuarında bir öküzün ağırlığı için 787 tahmin toplandı. Ortanca (ortadaki) tahmin 1.207 libre, gerçek ağırlık 1.198 libre: hata %1'den az.
- Neden? Bağımsız hatalar birbirini götürür.
- **Çeşitlilik tahmin teoremi** (Page, 2007): (ortalama − gerçek)^2 = bireysel hata karelerinin ortalaması − tahminlerin çeşitliliği. Çeşitlilik: tahminlerin kendi ortalamalarından sapma karelerinin ortalaması.
- Koşullar: **bağımsızlık** ve **çeşitlilik**. Herkes birbirini izlerse çeşitlilik çöker.

tahmin x	bireysel hata $(x - 50)^2$	çeşitlilik $(x - 54)^2$
40	100	196
45	25	81
55	25	1
60	100	36
70	400	256
ort. 54	130	114

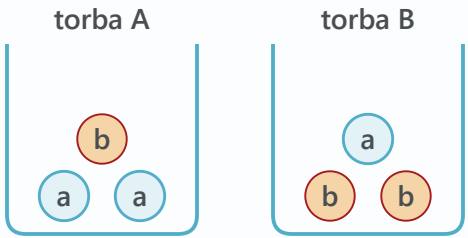
Gerçek 50: kalabalığın hatası $(54 - 50)^2 = 16 = 130 - 114$

Kalabalık, ortalama bir üyesinden her zaman en az onun kadar iyi tahmin eder; çeşitlilik ne kadar büyükse kazanç o kadar büyüktür.

TOPLU BİLGİ

Bilgi Şelaleleri: Herkes Haklı Olarak Yanılabilir

- **Model** (Bikhchandani, Hirshleifer ve Welch, 1992): torba yazı-turayla seçilir. Sırayla her kişi gizlice bir top görür, öncekilerin *tahminlerini* (toplarını değil) duyar ve torbayı tahmin eder. Her top, 2/3 olasılıkla torbanın kendi rengindedir.
- **Bayes:** bu yüzden her a, A lehine oranı 2 katına çıkarır: a'ların b'lerden f fazla olduğu durumda oran 2^f'e 1, yani $P(A) = 2^f / (2^f + 1)$. Eşitlikte ($P(A) = 1/2$) kişi kendi sinyalini izler.
- **Şelale:** öncekilerden çıkarılan fark $f = 2$ olunca, kendi b'sini gören kişi için $f = 1$ ve $P(A) = 2/3 > 1/2$: kendi sinyalini yok sayıp A der. Sonrakiler onun tahmininden hiçbir şey öğrenmez.
- **Kırılganlık:** şelale yalnızca iki sinyale dayanır; tek bir kamuya açık bilgi onu bozabilir. Moda, ürün yorumları, panik.



her çekiliş: doğru sinyal olasılığı 2/3

Laboratuvarda (Anderson ve Holt, 1997) bu torba deneyinde şelaleler, oluşabildikleri durumların çoğunda gerçekten oluştu.

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Çağrı `selale(sinyaller)` ; gerçek torba B. Öncekilerin tahminlerinden çıkarılan fark f = 0.

Adım 1 / 8

```
▶ 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
```

DEĞİŞKENLER

`sinyaller = [a, a, b, b, b, b]`

`f = 0` `tahmin = []`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Kişi 1, sinyal a: f = 0 ⇒ kendi sinyalini izler, **A**. Tahmin sinyali ele verir: f = 1, P(A) = 2/3.

Adım 2 / 8

```
1  def selale(sinyaller):
2      f = 0                    # tahminlerden çıkan a − b
3      tahmin = []
▶      for s in sinyaller:
5          if f >= 2: t = 'A'      # A şelalesi
6          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)
11     return tahmin
```

DEĞİŞKENLER

`s = a` `f = 1` `t = A` `tahmin = [A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Kişi 2, sinyal a: f = 1 ⇒ kendi a'sıyla P(A) = 0,8, **A**. f = 2.

Adım 3 / 8

```
1  def selale(sinyaller):
2      f = 0                    # tahminlerden çıkan a − b
3      tahmin = []
▶      for s in sinyaller:
5          if f >= 2: t = 'A'      # A şelalesi
6          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)
11     return tahmin
```

DEĞİŞKENLER

`s = a` `f = 2` `t = A`

`tahmin = [A, A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Kişi 3, sinyal b: f = 2 ⇒ kendi b'siyle bile P(A) = 2/3 ⇒ **A**. Şelale başladı; f artık değişmez.

Adım 4 / 8

```
1  def selale(sinyaller):
2      f = 0                    # tahminlerden çıkan a − b
3      tahmin = []
       for s in sinyaller:
           if f >= 2: t = 'A'      # A şelalesi
6          elif f <= -2: t = 'B'   # B şelalesi
7          else:
8              t = 'A' if s == 'a' else 'B'
9              f += 1 if s == 'a' else -1
           tahmin.append(t)
11     return tahmin
```

DEĞİŞKENLER

`s = b` `f = 2` `t = A`

`tahmin = [A, A, A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Kişi 4, sinyal b: aynı hesap ⇒ **A**. Onun tahmini de bilgi taşımaz.

Adım 5 / 8

```
1  def selale(sinyaller):
2      f = 0                    # tahminlerden çıkan a − b
3      tahmin = []
▶      for s in sinyaller:
▶          if f >= 2: t = 'A'      # A şelalesi
6          elif f <= -2: t = 'B'   # B şelalesi
7          else:
8              t = 'A' if s == 'a' else 'B'
9              f += 1 if s == 'a' else -1
▶          tahmin.append(t)
11     return tahmin
```

DEĞİŞKENLER

`s = b` `f = 2` `t = A`

`tahmin = [A, A, A, A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Kişi 5, sinyal b ⇒ **A**.

Adım 6 / 8

```
1  def selale(sinyaller):
2      f = 0                    # tahminlerden çıkan a − b
3      tahmin = []
▶      for s in sinyaller:
▶          if f >= 2: t = 'A'      # A şelalesi
6          elif f <= -2: t = 'B'   # B şelalesi
7          else:
8              t = 'A' if s == 'a' else 'B'
9              f += 1 if s == 'a' else -1
▶          tahmin.append(t)
11     return tahmin
```

DEĞİŞKENLER

`s = b` `f = 2` `t = A`

`tahmin = [A, A, A, A, A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



Kişi 6, sinyal b ⇒ **A**.

Adım 7 / 8

```
1  def selale(sinyaller):
2      f = 0                    # tahminlerden çıkan a − b
3      tahmin = []
▶      for s in sinyaller:
▶          if f >= 2: t = 'A'      # A şelalesi
6          elif f <= -2: t = 'B'   # B şelalesi
7          else:
8              t = 'A' if s == 'a' else 'B'
9              f += 1 if s == 'a' else -1
▶          tahmin.append(t)
11     return tahmin
```

DEĞİŞKENLER

`s = b` `f = 2` `t = A`

`tahmin = [A, A, A, A, A, A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Kod İzleme: Bir Şelale Nasıl Başlar?



return: herkes A dedi, oysa 6 sinyalin 4'ü b. Sinyaller birleştirilseydi f = −2, P(A) = 0,2 ⇒ B. Her kişi rasyoneldi; bilgi birikmedi.

Adım 8 / 8

```
1  def selale(sinyaller):
2      f = 0                  # tahminlerden çıkan a − b
3      tahmin = []
4      for s in sinyaller:
5          if f >= 2: t = 'A'      # A şelalesi
6          elif f <= -2: t = 'B'   # B şelalesi
7          else:
8              t = 'A' if s == 'a' else 'B'
9              f += 1 if s == 'a' else -1
10         tahmin.append(t)
▶      return tahmin
```

DEĞİŞKENLER

`tahmin = [A, A, A, A, A, A]`

ÇAĞRI YIĞINI

`selale(sinyaller)`

TOPLU BİLGİ

Condorcet Jüri Teoremi: Çoğunluk Ne Kadar Akıllı?

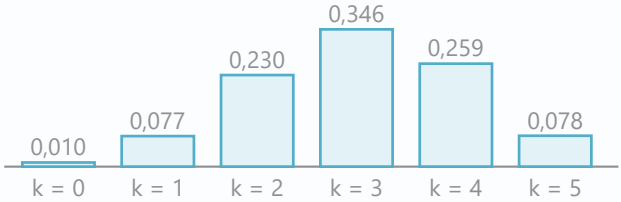
- **Soru:** bir doğru cevap var (sanık suçlu mu, değil mi?). Çoğunluk oyu onu bulma olasılığını nasıl değiştirir? Tercihlerin döngüye girdiği Condorcet paradoksundan farklı bir sorudur.
- **Teorem** (Condorcet, 1785): n (tek) seçmen *bağımsız* oy verir ve her biri $p > 1/2$ olasılıkla doğruyu seçerse, çoğunluğun doğru olma olasılığı n ile artar ve 1'e gider.
- $p < 1/2$ ise tersine: kalabalık büyüdükçe çoğunluk neredeyse kesin yanılır.
- $P_n = \Sigma_{k > n/2}\, C(n, k) \cdot p^k (1 - p)^{n - k}$

n	p = 0,6	p = 0,4
1	0,600	0,400
3	0,648	0,352
5	0,683	0,317
11	0,753	0,247
51	0,926	0,074
101	0,979	0,021

Çoğunluğun doğru karar verme olasılığı

TOPLU BİLGİ

Kod İzleme: Beş Kişilik Jüri





toplam = 0,0000

Çağrı `juri(5, 0.6)` : 5 seçmenden en az 3'ü doğru oy verirse çoğunluk doğrudur. k: doğru oy sayısı.

Adım 1 / 5

```
1 from math import comb
▶ def juri(n, p):
▶     toplam = 0
4     for k in range(n // 2 + 1, n + 1):
5         terim = comb(n, k) * p**k * (1 - p)**(n - k)
6         toplam += terim
7     return toplam
▶ print(round(juri(5, 0.6), 4))
```

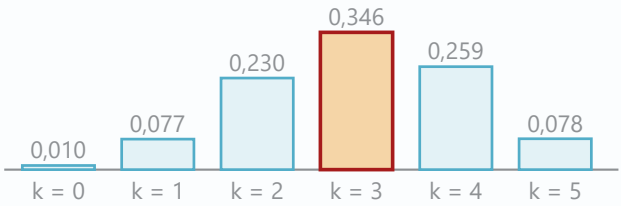
DEĞİŞKENLER

`n = 5` `p = 0.6` `toplam = 0`

ÇIKTI

TOPLU BİLGİ

Kod İzleme: Beş Kişilik Jüri





toplam = 0,3456

k = 3: C(5, 3)·0,6^3·0,4^2 = 10·0,2160·0,1600 = **0,3456**. Toplam 0,3456.

Adım 2 / 5

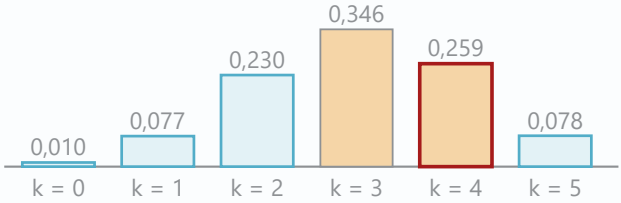
```
1 from math import comb
2 def juri(n, p):
3     toplam = 0
▶     for k in range(n // 2 + 1, n + 1):
▶         terim = comb(n, k) * p**k * (1 - p)**(n - k)
▶         toplam += terim
7     return toplam
8 print(round(juri(5, 0.6), 4))
```

DEĞİŞKENLER

`k = 3` `terim = 0.3456` `toplam = 0.3456`

ÇIKTI

Kod İzleme: Beş Kişilik Jüri





toplam = 0,6048

k = 4: C(5, 4)·0,6^4·0,4^1 = 5·0,1296·0,4000 = **0,2592**. Toplam 0,6048.

Adım 3 / 5

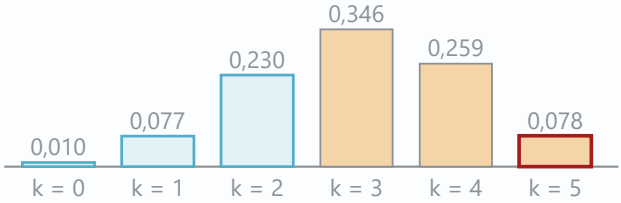
```
1 from math import comb
2 def juri(n, p):
3     toplam = 0
▶     for k in range(n // 2 + 1, n + 1):
▶         terim = comb(n, k) * p**k * (1 - p)**(n - k)
▶         toplam += terim
7     return toplam
8 print(round(juri(5, 0.6), 4))
```

DEĞİŞKENLER

`k = 4` `terim = 0.2592` `toplam = 0.6048`

ÇIKTI

Kod İzleme: Beş Kişilik Jüri





toplam = 0,6826

k = 5: C(5, 5)·0,6^5·0,4^0 = 1·0,0778·1 = **0,0778**. Toplam 0,6826.

Adım 4 / 5

```
1 from math import comb
2 def juri(n, p):
3     toplam = 0
▶     for k in range(n // 2 + 1, n + 1):
▶         terim = comb(n, k) * p**k * (1 - p)**(n - k)
▶         toplam += terim
7     return toplam
8 print(round(juri(5, 0.6), 4))
```

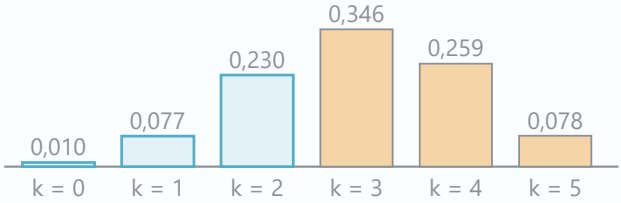
DEĞİŞKENLER

`k = 5` `terim = 0.0778` `toplam = 0.6826`

ÇIKTI

TOPLU BİLGİ

Kod İzleme: Beş Kişilik Jüri





toplam = 0,6826

return: P_5 = 0,6826 > 0,6. Beş bağımsız ve orta yetkinlikte seçmen, tek bir seçmenden daha iyi karar verir.

Adım 5 / 5

```
1 from math import comb
2 def juri(n, p):
3     toplam = 0
4     for k in range(n // 2 + 1, n + 1):
5         terim = comb(n, k) * p**k * (1 - p)**(n - k)
6         toplam += terim
▶     return toplam
▶ print(round(juri(5, 0.6), 4))
```

DEĞİŞKENLER

`toplam = 0.6826`

ÇIKTI

```
0.6826
```

TOPLU BİLGİ

Jüri Teoremi Ne Zaman Çöker?

Bağımlılık

5 kişilik jüride 4 kişi bir kanaat önderinin oyunu izlerse çoğunluk o kişinin oyudur: P = 0,6. Bağımsız oylarda 0,683'tü. Şelaleler ve ortak bilgi kaynakları bağımsızlığı bozar.

Yetkinlik

p < 1/2 olan bir konuda (yanıltıcı bir ipucu) kalabalık büyüdükçe yanılgı kesinleşir: p = 0,4 iken 101 kişide çoğunluk yalnızca %2,1 olasılıkla doğrudur.

Stratejik oy

Seçmen, oyunun yalnızca belirleyici olduğu durumda önemli olduğunu düşünürse kendi sinyaline aykırı oy vermesi rasyonel olabilir (Austen-Smith ve Banks, 1996). Oybirliği kuralında bu, masumu mahkûm etme olasılığını artırabilir (Feddersen ve Pesendorfer, 1998).

Kalabalığın bilgeliği, şelaleler ve jüri teoremi aynı dersi verir: bir grubun bilgisi, üyelerinin **bağımsız** bilgisinin bir araya gelebildiği ölçüde değerlidir.

ÖZET

Sık Yapılan Hatalar: Kolektif Karar

"Kalabalık her zaman bilgedir."
Bağımsızlık ve $p > 1/2$ gerekir. Şelalede altı kişinin altısı da yanıldı.

"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.

Özet

Beklenti kuramı

Referans noktası, λ ≈ 2,25, olasılık ağırlığı: ±100 TL bahsi reddedilir, kabul eşiği ≈ 251 TL; bağış etkisi ve çerçeveleme aynı kökten gelir.

Sınırlı rasyonel denge

Kuantal tepki: iyi seçenek daha sık seçilir; asimetrik yazı-turada Nash'in 1/2'si yerine büyük kazanca kayma.

Sosyal tercihler

Fehr–Schmidt: α = 1 olan yanıtlayan 10'dan 10/3'ün altındaki teklifi reddeder; güven oyununda karşılıklılık.

Ortak mallar

4 çobanlı dengede toplam 1.600, optimum 2.500; vergi, kota ve Ostrom'un yerel kurumları.

Kamu malı ve ceza

Katkılar %40–60'tan başlayıp erir; ceza işbirliğini korur ama ikinci derece beleşçilik ve antisosyal ceza doğurabilir.

Toplu bilgi

Çeşitlilik teoremi, şelalede bilginin birikmemesi, jüri teoremi (5 kişi, p = 0,6 ⇒ 0,683).

ÖZET

Etkileşimli Oyunlar

Kamu Malı Oyunu: temel

Dört oyuncu, çarpan 1,6: katkı, beleşçilik ve grup refahı.

Eşikli proje

Toplam katkı 22'ye ulaşırsa herkese 12 ek jeton: katkı bir koordinasyon sorununa dönüşür.

Cezalı katkı

3 jetondan az katana 2 jeton ceza: bir kuralın dengeyi nasıl kaydırdığı.

Oyunda üç düzey vardır: cömert botlar, karışık botlar ve önceki turun ortalaması yüksekse katkısını azaltan uyarlanır beleşçiler.

ÖZET

Örnek Problemler

1. Beklenti kuramı parametreleriyle +200 / −100 yazı-tura bahsi kabul edilir mi?
2. Ortak mallar modelinde n = 9 çoban için denge G* ve toplam kazanç? Optimumu sağlayan vergi?
3. Kamu malı oyununda cezalı senaryoda ceza 2 yerine 1 jeton olsaydı denge katkısı kaç olurdu?
4. Fehr–Schmidt modelinde α = 2 olan yanıtlayan 10 jetondan en az kaçını ister?
5. Condorcet jüri teoreminde p = 0,6 iken 3 kişilik jürinin çoğunluğu hangi olasılıkla doğrudur?

▶ **Cevapları göster**

Kaynaklar

- H. A. Simon, "A behavioral model of rational choice", *Quarterly J. Economics* 69, 1955.
- D. Kahneman, A. Tversky, "Prospect theory: An analysis of decision under risk", *Econometrica* 47(2), 1979.
- A. Tversky, D. Kahneman, "The framing of decisions and the psychology of choice", *Science* 211, 1981; "Advances in prospect theory", *J. Risk and Uncertainty* 5, 1992.
- D. Kahneman, J. L. Knetsch, R. H. Thaler, "Experimental tests of the endowment effect and the Coase theorem", *J. Political Economy* 98(6), 1990.
- R. D. McKelvey, T. R. Palfrey, "Quantal response equilibria for normal form games", *Games and Economic Behavior* 10(1), 1995; J. K. Goeree, C. A. Holt, "Ten little treasures of game theory and ten intuitive contradictions", *American Economic Review* 91(5), 2001.
- E. Fehr, K. M. Schmidt, "A theory of fairness, competition, and cooperation", *Quarterly J. Economics* 114(3), 1999; C. Engel, "Dictator games: A meta study", *Experimental Economics* 14(4), 2011.
- J. Berg, J. Dickhaut, K. McCabe, "Trust, reciprocity, and social history", *Games and Economic Behavior* 10(1), 1995; M. Rabin, "Incorporating fairness into game theory and economics", *American Economic Review* 83(5), 1993.
- G. Hardin, "The tragedy of the commons", *Science* 162, 1968; E. Ostrom, *Governing the Commons*, Cambridge University Press, 1990.

Kaynaklar (2/2)

- 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.
- U. Fischbacher, S. Gächter, E. Fehr, "Are people conditionally cooperative? Evidence from a public goods experiment", *Economics Letters* 71(3), 2001.
- E. Fehr, S. Gächter, "Cooperation and punishment in public goods experiments", *American Economic Review* 90(4), 2000; "Altruistic punishment in humans", *Nature* 415, 2002.
- B. Herrmann, C. Thöni, S. Gächter, "Antisocial punishment across societies", *Science* 319, 2008.
- F. Galton, "Vox populi", *Nature* 75, 1907; S. E. Page, *The Difference*, Princeton University Press, 2007.
- S. Bikhchandani, D. Hirshleifer, I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades", *J. Political Economy* 100(5), 1992; L. R. Anderson, C. A. Holt, "Information cascades in the laboratory", *American Economic Review* 87(5), 1997.
- Marquis de Condorcet, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, 1785; D. Austen-Smith, J. S. Banks, "Information aggregation, rationality, and the Condorcet jury theorem", *American Political Science Review* 90(1), 1996; T. Feddersen, W. Pesendorfer, "Convicting the innocent", *American Political Science Review* 92(1), 1998.
- R. Nagel, "Unraveling in guessing games: An experimental study", *American Economic Review* 85(5), 1995.
- J. C. Harsanyi, "Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility", *J. Political Economy* 63, 1955; J. Rawls, *A Theory of Justice*, Harvard University Press, 1971.

Son

Bölüm 12: Davranışsal ve Kolektif Karar

Sonraki bölüm: Bölüm 13: Öğrenme ve Adaptasyon

Çalışma soruları · Terim sözlüğü