Yöneylem Araştırması III

Doç. Dr. Hakan ÇERÇİOĞLU
cercioglu@gazi.edu.tr

BÖLÜM I: Hedef Programlama

- HEDEF PROGRAMLAMAYA GİRİŞ
- ÖNCELİKSİZ HEDEF PROGRAMLAMA
- ÖNCELİKLİ HEDEF PROGRAMLAMA
- HEDEF PROGRAMLAMA MODELLERİNİN ÇÖZÜMÜ

✵Hedef Programlamaya Giriş

- Bugüne kadar anlatılan bütün DP modellerinde tek bir amaç üzerinde duruldu. Yani amaç fonksiyonundaki bütün karar değişkenleri ölçülebilen aynı birime sahipti.
- Bazı durumlarda aynı birime çevrilemeyen birçok amaç birlikte isteniyor olabilir.
- Bu tip problemlerin çözümünde 1972 yılında Lee ve Ignizio'nun geliştirdiği amaç programlama yaklaşımı kullanılmaktadır.
- Amaç programlama, belirli kararlar çerçevesinde farklı ve çelişen amaçların en iyilenmesini aynı anda sağlayan matematiksel bir yöntemdir.

✵Hedef Programlama Problemlerinin Modellenmesi

- Verilen herhangi bir problemin formülasyonunda mümkün olduğu kadar sağlanması gereken ve geçici olabilmekle birlikte kesin olarak belirlenen ihtiyaca hedef denir.
- Göreceli önem derecesine göre ağırlıklandırılan birçok hedeften negatif, pozitif veya her iki yöndeki sapmaları eş zamanlı olarak minimize etmeyi amaçlayan çok amaçlı doğrusal programlama çözüm tekniğine Hedef Programlama denir.
- DP da olduğu gibi amaç kısıt denklemleride 3 şekilde karşımıza çıkabilir;

Hedef Programlama Problemlerinin Modellenmesi

1. $f_1(x) = \sum_{j=1}^{n} a_{ij} x_j \le b_i$
2. $f_2(x) = \sum_{j=1}^{n} a_{ij} x_j \ge b_i$
3. $f_3(x) = \sum_{j=1}^{n} a_{ij} x_j = b_i$

- Örneğin;
- 1nci kısıtta b_i'yi aşan her bir birimin cezasının 3,
- 2nci kısıtta b_i'nin altında kalan her bir birimin cezasının 5,
- 3ncü kısıtta ise b_i'yi aşan her bir birimin cezasının 2, b_i'nin altında kalan her bir birimin cezasının ise 4 olduğunu varsayalım.

- Görüldüğü gibi sağ taraf sabitleri sabit bir değere sahip değildir. Hatta bütün kısıtların aynı anda sağlanması beklenmemelidir.
- Bunun yerine yönetimin amaçlarına mümkün olduğunca yaklaşılmaya çalışılmalıdır.

Hedef Programlama Problemlerinin Modellenmesi

- Bütün amaçları içeren amaç fonksiyonu;

$$\begin{aligned} \text{Min } Z = \; & 3(f_1(x) - bi)^+ + \\ & 5(f_2(x) - bi)^- + \\ & 2(f_3(x) - bi)^+ + 4(f_3(x) - bi)^- \end{aligned}$$

- Oranlılık özelliği sağlanmadığı için amaç fonksiyonu *doğrusal değildir*. Parantez içerisindeki ifadeler doğrusaldır. Amaç fonksiyonu doğrusal olmadığı için lineer hale dönüştürmek gerekir. Bunun içinde yardımcı bir değişkene ihtiyaç vardır. Bu yardımcı değişkene *d* (deviation=sapma) dersek

$$\begin{aligned} d_1 &= f_1(x) - b_i \quad , \quad d_1 = d_1^+ - d_1^- \\ d_2 &= f_2(x) - b_i \quad , \quad d_2 = d_2^+ - d_2^- \\ d_3 &= f_3(x) - b_i \quad , \quad d_3 = d_3^+ - d_3^- \end{aligned}$$

- Yukarıda verilen *1*nci kısıt için pozitif yöndeki sapmayı ifade eden d_1^+, *2*nci kısıt için *negatif* yöndeki sapmayı ifade eden d_2^-, *3*ncü kısıt için ise her iki yöndeki sapmaları ifade eden d_3^- ve d_3^+ amaç fonksiyonunda yer alacaktır. Bu değişkenler amaçtan sapmaları gösterir.

Hedef Programlama Problemlerinin Modellenmesi

Min Z =	$3d_1^+ + 5d_2^- + 2d_3^+ + 4d_3^-$	
S.T.	$f_1(x) - (d_1^+ - d_1^-) = b_i$	($\leq$ olduğu için d_1^+)
	$f_2(x) - (d_2^+ - d_2^-) = b_i$	($\geq$ olduğu için d_2^-)
	$f_3(x) - (d_3^+ - d_3^-) = b_i$	(= olduğu için d_3^+ ve d_3^-)
	$x_j \geq 0, d_k^+ \geq 0, d_k^- \geq 0$	$(j = 1,2,\ldots,n; k = 1,2,\ldots,m)$

- Bu örneği genellemek istersek: n adet değişken ve m adet hedeften oluşan hedef programlama modelinin genel yapısı aşağıdaki gibi olur.

$$\min Z = d_1^- + d_1^+ + d_2^- + d_2^+ + \cdots + d_m^- + d_m^+$$

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + d_1^- - d_1^+ = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + d_2^- - d_2^+ = b_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + d_m^- - d_m^+ = b_m$$

$$x_j \geq 0 \qquad j=1,2,..n$$

$$d_i^-, d_i^+ \geq 0 \qquad i=1,2,..m$$

Hedef Programlama Problemlerinin Modellenmesi

- Bu model DP haline dönüştüğü için simpleks metodu ile çözülebilir.

ÖZET

Orjinal Amaç Kısıt Denklemi	Amaç Programlama Gösterimi	Minimize Edilecek Sapma Değişkeni
$f(x) \leq b_i$	$f(x) + d_i^- - d_i^+ = b_i$	d_i^+
$f(x) \geq b_i$	$f(x) + d_i^- - d_i^+ = b_i$	d_i^-
$f(x) = b_i$	$f(x) + d_i^- - d_i^+ = b_i$	d_i^-, d_i^+

- Yukarıda açıklandığı gibi hedef programlama modelinin genel yapısındaki amaç fonksiyonunda yapılabilecek değişikliklere göre, hedef programlama iki gruba ayrılır;
  1. Önceliksiz (Ağırlıklı) Hedef Programlama (Non-Preemptive Goal Programming)
  2. Öncelikli Hedef Programlama (Preemptive Goal Programming)

  Şimdi bu iki durumu sırasıyla inceleyelim;

1.Önceliksiz (Ağırlıklı) Hedef Programlama

- Ağırlıklı hedef programlama olarak da ifade edilen önceliksiz hedef programlamada her hedef için bir ağırlık belirlenerek bu ağırlıklara göre amaç fonksiyonu oluşturulur.
- m adet hedeften oluşan bir hedef programlama probleminin i.nci hedefi

  $$min\ G_i\ ,\ (i=1,2,\ldots m)$$

  olarak ifade edildiğini varsayalım. Buna göre problemin amaç fonksiyonu

  $$min\ Z=w_1G_1+w_2G_2+\ldots+w_mG_m$$

  şeklinde tanımlanır. Burada w_i, karar vericinin her hedef için düşündüğü göreli önemi yansıtır. Örneğin her hedef için $w_i=1$ ise hedefler eşit ağırlıklı demektir. Ağırlık değerleri genellikle subjektif olarak belirlenir. Şimdi önceliksiz hedef programlamayı bir örnek üzerinde açıklayalım:

1.Önceliksiz (Ağırlıklı) Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)

- Bir otomobil firması yeni ürettiği bir model için televizyonda reklam yayınlamayı planlamaktadır. Otomobil firmasının reklam şirketine bildirdiği hedefler:

1. reklamı en az 40 milyon yüksek gelirli izlemelidir.(YG hedefi)
2. reklamı en az 60 milyon orta gelirli izlemelidir.(OG hedefi)
3. reklamı en az 35 milyon düşük gelirli izlemelidir.(DG hedefi)

- Reklam şirketi futbol maçı veya sinema arasında olmak üzere iki reklam kuşağını dikkate alacaktır. Otomobil şirketinin reklam bütçesi ise en fazla 600 birimdir. Reklamın kuşaklara göre bir dakikasının maliyeti ve dakikada ulaşılabilecek izleyici sayısı aşağıdaki tabloda verilmiştir. Bu verilere göre ve yukarıdaki üç hedefi dikkate alacak şekilde reklam planlaması yapılacaktır

	İzleyici Sayısı (milyon kişi / dk.)			Maliyet (birim / dk.)
	YG	OG	DG	
Futbol	7	10	5	100
Sinema	3	5	4	60

Önceliksiz Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)

	İzleyici Sayısı (milyon kişi / dk.)			Maliyet (birim / dk.)
	YG	OG	DG	
Futbol	7	10	5	100
Sinema	3	5	4	60

- Karar Değişkenleri:

- Bu problemde hangi reklam kuşağında ne kadar süreyle reklam yayınlanacağına karar verileceğine göre karar değişkenleri ;

X_1 Futbol arasında yayınlanacak reklam süresi (dk.)
X_2 Sinema arasında yayınlanacak reklam süresi (dk.)

- Buna göre aşağıdaki DP modelinin herhangi bir uygun çözümü otomobil firmasının hedeflerini gerçekleştirir.

Önceliksiz Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)

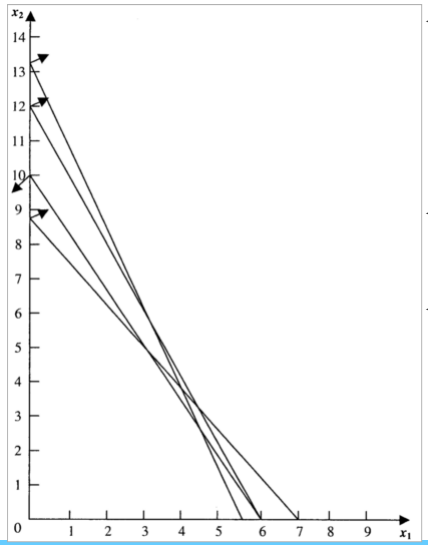
	İzleyici Sayısı (milyon kişi / dk.)			Maliyet (birim / dk.)
	YG	OG	DG	
Futbol	7	10	5	100
Sinema	3	5	4	60

- MODEL:

$$
\begin{aligned}
\text{Min (veya Maks) } z = \; & 0x_1 + 0x_2 \text{ (veya herhangi bir amaç fonksiyonu)} \\
& 7x_1 + 3x_2 \geq 40 \text{ (YG kısıtı)} \\
& 10x_1 + 5x_2 \geq 60 \text{ (OG kısıtı)} \\
& 5x_1 + 4x_2 \geq 35 \text{ (DG kısıtı)} \\
& 100x_1 + 60x_2 \leq 600 \text{ (Bütçe Kısıtı)} \\
& x_1 , x_2 \geq 0
\end{aligned}
$$

Önceliksiz Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)



- Ancak şekilde görüldüğü gibi, hem bütçe kısıtını, hem de diğer üç hedefi ortak olarak sağlayan hiç bir nokta olmadığından bu problemin uygun çözüm bölgesi bulunmamaktadır.
- Bu durumda hedeflerin karşılanması mümkün olmayacaktır.
- Bunun üzerine reklam şirketi otomobil firmasından her hedef için, hedeften bir birim uzaklaşmanın firmayı ne kadarlık bir zarara uğratacağını bildirmesini ister. Otomobil firmasından gelen bilgiler aşağıda verilmiştir.

Önceliksiz Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)

1. Reklamı izlemeyen 40 milyonun altındaki her 1 milyon (YG) için, firmanın satış gelirlerinde 200 birim kayıp ortaya çıkmaktadır.
2. Reklamı izlemeyen 60 milyonun altındaki her 1 milyon (OG) için, firmanın satış gelirlerinde 100 birim kayıp ortaya çıkmaktadır.
3. Reklamı izlemeyen 35 milyonun altındaki her 1 milyon (DG) için, firmanın satış gelirlerinde 50 birim kayıp ortaya çıkmaktadır.

Önceliksiz Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)

- Sapma Değişkenleri;

d_i^- = inci hedeften eksik olan miktar (negatif yöndeki sapma miktarı) (i=1, 2, 3)
d_i^+ = inci hedeften fazla olan miktar (pozitif yöndeki sapma miktarı) (i=1, 2, 3)

- Firmanın hedeflerinde meydana gelebilecek istenmeyen yöndeki sapmaları minimize edecek hedef prog. modeli;

$$
\begin{array}{lrcrcrcrcrcrcl}
\text{Min } z = & 200d_1^- & + & 100d_2^- & + & 50d_3^- & & & & & & & & \text{(Amaç fonksiyonu)} \\
 & 7x_1 & + & 3x_2 & + & d_1^- & - & d_1^+ & & & & & = 40 & \text{(YG kısıtı)} \\
 & 10x_1 & + & 5x_2 & + & d_2^- & - & d_2^+ & & & & & = 60 & \text{(OG kısıtı)} \\
 & 5x_1 & + & 4x_2 & + & d_3^- & - & d_3^+ & & & & & = 35 & \text{(DG kısıtı)} \\
 & 100x_1 & + & 60x_2 & & & & & & & & & \leq 600 & \text{(Bütçe Kısıtı)} \\
 & x_j,\ d_i^-,\ d_i^+ & & & & & & & & & & & \geq 0 & (j=1, 2)\ (i=1, 2, 3)
\end{array}
$$

- OPTİMAL ÇÖZÜM:

z=250, x_1=6, x_2=0, d_1^- = d_2^- = d_2^+= d_3^+=0 ve d_1^+=2, d_3^- =5

- $i$nci hedefe ait değişkenin amaç fonksiyonu katsayısına $i$nci hedefin ağırlığı denir ve hedefin önem derecesini gösterir.

Önceliksiz Hedef Programlama

ÖRNEK UYGULAMA (REKLAM ŞİRKETİ PROBLEMİ)

- Reklam bütçesinde ne kadarlık bir artış yapılırsa 3ncü hedef de tam olarak sağlanabilir?

- Bu maksatla, bütçe kısıtı da bir hedef haline dönüştürülmelidir. Burada dikkat edilmesi gereken bütçe hedefinin pozitif yöndeki sapma miktarının minimum yapılacağıdır. Ayrıca bütçe kısıtının pozitif yöndeki sapma miktarı d_4^+ 3ncü hedeften de önemsiz bir katsayı ile mesela 1) amaç fonksiyonuna eklenmelidir. Bu değişikliklerden sonra elde edilen model ve optimal çözümü:

$$
\begin{array}{lllllll}
\text{Min } z = & 200d_1^- + & 100d_2^- + & 50d_3^- + & d_4^+ & & \text{(Amaç fonksiyonu)} \\
& 7x_1 + & 3x_2 + & d_1^- - & d_1^+ & = 40 & \text{(YG kısıtı)} \\
& 10x_1 + & 5x_2 + & d_2^- - & d_2^+ & = 60 & \text{(OG kısıtı)} \\
& 5x_1 + & 4x_2 + & d_3^- - & d_3^+ & = 35 & \text{(DG kısıtı)} \\
& 100x_1 + & 60x_2 + & d_4^- - & d_4^+ & = 600 & \text{(Bütçe Kısıtı)} \\
& x_j,\ d_i^-,\ d_i^+ & & & & \geq 0 & (j=1, 2)\ (i=1, 2, 3, 4)
\end{array}
$$

- OPTİMAL ÇÖZÜM:

$z=33.33,\quad x_1=4.33,\quad x_2=3.33,\quad d_1^+=0.33,\quad d_4^+=33.33$ ve $d_1^-=d_2^-=d_3^-=d_4^-=d_2^+=d_3^+=0$

2.Öncelikli Hedef Programlama

- Yukarıdaki örnekte otomobil firmasının hedeflerinin göreceli olarak önemini kesin olarak belirleyebildiği kabul edilmektedir. Örneğin, YG hedefinin OG hedefinin 200/100=2 kat öneme sahip olduğu ve OG hedefinin de DG hedefine göre 2 kat öneme sahip olduğu belirlenmiş ve model ona göre kurulmuştu. Gerçek hayatta karar verici çoğu zaman hedeflerin göreceli önemlerini kesin olarak belirleyemez. Bu durumda Öncelikli Hedef Programlama kullanılması daha uygun olur. Öncelikli Hedef Programlamada karar vericinin en önemliden (1. hedeften) en önemsize (m.ci hedef) doğru hedeflerini sıralaması gerekmektedir. i hedefine ait değişkenin amaç fonksiyonu katsayısı olarak P_i kullanılacak olursa;

$$P_1 >> P_2 >> P_3 >> \ldots >> P_m$$

olduğu varsayılır. Yani 1. hedefin ağırlığı 2. hedefin ağırlığından çok büyük olmak üzere, hedefler arasında öncelikler oluşturulur.

Öncelikli Hedef Programlama

- Reklam şirketi ile ilgili örnekte öncelikler sırasıyla YG, OG ve DG şeklinde ise öncelikli hedef programlama modeli

$$\min Z = P_1 d_1^- + P_2 d_2^- + P_3 d_3^-$$

$$7x_1 + 3x_2 + d_1^- - d_1^+ = 40 \quad \text{(YG kısıtı)}$$

$$10x_1 + 5x_2 + d_2^- - d_2^+ = 60 \quad \text{(OG kısıtı)}$$

$$5x_1 + 4x_2 + d_3^- - d_3^+ = 35 \quad \text{(DG kısıtı)}$$

$$100x_1 + 60x_2 \leq 600$$

$$x_j \geq 0 \qquad j=1,2$$

$$d_i^-, d_i^+ \geq 0 \qquad i=1,..3$$

- Şimdi konunun daha iyi anlaşılabilmesi için teçhizat alım probleminin öncelikli hedef programlama modelinin kurulması gösterelim.

Öncelikli Hedef Programlama

ÖRNEK UYGULAMA (TEÇHİZAT ALIM PROBLEMİ)

- Bir otomotiv yan sanayi firması üretiminde kullanmak üzere dört cins makina ve teçhizatın (torna tezgahı, freze tezgahı, kumpas ve rontgen cihazı) alımını planlanmakta olup bunun için toplam 3500 birim ödenek ayırmıştır. Her cins makina ve teçhizatın maliyeti (birim satış fiyatı), yıllık bakım ve işletme gideri ve sağlayacağı faydayı gösteren etkinlik puanı ile ihtiyaç miktarı aşağıdaki tabloda verilmiştir. Bu proje ile ilgili hedefler öncelik sırasına göre aşağıda verilmiştir:

1. Toplam etkinliğin en az 80 puan olması.
2. Yıllık toplam bakım ve işletme giderinin mümkün olduğu kadar 70 birimi aşmaması.
3. Torna tezgahı ihtiyacının mümkün olduğu kadar tam karşılanması.
4. Diğer makina ve teçhizat ihtiyaçlarının ise ihtiyaç miktarından az olmayacak şekilde karşılanması.

Buna göre öncelikli hedef programlama modelini oluşturunuz.

Öncelikli Hedef Programlama

ÖRNEK UYGULAMA (TEÇHİZAT ALIM PROBLEMİ)

Cinsi	Birim Maliyet	Bakım – İşletme Gideri	Etkinlik Puanı	İhtiyaç
Torna Tezgahı	600	4	6	5
Freze Tezgahı	250	3	4	5
Kumpas	150	2	5	10
Rontgen Cihazı	40	1	3	20

- KARAR DEĞİŞKENLERİ:
  - X_1 Satın alınacak torna tezgahı miktarı.
  - X_2 Satın alınacak freze tezgahı miktarı.
  - X_3 Satın alınacak kumpas miktarı.
  - X_4 Satın alınacak rontgen cihazı miktarı.
- SAPMA DEĞİŞKENLERİ:
  - d_i^- *i*nci hedeften negatif yönde sapma miktarı ($i=1,2,\ldots,6$)
  - d_i^+ *i*nci hedeften pozitif yönde sapma miktarı ($i=1,2,\ldots,6$)

Öncelikli Hedef Programlama

ÖRNEK UYGULAMA (TEÇHİZAT ALIM PROBLEMİ)

Cinsi	Birim Maliyet	Bakım – İşletme Gideri	Etkinlik Puanı	İhtiyaç
Torna Tezgahı	600	4	6	5
Freze Tezgahı	250	3	4	5
Kumpas	150	2	5	10
Rontgen Cihazı	40	1	3	20

- KISITLAR:

$$6x_1 + 4x_2 + 5x_3 + 3x_4 + d_1^- - d_1^+ = 80 \quad \text{(Etkinlik Kısıtı)}$$
$$4x_1 + 3x_2 + 2x_3 + x_4 + d_2^- - d_2^+ = 70 \quad \text{(Bakım-İşletma Kısıtı)}$$
$$x_1 + d_3^- - d_3^+ = 5 \quad \text{(Torna Tezgahı Kısıtı)}$$
$$x_2 + d_4^- - d_4^+ = 5 \quad \text{(Freze Tezgahı Kısıtı)}$$
$$x_3 + d_5^- - d_5^+ = 10 \quad \text{(Kumpas Kısıtı)}$$
$$x_4 + d_6^- - d_6^+ = 20 \quad \text{(Rontgen Cihazı Kısıtı)}$$
$$600x_1 + 250x_2 + 150x_3 + 40x_4 \le 3500 \quad \text{(Bütçe Kısıtı)}$$

Öncelikli Hedef Programlama

ÖRNEK UYGULAMA (TEÇHİZAT ALIM PROBLEMİ)

AMAÇ FONKSİYONU:

1.Hedef: Toplam etkinliğin en az 80 puan olması	$\text{Min } z_1 = d_1^-$
2.Hedef: Yıllık toplam bakım-işletme maliyetinin en fazla 70 birim olması	$\text{Min } z_2 = d_2^+$
3.Hedef: Torna tezgahı ihtiyacının tam olarak karşılanması	$\text{Min } z_3 = d_3^- + d_3^+$
4.Hedef: Diğer makina ve teçhizatların en az ihtiyaç miktarı kadar karşılanması	$\text{Min } z_4 = d_4^- + d_5^- + d_6^-$

$P_1 >> P_2 >> P_3 >> P_4$

olmak üzere problemin bütünleşik amaç fonksiyonu:

$$\text{Min } z = P_1d_1^- + P_2d_2^+ + P_3d_3^- + P_3d_3^+ + P_4d_4^- + P_4d_5^- + P_4d_6^-$$

Öncelikli Hedef Programlama

ÖRNEK UYGULAMA (TEÇHİZAT ALIM PROBLEMİ)

HEDEF PROGRAMLAMA MODELİ VE OPTİMAL ÇÖZÜMÜ:

$$\text{Min } z = P_1d_1^- + P_2d_2^+ + P_3d_3^- + P_3d_3^+ + P_4d_4^- + P_4d_5^- + P_4d_6^-$$

$$\begin{aligned}
6x_1 + 4x_2 + 5x_3 + 3x_4 + d_1^- - d_1^+ &= 80 \\
4x_1 + 3x_2 + 2x_3 + x_4 + d_2^- - d_2^+ &= 70 \\
x_1 + d_3^- - d_3^+ &= 5 \\
x_2 + d_4^- - d_4^+ &= 5 \\
x_3 + d_5^- - d_5^+ &= 10 \\
x_4 + d_6^- - d_6^+ &= 20 \\
600x_1 + 250x_2 + 150x_3 + 40x_4 &\leq 3500 \\
x_j,\ d_i^-,\ d_i^+\ (j=1, 2, 3, 4)\ (i=1, 2, \ldots, 6) &\geq 0
\end{aligned}$$

$x_1=4$, $x_2=0$, $x_3=2$, $x_4=20$, $d_1^+=14$, $d_2^-=30$, $d_3^-=1$, $d_4^-=5$, $d_5^-=8$ ve $d_1^-= d_6^-= d_2^+= d_3^+= d_4^+= d_5^+= d_6^+=0$

Hedef Programlama Modellerinin Çözümü

- Önceliksiz Hedef Programlama Modeli yapı olarak Doğrusal Programlama modeline çok benzediğinden, normal simpleks metodu veya bilgisayar yazılımlarını (lındo, lıngo vb.) kullanarak çözülebilir.

- Öncelikli hedef programlama modellerinin çözümünde bazı farklılıklar vardır. Bu farklılıklar aşağıdaki başlıklar altında incelenecektir.

  1. Hedef programlama modellerinin simpleks algoritması ile çözümü

  2. Hedef programlama modellerinin lindo ile çözümü

Hedef Programlama Modellerinin Çözümü

1. HEDEF PROGRAMLAMADA SİMPLEKS ALGORİTMASI

- Öncelikli hedef programlama problemleri aşağıda simpleks metodundan farklılıkları belirtilen ve simpleks metodunun bir uzantısı olan hedef programlama simpleksi ile çözülebilir.

1. Normal simpleks tablosunda sadece bir 0) satırı var iken, *m* hedef bulunan bir hedef programlama simpleks tablosunda her hedef için bir (0) satırı olmak üzere toplam *m* adet (0) satırı vardır. Her amaç fonksiyonu ilgili hedefteki istenmeyen yöndeki sapmadan doğan cezayı (maliyeti) temsil etmektedir. Buna göre herhangi bir *i* hedefinin amaç fonksiyonu değerinin sıfırdan büyük olması $Z_i>0$ *i* hedefinde istenmeyen yönde bir sapma olduğunu gösterir. Z_i değeri sıfıra yaklaştıkça hedefe yaklaşılmakta olup, $Z_i=0$ ise *i* hedefi elde edilmiş demektir.

Hedef Programlama Modellerinin Çözümü

1. HEDEF PROGRAMLAMADA SİMPLEKS ALGORİTMASI

1. Hedef programlama probleminde giren değişken aşağıda açıklandığı şekilde belirlenir.
   - Henüz elde edilmemiş en öncelikli hedef bulunur. Başka bir ifade ile $z_i>0$ olan en öncelikli i hedefi bulunur.
   - *i* hedefinin (0) satırında katsayısı en pozitif olan değişken giren değişken olarak seçilir. Ancak bu hedeften daha öncelikli bir hedefin (0) satırındaki aynı değişkenin katsayısı negatif ise o değişken temele giremez. Bu durumda bir sonraki (0) satırına (*i+1* hedefine) geçilir ve *i+1* hedefine yaklaşılmaya çalışılır.

2. Herhangi bir satır işlemi yapıldığında bu işlem mutlaka her hedefin (0) satırına uygulanır.

3. Eğer bütün hedefler elde edilmiş ise ($z_1=z_2=\ldots=z_m=0$ ise) veya elde edilmemiş her *i* hedefi için, temele girebilecek ve z_i değerini azaltabilecek bütün değişkenler i'den daha öncelikli bir hedefdeki sapmayı artırıyorsa optimal çözüm elde edilmiş demektir.

Hedef Programlama Modellerinin Çözümü

1. HEDEF PROGRAMLAMADA SİMPLEKS ALGORİTMASI (REKLAM ŞİRKETİ PROBLEMİ)

- Reklam şirketi örneğindeki hedeflerin öncelikleri sırası ile (YG) hedefi, (OG) hedefi ve (DG) hedefi olarak kabul edilirse, bu problemin öncelikli hedef programlama modeli aşağıdaki şekilde olur.

$$
\begin{array}{lrcrcrcrcll}
\text{Min } z = & P_1 d_1^- & + & P_2 d_2^- & + & P_3 d_3^- & & & & & \text{(Amaç fonksiyonu)} \\
 & 7x_1 & + & 3x_2 & + & d_1^- & - & d_1^+ & = & 40 & \text{(YG kısıtı)} \\
 & 10x_1 & + & 5x_2 & + & d_2^- & - & d_2^+ & = & 60 & \text{(OG kısıtı)} \\
 & 5x_1 & + & 4x_2 & + & d_3^- & - & d_3^+ & = & 35 & \text{(DG kısıtı)} \\
 & 100x_1 & + & 60x_2 & & & & & \leq & 600 & \text{(Bütçe Kısıtı)} \\
 & x_j,\ d_i^-,\ d_i^+ & & & & & & & \geq & 0 & (j=1, 2)\ (i=1, 2, 3)
\end{array}
$$

- Her bir hedef için amaç fonksiyonu satırı;

$$
\begin{array}{ll}
\text{1nci hedefin (0) satırı} : & z_1 - P_1 d_1^- = 0 \\
\text{2nci hedefin (0) satırı} : & z_2 - P_2 d_2^- = 0 \\
\text{3ncü hedefin (0) satırı} : & z_3 - P_3 d_3^- = 0
\end{array}
$$

Hedef Programlama Modellerinin Çözümü

1. HEDEF PROGRAMLAMADA SİMPLEKS ALGORİTMASI (REKLAM ŞİRKETİ PROBLEMİ)

Satır	TD	Katsayılar									STD
		x_1	x_2	d_1^+	d_2^+	d_3^+	d_1^-	d_2^-	d_3^-	s_4	
0 (YG)	z_1	0	0	0	0	0	$-P_1$	0	0	0	0
0 (OG)	z_2	0	0	0	0	0	0	$-P_2$	0	0	0
0 (DG)	z_3	0	0	0	0	0	0	0	$-P_3$	0	0
(YG)	?	7	3	−1	0	0	1	0	0	0	40
(OG)	?	10	5	0	−1	0	0	1	0	0	60
(DG)	?	5	4	0	0	−1	0	0	1	0	35
Bütçe	s_4	100	60	0	0	0	0	0	0	1	600

XP_1 XP_2 XP_3

Satır	TD	Katsayılar									STD
		x_1	x_2	d_1^+	d_2^+	d_3^+	d_1^-	d_2^-	d_3^-	s_4	
0 (YG)	z_1	$7P_1$	$3P_1$	$-P_1$	0	0	0	0	0	0	$40P_1$
0 (OG)	z_2	$10P_2$	$5P_2$	0	$-P_2$	0	0	0	0	0	$60P_2$
0 (DG)	z_3	$5P_3$	$4P_3$	0	0	$-P_3$	0	0	0	0	$35P_3$
(YG)	d_1^-	7	3	−1	0	0	1	0	0	0	40 (40/7 *min.*)
(OG)	d_2^-	10	5	0	−1	0	0	1	0	0	60 (60/10)
(DG)	d_3^-	5	4	0	0	−1	0	0	1	0	35 (35/5)
Bütçe	s_4	100	60	0	0	0	0	0	0	1	600 (600/100)

Hedef Programlama Modellerinin Çözümü

1. HEDEF PROGRAMLAMADA SİMPLEKS ALGORİTMASI (REKLAM ŞİRKETİ PROBLEMİ)

Satır	TD	Katsayılar									STD
		x_1	x_2	d_1^+	d_2^+	d_3^+	d_1^-	d_2^-	d_3^-	s_4	
0 (YG)	z_1	0	0	0	0	0	$-P_1$	0	0	0	0
0 (OG)	z_2	0	$5P_2/7$	$10P_2/7$	$-P_2$	0	$-10P_2/7$	0	0	0	$20P_2/7$
0 (DG)	z_3	0	$13P_3/7$	$5P_3/7$	0	$-P_3$	$-5P_3/7$	0	0	0	$45P_3/7$
(YG)	x_1	1	3/7	−1/7	0	0	1/7	0	0	0	40/7
(OG)	d_2^-	0	5/7	10/7	−1	0	−10/7	1	0	0	20/7
(DG)	d_3^-	0	13/7	5/7	0	−1	−5/7	0	1	0	45/7
Bütçe	s_4	0	120/7	100/7	0	0	−100/7	0	0	1	200/7

1'inci hedef elde edildi

Hedef Programlama Modellerinin Çözümü

1. HEDEF PROGRAMLAMADA SİMPLEKS ALGORİTMASI (REKLAM ŞİRKETİ PROBLEMİ)

Satır	TD	Katsayılar									STD
		x_1	x_2	d_1^+	d_2^+	d_3^+	d_1^-	d_2^-	d_3^-	s_4	
0 (YG)	z_1	0	0	0	0	0	$-P_1$	0	0	0	0
0 (OG)	z_2	0	$-P_2$	0	$-P_2$	0	0	0	0	$-P_2/10$	0
0 (DG)	z_3	0	P_3	0	0	$-P_3$	0	0	0	$-P_3/20$	$5P_3$
(YG)	x_1	1	3/5	0	0	0	0	0	0	−1/100	6
(OG)	d_2^-	0	−1	0	−1	0	0	1	0	−1/10	0
(DG)	d_3^-	0	1	0	0	−1	0	0	1	−1/20	5
Bütçe	d_1^+	0	6/5	1	0	0	−1	0	0	7/100	2

- 3ncü hedef için X_2 nin temele girmesi, daha öncelikli bir hedef olan 2nci hedeften uzaklaşmaya yol açar. çünkü 2nci hedefin X_2 sütununda negatif kaysayı bulunmaktadır. Ayrıca 3ncü hedefin (0) satırında pozitif değere sahip başka katsayı olmadığından elde edilen bu tablo aynı zamanda optimal çözüm tablosudur. modelin optimal çözümü:

$$z=5P_3,\ x_1=6,\ x_2=0,\ d_1^- = d_2^- = d_2^+ = d_3^+=0,\ d_1^+=2,\ d_3^- =5$$

Hedef Programlama Modellerinin Çözümü

2. ÖNCELİKLİ HEDEF PROGRAMLAMA PROBLEMLERİNİN BİLGİSAYAR İLE ÇÖZÜMÜ

ADIM 1 :Hedef amaç fonksiyonları, önceliklerine göre büyükten küçüğe sıralanır.

ADIM 2 :Çözümü yapılmamış önceliği en büyük olan hedefin amaç fonksiyonu modelin amaç fonksiyonu olarak alınır ve optimal çözüm bulunur.

ADIM 3 :Bütün amaç fonksiyonları çözülmüşse işlem durdurulur. Bu durumda en son öncelikli hedefin optimal çözümü, öncelikli hedef programlama probleminin optimal çözümüdür. Eğer çözülmemiş amaç fonksiyonu varsa Adım 4'e gidilir.

ADIM 4 :Çözülen amaç fonksiyonu, bulunan optimal amaç fonksiyonu değerine eşit olacak şekilde alınarak modele kısıt olarak eklenir ve Adım 2'ye gidilir.

Hedef Programlama Modellerinin Çözümü

2. ÖNCELİKLİ HEDEF PROGRAMLAMA PROBLEMLERİNİN BİLGİSAYAR İLE ÇÖZÜMÜ (REKLAM ŞİRKETİ PROBLEMİ)

Öncelik Sırasında Amaç Fonksiyonları

$$\begin{aligned} \text{Min } z_1 &= d_1^- \\ \text{Min } z_2 &= d_2^- \\ \text{Min } z_2 &= d_3^- \end{aligned}$$

Başlangıç Kısıtları

$$\begin{array}{rcrcrcrl} 7x_1 & + & 3x_2 & + & d_1^- & - & d_1^+ & = 40 \\ 10x_1 & + & 5x_2 & + & d_2^- & - & d_2^+ & = 60 \\ 5x_1 & + & 4x_2 & + & d_3^- & - & d_3^+ & = 35 \\ 100x_1 & + & 6x_2 & & & & & \leq 600 \\ x_j,\ d_i^-,\ d_i^+ & & (j=1,2)\ (i=1,2,3) & & & & & \geq 0 \end{array}$$

Hedef Programlama Modellerinin Çözümü

2. ÖNCELİKLİ HEDEF PROGRAMLAMA PROBLEMLERİNİN BİLGİSAYAR İLE ÇÖZÜMÜ (REKLAM ŞİRKETİ PROBLEMİ)

BİRİNCİ HEDEF

$$\begin{array}{rcrcrcrl} \text{Min } z_1 = d_1^- & & & & & & & \\ 7x_1 & + & 3x_2 & + & d_1^- & - & d_1^+ & = 40 \\ 10x_1 & + & 5x_2 & + & d_2^- & - & d_2^+ & = 60 \\ 5x_1 & + & 4x_2 & + & d_3^- & - & d_3^+ & = 35 \\ 100x_1 & + & 6x_2 & & & & & \leq 600 \\ x_j,\ d_i^-,\ d_i^+ & & (j=1,2)\ (i=1,2,3) & & & & & \geq 0 \end{array}$$

- Bu modelin optimal çözümü
  $z_1=d_1^-=0$
  elde edilir. Yani 1nci hedef sağlanmıştır. 1nci hedefi bozmadan 2nci hedefi elde etmek için
  $d_1^-=0$
  Kısıtı modele eklenerek 2nci hedefin istenmeyen yöndeki sapması d_2^- değişkeni` minimize edilir.

Hedef Programlama Modellerinin Çözümü

2. ÖNCELİKLİ HEDEF PROGRAMLAMA PROBLEMLERİNİN BİLGİSAYAR İLE ÇÖZÜMÜ (REKLAM ŞİRKETİ PROBLEMİ)

İKİNCİ HEDEF

$$
\begin{aligned}
\text{Min } z_2 = \; & d_2^- \\
7x_1 + 3x_2 + d_1^- - d_1^+ &= 40 \\
10x_1 + 5x_2 + d_2^- - d_2^+ &= 60 \\
5x_1 + 4x_2 + d_3^- - d_3^+ &= 35 \\
100x_1 + 6x_2 &\le 600 \\
\boxed{d_1^- = 0} & \\
x_j,\; d_i^-,\; d_i^+ \; (j=1, 2)\; (i=1, 2, 3) &\ge 0
\end{aligned}
$$

- Bu modelin optimal çözümü
  $z_2=d_2^-=0$
  elde edilir. Yani 2nci hedef de sağlanmıştır. 1nci ve 2nci hedefi bozmadan 3ncü hedefi elde etmek için
  $d_2^-=0$
  kısıtı modele eklenerek 3ncü hedefin istenmeyen yöndeki sapması d_3^- değişkeni minimize edilir.

Hedef Programlama Modellerinin Çözümü

2. ÖNCELİKLİ HEDEF PROGRAMLAMA PROBLEMLERİNİN BİLGİSAYAR İLE ÇÖZÜMÜ (REKLAM ŞİRKETİ PROBLEMİ)

ÜÇÜNCÜ HEDEF

$$
\begin{aligned}
\text{Min } z_2 = \; & d_3^- \\
7x_1 + 3x_2 + d_1^- - d_1^+ &= 40 \\
10x_1 + 5x_2 + d_2^- - d_2^+ &= 60 \\
5x_1 + 4x_2 + d_3^- - d_3^+ &= 35 \\
100x_1 + 6x_2 &\le 600 \\
\boxed{d_1^- = 0} & \\
\boxed{d_2^- = 0} & \\
x_j,\; d_i^-,\; d_i^+ \; (j=1, 2)\; (i=1, 2, 3) &\ge 0
\end{aligned}
$$

- Bu modelin optimal çözümü
  $z = 5$, $X_1=6$, $X_2=0$, $d_1^-= d_2^-= d_1^+= d_2^+= d_3^+=0$ VE $d_1^+=2$, $d_3^-=5$
  bulunur.
- Bu model gerçekleştirilmesi gereken son hedefi ifade ettiğine göre, bu modelin optimal çözümü aynı zamanda öncelikli hedef programlama modelinin optimal çözümüdür.

BÖLÜM II: Oyun Teorisi

- OYUN TEORİSİNE GİRİŞ
- TEMEL KAVRAMLAR
- İKİ KİŞİLİ SIFIR TOPLAMLI OYUNLAR
- İKİ KİŞİLİ SABİT TOPLAMLI OYUNLAR
- İKİ KİŞİLİ SABİT TOPLAMLI OLMAYAN OYUNLAR

Oyun Teorisine Giriş

- Bu bölümde çeşitli belirsizlik ortamlarında iki ya da daha fazla rakip arasında gelişen ve her birinin diğerlerine karşı en iyi hareket tarzını bulmaya çalıştığı karar verme süreçleri birer oyun mantığıyla ele alınıp bunlarla ilgili çözüm teknikleri incelenecektir

Oyun Teorisine Giriş

- Oyun teorisi ilk olarak 1928 yılında John Von Neumann tarafından ortaya konmuş olup halen oldukça karmaşık yapıdaki rekabet ortamlarına yönelik araştırmalar devam etmekte olup bu kısımda en basit durum olan iki kişili oyunlar ve n-kişili oyunlar üzerinde durulacaktır.
- Oyun teorisine, çok kişili karar teorisi adı da verilmektedir .

Temel Kavramlar

- Oyuncu
- Oyunlardaki karar vericilerden (taraflardan) her birine oyuncu denir.

- Strateji
- Oyuncuların uygulayabilecekleri hareket tarzlarına strateji denir. Her bir oyuncu için en az iki olmak üzere stratejiler sonlu yada sonsuz sayıda olabilir.

- Kazanç
- Oyuncuların kullanacakları stratejilere bağlı olarak elde edecekleri değerdir. Kazanç değeri pozitif (kar) ya da negatif (zarar) olabilir.

Temel Kavramlar

Oyun Durumlarının Türleri

- Rekabetin veya çatışmanın söz konusu olduğu oyun durumları çeşitli sınıflara ayrılabilir. Bir sınıflandırma, oyuncu denilen karar vericinin durumuna göre yapılır. İki oyuncunun bulunduğu oyunlara iki-kişili oyunlar ikiden fazla oyuncunun olduğu oyunlara n-kişili oyunlar denir.
- Oyunlar, oyuncuların kazançları ve kayıpları cinsinden sonuçlarına göre de; Oyuncuların stratejilerinin herhangi bir kombinasyonunda (yani oyuncular hangi stratejiyi seçerlerse seçsinler), oyuncuların kazançlarının toplamının bir c sabit değerine eşit olduğu oyunlara sabit-toplamlı oyunlar, diğerlerine de sabit toplamlı olmayan oyunlar denir.

Temel Kavramlar

Oyun Durumlarının Türleri

- Oyuncuların kazanç ve kayıpları toplamı sıfır ise, oyuna sıfır-toplamlı oyun denir (c=0 olan sabit toplamlı oyun).
- İki-kişili sıfır-toplamlı bir oyunda, oyunculardan birinin kazancı diğerinin kaybına eşittir. İki kişilik sıfır toplamlı oyunlar matematiksel olarak basit olduklarından oyun teorisinin ilkelerinin tanıtılmasında en sık kullanılan oyun türüdür. Bu bölümde de geniş olarak bu konuya yer verilecektir.

Temel Kavramlar

- Bütün oyun problemleri aşağıdaki kabullere göre incelenir.

VARSAYIMLAR

1. Oyuncular oldukça mantıklı kişilerdir.
2. Oyuncular sadece kendi faydalarını artıracak stratejileri seçerler.
3. Oyuncular riske girmeden kendileri için garanti olan en iyi kazancı elde etmeye çalışırlar.
4. Oyunlarda belirsizlik hakimdir, yani oyuncular oyuna başlamadan önce rakibinin hangi stratejiyi kullanacağını bilmezler.

İki Kişili Sıfır Toplamlı Oyunlar

- İki-kişili sıfir-toplamlı oyunların karakteristikleri
  1. İki adet oyuncu vardır ve bunlar satır oyuncusu ve sütun oyuncusu olarak adlandırılırlar.
  2. Oyunculardan birinin kazancı (karı) diğerinin kaybına (zararına) eşittir, oyuncular hangi stratejilerini seçerse seçsinler iki oyuncunun kazançları toplamı sıfır yapar, yani bir oyuncunun kazancı diğer oyuncudan gelmektedir.
  3. Satır oyuncusu toplam m adet stratejiden birini kullanırken sütun oyuncusu da aynı anda n adet stratejiden birini kullanır.

İki Kişili Sıfır Toplamlı Oyunlar

- Oyun problemlerinde oyuncuların bütün stratejilerine karşılık gelen kazanç değerleri bir matris şeklinde gösterilir ve bu matrise kazanç matrisi adı verilir.
- İki-kişili sıfır-toplamlı bir oyunun kazanç matrisinin genel yapısı aşağıdaki şekilde verilmiştir. Matristeki kazanç değerleri genellikle satır oyuncusunun kazancı olarak ifade edilir. Toplam sıfır olduğuna göre sütun oyuncusunun kazancı, kazanç matrisindeki değerlerin -1 ile çarpılmasıyla elde edilir.

		Sütun Oyuncusunun Stratejisi			
		Strateji 1	Strateji 2	…	Strateji *n*
Satır Oyuncusunun Stratejileri	Strateji 1	a_{11}	a_{12}	…	a_{1n}
	Strateji 2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	…	:
	Strateji *m*	a_{m1}	a_{m2}	…	a_{mn}

- Satır oyuncusu *i* stratejisini ve sütun oyuncusu *j* stratejisini seçerse, satır oyuncusu a_{ij} kazancını elde ederken sütun oyuncusu a_{ij} miktarını kaybeder.

İki Kişili Sıfır Toplamlı Oyunlar

- Örnek (Tek-Çift oyunu)
- Bu oyunda her iki oyuncu aynı anda ya tek yada çift parmağını gösterir. Parmak sayılarının toplamı tek sayı olursa, (yani oyuncular farklı sayıda parmak gösterirse) satır oyuncusu, çift sayı olursa sütun oyuncusu kazanır. Kazanan oyuncu kaybedenden 1 birim alır. Her iki oyuncunun da iki hareket tarzı (yani stratejisi) vardır: bir parmak veya iki parmak göstermek.

		Sütun Oyuncusu	
		1	2
Satır Oyuncusu	1	-1	1
	2	1	-1

İki Kişili Sıfır Toplamlı Oyunlar

- Örnek (Tek-Çift oyunu)
- İki kişi arasında oynanan bir oyunda oyuncular aynı anda hem bir ya da iki parmak göstermekte, hem de "tek" ya da "çift" diye bağırarak rakibinin parmak sayısını tahmin etmektedirler Karşısındakinin parmak sayısını doğru tahmin eden oyuncu her iki oyuncunun parmak sayılarının toplamı kadar puan kazanmakta, yanlış tahminde bulunan oyuncu ise toplam parmak sayısı kadar puan kaybetmektedir. Her iki oyuncu da doğru tahmin etmiş ya da her ikisi de bilememiş ise beraberlik söz konusu olup oyuncular sıfır puan almaktadırlar.
- Burada her iki oyuncu için kullanabilecek dört strateji vardır:
  - 1nci strateji :tek parmak gösterip "tek" söylemek (TT)
  - 2nci strateji :tek parmak gösterip "çift" söylemek (TÇ)
  - 3ncü strateji :çift parmak gösterip "tek" söylemek (ÇT)
  - 4ncü strateji :çift parmak gösterip "çift" söylemek (ÇÇ)

İki Kişili Sıfır Toplamlı Oyunlar

- Örnek (Tek-Çift oyunu)
- Değerler satır oyuncusunun kazançlarını göstermek üzere stratejilere bağlı kazanç matrisi aşağıdaki şekilde düzenlenir.

		Sütun Oyuncusu			
		1 (TT)	2 (TÇ)	3 (ÇT)	4 (ÇÇ)
Satır Oyuncusu	1 (TT)	0	2	-3	0
	2 (TÇ)	-2	0	0	3
	3 (ÇT)	3	0	0	-4
	4 (ÇÇ)	0	-3	4	0

İki Kişili Sıfır Toplamlı Oyunlar

Denge Noktası ve Kararlı Oyunlar

- Sonraki slaytta verilen kazanç matrisinde iki kişili sıfır toplamlı oyunun her iki oyuncu tarafından nasıl en iyi şekilde oynayacağını, yani satır ve sütun oyuncularının hangi stratejileri kullanmaları gerektiğini inceleyelim. Burada her oyuncu rakibinin kendi kullanacağı stratejiyi bildiğini kabul ederek kendisi için en iyi hareket tarzını seçmeye çalışacaktır.

İki Kişili Sıfır Toplamlı Oyunlar

Denge Noktası ve Kararlı Oyunlar

- Genel olarak oyuncuların herbirinin en iyi stratejiyi belirlemede kullanacağı yaklaşım şu şekilde açıklanabilir;
  - Satır oyuncusu kendi stratejilerinin her biri için, kazanabileceği minimum kazancı saptar ve bunlar arasından maksimum değerli kazancın bulunduğu stratejiyi seçer.
  - Sütun oyuncusu ise her bir stratejisinden kaybedebileceği, maksimum değerleri saptar ve bu maksimum kayıplar arasından minimum kaybın bulunduğu stratejiyi seçer.
- BÖYLECE HER OYUNCU, RAKİBİ NE SEÇERSE SEÇSİN KENDİ STRATEJİSİ İLE BELİRLEDİĞİ SONUÇTAN DAHA KÖTÜSÜ İLE KARŞILAŞAMAMAYI GARANTİ ALTINA ALIR.

İki Kişili Sıfır Toplamlı Oyunlar

Denge Noktası ve Kararlı Oyunlar

- Satır oyuncusu minimum beklenen kazancını maksimum yapan stratejiyi (maksimin), sütun oyuncusu ise maksimum beklenen kaybını minimum yapan stratejiyi (minimaks) seçer.

- Aşağıdaki koşulu sağlayan oyuna denge noktasına sahiptir denir.

maks(satır minimumları)=min(sütun maksimumları)

maksimin=minimaks

- Denge noktasına sahip oyunlara kararlı oyun denir ve yukarıdaki eşitliği sağlayan nokta oyunun denge noktasını oluşturur.

İki Kişili Sıfır Toplamlı Oyunlar

Denge Noktası ve Kararlı Oyunlar

Örnek

		Sütun Oyuncusunun Stratejisi			Satır Minimumu
		Strateji 1	Strateji 2	Strateji 3	
Satır Oyuncusunun Stratejisi	Strateji 1	4	4	10	4
	Strateji 2	2	3	1	1
	Strateji 3	6	5	7	5
Sütun Maksimumu		6	5	10	

maksimin=minimaks=5 olduğundan matristeki 3ncü satır ile 2nci sütun, (3,2) noktası, oyunun denge noktasıdır.

İki Kişili Sıfır Toplamlı Oyunlar

Denge Noktası ve Kararlı Oyunlar

- Saf strateji
- Denge noktasını oluşturan satır, satır oyuncusunun saf stratejisi, sütun ise sütun oyuncusunun saf stratejisidir.
- Denge noktasına sahip oyunların optimum çözümüne göre her oyuncu sadece saf stratejisini kullanır Saf stratejisini kullanmayan oyuncunun durumunda iyileşme söz konusu olmaz

- Oyunun değeri (v)
- Oyunun optimum çözümüne göre, satır oyuncusunun kazanacağı ve sütun oyuncusunun kaybedeceği değere oyunun değeri denir. Sıfır toplamlı oyunlarda her iki oyuncu için de oyunun değeri aynıdır. Dengeli oyunlarda oyunun değeri denge noktasındaki kazanç değerine eşittir.
- Bir önceki örnekte oyunun değeri v=5 dir.

İki Kişili Sıfır Toplamlı Oyunlar

Denge Noktası ve Kararlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Herhangi bir oyuncunun herhangi bir *i* stratejisi her zaman (rakibin bütün hareket tarzlarına karşı) en az diğer bir *i'* stratejisinin sağladığı faydayı sağlıyor ve rakibin en az bir stratejisi karşısında da *i'* stratejisinden daha iyi bir fayda sağlıyor ise *i* stratejisi *i'* stratejisine göre üstündür denir ve *i'* stratejisini alt eder (saf dışı bırakır). Alt edilen strateji kazanç matrisinden çıkarılarak bundan sonraki işlemlerde göz önünde bulundurulmaz.

İki Kişili Sıfır Toplamlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Örnek (Reklam Kampanyası)
- Rakip iki otomobil firması kış dönemi satışlarını artırmak amacıyla ekim ayında iki hafta süre ile reklam faaliyetlerine başlamayı planlamaktadır. Firmalar radyo veya televizyon olmak üzere iki ortam üzerinde yoğunlaşmakta olup reklam ya her birinde birer hafta yayınlanacak, ya da her iki hafta aynı ortamda yayınlanacaktır. Bu modeldeki stratejiler ve kazanç matrisi aşağıda verilmiştir. Kazanç matrisindeki değerler bin kişi olarak satır oyuncusunun kazancını göstermektedir.
- 1nci strateji:1 hafta radyo 1 hafta televizyonda reklam
- 2nci strateji:2 hafta radyoda reklam yayınlamak
- 3ncü strateji:2 hafta televizyonda reklam yayınlamak

İki Kişili Sıfır Toplamlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Örnek (Reklam Kampanyası)

		Sütun Oyuncusu		
		1	2	3
Satır Oyuncusu	1	1	2	4
	2	1	0	5
	3	1	1	-1

- **Kazanç matrisi incelenirse sütun oyuncusunun üstün bir stratejisinin olmadığı görülür.**
- **Satır oyuncusunun ise 1nci stratejisi 3ncü stratejisine göre üstündür, çünkü; 1nci strateji her zaman (yani sütun oyuncusu ne yaparsa yapsın) en az 3ncü stratejinin sağladığı kazancı (3ncü stratejinin sağladığı kazanca eşit veya daha fazla kazanç) sağlamaktadır (1>1, 2>1, 4>-1). Bu yüzden satır oyuncusunun 3ncü stratejisini elinde bulundurmasının hiç bir anlamı yoktur, yani bu strateji 1nci strateji tarafından alt edilmiştir. Kazanç matrisinden çıkartılır.**

İki Kişili Sıfır Toplamlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Örnek (Reklam Kampanyası)

		Sütun Oyuncusu		
		1	2	3
Satır Oyuncusu	1	1	2	4
	2	1	0	5

- **Kazanç matrisi incelenirse sütun oyuncusunun 3ncü stratejisi 1 ve 2nci stratejileri tarafından alt edilir Bu nedenle 3ncü sütunda matristen çıkartılır.**

İki Kişili Sıfır Toplamlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Örnek (Reklam Kampanyası)

		Sütun Oyuncusu	
		1	2
Satır Oyuncusu	1	1	2
	2	1	0

Kazanç matrisi incelenirse satır oyuncusunun 2nci stratejisi 1nci stratejisi tarafından alt edilir. Bu nedenle 2nci satırda matristen çıkartılır.

İki Kişili Sıfır Toplamlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Örnek (Reklam Kampanyası)

		Sütun Oyuncusu	
		1	2
Satır Oyuncusu	1	1	2

- **Kazanç matrisi incelenirse sütun oyuncusunun 2nci stratejisi 1nci stratejisi tarafından alt edilir Bu nedenle 2nci sütunda matristen çıkartılır.**

İki Kişili Sıfır Toplamlı Oyunlar

- Üstün (Dominant) Stratejiler ve Alt Etme
- Örnek (Reklam Kampanyası)

		Sütun Oyuncusu
		1
Satır Oyuncusu	1	1

- **Sonuç olarak her iki oyuncu da 1nci stratejilerini (saf stratejiler) oynamalı yani 1 hafta radyoda ve 1 hafta televizyonda reklam yayınlamalıdırlar. Bunun sonucunda satır oyuncusu sütun oyuncusundan 1000 müşteri kazanır, yani oyunun değeri 1'dir**
- **Yukarıdaki örnek problemde (1, 1) noktası bir denge noktasıdır. Bu minimaks-maksimin kriterini kullanarak**
- **Maksimin = minimaks = 1**
- **eşitliğinden de görülebilir. Üstün stratejilerin varlığı ve alt etme işlemi bizi bu denge noktasına getirmiştir.**

İki Kişili Sıfır Toplamlı Oyunlar

Kararsız oyunlar

- Denge noktası bulunmayan iki-kişili sıfır-toplamlı oyunlar kararsız oyun olarak adlandırılır. Bu bölümde denge noktası bulunmayan kararsız oyunlarda optimal stratejilerin nasıl bulunacağı incelenecektir.
- Kararsız oyunlarda oyuncular stratejilerinin olasılık dağılımını (her bir stratejinin kullanılma olasılığını ya da oranını) saptayarak bu olasılıklara göre stratejilerini kullanırlar.

			Sütun Oyuncusu			
	Olasılık		y_1	y_2	…	y_n
		Strateji	1	2	…	n
Satır Oyuncusu	x_1	1	a_{11}	a_{12}	…	a_{1n}
	x_2	2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	:	…	:
	x_m	m	a_{m1}	a_{m2}	…	a_{mn}

İki Kişili Sıfır Toplamlı Oyunlar

Kararsız oyunlar

- Satır oyuncusunun beklenen kazanç değeri:

$$v = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_i y_j = a_{11}x_1y_1 + a_{12}x_1y_2 + \cdots + a_{mn}x_m y_n$$

			Sütun Oyuncusu			
	Olasılık		y_1	y_2	…	y_n
		Strateji	1	2	…	n
Satır Oyuncusu	x_1	1	a_{11}	a_{12}	…	a_{1n}
	x_2	2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	:	…	:
	x_m	m	a_{m1}	a_{m2}	…	a_{mn}

- Satır oyuncusunun m adet stratejisi ve sütun oyuncusunun n adet stratejisinin olduğunu kabul ederek aşağıdaki tanımları yapalım;
- x_i= satır oyuncusunun i stratejisini kullanma olasılığı (oranı) i=1,2,...m
- y_j= sütun oyuncusunun j stratejisini kullanma olasılığı (oranı) j=1,2,...n

İki Kişili Sıfır Toplamlı Oyunlar

- Kararsız oyunlar

- x_i ve y_j olasılık olduğuna göre;

1. $0 \le x_i \le 1$ (i=1,2,...m) , $0 \le y_j \le 1$ (j=1,2,...,n)
2. $\sum_{i=1}^{m} x_i = \sum_{j=1}^{n} x_j$

İki Kişili Sıfır Toplamlı Oyunlar

- Kararsız oyunlar
- Satır oyuncusunun x_i ve sütun oyuncusunun y_j olasılıklarına Karma strateji adı verilir.
- Karma stratejilerin belirlenmesinde de Minimaks-Maksimin kriteri kullanılır. Satır oyuncusu Maksimin kriterine göre minimum beklenen kazancını maksimum yapan karma stratejiyi, sütun oyuncusu ise Minimaks kriterine göre maksimum beklenen kaybını minimum yapan karma stratejiyi seçer.
- Karma stratejilerle oynanacak bir oyunda *minimaks=maksimin=v* (oyunun değeri) eşitliğini sağlayan stratejiler optimum karma stratejilerdir.

İki Kişili Sıfır Toplamlı Oyunlar

- Grafik Çözüm Metodu (2x2,mx2 ve 2xn Boyutlu oyunlar)
- Oyunculardan birisinin yalnızca iki stratejisi varsa o zaman grafik metodunu kullanarak optimal KARMA STRATEJİLER bulunabilir. Eğer her oyuncunun ikiden fazla stratejisi varsa (alt edilebilecek bütün stratejiler çıkarıldıktan sonra) bu durumda ileride görülebileceği gibi oyun bir DOĞRUSAL PROGRAMLAMA modeli olarak yazılıp çözülebilir. Örneğin satır oyuncusunun iki stratejisi varsa karma stratejisi (x_1,x_2) olasılıkları ile tanımlanır ve durumda $x_2 = 1-x_1$ olasılığından tek değişken olan x_1'in optimal değerini bulmak gerekmektedir. Bunun için öncelikle rakibin (sütun oyuncusu) her bir saf stratejisine karşılık gelen kazancı x_1'in fonksiyonu olarak çizilir ve bu grafikte minimum kazancı maksimum yapan nokta yani maksimin noktası belirlenir. Sütun oyuncusu için çözüm yapılıyorsa bu sefer maksimum kaybı minimum yapan nokta yani minimaks noktası bulunur.

İki Kişili Sıfır Toplamlı Oyunlar

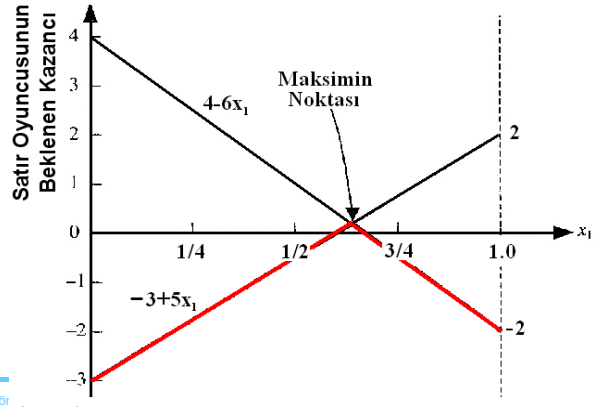
- Grafik Çözüm Metodu
- İki-kişili sıfır-toplamlı bir oyunun kazanç matrisi aşağıdaki tabloda verilmiştir. Matrise bakıldığında oyunda bir denge noktası bulunmadığı görülür. O halde her iki oyuncunun da karma strateji kullanması yani satır oyuncusunun (x_1, x_2) olasılıklarını, sütun oyuncusunun ise (y_1, y_2) olasılıklarını bulması gerekmektedir.

			Sütun Oyuncusu	
	Olasılık		y_1	$y_2=1-y_1$
		Strateji	1	2
Satır Oyuncusu	x_1	1	-2	2
	$x_2=1-x_1$	2	4	-3

İki Kişili Sıfır Toplamlı Oyunlar

Grafik Çözüm Metodu (Satır Oyuncusu)

Sütun Oyuncusunun Seçtiği Strateji	Satır Oyuncusunun Beklenen Kazancı
1	$-2x_1+4(1-x_1)=4-6x_1$
2	$2x_1-3(1-x_1)=-3+5x_1$



Maksimin noktası

$$-3+5x_1=4-6x_1 \Rightarrow x_1=7/11$$
$$x_2=1-x_1 \Rightarrow x_2=4/11$$

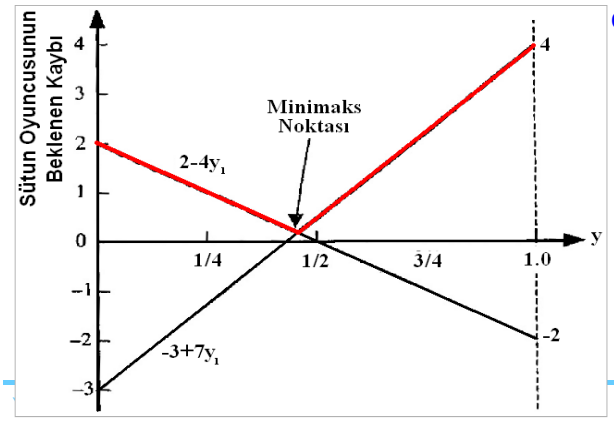
Oyunun değeri

$$v=-3+5(7/11)=2/11$$

İki Kişili Sıfır Toplamlı Oyunlar

Grafik Çözüm Metodu (Sütun Oyuncusu)

Satır Oyuncusunun Seçtiği Strateji	Sütun Oyuncusunun Beklenen Kaybı
1	$-2y_1+2(1-y_1)=2-4y_1$
2	$4y_1-3(1-y_1)=-3+7y_1$



Minimaks noktası

$$2-4y_1=-3+7y_1 \Rightarrow y_1=5/11$$
$$y_2=1-y_1 \Rightarrow y_2=6/11$$

Oyunun değeri

$$v=-3+7(5/11)=2/11$$

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile Modelleme (Satır oyuncusunun DP modeli)
- Sütun oyuncusu herhangi bir *j* stratejisini seçerse satır oyuncusunun kazancı $a_{1j}x_1+a_{2j}x_2+\ldots+a_{mj}x_m$ olur.
- Sütun oyuncusu bütün stratejileri için satır oyuncusunun beklenen kazancını (kendi kaybını) minimum seviyede tutmak isteyecektir. Yani satır oyuncusunun beklenen kazancı *v* ise;
- $v \leq \min\{a_{11}x_1+a_{21}x_2+\ldots+a_{m1}x_m,\ a_{12}x_1+a_{22}x_2+\ldots+a_{m2}x_m,\ldots\ldots,\ a_{1n}x_1+a_{2n}x_2+\ldots+a_{mn}x_m\}$
- Eşitsizlik genellenirse;

$$\sum_{i=1}^{m} a_{ij}x_i \geq v \quad (j=1,2,\cdots,n)$$

			Sütun Oyuncusu			
	Olasılık		y_1	y_2	…	y_n
		Strateji	1	2	…	*n*
Satır Oyuncusu	x_1	1	a_{11}	a_{12}	…	a_{1n}
	x_2	2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	:	…	:
	x_m	*m*	a_{m1}	a_{m2}	…	a_{mn}

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile Modelleme (Satır oyuncusunun DP modeli)
- Satır oyuncusu bu minimum kazançlar arasından maksimum olanını elde edecek yani modelin amaç fonksiyonu *maks z=v* olacaktır. Buna göre satır oyuncusunun DP modeli;

$$\begin{array}{llllll}
Maks\ Z = & v & & & & \\
S.T. & a_{11}x_1 & +a_{21}x_2 & +\cdots & +a_{m1}x_m & \geq v \\
 & a_{12}x_1 & +a_{22}x_2 & +\cdots & +a_{m2}x_m & \geq v \\
 & \vdots & \vdots & & \vdots & \vdots \\
 & a_{1n}x_1 & +a_{2n}x_2 & +\cdots & +a_{mn}x_m & \geq v \\
 & x_1 & +x_2 & +\cdots & +x_m & =1 \\
 & & & & x_i & \geq 0\ (i=1,2,\cdots,m) \\
 & & & & v & Sınırsız
\end{array}$$

			Sütun Oyuncusu			
	Olasılık		y_1	y_2	…	y_n
		Strateji	1	2	…	*n*
Satır Oyuncusu	x_1	1	a_{11}	a_{12}	…	a_{1n}
	x_2	2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	:	…	:
	x_m	*m*	a_{m1}	a_{m2}	…	a_{mn}

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile Modelleme (Sütun oyuncusunun DP modeli)
- Satır oyuncusu herhangi bir *i* stratejisini seçerse sütun oyuncusunun kaybı $a_{i1}y_1+a_{i2}y_2+\ldots+a_{in}y_n$ olur.
- Satır oyuncusu bütün stratejileri için sütun oyuncusunun beklenen kaybını (kendi kazancını) maksimum seviyede tutmak isteyecektir. Yani sütun oyuncusunun beklenen kaybıı *w* ise;
- $w \geq max\{a_{11}y_1+a_{12}y_2+\ldots+a_{1n}y_n\ ,\ \ a_{21}y_1+a_{22}y_2+\ldots+a_{2n}y_n\ ,\ldots\ldots\ ,\ a_{m1}y_1+a_{m2}y_2+\ldots+a_{mn}y_n\}$
- Eşitsizlik genellenirse;

$$\sum_{j=1}^{n} a_{ij}y_j \leq w \quad (i=1,2,\cdots,m)$$

	Olasılık		Sütun Oyuncusu			
			y_1	y_2	…	y_n
		Strateji	1	2	…	*n*
Satır Oyuncusu	x_1	1	a_{11}	a_{12}	…	a_{1n}
	x_2	2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	:	…	:
	x_m	*m*	a_{m1}	a_{m2}	…	a_{mn}

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile Modelleme (Sütun oyuncusunun DP Modeli)
- Sütun oyuncusu bu maksimum kayıplar arasından minimum olanını elde edecek yani modelin amaç fonksiyonu *Min z=w* olacaktır. Buna göre sütun uyuncusunun DP modeli;

$$\begin{array}{lllll}
Min\ Z = & w & & & \\
S.T. & a_{11}y_1 & +a_{12}y_2 & +\cdots & +a_{1n}y_n \leq w \\
 & a_{21}y_1 & +a_{22}y_2 & +\cdots & +a_{2n}y_n \leq w \\
 & \vdots & \vdots & & \vdots \quad \vdots \\
 & a_{m1}y_1 & +a_{m2}y_2 & +\cdots & +a_{mn}y_n \leq w \\
 & y_1 & +y_2 & +\cdots & +y_n = 1 \\
 & & & & y_j \geq 0\ (j=1,2,\cdots,n) \\
 & & & & w \quad Sınırsız
\end{array}$$

	Olasılık		Sütun Oyuncusu			
			y_1	y_2	…	y_n
		Strateji	1	2	…	*n*
Satır Oyuncusu	x_1	1	a_{11}	a_{12}	…	a_{1n}
	x_2	2	a_{21}	a_{22}	…	a_{2n}
	:	:	:	:	…	:
	x_m	*m*	a_{m1}	a_{m2}	…	a_{mn}

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme
- Satır ve sütun oyuncularının DP modelleri bir birinin dualidir. Dolayısıyla;
  - modellerden birisi çözülüp diğerinin çözümü gölge fiyatlardan faydalanılarak bulunur.
  - her iki model içinde oyunun değeri eşit olup *v=w* dir.
- İki kişili-sıfır toplamlı bir oyun için;

  maksimin≤oyunun değeri(v)≤minimaks
- Buna göre;
  - Maksimin≥0 ve Minimaks≥0 ise v≥0'dır. bu durumda satır oyuncusunun modeline v≥0 ve sütun oyuncusunun modeline de w≥0 kısıtları eklenerek model çözülebilir.
  - Maksimin≤0 ve Minimaks≥0 ise oyunun değeri negatif veya pozitif olabilir (sınırsızdır).
  - Maksimin≤0 ve Minimaks≤0 ise oyunun değeri negatif olacaktır (yine sınırsızdır).
- Son iki durumda sınırsız değişkenlerden kurtulmak için;
  - Kazanç matrisinin bütün elemanlarına matristeki en küçük negatif sayının mutlak değeri eklenir ve model çözülür.
  - Bu işlem optimal çözüm olasılıklarını değiştirmezken, oyunun değeri eklenen sayı kadar fazla elde edilir.

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- İki kişi arasında oynanan bir oyunda oyuncular aynı anda elleri ile taş, kağıt ya da makas işaretini gösterirler. Makas kağıda göre, kağıt taşa göre ve taş ise makasa göre üstün olup, üstün olan işareti gösteren oyuncu diğerinden 1 puan kazanır. Bu problemin oyun modelini geliştiriniz ve çözüm yöntemini açıklayınız.

- Çözüm
- Problem, iki kişili-sıfır toplamlı bir oyun problemidir. Her iki oyuncununda üç stratejisi vardır.
  - taş göstermek,
  - kağıt göstermek,
  - makas göstermek.

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

Oyunun Kazanç Matrisi

		Sütun Oyuncusu			Satır
	Strateji	Taş	Kağıt	Makas	Min.
Satır Oyuncusu	Taş	0	-1	1	-1
	Kağıt	1	0	-1	-1
	Makas	-1	1	0	-1
Sütun Maks.		1	1	1	-

- Oyunda bir denge noktası bulunmamakta ve hiç bir strateji de alt edilememektedir. Oyunun tek çözüm yolu, oyuncuların DP modelini oluşturarak oyuncuların karma stratejilerini ve oyunun değerini belirlemektir.
- Kazanç matrisinde Maksimin=-1 olduğu için kazanç matrisinin bütün elemanlarına en küçük negatif sayının mutlak değeri yani +1 eklenerek oyunun yeni kazanç matrisi bulunur.

		Sütun Oyuncusu			Satır
	Strateji	Taş	Kağıt	Makas	Min.
Satır Oyuncusu	Taş	1	0	2	0
	Kağıt	2	1	0	0
	Makas	0	2	1	0
Sütun Maks.		2	2	2	-

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli
- (x_1, x_2, x_3) olasılıkları satır oyuncusunun karma stratejileri olmak üzere sütun oyuncusunun her hareket tarzına karşılık kazancı aşağıdaki tabloda verilmiştir.

Sütun Oyuncusunun Stratejisi	Satır Oyuncusunun Beklenen Kazancı
Taş	x_1+2x_2
Kağıt	x_2+2x_3
Makas	$2x_1+x_3$

- Sütun oyuncusu satır oyuncusuna $\min\{x_1+2x_2, x_2+2x_3, 2x_1+x_3\}$ değerine eşit beklenen kazancı verecek stratejiyi seçecektir. Satır oyuncusu karma stratejilerini öyle belirlemelidir ki $\min\{x_1+2x_2, x_2+2x_3, 2x_1+x_3\}$ değerini (yani kazancının üst sınırını) mümkün olduğu kadar yüksek tutsun

		Sütun Oyuncusu			Satır
	Strateji	Taş	Kağıt	Makas	Min.
Satır Oyuncusu	Taş	1	0	2	0
	Kağıt	2	1	0	0
	Makas	0	2	1	0
Sütun Maks.		2	2	2	-

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli
- Buna göre satır oyuncusu optimal karma stratejisini belirlemek için aşağıdaki DP modelini kullanacaktır.

$$\text{Maks } z = v'$$
$$\begin{aligned} x_1 + 2x_2 - v' &\geq 0 \\ x_2 + 2x_3 - v' &\geq 0 \\ 2x_1 + x_3 - v' &\geq 0 \\ x_1 + x_2 + x_3 &= 1 \\ x_1, x_2, x_3, v' &\geq 0 \end{aligned}$$

OPTİMAL ÇÖZÜM
$(x_1,x_2,x_3)=(1/3,1/3,1/3)$, v'=1

OYUNUN DEĞERİ v=v'-1=0

		Sütun Oyuncusu			Satır
	Strateji	Taş	Kağıt	Makas	Min.
Satır Oyuncusu	Taş	1	0	2	0
	Kağıt	2	1	0	0
	Makas	0	2	1	0
Sütun Maks.		2	2	2	-

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Sütun Oyuncusunun Modeli
- (*y1, y2, y3*) olasılıkları sütun oyuncusunun karma stratejileri olmak üzere satır oyuncusuna benzer yaklaşım uygulandığında sütun oyuncusunun DP modeli aşağıdaki şekilde olacaktır.

$$\text{Min } z = w'$$
$$\begin{aligned} y_1 + 2y_3 - w' &\leq 0 \\ 2y_1 + y_2 - w' &\leq 0 \\ 2y_2 + y_3 - w' &\leq 0 \\ y_1 + y_2 + y_3 &= 1 \\ y_1, y_2, y_3, w' &\geq 0 \end{aligned}$$

OPTİMAL ÇÖZÜM
$(y_1,y_2,y_3)=(1/3,1/3,1/3)$, w'=1

OYUNUN DEĞERİ w=w'-1=0

		Sütun Oyuncusu			Satır
	Strateji	Taş	Kağıt	Makas	Min.
Satır Oyuncusu	Taş	1	0	2	0
	Kağıt	2	1	0	0
	Makas	0	2	1	0
Sütun Maks.		2	2	2	-

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli (Simpleks Yöntemi ile çözümü)
- Modelde görülebileceği gibi x1+x2+x3=1 olması özelliğinden dolayı herhangi bir x değişkenini diğerleri cinsinden ifade edebiliriz. Kolaylık olması açısından x3 değişkenini 1-x1-x2 şekline dönüştürelim. Modelimiz sırasıyla aşağıdaki forma dönüşecektir.

$$\begin{aligned} &maks\ z = v' \\ &x_1 + 2x_2 - v' \geq 0 \\ &x_2 + 2(1 - x_1 - x_2) - v' \geq 0 \\ &2x_1 + (1 - x_1 - x_2) - v' \geq 0 \\ &x_1, x_2 \geq 0 \\ &v'\ sınırsız \end{aligned}$$

$$\begin{aligned} &maks\ z = v' \\ &v' - x_1 - 2x_2 \leq 0 \\ &v' + 2x_1 + x_2 \leq 2 \\ &v' - x_1 + x_2 \leq 1 \\ &x_1, x_2 \geq 0 \\ &v'\ sınırsız \end{aligned}$$

		Sütun Oyuncusu			Satır
	Strateji	Taş	Kağıt	Makas	Min.
Satır Oyuncusu	Taş	1	0	2	0
	Kağıt	2	1	0	0
	Makas	0	2	1	0
Sütun Maks.		2	2	2	-

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli (Simpleks yöntemi ile çözümü)
- Elde edilen modeli simpleks tablosuna koyduğumuzda, aşağıdaki formu elde ederiz.

	x1	x2	v'	S1	S2	S3	STD
z	0	0	-1	0	0	0	0
S1	-1	-2	1	1	0	0	0
S2	2	1	1	0	1	0	2
S3	-1	1	1	0	0	1	1

- Simpleks algoritmasını uyguladığımızda giren değişkeni inceleyelim. Bunun için amaç fonksiyonu satırında en küçük değerli sütun seçilir. Amaç fonksiyon satırında tek negatif değerli sütun v' sütunu olduğunda bu sütun seçilir..

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli (Simpleks yöntemi ile çözümü)

	x1	x2	v'	S1	S2	S3	STD	Min.oran testi
z	0	0	-1	0	0	0	0	
S1	-1	-2	1	1	0	0	0	0/1
S2	2	1	1	0	1	0	2	2/1
S3	-1	1	1	0	0	1	1	1/1

	x1	x2	v'	S1	S2	S3	STD	Min.oran testi
z	-1	-2	0	1	0	0	0	
v'	-1	-2	1	1	0	0	0	-
S2	3	3	0	-1	1	0	2	2/3
S3	0	3	0	-1	0	1	1	1/3

Yöneylem Araştırması III

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli (Simpleks yöntemi ile çözümü)

	x1	x2	v'	S1	S2	S3	STD	Min.oran testi
z	-1	0	0	1/3	0	2/3	2/3	
v'	-1	0	1	1/3	0	2/3	2/3	-
S2	3	0	0	0	1	-1	1	1/3
x2	0	1	0	-1/3	0	1/3	1/3	-

	x1	x2	v'	S1	S2	S3	STD
z	0	0	0	1/3	1/3	1/3	1
v'	0	0	1	1/3	1/3	1/3	1
x1	1	0	0	0	1/3	-1/3	1/3
x2	0	1	0	-1/3	0	1/3	1/3

Yöneylem Araştırması III

İki Kişili Sıfır Toplamlı Oyunlar

- DP ile modelleme

Örnek (Taş-Kağıt-Makas Oyunu)

- Satır Oyuncusunun Modeli
- (*x1, x2, x3*) olasılıkları satır oyuncusunun karma stratejileri olmak üzere tablodan elde edilen bilgileri çekersek;

OPTİMAL ÇÖZÜM

v'= 1

x_i = (1/3,1/3,1/3),

OYUNUN DEĞERİ v=v'-1=0

- (*y1, y2, y3*) olasılıkları sütun oyuncusunun karma stratejileri olmak üzere tablodan elde edilen bilgileri çekersek amaç fonksiyonu satırındaki aylak değişkenlerin katsayılarının ifade ettiği durumlar ;

OPTİMAL ÇÖZÜM

y_i = S_i =(1/3,1/3,1/3), w=v=0

		Sütun Oyuncusu			*Satır Min.*
	Strateji	Taş	Kağıt	Makas	
Satır Oyuncusu	Taş	*1*	*0*	*2*	*0*
	Kağıt	*2*	*1*	*0*	*0*
	Makas	*0*	*2*	*1*	*0*
Sütun Maks.		*2*	*2*	*2*	-

İki Kişili Sıfır Toplamlı Oyunlar

- Özet
- İki kişili-sıfır toplamlı oyunların çözümü aşağıdaki şekilde özetlenebilir.

- Adım 1: Denge noktası bulunup bulunmadığını kontrol edin, denge noktası yok ise adım 2'ye gidin, varsa bulunan strateji saf stratejidir. Her iki oyuncu için bu nokta optimal karardır.

- Adım 2: Satır ve sütun oyuncusu için alt edilebilecek stratejileri kazanç matrisinden çıkartın ve adım 3 e gidin.

- Adım 3: Kazanç matrisi 2x2 lik bir matris ise grafik metodunu değilse DP metodunu kullanarak çözümü bulun.

İki Kişili Sabit Toplamlı Oyunlar

- Bu tip oyunlarda oyuncular birbirinden alışveriş yapmayıp sabit bir kaynaktan mümkün olduğu kadar fazla pay almak amacıyla rekabet ederler.

- İki kişili-sabit-toplamlı oyun, iki kişi arasında oynanan ve her bir oyuncunun herhangi bir stratejisini kullanması durumunda satır oyuncusu ile sütun oyuncusunun kazançları toplamının *c* gibi sabit bir değere eşit olduğu oyundur.

- Satır oyuncusunun kazancındaki bir birimlik artış sütun oyuncusunun kazancında bir birim azalışa neden olur. Bu oyunlarda da optimal stratejiler ve oyunun değeri iki kişili-sıfır toplamlı oyunlarda kullanılan metotlarla belirlenebilir.

İki Kişili Sabit Toplamlı Oyunlar

- Örnek (İki Televizyon Kanalının Rekabeti)
- İki televizyon kanalı, 20:00-21:00 zaman dilimi süresince 100 milyon kişilik izleyici kitlesini çekmek için rekabet halindedirler. İki kanal, bu zaman diliminde yayınlayacakları programı aynı anda duyurmak zorundadırlar. Her kanalın muhtemel seçenekleri ile her seçenek için 1nci kanalın izleyici sayısı (milyon olarak) aşağıdaki tabloda verilmiştir.

		İkinci Kanal		
		Western Filmi	Klasik Müzik	Güldürü
Birinci Kanal	Western Filmi	35	15	60
	Klasik Müzik	45	58	50
	Güldürü	38	14	70

- Örneğin, her iki kanal filmi seçerse, kazanç matrisi 35 milyon seyircinin 1nci kanalı ve 100-35=65 milyon seyircinin 2nci kanalı tercih edeceğini göstermektedir.

İki Kişili Sabit Toplamlı Oyunlar

- Örnek (İki Televizyon Kanalının Rekabeti)
- Bu oyunda c=100 dür.

Satır oyuncusu için maksimin=maks(15,45,14)=45 (klasik müzik)
sütun oyuncusu için minimaks=min(45,58,70)=45 (western filmi)

Maksimin=Minimaks=45 (oyunda denge noktası var)

- Oyunun çözümü

1nci kanal için oyunun değeri 45 milyon seyirci, optimal saf strateji Klasik müzik,
2nci kanal için oyunun değeri 100-45=55 milyon seyirci, optimal saf strateji Western filmidir.

İki Kişili Sabit Toplamlı Olmayan Oyunlar

- Örnek (Mahkumun Açmazı)
- Firar eden ve bir soyguna karışan iki mahkum yeniden yakalanmış ve yeni suçlarından yargılanmayı beklemektedirler. Suçlu olmalarına rağmen, savcı onları mahkum ettirmek için yeterli delil olmadığını düşünmektedir. Bu yüzden savcı mahkumları suçu itiraf etmeye ve diğeri aleyhinde tanıklık yapmaya zorlamak için, her mahkuma şunu söyler:
- "Eğer sadece biriniz itiraf eder ve arkadaşının aleyhine tanıklık yaparsa, inkar eden kesinlikle 20 yıl hapis cezasına mahkum edilirken itiraf eden serbest kalır Her ikiniz de itiraf ederseniz, 5'er yıl hapse mahkum olursunuz. Hiç biriniz itiraf etmezseniz, her ikiniz de önceki suçunuzun devamı olarak 1'er yıl hapis cezası alırsınız."
- Mahkumlar mahkeme önüne çıkıncaya kadar kesinlikle birbirleri ile görüşemeyeceklerdir Buna göre mahkumlar ne yapmalıdır?

İki Kişili Sabit Toplamlı Olmayan Oyunlar

- Örnek (Mahkumun Açmazı)
- Aşağıdaki tablo oyunun kazanç matrisini göstermektedir. Bu matriste parantez içindeki ilk rakam satır oyuncusunun, ikinci rakamsa sütun oyuncusunun kazancını gösterir.

		İkinci Mahkum	
		İtiraf	İnkar
Birinci Mahkum	İtiraf	(-5,-5)	(0,-20)
	İnkar	(-20,0)	(-1,-1)

- Matristeki kazanç toplamları -2, -10 ve -20 olarak değiştiğinden problem iki kişili sabit toplamlı olmayan bir oyun problemidir.
- Her iki oyuncu içinde itiraf stratejisi inkar stratejisine göre daha üstündür Dolayısıyla her iki oyuncu içinde inkar stratejileri alt edilir
- İki-kişili sıfır-toplamlı oyunlarda olduğu gibi, oyuncuların tek taraflı olarak strateji değiştirdiklerinde daha iyi bir fayda elde edemeyecekleri stratejilere denge noktası denir. Bu sebeple (-5, -5) bir denge noktasıdır.

İki Kişili Sabit Toplamlı Olmayan Oyunlar

- Örnek 2 (Reklam Kampanyası)
- Bir şehirdeki iki rakip lokantanın (a ve b) yıllık satış gelirleri toplamı 240 birimdir. Lokantalar gelecek yıl için reklam bütçelerini planlamaktadırlar. Her iki lokanta da reklam için 6 veya 10 birim para ayırabilecektir. Eğer birisi reklam için diğerinden daha fazla harcarsa çok harcayan lokanta toplam satış gelirinin 190 birimini elde edecektir. Her ikisi de aynı miktarda harcarsa karı eşit olarak paylaşacaklardır. Her bir birim satış 0.1 birim kar bırakmaktadır.
- Her iki lokanta da net karını (satış geliri-reklam harcamaları) maksimum yapmak istiyorsa bu oyunun denge noktası ne olur?

İki Kişili Sabit Toplamlı Olmayan Oyunlar

- Örnek 2 (Reklam Kampanyası)
- Bu oyunda kâra göre oluşturulan kazanç matrisi aşağıda verilmiştir.

		B Lokantası	
		10	6
A Lokantası	10	(2,2)	(9,-1)
	6	(-1,9)	(6,6)

- Matristeki kazanç toplamları 4, 8 ve 12 olarak değiştiğinden problem iki kişili sabit toplamlı olmayan bir oyun problemidir.
- Her iki oyuncu içinde 10 stratejisi 6 ya göre daha üstündür Dolayısıyla her iki oyuncu içinde 6 stratejileri alt edilir
- (2, 2) noktası denge noktasını gösterir.

BÖLÜM III: Dinamik Programlama

- DİNAMİK PROGRAMLAMAYA GİRİŞ
- DİNAMİK PROGRAMLAMANIN ESASLARI
  - TEMEL ÖZELLİKLERİ
  - MATEMATİKSEL FORMÜLASYON VE HESAPLAMA METODU
- EN KISA YOL PROBLEMLERİ
- SIRT ÇANTASI PROBLEMLERİ
- ÜRETİM-SATIN ALMA PROBLEMLERİ

Dinamik Programlamaya Giriş

- 1951 yılında Bellman tarafından geliştirilmiştir.
- Dinamik programlama, problemleri safhalara ayırarak çözen ve pek çok alanda kullanılabilen genel bir yaklaşımdır.
- Dinamik programlama, birbirine bağlı bir dizi karar vermeyi gerektiren ve problem parametrelerinin safhadan safhaya değiştiği çok safhalı veya çok dönemli problemlerin çözümünde kullanılabilecek en uygun yaklaşımdır.
- Dinamik programlama problemlerinin standart bir matematiksel formülasyonu yoktur ve kullanılan formüller problemin özelliğine göre geliştirilir.

Dinamik Programlamanın Esasları

- Genel olarak dinamik programlama yaklaşımı aşağıdaki işlemleri kapsar.
  - Ayrıştırma: Ana problemi, her biri bir karar gerektiren alt problemlere yani safhalara ayırma işlemidir.
  - Safhaların çözümü: Her safhada, belirlenen amaca göre alternatiflerden en iyisine karar verme işlemidir.
  - Birleştirme: Her safhada elde edilen sonuçları birleştirerek ana problemin optimal çözümünün bulunması işlemidir.
- Problemin parametrelerinin (yani durumların ve karar değişkenlerinin) veriliş biçimine göre dinamik programlama problemleri iki grupta düşünülebilir;
  - Deterministik dinamik programlama problemleri
  - Olasılıklı dinamik programlama problemleri

Dinamik Programlamanın Esasları

- Temel özellikler
- Dinamik programlama problemlerini karakterize eden ve dinamik programlama formülasyonunda bulunan temel özellikler aşağıdaki gibi açıklanabilir.
  - Safha: Her problem öncelikle her birisi bir karar gerektiren alt problemlere yani safhalara ayrılır. Safha, bir zaman dilimine veya karar verilmesi gereken bir konuma karşılık gelir.
  - Durum: Her safhada safhanın başlangıç koşullarını gösterir.
  - Karar (hareket tarzı): Herhangi bir safhada, mümkün olan her durum için uygulanabilecek çeşitli kararlar söz konusudur. Karar, herhangi bir safhada bulunulan durumu bir sonraki safhanın başlangıç durumuna dönüştürür.

Dinamik Programlamanın Esasları

- Temel özellikler
  - Getiri (veya maliyet): Dinamik programlama problemlerinde herhangi bir safhadaki getiri, safha getirisi ve toplam getiri olarak değerlendirilebilir.
    - Safha getirisi (veya maliyeti): Herhangi bir safhanın getirisi veya maliyeti, bulunulan duruma ve uygulanan hareket tarzına göre kazanılan veya kaybedilen net değerdir (kar, mesafe, süre, zaiyat, para, vs). Getiri veya maliyet fonksiyonu, doğrusal programlamadaki amaç fonksiyonuna eşdeğer bir performans ölçüsüdür.
    - Toplam getiri veya maliyet: Herhangi bir safhadaki (*n*nci safha) toplam getiri veya maliyet, o safhanın getirisi veya maliyeti ile bir sonraki safhanın (*n+1*'nci safha) toplam getirisi veya maliyetinin toplamından oluşan kümülatif (yani birikimli) bir değerdir.
  - Ardışık ilişki: Safhalar arasında kararların, durumların ve maliyet veya getiri fonksiyonlarının birbiriyle ilişkisini sağlayan mekanizmadır. Ardışık ilişki, mevcut safhadaki maliyet veya getiri fonksiyonuna bir sonraki safhanın katkısını belirler.

Dinamik Programlamanın Esasları

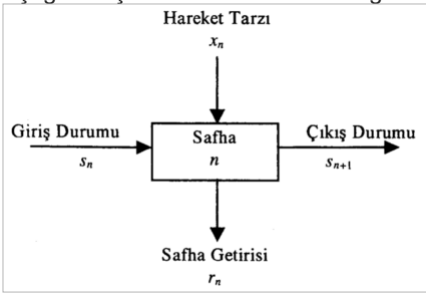
- Temel özellikler
  - Optimallik prensibi :Sistemin herhangi bir safhadaki mevcut durumu biliniyor ise, kalan safhalarda uygulanacak olan optimal hareket tarzı, önceki safhalarda uygulanmış olan hareket tarzlarından bağımsızdır. Markov özelliği olarak bilinen bu özelliğe sahip olmayan problemler dinamik programlama problemi olarak modellenemezler.

Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- Dinamik programlama, problemin ilk veya en son safhası için optimal hareket tarzını bularak çözüm sürecine başlar. Ardışık ilişkiyi kullanarak ileriye veya geriye doğru safha safha ilerler ve bitiş veya başlangıç safhasının optimal çözümünü bulur.
- Burada geriye doğru hesaplama üzerinde durulacaktır.
- N :safha sayısı
- n :Mevcut safha ($n=1, 2, ..., N$)
- S_n :n nci Safhada bulunulan durum
- X_n :n nci Safhanın karar değişkeni (yani hareket tarzı)
- X_n^* :$X_n$in optimal değeri (optimal hareket tarzı)
- r_n :n nci Safhanın getirisi(veya maliyeti)
- f_n :n nci Safhadaki toplam getiri veya maliyet

Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- Dinamik programlama formülasyonunun unsurları arasındaki ilişki aşağıdaki şekilde sembolik olarak gösterilmiştir



- Getiri veya maliyet fonksiyonları (r_n ve f_n), Aslında *n*nci safhada bulunulan duruma (S_n) ve uygulanan hareket tarzına (X_n) bağlı fonksiyonlardır. Bu yüzden safha getirisi veya maliyeti aşağıdaki şekilde gösterilebilir ve bu ifade *n*nci safhadaki herhangi bir durum-karar bileşimine (S_n,X_n) karşılık gelen getiriyi veya maliyeti gösterir.

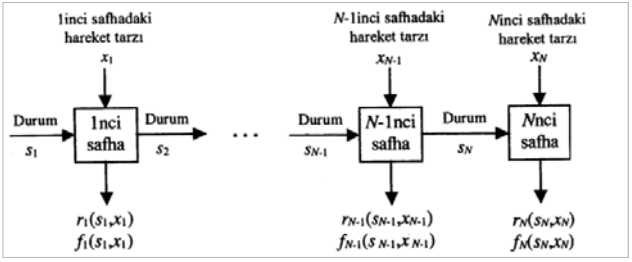
$$r_n(s_n,x_n)$$

Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- Problemin safhalarını birbirine bağlayabilmek için sadece *n*nci Safhanın getirisini veya maliyetini değil, *n+1*nci safhadaki toplam getiriyi veya maliyeti de temsil eden birikimli bir getiri veya maliyet fonksiyonuna ihtiyaç vardır. Bu fonksiyon yukarıda tanımlanan toplam getiri veya maliyet fonksiyonudur ve

$$f_n(s_n,x_n)$$

şeklinde gösterilir. Aşağıdaki şekilde *n* safhalı bir problemin safhaları, durumları ve getiri veya maliyet fonksiyonları arasındaki ilişki gösterilmektedir.



Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- *n*nci safhadaki toplam getiri veya maliyet fonksiyonu $f_n(S_n,X_n)$, *n+1*nci safhadaki toplam getiri fonksiyonu ($f_{n+1}(S_{n+1},X_{n+1})$) kullanılarak ifade edilir.
- Dinamik programlamadaki ardışık getiri veya maliyet fonksiyonunun genel yapısı;

$$f_n(s_n,x_n) = r_n(s_n,x_n) + f_{n+1}(s_{n+1},x_{n+1})$$

- Diyelim ki, $f_n^*(S_n,X_n)$, S_n durumu ve X_n hareket tarzı için elde edilecek optimal toplam getiri veya maliyet olsun. Buna göre yukarda verilen denklemdeki ardışık ilişki aşağıdaki şekle dönüşür;

$$f_n(s_n,x_n) = r_n(s_n,x_n) + f^*_{n+1}(s_{n+1},x_{n+1})$$

- Toplam getiri veya maliyet fonksiyonu yukarıda olduğu gibi toplama (+) değil de çarpma (*) ilişkisine dayalı ise, bu ilişki aşağıdaki şekilde yazılabilir;

$$f_n(s_n,x_n) = r_n(s_n,x_n) * f^*_{n+1}(s_{n+1},x_{n+1})$$

Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- Herhangi bir safhada (*n*nci safha) bulunulan başlangıç durumu (S_n) ve uygulanan hareket tarzına (X_n) göre o safhanın çıkış durumu, yani *n+1*nci safhanın başlangıç durumu (S_{n+1}) belirlenir.
- O halde dinamik programlama probleminin safhaları ve durumları arasındaki geçişi tanımlayan bir fonksiyona ihtiyaç vardır. Bu fonksiyona Safha Dönüşüm Fonksiyonu ($t(S_n,X_n)$) adı verilir ve herhangi bir safhanın girdisini (başlangıç durumu) çıktıya (çıkış durumu) dönüştürür. Bu çıktı ise bir sonraki safhanın girdisini oluşturur. Aradaki bu ilişki aşağıdaki gibi yazılabilir:

$$s_{n+1} = t(s_n,x_n)$$

Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- Ele alınacak dinamik programlama problemleri çözülürken her safha için aşağıdaki tablo formatı kullanılacaktır.

x_n / s_n	$f_n(s_n, x_n)$	$f^*_n(s_n)$	x^*_n

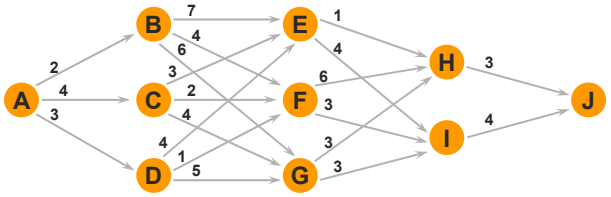
- n nci safhanın çözüm tablosunda her durum (S_n) için birer satır, her hareket tarzı (X_n) için birer sütun oluşturulacak ve tabloda S_n satırı ile X_n sütununa karşılık gelen yere Toplam getiri veya Maliyet fonksiyonunun değeri $f_n(S_n,X_n)$ yazılacaktır. Ayrıca tablonun en sağındaki iki sütun ise sırasıyla, her durum için optimal getiriyi veya maliyeti, yani $f_n^*(S_n)$ değerini ve optimal hareket tarzını (X_n^*) gösterecektir.

Dinamik Programlamanın Esasları

- Matematiksel formülasyon ve hesaplama metodu
- Özet olarak, herhangi bir dinamik programlama problemi çözülürken aşağıdaki sorular cevaplanmalıdır.
  - Problemin safhaları nelerdir?
  - Her safhada sistemin bulunabileceği durumlar nelerdir?
  - Uygulanacak hareket tarzları (karar değişkenleri) nelerdir?
  - Safha dönüşüm fonksiyonunun yapısı nasıldır?
  - Toplam getiri fonksiyonunun yapısı ve ardışık ilişki nasıldır?

En Kısa Yol Problemleri

- Posta arabası problemi
- Aşağıdaki şebekede düğümler şehirleri, oklar ise güzergahları göstermektedir. Başlangıçta A şehrinde bulunan bir posta arabasının en kısa mesafeyle J şehrine ulaşabilmesi için hangi yolu takip etmesi gerekir.



En Kısa Yol Problemleri

- Posta arabası problemi (Formülasyon)
- Karar değişkenleri X_n (n=1, 2, 3, 4) n safhasında (n nci Posta arabası yolculuğunda) varış noktalarını göstersin. Buna göre seçilecek güzergah $A \rightarrow X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$ ve $X_4 = J$ olacaktır.
- Ardışık ilişki: $f_n(S_n, X_n)$, S_n durumunda bulunan posta arabasının, n safhasına başlamaya hazır olduğu ve gideceği yer olarak X_n'i seçmesi halinde, kalan safhalardaki bütün yolların en kısası olsun

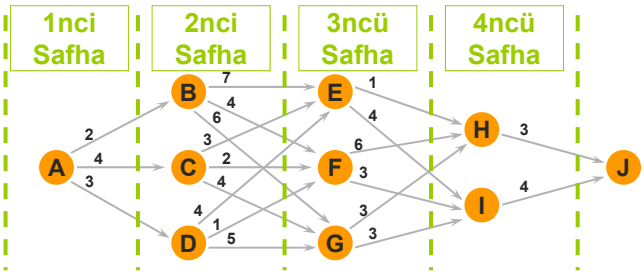
$$f_n(s_n, x_n) = c_{s_n x_n} + f^*_{n+1}(x_n)$$

- Burada C_{SnXn} S_n durumunda X_n hareket tarzının seçilmesi sonucu ilgili safhada katedilecek mesafedir.
- S_n ve n için X^*_n, $f_n(S_n, X_n)$'i minimize eden herhangi bir x_n değerini göstersin. $f^*_n(S_n)$ ise $f_n(S_n, X_n)$'nin minimum değeri olsun.

$$f^*_n(s_n) = \min_{x_n} f_n(s_n, x_n) = f_n(s_n, x^*_n)$$

En Kısa Yol Problemleri

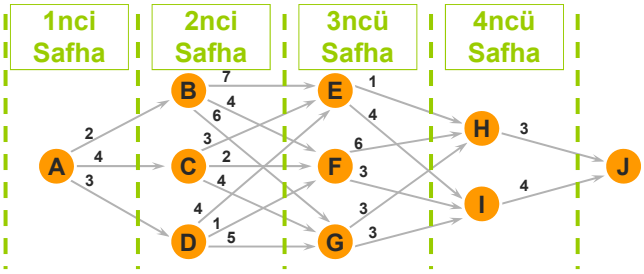
- Posta arabası problemi (Formülasyon)



- Problem 4 safhadan oluşmaktadır (*n=1, 2, 3, 4*).
- Her safhanın başlangıç düğümleri ilgili safhanın durumlarını ifade etmektedir. Örneğin n=3 İçin E, F ve G 3 ncü safhanın durumlarıdır
- 3ncü safhada bulunan E durumu için 2 hareket tarzı söz konusudur İlk hareket tarzı E→H yolunu kullanarak, ikinci hareket tarzı ise E→I yolunu kullanarak 3ncü safhayı geçmektir.
- Her bir safhada amaç; safhanın her bir durumu için son düğüme en kısa mesafeyle gidilmesini sağlayan hareket tarzının belirlenmesidir.

En Kısa Yol Problemleri

- Posta arabası problemi (Çözüm)
- n=4 için

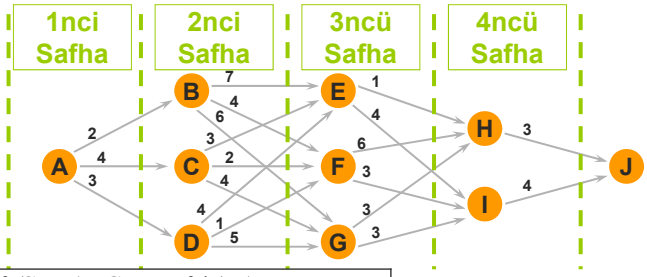


n=4; $f_4(S_4,X_4)=C_{S4X4}$			
S_4 \ X_4	$f_4(S_4,X_4)$	$f_4^*(S_4)$	X_4^*
	J		
H	3	3	J
I	4	4	J

En Kısa Yol Problemleri

Posta arabası problemi (Çözüm)

- n=3 için



$n=3;\ f_3(S_3,X_3)=C_{S3X3}+f_4^*(X_3)$

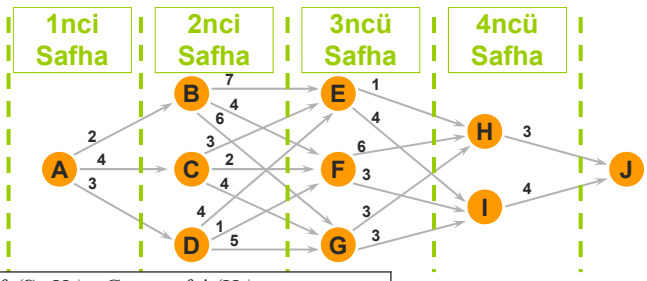
S_3 \ X_3	$f_3(S_3,X_3)$ H	$f_3(S_3,X_3)$ I	$f_3^*(S_3)$	X_3^*
E	4	8	4	H
F	9	7	7	I
G	6	7	6	H

Örneğin S_3=E için
X_3=H:$f_3(E,H)=C_{EH}+f_4^*(H)=1+3=4^*$
X_3=I:$f_3(E,I)=C_{EI}+f_4^*(I)=4+4=8$

En Kısa Yol Problemleri

Posta arabası problemi (Çözüm)

- n=2 için



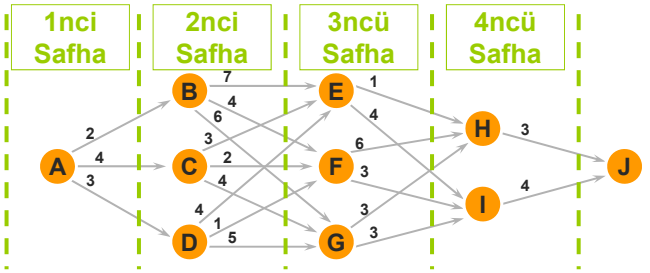
$n=2;\ f_2(S_2,X_2)=C_{S2X2}+f_3^*(X_2)$

S_2 \ X_2	$f_2(S_2,X_2)$ E	$f_2(S_2,X_2)$ F	$f_2(S_2,X_2)$ G	$f_2^*(S_2)$	X_2^*
B	11	11	12	11	E/F
C	7	9	10	7	E
D	8	8	11	8	E/F

Örneğin S_2=C için
X_2=E:$f_2(C,E)=C_{CE}+f_3^*(E)=3+4=7^*$
X_2=F:$f_2(C,F)=C_{CF}+f_3^*(F)=2+7=9$
X_2=G:$f_2(C,G)=C_{CG}+f_3^*(G)=4+6=10$

En Kısa Yol Problemleri

- Posta arabası problemi (Çözüm)
- n=1 için

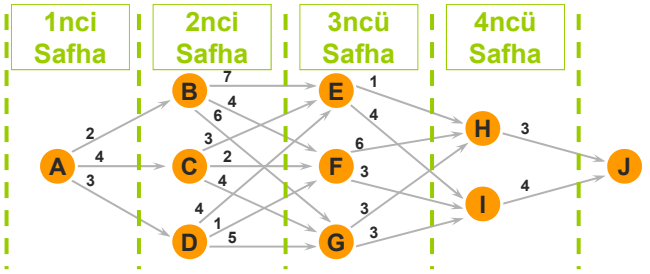


$n=1;\ f_1(S_1,X_1)=C_{S1X1}+f_2^*(X_1)$					
S_1 \ X_1	$f_1(S_1,X_1)$			$f_1^*(S_1)$	X_1^*
	B	C	D		
A	13	11	11	11	C/D

Örneğin S_1=A
X_1=B:$f_1(A,B)=C_{AB}+f_2^*(B)=2+11=13$
X_1=C:$f_1(A,C)=C_{AC}+f_2^*(C)=4+7=11^*$
X_1=D:$f_1(A,D)=C_{AD}+f_2^*(D)=3+8=11^*$

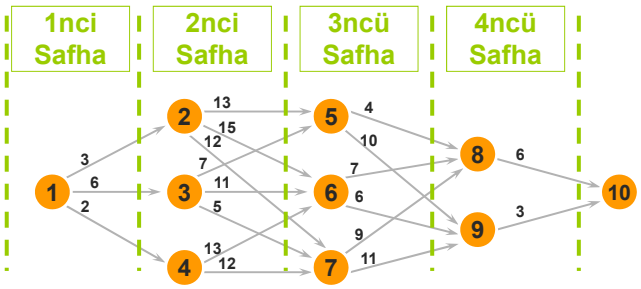
En Kısa Yol Problemleri

- Posta arabası problemi (Çözüm)
- Optimal çözümün bulunması
- *n=1* Tablosuna göre, posta arabası ilk olarak C veya D durumuna gitmelidir. $X_1^*=C$ 'yi seçerse; *n=2* için $X_2^*=E$ *n=3* için $X_3^*=H$ ve *n=4* için $X_4^*=J$ bulunur. Öyleyse optimal güzergahlardan birisi A→C→E→H→J olarak bulunur. $X_1^*=D$ seçilmesi durumunda diğer iki optimal güzergah A→D→E→H→J ve A→D→F→I→J'dir. Hepsinin toplam mesafesi $f_1^*(A)=11$'dir.



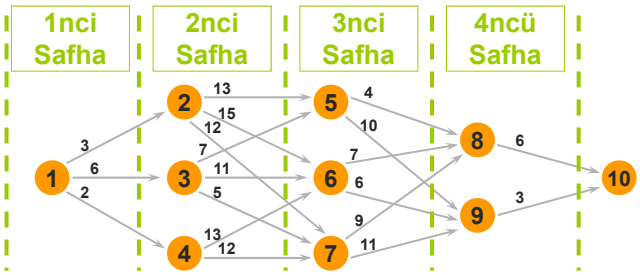
En Kısa Yol Problemleri

- Örnek 2
- Aşağıdaki şebekede oklar üzerindeki rakamlar saat biriminde iki düğüm arasındaki mesafeyi göstermektedir. 1 düğümünden 10 düğümüne giden en kısa süreli yolu bulunuz.



En Kısa Yol Problemleri

- Örnek 2

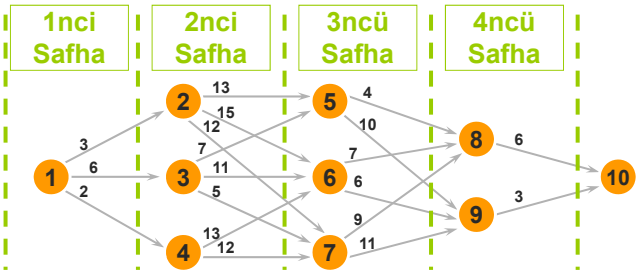


$n=4;\ f_4(S_4,X_4)=C_{S4X4}$			
S_4 \ X_4	$f_4(S_4,X_4)$ 10	$f_4^*(S_4)$	X_4^*
8	6	6	10
9	3	3	10

- $f_4(8,10)=C_{810}=6$
- $f_4(9,10)=C_{910}=3$

En Kısa Yol Problemleri

- Örnek 2

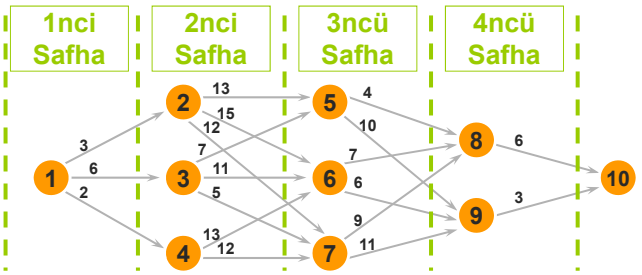


$n=3;\ f_3(S_3,X_3)=C_{S3X3}+f_4^*(X_3)$				
S_3 \ X_3	$f_3(S_3,X_3)$		$f_3^*(S_3)$	X_3^*
	8	9		
5	10	13	10	8
6	13	9	9	9
7	15	14	14	9

- f3(5,8)=C58+f4*(8)=4+6=10
  f3(5,9)=C59+f4*(9)=10+3=13
- f3(6,8)=C68+f4*(8)=7+6=13
  f3(6,9)=C69+f4*(9)=6+3=9
- f3(7,8)=C78+f4*(8)=9+6=15
  f3(7,9)=C79+f4*(9)=11+3=14

En Kısa Yol Problemleri

- Örnek 2

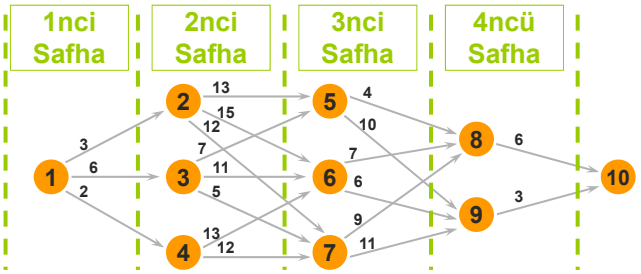


$n=2;\ f_2(S_2,X_2)=C_{S2X2}+f_3^*(X_2)$					
S_2 \ X_2	$f_2(S_2,X_2)$			$f_2^*(S_2)$	X_2^*
	5	6	7		
2	23	24	26	23	5
3	17	20	19	17	5
4	-	22	26	22	6

- f2(2,5)=C25+f3*(5)=13+10=23
  f2(2,6)=C26+f3*(6)=15+9=24
  f2(2,7)=C27+f3*(7)=12+14=26
- f2(3,5)=C35+f3*(5)=7+10=17
  f2(3,6)=C36+f3*(6)=11+9=20
  f2(3,7)=C37+f3*(7)=5+14=19
- f2(4,6)=C46+f3*(6)=13+9=22
  f2(4,7)=C47+f3*(7)=12+14=26

En Kısa Yol Problemleri

Örnek 2



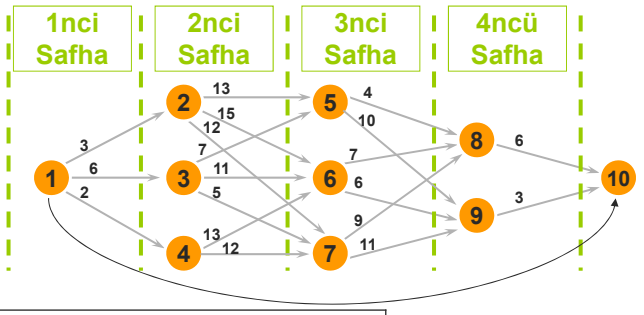
$n=1;\ f_1(S_1,X_1)=C_{S1X1}+f_2^*(X_1)$					
S_1 \ X_1	$f_1(S_1,X_1)$			$f_1^*(S_1)$	X_1^*
	2	3	4		
1	26	23	24	23	3

❋ f1(1,2)=C12+f2*(2)=3+23=26
f1(1,3)=C13+f2*(3)=6+17=23
f1(1,4)=C14+f2*(4)=2+22=24

$f_1^*(S_1)=23$ (En kısa süre) ve yol:1→3→5→8→10

En Kısa Yol Problemleri

Özel Durumlar



$n=3;\ f_3(S_3,X_3)=C_{S3X3}+f_4^*(X_3)$				
S_3 \ X_3	$f_3(S_3,X_3)$		$f_3^*(S_3)$	X_3^*
	8	9		
5	∞	13	13	9
6	13	9	9	9
7	15	14	14	9

Sırt Çantası Problemleri

- Bir sırt çantasına yüklenecek çeşitli tipte malzemeler olduğunu kabul edelim. Her bir malzemenin belirli bir ağırlığı veya hacmi ve aynı zamanda sağlayacağı faydasını gösteren bir değeri vardır.
- Öbür yandan sırt çantasının da bir ağırlık veya hacim kapasitesi vardır.
- Amaç çantaya konan malzemelerden en fazla toplam faydayı elde edebilmek için, çantanın kapasitesini aşmamak koşuluyla, hangi malzemeden ne kadar yükleneceğine karar vermektir.
- Sırt çantası problemi yaklaşımını çeşitli alanlardaki problemlere (özellikle dağıtım problemlerine) uygulamak mümkündür.

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)
- Gelecek yıl için sermaye artırımı ile ilgili projelerini planlayan bir şirket, bu projeler için toplam 10 birim bütçe ayırmıştır. Şirketin başlatmayı düşündüğü 4 proje olup, her projenin maliyeti ile sağlayacağı gelirinin net bugünkü değeri (NBD) aşağıdaki tabloda verilmiştir.
- Projelerden elde edilecek toplam net bugünkü değeri maksimum yapmak için hangi projelerin seçilmesi gerekir.

Proje	Maliyet (Birim)	NBD (Birim)
1	3	3
2	3	6
3	7	10
4	5	5

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)
- Bu problemin 0-1 tamsayılı DP modeli aşağıda verilmiştir.

$$
\begin{array}{lllllll}
Maks\ Z = & 3x_1 & +6x_2 & +10x_3 & +5x_4 & \\
S.T. & 3x_1 & +3x_2 & +7x_3 & +5x_4 & \leq 10 \\
 & x_1, & x_2, & x_3, & x_4 & 0\ veya\ 1
\end{array}
$$

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)
- Formülasyon

- Safhalar: Bu problemde, her bir projenin seçilip seçilmeyeceğine karar verileceğinden projeler problemin safhaları olacaktır. Yani, *n*nci safha=*n*nci proje (*n=1, 2, 3, 4*).
- Durumlar:Bir safhadan diğerine geçerken değişecek olan, eldeki sermaye miktarıdır. O halde S_n=*n*nci safhada eldeki sermaye miktarı.
- Karar değişkenleri (Hareket tarzları):*n*nci safhadaki hareket tarzı *n*nci projeyi seçmek ya da seçmemek olacaktır. Yani her safhada X_n*=0 veya1*

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)
- Formülasyon
- Safha dönüşüm fonksiyonu: C_n, $n$nci projenin maliyetini göstersin. Buna göre safha dönüşüm fonksiyonu $S_{n+1}=t(S_n,X_n)=S_n-C_nX_n$
- Ardışık ilişki: P_n, $n$nci projeden elde edilen net şimdiki değeri ifade ederse, safha getiri fonksiyonunu;

$$r_n(s_n,x_n) = p_nx_n$$

Toplam getiri fonksiyonu;

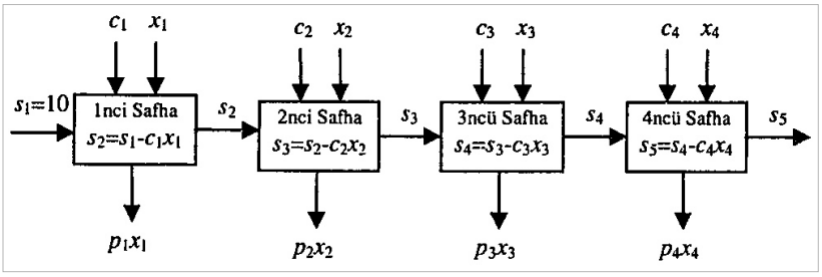
$$f_n(s_n,x_n) = p_nx_n + f^*_{n+1}(s_n-c_nx_n)$$

ve $n$nci safhada S_n durumundaki en iyi toplam getiri;

$$f^*_n(s_n) = \underset{x_n=0 \text{ veya } 1}{\text{maks}} \{p_nx_n + f^*_{n+1}(s_n-c_nx_n)\}$$

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)
- Formülasyon
- Problemin unsurları arasındaki ilişki aşağıdaki şekilde gösterilmiştir.



Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)

ÇÖZÜM (n=4)

Durumların Belirlenmesi	Önceki Projeler			Gerekli Sermaye	Kalan Sermaye
	1	2	3		
	0	1	1	10	0
	0	0	1	7	3
	1	1	0	6	4
	1	0	0	3	7
	0	0	0	0	10

Proje	Maliyet (Birim)	NBD (Birim)
1	3	3
2	3	6
3	7	10
4	5	5

n=4; $f_4(S_4,X_4)=5X_4$
(X_4=0 veya 1)

S_4 \ X_4	$f_4(S_4,X_4)$ 0	$f_4(S_4,X_4)$ 1	$f_4^*(S_4)$	X_4^*
0	0	-	0	0
3	0	-	0	0
4	0	-	0	0
7	0	5	5	1
10	0	5	5	1

- f4(0,0)=5(0)=0
  f4(0,1)=sermaye yetmez
- f4(3,0)=5(0)=0
  f4(3,1)=sermaye yetmez
- f4(4,0)=5(0)=0
  f4(4,1)=sermaye yetmez
- f4(7,0)=5(0)=0
  f4(7,1)=5(1)=5
- f4(10,0)=5(0)=0
  f4(10,1)=5(1)=5

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)

ÇÖZÜM (n=3)

Durumların Belirlenmesi	Önceki Projeler		Gerekli Sermaye	Kalan Sermaye
	1	2		
	1	1	6	4
	1	0	3	7
	0	0	0	10

Proje	Maliyet (Birim)	NBD (Birim)
1	3	3
2	3	6
3	7	10
4	5	5

n=3 $f_3(S_3,X_3)=10X_3+f_4^*(S_3-7X_3)$
(X_3=0 veya 1)

S_3 \ X_3	$f_3(S_3,X_3)$ 0	$f_3(S_3,X_3)$ 1	$f_3^*(S_3)$	X_3^*
4	0	-	0	0
7	5	10	10	1
10	5	10	10	1

- f3(4,0)=10(0)+f4*(4-7(0))=0+f4*(4)=0
  f3(4,1)=sermaye yetmez
- f3(7,0)=10(0)+f4*(7-7(0))=0+f4*(7)=5
  f3(7,1)=10(1)+f4*(7-7(1))=10+f4*(0)=10
- f3(10,0)=10(0)+f4*(10-7(0))=0+f4*(7)=5
  f3(10,1)=10(1)+f4*(10-7(0))=10+f4*(3)=10

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)

ÇÖZÜM (n=2)

Durumların Belirlenmesi	Önceki Projeler	Gerekli Sermaye	Kalan Sermaye
	1		
	1	3	7
	0	0	10

Proje	Maliyet (Birim)	NBD (Birim)
1	3	3
2	3	6
3	7	10
4	5	5

n=2; $f_2(S_2,X_2)=6X_2+f_3^*(S_2-3X_2)$
$(X_2=0$ *veya* $1)$

S_2 \ X_2	$f_2(S_2,X_2)$ 0	$f_2(S_2,X_2)$ 1	$f_2^*(S_2)$	X_2^*
7	10	6	10	0
10	10	16	16	1

- f2(7,0)=6(0)+f3*(7-3(0))=0+f3*(7)=10
  f2(7,1)=6(1)+f3*(7-3(1))=6+f3*(4)=6
- f2(10,0)=6(0)+f3*(10-3(0))=0+f3*(10)=10
  f2(10,1)=6(1)+f3*(10-3(1))=6+f3*(7)=16

Sırt Çantası Problemleri

- Örnek (Sermaye Bütçeleme Problemi)

ÇÖZÜM (n=1)

n=1; $f_1(S_1,X_1)=3X_1+f_2^*(S_1-3X_1)$
$(X_1=0$ *veya* $1)$

S_1 \ X_1	$f_1(S_1,X_1)$ 0	$f_1(S_1,X_1)$ 1	$f_1^*(S_1)$	X_1^*
10	16	13	16	0

- f1(10,0)=3(0)+f2*(10-3(0))=0+f2*(10)=16
  f1(10,1)=3(1)+f2*(10-3(1))=3+f2*(7)=13

- OPTİMAL ÇÖZÜM;
- $S_1=10$ için $X_1^*=0$, $S_2=S_1-3X_1^*=10-3(0)=10$
- $S_2=10$ için $X_2^*=1$, $S_3=S_2-3X_2^*=10-3(1)=7$
- $S_3=7$ için $X_3^*=1$, $S_4=S_3-7X_3^*=7-7(1)=0$
- $S_4=0$ için $X_4^*=0$

- Bu çözüme *(0,1,1,0)* göre *2* ve *3*ncü projeler seçilecek ve elde edilecek en yüksek toplam net şimdiki değer *f1*(10)=16* birim olacaktır.

Sırt Çantası Problemleri

- Örnek (Havayolu Taşımacılığı)
- Bir havayolu şirketi konteynırlarla üç cins malzeme taşıyacaktır. Şirketin elindeki kargo uçağının ağırlık kapasitesi 7 tondur. Malzemenin cinsine göre bir konteynırın ağırlığı ile konteynır başına elde edilecek kar ve taşınacak konteynır miktarı aşağıdaki tabloda verilmiştir.
- Şirketin amacı en fazla karı elde etmektir. Buna göre hangi tip kargodan kaç konteynır taşınacağını bulunuz

Kargo	Ağırlık (ton/Konteynır)	Kar (birim/konteynır)	Miktar (Konteynır)
Termos	1	60	5
Televizyon	2	90	3
Portatif Çadır	3	150	3

Sırt Çantası Problemleri

- Örnek (Havayolu Taşımacılığı)
- Bu problemin tamsayılı DP modeli aşağıda verilmiştir.

$$
\begin{array}{llllll}
Maks\ Z = & 60x_1 & +90x_2 & +150x_3 & & \\
S.T. & x_1 & +2x_2 & +3x_3 & \leq 7 \\
 & x_1 & & & \leq 5 \\
 & & x_2 & & \leq 3 \\
 & & & x_3 & \leq 3 \\
 & x_1, & x_2, & x_3 & \geq 0 \text{ ve tamsayı}
\end{array}
$$

Kargo	Ağırlık	Kar	Miktar
Termos	1	60	5
Televizyon	2	90	3
Portatif Çadır	3	150	3

Sırt Çantası Problemleri

- Örnek (Havayolu Taşımacılığı)
- Formülasyon

- Safhalar: *n*nci safha= *n*nci tip kargodur (*n= 1, 2, 3*).
- Durumlar: *nnci* safhada eldeki boş uçak kapasitesidir.
- Karar değişkenleri: X_n=*n*nci tip kargodan taşınacak konteynır miktarıdır.
- Safha dönüşüm fonksiyonu: C_n=*n*nci tip bir konteynır kargonun ağırlığı olmak üzere;

$$s_{n+1} = t(s_n, x_n) = s_n - c_n x_n$$

- Ardışık ilişki: P_n= *n*nci tip bir konteynır kargodan elde edilecek kar miktarı olmak üzere;

$$f_n(s_n, x_n) = p_n x_n + f^*_{n+1}(s_n - c_n x_n)$$

$$f^*_n(s_n) = \underset{x_n}{\text{maks}}\ f_n(s_n, x_n) = \underset{x_n}{\text{maks}}\ \{ p_n x_n + f^*_{n+1}(s_n - c_n x_n)\}$$

Sırt Çantası Problemleri

- Örnek (Havayolu Taşımacılığı)
- Çözüm (n=3)

n=3 (portatif çadır); $f_3(s_3, x_3) = 150x_3$

s_3 \ x_3	$f_3(s_3,x_3)$ 0	1	2	$f^*_3(s_3)$	x^*_3
0,1,2	0	–	–	0	0
3,4,5	0	150	–	150	1
6,7	0	150	300	300	2

- **S3=6,7 hesaplamaları**
  f3(6,0)=f3(7,0)= 0
  f3(6,1)=f3(7,1)= 150(1)= 150
  f3(6,2)=f3(7,2)= 150(2)= 300

Kargo	Ağırlık	Kar	Miktar
Termos	1	60	5
Televizyon	2	90	3
Portatif Çadır	3	150	3

Sırt Çantası Problemleri

- Örnek (Havayolu Taşımacılığı)
- Çözüm (n=2)

n=2 (televizyon); $f_2(s_2,x_2) = 90x_2 + f^*_3(s_2-c_2x_2)$

s_2 \ x_2	$f_2(s_2,x_2)$ 0	1	2	3	$f^*_2(s_2)$	x^*_2
0.1	0	–	–	–	0	0
2	0	90	–	–	90	1
3	150	90	–	–	150	0
4	150	90	180	–	180	2
5	150	240	180	–	240	1
6	300	240	180	270	300	0
7	300	240	330	270	330	2

- **S2=7 hesaplamaları**
  f2(7,0)= 90(0)+f3*(7-2(0))= 0+300= 300
  f2(7,1)= 90(1)+f3*(7-2(1))= 90+150= 240
  f2(7,2)= 90(2)+f3*(7-2(2))= 180+150= 330
  f2(7,3)= 90(3)+f3*(7-2(3))= 270+0= 270

Kargo	Ağırlık	Kar	Miktar
Termos	1	60	5
Televizyon	2	90	3
Portatif Çadır	3	150	3

Sırt Çantası Problemleri

- Örnek (Havayolu Taşımacılığı)
- Çözüm (n=1)

n=1 (termos); $f_1(s_1,x_1) = 60x_1 + f^*_2(s_1-c_1x_1)$

s_1 \ x_1	$f_1(s_1,x_1)$ 0	1	2	3	4	5	$f^*_1(s_1)$	x^*_1
7	330	360	360	360	390	390	390	4 veya 5

- OPTİMAL ÇÖZÜM
- Alternatif 1
  S1=7 için *X1*=4* → *S2=S1-1(X1*)=7-1(4)=3*
  S2=3 için *X2*=0* → *S3=S2-2(X2*)=3-2(0)=3*
  S3=3 için *X3*=1*
- Alternatif 2
  S1=7 için *X1*=5* → *S2=S1-1(X1*)=7-1(5)=2*
  S2=2 için *X2*=1* → *S3=S2-2(X2*)=2-2(1) =0*
  S3=0 için *X3*=0*

Kargo	Ağırlık	Kar	Miktar
Termos	1	60	5
Televizyon	2	90	3
Portatif Çadır	3	150	3

- Her iki alternatifte de Toplam Kar 390 birimdir.

Üretim Satınalma Problemleri

- Üretim veya satın alma problemlerinde, belli dönemlere ait taleplerin karşılanması için izlenecek üretim veya satın alma politikaları belirlenir.
- Bu problemlerde her dönemdeki talep, üretim veya satın alma maliyetlerinin yanısıra, izlenecek stok politikası, ilk dönemin başında elde bulunan stok miktarları ile stok bulundurma maliyetleri belirleyici rol oynamaktadır.

Üretim Satınalma Problemleri

- Örnek (Satın alma-Stok problemi)
- Bir fabrika, özel bir çeşit kompresör alımıyla ilgili olarak Ocak, Şubat ve Mart aylarını kapsamak üzere üç aylık dönem için bir satın alma planı oluşturacaktır. Satın alma ve stok bulundurma ile ilgili bilgiler aylara göre aşağıdaki tabloda verilmiştir.

Ay	Talep	Kompresör Fiyatı (Ayın Başında)	Depolama Maliyeti
Ocak	3	150	12
Şubat	4	160	10
Mart	2	175	10

- Fabrika her ayın başında, talebi karşılayabilmek için gerekli sayıda kompresör satın almak zorundadır. Talep fazlası kompresörler stokta tutularak bir sonraki aya devredecektir. Ocak ayı başında fabrikanın elinde kompresör bulunmamakta ve Mart ayı sonunda da stok istenmemektedir. Buna göre minimum maliyet için hangi ay ne kadar kompresör satın alınmalıdır.

Üretim Satınalma Problemleri

- Örnek (Satın alma-Stok problemi)
- Formülasyon
- Safhalar: *n*nci safha *n*nci ay (*n=1, 2, 3*)
- Durumlar: S_n *n*nci ayın başında eldeki stok miktarı
- Karar değişkenleri: X_n *n*nci ayda satın alınacak kompresör miktarı
- Safha dönüşüm fonksiyonu: d_n *n*nci aydaki talep miktarı olmak üzere, $S_n+X_n-d_n$ değeri; başlangıç stok miktarı ile satın alınan kompresör toplamının o ayki talebin üzerinde olan kısmını (talep fazlasını) ifade etmektedir. o halde $S_n+X_n-d_n$, bir sonraki ayın başlangıç stok miktarına eşit olacaktır.yani;

$$s_{n+1} = t(s_n, x_n) = s_n + x_n - d_n$$

Üretim Satınalma Problemleri

- Örnek (Satın alma-Stok problemi)
- Formülasyon
- Ardışık ilişki: Her ayın toplam maliyeti, satın alma ve stok bulundurma maliyetlerinden oluşacaktır. Buna göre, o ayki satın alma maliyeti ile bir sonraki safhaya devredilecek olan kompresörler için stok bulundurma maliyetlerini göz önüne alacak bir maliyet fonksiyonuna ihtiyaç vardır. *n*nci aydaki birim kompresör fiyatı P_n, stok bulundurma maliyeti de C_n ile gösterilirse, safha maliyet fonksiyonu aşağıdaki gibi yazılabilir.

$$r_n(s_n, x_n) = p_n x_n + c_n(s_n + x_n - d_n)$$

Toplam maliyet fonksiyonu;

$$\begin{aligned} f_n(s_n, x_n) &= r_n(s_n, x_n) + f^*_{n+1}(s_n + x_n - d_n) \\ &= p_n x_n + c_n(s_n + x_n - d_n) + f^*_{n+1}(s_n + x_n - d_n) \end{aligned}$$

Optimal toplam maliyet

$$f^*_n(s_n) = \min_{x_n} f_n(s_n, x_n) = \min_{x_n} \{ p_n x_n + c_n(s_n + x_n - d_n) + f^*_{n+1}(s_n + x_n - d_n) \}$$

Üretim Satınalma Problemleri

- Örnek (Satın alma-Stok problemi)
- Çözüm(n=3)
- Mart ayı sonunda herhangi bir stok istenmediğine ve bu aydaki talep *2* olduğuna göre mart ayı başında stokta en fazla *2* birim kompresör olabilir. Buna göre *s_3=0,1 veya 2* olabilir.

n=3 (Mart); $f_3(S_3,X_3)=175X_3$

S_3 \ X_3	$f_3(S_3,X_3)$ 0	1	2	$f_3^*(S_3)$	X_3^*
0	-	-	350	350	2
1	-	175	-	175	1
2	0	-	-	0	0

- **f3(0,0) talebi karşılamaz**
  f3(0,1) talebi karşılamaz
  f3(0,2)=175(2)=350
- **f3(1,0) talebi karşılamaz**
  f3(1,1)=175(1)=175
  f3(1,2) dönem sonu stok kalır
- **f3(2,0)=175(0)=0**
  f3(2,1) dönem sonu stok kalır
  f3(2,2) dönem sonu stok kalır

Ay	Talep	Kompresör Fiyatı	Depolama Maliyeti
Ocak	3	150	12
Şubat	4	160	10
Mart	2	175	10

Üretim Satınalma Problemleri

- Örnek (Satın alma-Stok problemi)
- Çözüm(n=2)
- **Başlangıçta şubat ve mart ayı taleplerini karşılayacak şekilde en fazla 6 adet kompresöre sahip olunabileceği gibi başlangıçta stok olmayabilir.**

n=2 (Şubat); $f_2(S_2,X_2)=160X_2+10(S_2+X_2-d_2)+f_3^*(S_2+X_2-d_2)$

S_2 \ X_2	$f_2(S_2,X_2)$ 0	1	2	3	4	5	6	$f_2^*(S_2)$	X_2^*
0					990	985	980	980	6
1				830	825	820		820	5
2			670	665	660			660	4
3		510	505	500				500	3
4	350	345	340					340	2
5	185	180						180	1
6	20							20	0

Talebi Karşılamaz

Mart ayı sonunda depoda stok kalır

- **S_2=6 hesaplamaları**
  f2(6,0)=160(0)+10(6+0-4)+f3*(6+0-4)=0+20+0=20
  f2(6,1), f2(6,2), f2(6,3), f2(6,4), f3(6,5), f3(6,6)
  Mart sonunda stok kalır

Ay	Talep	Kompresör Fiyatı	Depolama Maliyeti
Ocak	3	150	12
Şubat	4	160	10
Mart	2	175	10

Üretim Satınalma Problemleri

- Örnek (Satın alma-Stok problemi)
- Çözüm(n=1)
- Bu safhada başlangıç stok miktarı sıfır olduğundan sadece bir durum söz konusudur, $S_1=0$ Satın alınacak miktar *3*'ten *9*'a kadar değişebilir.

$n=1$ (Ocak); $f_1(S_1,X_1)=150X_1+12(S_1+X_1-d_1)+f_2^*(S_1+X_1-d_1)$									
S_1 \ X_1	$f_1(S_1,X_1)$							$f_1^*(S_1)$	X_1^*
	3	4	5	6	7	8	9		
0	1430	1432	1434	1436	1438	1440	1442	1430	3

- **OPTİMAL ÇÖZÜM**
- ***S1=0* için *X1*=3, S2=S1+X1-d1=0+3-3=0***
- ***S2=0* için *X2*=6, S3=S2+X2-d2=0+6-4=2***
- ***S3=2* için *X3*=0***
- **Bu çözüme göre (Ocak, Şubat, Mart) = (*3, 6, 0*) Kompresör satın alınacak, minimum satın alma ve stok bulundurma maliyeti *f1*(s1)=1430* birim olacaktır.**

Ay	Talep	Kompresör Fiyatı	Depolama Maliyeti
Ocak	3	150	12
Şubat	4	160	10
Mart	2	175	10

BÖLÜM III: Markov Zincirleri

- **MARKOV ZİNCİRLERİNE GİRİŞ**
- **STOKASTİK SÜREÇLER**
- **MARKOV ZİNCİRLERİ**
- ***n* ADIMDA GEÇİŞ OLASILIKLARI**
- **DURUMLARIN SINIFLANDIRILMASI**
- **DENGE DURUMLARI**
- **YUTAN ZİNCİRLER**

Markov Zincirlerine Giriş

- Zaman içerisinde tamamen önceden kestirilemeyecek şekilde gelişen süreçlere stokastik süreçler denir.
- Bu belirsizliğe olayların tutarsızlığından kaynaklanan ve kontrol edilemeyen değişimler neden olur.
- Markov analizi Rus matematikçi Andrey Markov tarafından geliştirilmiş olup stokastik süreçlerin değerlendirilmesinde kullanılır.
- Markov analizi bir optimizasyon tekniği olmayıp, çeşitli karar durumlarında karar vermeye yardımcı olabilecek olasılıklı bilgiler sağlar.
- Markov analizi, zaman içerisinde bir durumdan diğer bir duruma olasılıklı olarak geçen sistemlere uygulanır.
- Örneğin bugün çalışan bir makinenin ertesi gün arızalanma olasılığı veya bir müşterinin kullandığı deterjan markasını değiştirme olasılığının hesaplanmasında kullanılabilir.

Stokastik Süreçler

- Stokastik süreç, rasgele sonuçlar doğuran bir olaylar serisidir.
- Stokastik süreçler olayların zamana göre değerlendirildiği süreçlerdir.
- Stokastik süreç, bir rasgele değişkenler kümesi (xt) ile tanımlanır. Burada t bilinen bir t kümesine ait zaman indisidir.
- Rasgele değişkenin aldığı her bir değere durum denir. Bu nedenle x_t için durum değişkeni ifadesi de kullanılır.
- Rasgele değişkenin alabileceği değerlerin tanımlandığı s uzayı durum uzayı olarak adlandırılır.
- Örneğin bir mağazada her saatin sonunda (t zamanında) gözlenen müşteri sayısı x_t olmak üzere;

$S=\{0,1,2,3,\ldots\}$ $T=\{1,2,3,4,5,6,7,8\}$

$\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8\}=\{2,5,3,6,10,4,3,5\}$

Stokastik Süreçler

- Durum uzayı s, sürekli değerlerden oluşuyorsa sürekli durumlu stokastik süreç, kesikli (tamsayılı) değerlerden oluşuyorsa kesikli durumlu stokastik süreç olarak adlandırılır.
- Zaman kümesi t, sürekli değerlerden oluşuyorsa sürekli zamanlı stokastik süreç, kesikli (tamsayılı) değerlerden oluşuyorsa kesikli zamanlı stokastik süreç olarak adlandırılır.

Stokastik Süreçler

- Örnek (İş Makinası)
- Bir firmada kullanılan pres makinası her günün sonunda kontrol edilmektedir. Makina çalışır durumda ise her hangi bir işlem yapılmamakta, arızalanmış ise o akşam tamir edilerek ertesi sabah çalışır durumda olması sağlanmaktadır. Bu makinanın durumuyla ilgili son 10 güne ait bilgiler aşağıdaki tabloda verilmiştir. (1:çalışıyor, 0:arızalı)

Gün	1	2	3	4	5	6	7	8	9	10
Durum	0	0	0	1	0	0	1	1	1	0

- Burada rasgele değişken (x_t) kesikli olup iki değerden birini alabilmektedir, s={0, 1}. Zaman kümesi t kesikli değerler almaktadır, t={1,2,3,…}. Öyleyse bu süreç kesikli durumlu ve kesikli zamanlı bir stokastik süreçtir.
- $x_t=\{x_1,x_2,x_3,\ldots,x_{10}\}$ sürecin genel genel gösterimi ve {0,0,0,1,0,0,1,1,1,0} ise sürecin aldığı değerlerdir.

Markov Zincirleri

Markov Özelliği

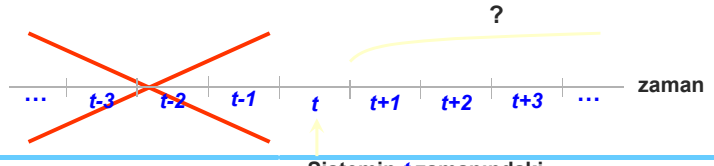
- Sistemin şimdiki durumu ve geçmişte bulunduğu durumlar biliniyor olsun, buna göre sistemin gelecekteki durumunun koşullu olasılığı şimdiki durumuna bağlı olup geçmişteki durumlardan bağımsızdır.

Markov süreçleri

- Markov özelliğini sağlayan stokastik süreçlere Markov süreçleri denir.

Markov zinciri

- Markov özelliğine sahip stokastik bir x_t süreci, eşit ve kesikli zaman aralıklarıyla ifade ediliyorsa ($t=0,1,2,3,\ldots$) Markov zinciri olarak adlandırılır.



Sistemin *t* zamanındaki durumu BİLİNİYOR.

Markov Zincirleri

Geçiş Olasılıkları

- $P(X_{t+1}=j \mid X_t=i)=P_{ij}$

Burada P_{ij}, sistemin herhangi bir dönemde (t döneminde) i durumunda iken bir sonraki dönemde ($t+1$ döneminde) j durumuna geçme olasılığıdır ve geçiş olasılığı olarak adlandırılır.

Başlangıç Olasılığı Vektörü

- q_i olasılığı, sistemin başlangıçta ($t=0$ anında) i durumunda bulunma olasılığı olsun, yani $P(X_0=i)=q_i$. Buna göre

$$q=[q_1, q_2,\ldots,q_S]$$

vektörü başlangıç olasılık dağılımını gösteren başlangıç olasılığı vektörü olarak adlandırılır.

- Markov zincirleri ile ilgili yapılacak incelemede başlangıç olasılıklarının da bilinmesi gerekmektedir.

Markov Zincirleri

- Örnek(İş Makinası Örneğinin Devamı)
- Stokastik sürecin son 10 günde aldığı değerler x_t={0,0,0,1,0,0,1,1,1,0} incelenirse, durumlar arasındaki geçiş miktarları aşağıdaki şekilde saptanabilir.

Durumdan (t Günü)	Durumuna (t+1 Günü)		
	Arızalı (0)	Çalışıyor (1)	
Arızalı (0)	3	2	5
Çalışıyor (1)	2	2	4

P_{00} :Herhangi bir günün sonunda makina arızalı iken bir sonraki günün sonunda da arızalı olma olasılığı ise;

$$P_{00} = 3/5$$
$$P_{01} = 2/5$$
$$P_{10} = 2/4$$
$$P_{11} = 2/4$$

$$P = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 0{,}6 & 0{,}4 \\ 0{,}5 & 0{,}5 \end{bmatrix} \end{array}$$

- Geçiş matrisinde satır toplamları 1 olmak zorundadır.
- Matrise göre;

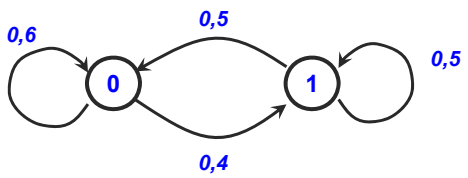
$$P_{00} = P(X_{11} = 0 \mid X_{10} = 0) = 0{,}6$$
$$P_{01} = P(X_{11} = 1 \mid X_{10} = 0) = 0{,}4$$
$$\vdots$$

Markov Zincirleri

- Örnek(İş Makinası Örneğinin Devamı)

$$P = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 0{,}6 & 0{,}4 \\ 0{,}5 & 0{,}5 \end{bmatrix} \end{array}$$

- Aşağıdaki şekilde geçiş matrisi geçiş diyagramı şeklinde gösterilmiştir. Düğümler durumları, arklar geçişleri, arklara ait rakamlar ise geçiş olasılıklarını gösterir.



Markov Zincirleri

Örnek(Arazi Hakimiyeti)

- Mavi kuvvetler ile kırmızı kuvvetler bölgedeki toplam 4 kritik araziyi elde etmek için muharebeye girişeceklerdir. Kritik arazilerden ikisi başlangıçta mavinin elindedir. Mavinin sonraki günlerde (t=1,2,…) yapılacak olan her bir muharebeyi kazanma olasılığı *p*, kaybetme olasılığı ise *1-p* olup, muharebeyi kazanırsa bir kritik araziyi ele geçirecek, kaybederse bir kritik araziyi kaybedecektir. Mavi 4 kritik araziyi de ele geçirirse veya elindeki bütün kritik arazileri kaybederse muharebeler sona erecektir. Her bir muharebeyi taraflardan birinin mutlaka kazanacağını kabul edelim.

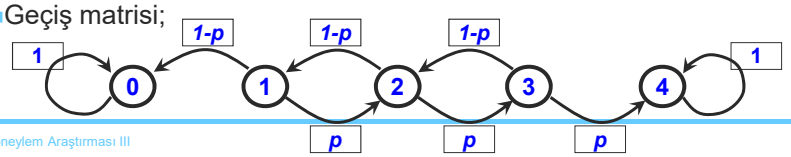
Markov Zincirleri

Örnek(Arazi Hakimiyeti)

- Bu stokastik süreç için rasgele değişken;
  X_t :t gününde yapılan muharebe sonunda mavinin elindeki kritik arazi sayısı,
- olarak tanımlanırsa $\{X_t, t=0,1,2,\ldots\}$ süreci bir markov zinciri oluşturur.
- Durum uzayı s={0,1,2,3,4}
- Geçiş matrisi;

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1-p & 0 & p & 0 & 0 \\ 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 1-p & 0 & p \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

- Geçiş matrisi;



n- Adımda Geçiş Olasılıkları

- Bir markov zinciri *m* zamanında *i* durumunda iken, *n* dönem sonra *j* durumunda bulunma olasılığı aşağıdaki gibi yazılabilir;

$$P(X_{m+n}=j \mid X_m=i)=P(X_n=j \mid X_0=i)=p_{ij}^{(n)}$$

Burada *Pij*, *i* durumundan *j* durumuna n-adımda geçiş olasılığı olarak adlandırılır.

n- Adımda Geçiş Olasılıkları

- Örnek(İş Makinası Örneğinin Devamı)

$$P=\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0,6 & 0,4 \\ 1 & 0,5 & 0,5 \end{array}$$

- 1-adımda geçiş olasılığı: 10ncu günün sonunda makine arızalandığına göre 11nci günde de arızalanma olasılığı nedir?
- Bu olasılık, geçiş matrisindeki 0 satın ile 0 sütununa karşılık gelen (0,0) elemanıdır;

$$p_{00}^{(1)}=P(X_{11}=0 \mid X_{10}=0)=p_{00}=0,6$$

n- Adımda Geçiş Olasılıkları

- Örnek(İş Makinası Örneğinin Devamı)

$$P = \begin{array}{c} \\ 0 \\ 1 \end{array}\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 0{,}6 & 0{,}4 \\ 0{,}5 & 0{,}5 \end{bmatrix} \end{array}$$

- 2-adımda geçiş olasılığı: 10ncu günün sonunda makine arızalandığına göre 12nci günde de arızalanma olasılığı nedir?
- Önce 10ncu günde 0 durumundan 11nci günde 0 ve 1 durumlarına, daha sonra 11nci günde 0 ve 1 durumlarından 12nci günde 0 durumuna geçilmelidir. Bu olasılık aşağıdaki gibi ifade edilebilir,

$$p_{00}^{(2)} = \sum_{k=0}^{1} \text{(0 durumundan } k \text{ durumuna geçiş olasılığı)} \times \text{(} k \text{ durumundan 0 durumuna geçiş olasılığı)}$$

$$\begin{aligned} p_{00}^{(2)} = P(X_{12}=0|X_{10}=0) &= P(X_{11}=0|X_{10}=0)\cdot P(X_{12}=0|X_{11}=0) + \\ &\quad P(X_{11}=1|X_{10}=0)\cdot P(X_{12}=0|X_{11}=1) \\ &= p_{00}\cdot p_{00} + p_{01}\cdot p_{10} \\ &= (0{,}6)\cdot(0{,}6)+(0{,}4)\cdot(0{,}5) \\ &= 0{,}56 \end{aligned}$$

n- Adımda Geçiş Olasılıkları

- Örnek(İş Makinası Örneğinin Devamı)

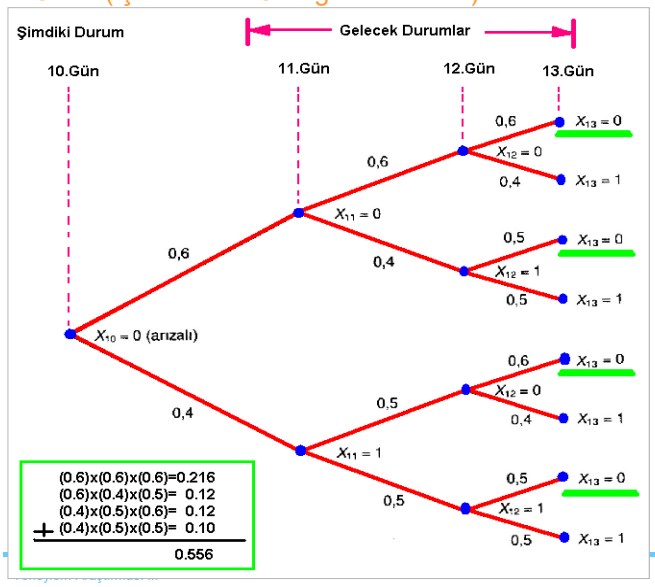
$$P = \begin{array}{c} \\ 0 \\ 1 \end{array}\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 0{,}6 & 0{,}4 \\ 0{,}5 & 0{,}5 \end{bmatrix} \end{array}$$

- 3-adımda geçiş olasılığı: 10ncu günün sonunda makine arızalandığına göre 13ncü günde de arızalanma olasılığı nedir?
- Önce 10ncu günde 0 durumundan 11nci günde 0 ve 1 durumlarına, daha sonra 11nci günde 0 ve 1 durumlarından 12nci günde 0 ve 1 durumlarına ve son olarak 12nci günde 0 ve 1 durumlarından 13ncü günde 0 durumuna geçilmelidir,

$$\begin{aligned} p_{00}^{(3)} = P(X_{13}=0|X_{10}=0) &= p_{00}\cdot p_{00}\cdot p_{00} + p_{00}\cdot p_{01}\cdot p_{10} \\ &\quad + p_{01}\cdot p_{10}\cdot p_{00} + p_{01}\cdot p_{11}\cdot p_{10} \\ &= (0{,}6)\cdot(0{,}6)\cdot(0{,}6)\cdot(0{,}6)\cdot(0{,}4)\cdot(0{,}5) \\ &\quad + (0{,}4)\cdot(0{,}5)\cdot(0{,}6)\cdot(0{,}4)\cdot(0{,}5)\cdot(0{,}5) \\ &= 0{,}556 \end{aligned}$$

n- Adımda Geçiş Olasılıkları

- Örnek(İş Makinası Örneğinin Devamı)



(0.6)x(0.6)x(0.6)=0.216
(0.6)x(0.4)x(0.5)= 0.12
(0.4)x(0.5)x(0.6)= 0.12
+ (0.4)x(0.5)x(0.5)= 0.10
0.556

Matris Yöntemi

- Yukarıdaki 2-adımda geçiş olasılığı $P_{00}^{(2)}$ ile ilgili örnek incelendiğinde yapılan işlemlerin aslında geçiş matrisinin 0 satırı ile 0 sütununun çarpımı olduğu görülür; yani $P_{00}^{(2)}$ değeri $P \cdot P = P^2$ matrisinin *(0, 0)* elemanıdır. Aynı şekilde $P_{00}^{(3)}$ olasılığının da P^3 matrisinin *(0, 0)* elemanı olduğu görülebilir.
- Bu işlemler n-adımda geçiş olasılığı için genelleştirilecek olursa Chapman-Kolmogorov denklemi olarak bilinen aşağıdaki denklem elde edilir.

$$p_{ij}^{(n)} = \sum_{k=0}^{s} p_{ik}^{(n-m)} \cdot p_{kj}^{(m)}$$

Aynı denklem matris şeklinde ifade edilecek olursa;

$$P^{(n)} = P^{(n-m)} \cdot P^{(m)}$$

- Chapman-Kolmogorov denkleminin m=1 için elde edilen ve en çok kullanılan özel şekli;

$$p_{ij}^{(n)} = \sum_{k=0}^{s} p_{ik}^{(n-1)} \cdot p_{kj}$$

$$P^{(n)} = P^{(n-1)} \cdot P$$

Matris Yöntemi

- n-adımda geçiş matrisi (*P(n)*),geçiş matrisi *P*'nin *nnci* kuvveti hesaplanarak bulunur.

$$P^{(n)}=P^{(n-1)}\cdot P=P\cdot P^{(n-1)}=P^{n}$$

Buradan da, *i* durumundan *j* durumuna, n-adımda geçiş olasılığı aşağıdaki şekilde elde edilebilir;

$$p_{ij}^{(n)}=P^{n} \text{ matrisinin } (i,j)\text{nci elamanı}$$

Matris Yöntemi

- Örnek (İş Makinası Örneğinin Devamı)
- Geçiş matrisi (veya tek adımda geçiş matrisi) daha önce elde edilmişti

$$P=\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0{,}6 & 0{,}4 \\ 1 & 0{,}5 & 0{,}5 \end{array}$$

sırasıyla 2 ve 3 adımda geçiş matrisleri;

$$P^2=P\bullet P=\begin{bmatrix}0{,}6 & 0{,}4\\ 0{,}5 & 0{,}5\end{bmatrix}\bullet\begin{bmatrix}0{,}6 & 0{,}4\\ 0{,}5 & 0{,}5\end{bmatrix}=\begin{bmatrix}0{,}56 & 0{,}44\\ 0{,}55 & 0{,}45\end{bmatrix}$$

$$P^3=P^2\bullet P=\begin{bmatrix}0{,}56 & 0{,}44\\ 0{,}55 & 0{,}45\end{bmatrix}\bullet\begin{bmatrix}0{,}6 & 0{,}4\\ 0{,}5 & 0{,}5\end{bmatrix}=\begin{bmatrix}0{,}556 & 0{,}444\\ 0{,}555 & 0{,}445\end{bmatrix}$$

- Örneğin 10ncu günde makine arızalı olduğuna göre; 12nci günde arızalı olma olasılığı $P_{00}^{(2)} = 0{,}56$ ve 13ncü günde arızalı olma olasılığı $P_{00}^{(3)} = 0{,}556$ olarak bulunur. Bu olasılıklar sırasıyla P^2 ve P^3 matrislerinin (0, 0) elemanlarıdır.

Koşulsuz Olasılıklar

$P(X_n=j|X_0=i)=p_{ij}^{(n)}$

- Yukarıdaki ifadede sistemin *n* geçişten sonra *j* durumunda olma olasılığı başlangıçtaki *i* durumuna bağlıdır.
- Markov zincirinin başlangıç (0 zamanındaki) durumu her zaman bilinmeyebilir.
- Bu durumda, başlangıç durumuna ait olasılık dağılımı biliniyorsa, *n* geçişten sonra *j* durumunun koşulsuz olasılığı, $U_j^{(n)}$, bulunabilir.

q_i : zincirin $t=0$ zamanında i durumunda bulunma olasılığı

$\mathbf{q}=[q_1 \quad q_2 \quad \ldots \quad q_s]$ başlangıç olasılığı vektörü ve

$u_j^{(n)}$: n zamanında j durumunda bulunma olasılığı

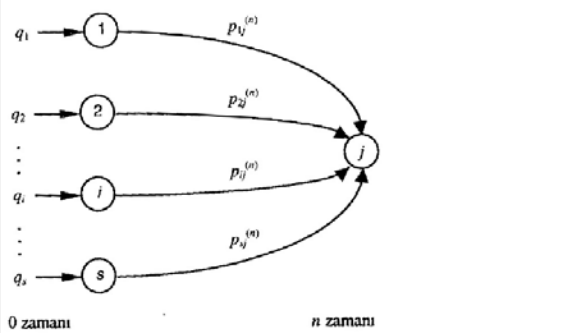
$$u_j^{(n)}=\sum_{i=1}^{s} \text{(başlangıçta } i \text{ durumunda bulunma olasılığı)}$$

$$\times \text{ (}n\text{ geçişte } i \text{ durumundan } j \text{ durumuna geçme olasılığı)}$$

$$=\sum_{i=1}^{s} q_i \, p_{ij}^{(n)}$$

$$= \mathbf{q}\,(\mathbf{P}^n \text{ matrisinin } j \text{ sütunu})$$

Koşulsuz Olasılıklar



Başlangıç Durumu Bilinmediğinde, n Zamanında j Durumunda Bulunma Olasılığının Saptanması

- $U^{(n)}$, bütün durumların *n* zamanındaki koşulsuz olasılıklarını gösteren koşulsuz olasılık vektörü olsun; yani,

$$U^{(n)}=[U_1^{(n)} \; U_2^{(n)} \; \ldots \; U_n^{(n)}]$$

- Bu durumda yukarıdaki işlemler, vektör-matris notasyonunu kullanarak aşağıdaki şekilde yazılabilir.

$$\mathbf{u}^{(n)}=\mathbf{q}\bullet\mathbf{P}^n$$

Koşulsuz Olasılıklar

- ÖRNEK(PAZAR PAYI ANALİZİ)
- A ve B marka kola üreticileri 1000 kişilik bir tüketici grubu üzerinde yılın 10ncu ve 11nci haftalarında yaptıkları bir araştırmanın sonuçları aşağıda verilmiştir.

		11nci hafta		
	Marka	(j) A	B	Toplam
10ncu hafta	(i)			
	A	360	40	400
	B	120	480	600
	Toplam	480	520	1000

1. B kola satın alan birisinin, iki hafta sonra A kola satın alma olasılığı nedir?
2. Acaba 13ncü haftada tüketicilerin yüzde kaçı A kola ve yüzde kaçı B kola satın alacaktır?

Koşulsuz Olasılıklar

- ÖRNEK(PAZAR PAYI ANALİZİ)

		11nci hafta		
	Marka	(j) A	B	Toplam
10ncu hafta	(i)			
	A	360	40	400
	B	120	480	600
	Toplam	480	520	1000

- Sistem iki durumlu bir markov zinciri olarak modellenebilir;
  - Durum 1 :tüketicinin A kola satın alması
  - Durum 2 tüketicinin B kola satın alması
- Durum değişkeni;

X_t tüketicinin t haftasında satın alacağı kola cinsi olarak tanımlanırsa $\{X_1, X_2, X_3, \ldots\}$ süreci bir markov zinciri oluşturur. bu zincirin geçiş matrisi aşağıda verilmiştir;

$$P = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} \end{array}$$

Koşulsuz Olasılıklar

- ÖRNEK(PAZAR PAYI ANALİZİ)

$$P = \begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} \end{array}$$

- Cevap 1
- $P(X_2=1|X_0=2)=p_{21}^{(2)}$ olasılığı sözkonusu olup bu **koşullu** olasılık P^2 matrisinin *(2, 1)*nci elemanıdır;

$$P^2 = P \bullet P = \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} \bullet \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} = \begin{bmatrix} 0{,}83 & 0{,}17 \\ 0{,}34 & 0{,}66 \end{bmatrix}$$

- Buna göre, $P_{21}^{(2)}=0{,}34$. Yani *B* kola satın alan bir tüketici *0,34* olasılıkla iki hafta sonra *A* kola alacaktır.

Koşulsuz Olasılıklar

- ÖRNEK(PAZAR PAYI ANALİZİ)

$$P = \begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} \end{array}$$

- Cevap 2
- Burada koşulsuz olasılıklar sorulmaktadır.
- Başlangıç zamanı 11nci hafta olacağından, 11nci haftadaki pazar payları (a kola için 480/1000 ve b kola için 520/1000) başlangıç olasılık dağılımını gösterir;

  q = [0,48 0,52]

- Yani *11*nci haftada tüketicilerin *%48*'i A kola ve *%52* si B kola satın almışlardır. Buna göre *13*ncü haftada pazar payları;

$$u^{(2)} = q \bullet P^2$$

$$= [\,0{,}48 \quad 0{,}52\,] \bullet \begin{bmatrix} 0{,}83 & 0{,}17 \\ 0{,}34 & 0{,}66 \end{bmatrix}$$

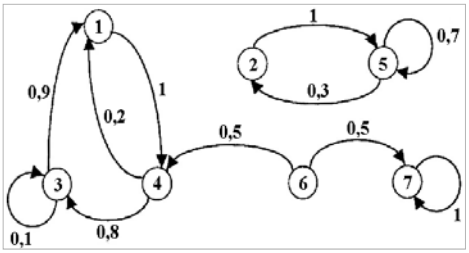
$$= [\,0{,}5752 \quad 0{,}4248\,]$$

- 13ncü haftada tüketicilerin %58'i A kola ve %42'si B kola satın alacaklardır.

Durumların Sınıflandırılması

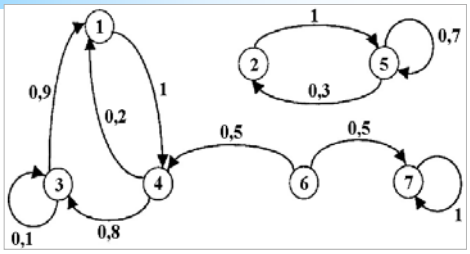
- n-adımda geçiş olasılıkları uzun geçişlerden sonra sabit bir değere yaklaşma eğilimi gösterirler.
- Markov zincirlerinin denge durumu ile ilgili bu özelliği incelemeden önce, aşağıdaki geçiş matrisi ve geçiş diyagramı üzerinde durumlarla ilgili bazı tanımlar incelenecektir.

$$P=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}\begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



Durumların Sınıflandırılması

$$P=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}\begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

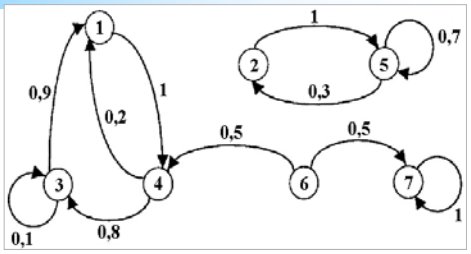


YOL

- *i* ve *j* durumları arasındaki yol, *i* durumundan başlayıp *j* durumunda sona eren, ve her birinin olasılığı sıfırdan farklı olan, geçişler zinciridir.
- *i* durumundan *j* durumuna giden bir yol varsa, *i* durumundan *j* durumuna ulaşılabilir denir.
- Örneğin, 6 durumundan 1 durumuna (6-4-1 yolu ile) ulaşılabilir, fakat 6 durumundan 2 durumuna ulaşılamaz.

Durumların Sınıflandırılması

$$P=\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 4 & 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

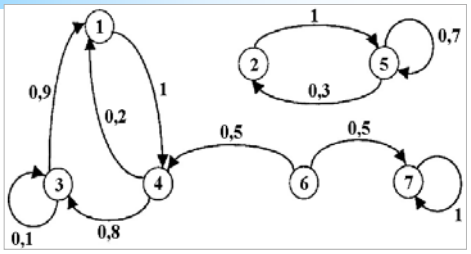


HABERLEŞEN DURUMLAR

- *i* durumundan *j* durumuna ve *j* durumundan da *i* durumuna ulaşılabiliyorsa *i* ve *j* durumları haberleşen durumlar olarak adlandırılır ve $i \leftrightarrow j$ şeklinde gösterilir.
- Örneğin, 3 ve 4 haberleşen durumlardır, öbür yandan 6 ve 7 birbiriyle haberleşemezler.

Durumların Sınıflandırılması

$$P=\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 4 & 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

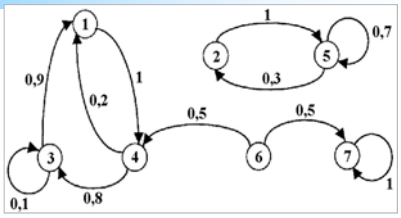


KAPALI KÜME

- Markov zincirinin herhangi bir durumlar kümesi *s*, eğer bu kümenin içindeki herhangi bir durumdan bu kümenin dışındaki hiç bir duruma geçilemiyorsa, kapalı küme olarak adlandırılır.
- Örneğin, $s_1=\{1,3,4,6,7\}$ ve $s_2=\{2,5\}$ kümelerinin ikisi de kapalı kümedir.

Durumların Sınıflandırılması

$$P=\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 4 & 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

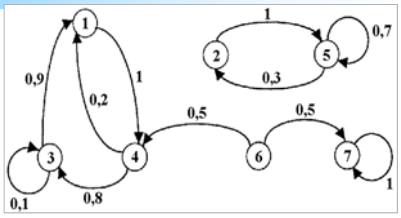


YUTAN DURUM

- *Pii = 1* ise *i* durumu yutan durum olarak adlandırılır. zincir *yutan* duruma geçtiği takdirde sürekli bu durumda kalır.
- Örneğin, 7 durumu yutan bir durumdur (sistem 7 durumuna geçerse bir daha asla çıkamaz).
- Yutan durumlar, modellenen sürecin sona erebileceği koşulları temsil ederler.
- Arazi hakimiyeti örneğindeki 0 ve 4 durumları yutan durumlardır, çünkü; mavi, kritik arazilerin tamamını kaybederse (0 durumu) veya dördünü de ele geçirirse (4 durumu) muharebeler sona erecektir.

Durumların Sınıflandırılması

$$P=\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 4 & 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

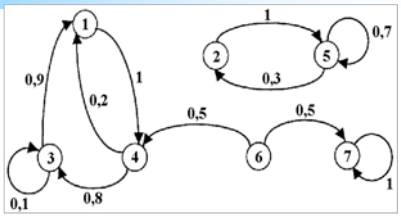


GEÇİCİ DURUM (GERİ DÖNMEYEN DURUM)

- *i* durumundan herhangi bir *j* durumuna ulaşılabiliyor fakat *j* durumundan *i* durumuna ulaşılamıyorsa (böyle bir *j* durumu varsa), *i* durumu geçici durum olarak adlandırılır.
- Örneğin, *6* durumu geçici bir durumdur (*6* durumundan *1* durumuna ulaşılabilir fakat *1* durumundan *6* durumuna ulaşılamaz).
- Uzun geçişlerden sonra geçici bir *i* durumunda bulunma olasılığı sıfırdır. geçici *i* durumuna her ulaşıldığında, bir daha geri dönmemek üzere bu durumu terkedip herhangi bir *j* durumuna gitme olasılığı vardır. yani eninde sonunda *j* durumuna ulaşılacak ve ondan sonra *i* durumuna geri dönülmeyecektir.

Durumların Sınıflandırılması

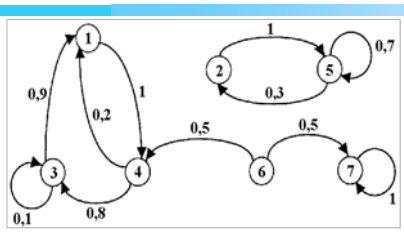
$$P=\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0,9 & 0 & 0,1 & 0 & 0 & 0 & 0 \\ 4 & 0,2 & 0 & 0,8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0,3 & 0 & 0 & 0,7 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0,5 & 0 & 0 & 0,5 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$



TEKRARLANAN DURUM (GERİ DÖNEN DURUM)

- Herhangi bir *i* durumu geçici bir durum değilse tekrarlanan durumdur.
- Herhangi bir tekrarlanan *i* durumundan hangi *j* durumuna gidilirse gidilsin *j* durumundan *i* durumuna geri dönen bir yol vardır.
- Örneğin, *6* durumu hariç bütün durumlar tekrarlanan durumlardır.

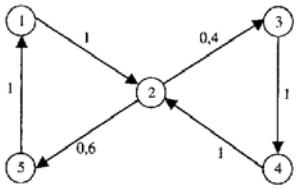
Durumların Sınıflandırılması



PERİYODİK DURUM

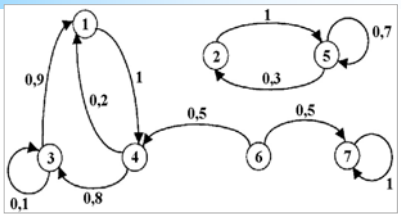
- Birden büyük bir tamsayı (k>1) ve bu tamsayının katı adımlarla (k,2k,3k,...) tekrarlayan *i* durumu periyodik bir durumdur ve *k* değeri bu durumun periyodunu gösterir.

- Tekrarlayan bir durum periyodik değilse aperiyodik olarak adlandırılır.
- yukardaki şekilde bütün durumlar aperiyodik olup, aşağıdaki markov zincirinde ise bütün durumlar periyodiktir.
- örneğin 1 durumu; 1-2,2-5,5-1 yolu ile 3 adımda ve 1-2,2-3,3-4,4-2, 2-5,5-1 yolu ile 6 adımda tekrarlamaktadır. bu şekildeki her bir durumun periyodu 3 tür.



Durumların Sınıflandırılması

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0{,}9 & 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 0{,}2 & 0 & 0{,}8 & 0 & 0 & 0 & 0 \\ 0 & 0{,}3 & 0 & 0 & 0{,}7 & 0 & 0 \\ 0 & 0 & 0 & 0{,}5 & 0 & 0 & 0{,}5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



ERGODİK MARKOV ZİNCİRİ

- Bir markov zincirindeki bütün durumlar tekrarlanan, aperiyodik ve birbiriyle haberleşen durumlar ise bu zincir ergodik bir zincirdir.
- Arazi hakimiyeti örneği ergodik olmayan bir zincirdir (örneğin 3 ve 4 durumları birbiriyle haberleşemezler).
- İş makinesi örneği ise ergodik bir zincirdir.

Denge Durumu

- *n*-adımda geçiş olasılıkları, yeterince uzun bir geçiş sürecinden sonra sabit bir değere yaklaşma eğilimi gösterirler, yani kararlı bir hale gelirler.
- Aşağıda iş makinesi örneğine ait hesaplamalar verilmiştir.

n	P^n
1	$\begin{bmatrix} 0{,}60 & 0{,}40 \\ 0{,}50 & 0{,}50 \end{bmatrix}$
2	$\begin{bmatrix} 0{,}56 & 0{,}44 \\ 0{,}55 & 0{,}45 \end{bmatrix}$
3	$\begin{bmatrix} 0{,}556 & 0{,}444 \\ 0{,}555 & 0{,}445 \end{bmatrix}$
4	$\begin{bmatrix} 0{,}5556 & 0{,}4444 \\ 0{,}5555 & 0{,}4445 \end{bmatrix}$
5	$\begin{bmatrix} 0{,}55556 & 0{,}44444 \\ 0{,}55555 & 0{,}44445 \end{bmatrix}$

- Örneğin. $P_{11}=0.6$; $P_{11}^{(2)}=0{,}56$; $P_{11}^{(3)}=0{,}556$; $P_{11}^{(4)}=0{,}5556$ ve $P_{11}^{(5)}=0.55556$ serisi incelendiğinde, $P_{11}^{(n)}$ olasılığındaki değişim miktarının her adımda gittikçe azaldığı görülür.
- Bu da bize n değeri arttıkça $P_{11}^{(n)}$ olasılığının sabit bir değere yaklaştığını gösterir.

Denge Durumu

DENGE DURUMU OLASILIKLARI

TEOREM

- P, *s* durumlu ergodik bir markov zincirinin geçiş matrisi olmak üzere;

$$\lim_{n\to\infty} P^n = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ s \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & s \end{array} \\ \begin{bmatrix} \pi_1 & \pi_2 & \cdots & \pi_s \\ \pi_1 & \pi_2 & \cdots & \pi_s \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_s \end{bmatrix} \end{array}$$

yani her hangi bir *i* başlangıç durumu için;

$$\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j$$ olacaktır.

- Uzun geçişlerden sonra ($n\to\infty$), markov zinciri dengeye ulaşmakta ve *j* durumunda bulunma olasılığı π_j, başlangıç durumundan bağımsız olmaktadır. Bu π olasılığına denge durumu olasılığı (kararlı olasılık) denir.
- $\pi=[\pi_1,\pi_2,\ldots,\pi_s]$ vektörü denge durumundaki olasılık dağılımını gösterir.

Denge Durumu

DENGE DURUMU OLASILIKLARININ BULUNMASI

$$\pi = \pi P$$
$$\pi_1+\pi_2+\ldots+\pi_s=1$$

Denklem sistemleri ortak çözülerek denge durumu olasılıkları bulunabilir.

ÖRNEK 1

$$P = \begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} \end{array}$$

Kola örneği geçiş matrisi için denge durumu olasılıklarını bulunuz.

$$[\pi_1 \quad \pi_2] = [\pi_1 \quad \pi_2] \cdot \begin{bmatrix} 0{,}90 & 0{,}10 \\ 0{,}20 & 0{,}80 \end{bmatrix} \rightarrow \begin{array}{l} \pi_1 = 0{,}90\pi_1 + 0{,}20\pi_2 \\ \pi_2 = 0{,}10\pi_1 + 0{,}80\pi_2 \end{array}$$

ve $\pi_1+\pi_2=1$ ortak çözülürse

$\pi_1 = 2/3$, $\pi_2 = 1/3$

- Buna göre, uzun bir zaman sonra herhangi bir tüketici 2/3 olasılıkla A kola ve 1/3 olasılıkla B kola satın alacaktır. Başka bir ifadeyle tüketicilerin 2/3'ü A ve 1/3'ü B kolayı tercih edeceklerdir. Bu olasılıklar A kola ve B kolanın uzun zaman sonra ulaşılacak denge durumundaki pazar paylarıdır.

Denge Durumu

DENGE DURUMU OLASILIKLARININ BULUNMASI

ÖRNEK 2 (MARKA DEĞİŞTİRME MODELİNİN DEVAMI)

- Diyelim ki, toplam 10 milyon kola tüketicisi olup, her tüketici haftada bir şişe kola satın almaktadır (52 hafta=1 yıl). Bir şişe kolanın üretici firmaya maliyeti 10 birim ve satış fiyatı 20 birimdir. Bilindiği gibi A kola satın alan bir tüketicinin bir sonraki hafta B kolaya geçme olasılığı 0.10'dur. Bir reklam şirketi, A kola satın alan tüketicilerden bir sonraki hafta B kolaya geçenlerin oranını %10'dan %5'e düşüreceğini ileri sümekte ve bunun için yılda $5x10^8$ birim istemektedir. Acaba A kola firması reklam şirketiyle anlaşma yapmalı mıdır?
- ÇÖZÜM
- π_1=2/3 A kola firmasının şu anki yıllık karı;
  (2/3)(52)(10,000,000)(20-10)=$3.467x10^9$
- Reklam şirketi aşağıdaki geçiş matrisini teklif etmektedir;

$$P_1 = \begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 0{,}95 & 0{,}05 \\ 0{,}20 & 0{,}80 \end{bmatrix} \end{array}$$

Yeni Denge Durumu Olasılıkları

$$\left.\begin{array}{l} \pi_1 = 0{,}95\pi_1 + 0{,}20\pi_2 \\ \pi_2 = 0{,}05\pi_1 + 0{,}80\pi_2 \\ \pi_1 + \pi_2 = 1 \end{array}\right\} \Rightarrow \pi_1 = 0{,}8 \text{ ve } \pi_2 = 0{,}2$$

(0.8)(52)(10,000,000)(20-10)-($5x10^8$)=$3.660x10^9$ → Kabul etmeli

Denge Durumu

ORTALAMA İLK GEÇİŞ ZAMANLARI

- Ergodik bir Markov zinciri için m_{ij}, *i* durumunda iken *j* durumuna ilk kez ulaşmak için gerekli geçiş sayısının beklenen değeri olsun. Bu durumda m_{ij}, *i* durumundan *j* durumuna ortalama ilk geçiş zamanı olarak adlandırılır.
- Marka değiştirme örneğine göre, m_{12} herhangi bir günde a kola satın alan birisinin ilk olarak b kola almadan önce satın alacağı ortalama kola miktarıdır.
- Diyelim ki şu anda *i* durumundayız. o halde, P_{ij} olasılığıyla *i* durumundan *j* durumuna bir geçişte gidilecektir. öbür yandan, P_{ik} olasılığıyla $(k \neq j)$, *k* durumuna geçilecek ve bu durumda ise *i* durumundan *j* durumuna ortalama $1+m_{kj}$, geçişte gidilecektir. Bu açıklamaya göre;

$$m_{ij} = p_{ij}(1) + \sum_{k \neq j} p_{ik}(1+m_{kj}) \quad \textbf{Burada} \quad p_{ij} + \sum_{k \neq j} p_{ik} = 1 \Rightarrow \boxed{m_{ij} = 1 + \sum_{k \neq j} p_{ik}\, m_{kj}}$$

- ORTALAMA TEKRARLANMA ZAMANI(m_{ii}): Sistem *i* durumunda iken tekrar *i* durumuna dönmek için geçecek ortalama geçiş sayısı aşağıdaki denklem kullanılarak hesaplanır;

$$m_{ii} = \frac{1}{\pi_i}$$

Denge Durumu

ORTALAMA İLK GEÇİŞ ZAMANLARI

ÖRNEK

- Kola örneğinde denge durumu olasılıkları, $\pi_1=2/3$ ve $\pi_2=1/3$ bulunmuştu. Ortalama tekrarlanma zamanları,

$$m_{11}=\frac{1}{\pi_1}=\frac{1}{(2/3)}=1{,}5 \quad \text{ve} \quad m_{22}=\frac{1}{\pi_2}=\frac{1}{(1/3)}=3$$

Örneğin, en son b kola satın alan birisi tekrar b kola alıncaya kadar ortalama 3 şişe kola satın alacaktır.

- Ortalama ilk geçiş zamanları;

$$m_{12}=1+p_{11}m_{12} = 1+0{,}9m_{12}$$

$$m_{21}=1+p_{22}m_{21} = 1+0{,}8m_{21}$$

Buradan da $m_{12}=10$ ve $m_{21}=5$ değerleri bulunur. Örneğin, en son a kola içen birisi b kola alıncaya kadar ortalama 10 şişe kola satın alacaktır, yani 10 şişe daha a kola içecektir.

Yutan Zincirler

- Bir markov zincirinin durumlarından bazıları yutan ve diğerleri de geçici durumlar ise bu markov zinciri yutan zincir olarak adlandırılır.
- Yutan zincirlerde, herhangi bir geçici durumdan eninde sonunda yutan bir duruma geçilir ve süreç sona erer
- Yutan zincirlerde yutan durumlara geçme olasılıkları hesaplanarak sürecin hangi yutan durumda sona ereceği bulunabilir.

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

- Müşterilerine kredi ile alışveriş olanağı sunan bir mağaza, alacakların tahsili için son ödeme tarihinden sonra 2 hafta beklemekte ve bu süre içerisinde borcunu ödemeyen müşterileri cezalı müşteriler kategorisine koyup alacak tahsili için yasal yollara başvurmaktadır. iki hafta içerisinde borcunu ödeyen bir müşteri yutan bir duruma geçmiş olacaktır. öbür yandan borcunu iki haftadan fazla geciktiren bir müşteri de yine yutan bir duruma geçmiş olacaktır. her haftanın başında müşterilerin hesapları incelenip aşağıdaki durumlara göre değerlendirimektedir (son ödeme tarihinden itibaren 1-7 gün geciken borç bir hafta gecikmiş. 8-14 gün geciken borç iki hafta gecikmiş olarak değerlendirilecektir).

DURUM 1	:BORCUNU ÖDEMİŞ
DURUM 2	:BORCUNU BİR HAFTA GECİKTİRMİŞ
DURUM 3	:BORCUNU İKİ HAFTA GECİKTİRMİŞ
DURUM 4	:CEZALI (YASAL İŞLEM GEREKTİRİYOR)

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

- Geçmiş dönemlere ait verinin değerlendirilmesi sonunda, haftalık olarak mağazanın alacaklarının durumunu gösteren aşağıdaki geçiş matrisi elde edilmiş olsun.

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0{,}70 & 0 & 0{,}30 & 0 \\ 0{,}50 & 0 & 0 & 0{,}50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

- Bu matrise göre, 1 ve 4 durumları yutan durumlar, 2 ve 3 durumları ise geçici durumlardır.

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

- Eğer *m* adet yutan durum ve *s-m* adet geçici durum olduğu kabul edilirse, yutan zincirlerin geçiş matrisleri aşağıdaki gibi yazılabilir.

$$P = \begin{array}{c} \\ s\text{-}m \text{ satır} \\ m \text{ satır} \end{array} \begin{array}{c} \begin{array}{cc} s\text{-}m \text{ sütun} & m \text{ sütun} \end{array} \\ \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} \end{array}$$

- BURADA;

  Q :geçici durumlar arasındaki geçişi gösteren *(s-m)x(s-m)* boyutlu bir matris,

  R :geçici durumlardan yutan durumlara geçişi gösteren *(s-m)xm* boyutlu bir matris,

  0 : utan bir durumdan geçici durumlara geçilemiyeceğini gösteren ve bütün elemanları sıfır olan *mx(s-m)* boyutlu sıfır matrisi,

  I :yutan bir duruma geçildiğinde sürekli o durumda kalınacağını, yani yutan durumdan asla çıkılamıyacağını gösteren *mxm* boyutlu birim matristir.

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

MARKOV ZİNCİRİNİN TEMEL MATRİSİ

- Temel matris, sistemin yutan bir duruma geçmeden önce, herhangi bir geçici durumda ortalama olarak ne kadar kalacağını göstermekte olup aşağıdaki formülasyonla hesaplanır;

$$T=(I-Q)^{-1}$$

- Sistem eğer başlangıçta geçici *i* durumunda bulunuyorsa, yutulmadan önce geçici *j* durumunda harcayacağı ortalama dönem sayısı temel matrisin($(I-Q)^{-1}$), *(i,j)*nci elemanıdır.
- Sistem yutulmadan önce geçici *i* durumunda harcayacağı ortalama süre temel matrisin *i*nci satırının toplamına eşittir.
- Sistem eğer başlangıçta geçici *i* durumunda bulunuyorsa, sonunda yutan *j* durumuna geçme olasılığı $(I-Q)^{-1}$ matrisinin *(i,j)*nci elemanıdır.

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

- Buna göre s=4 ve m=2 olmak üzere örnek problemin geçiş matrisi;

$$P = \begin{array}{c} \\ 2 \\ 3 \\ 1 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 2 & 3 & 1 & 4 \end{array} \\ \left[\begin{array}{cc|cc} 0 & 0{,}30 & 0{,}70 & 0 \\ 0 & 0 & 0{,}50 & 0{,}50 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

$$Q = \begin{array}{c} \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0{,}30 \\ 0 & 0 \end{bmatrix} \end{array} \qquad 0 = \begin{array}{c} \\ 1 \\ 4 \end{array}\begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array}$$

$$R = \begin{array}{c} \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 4 \end{array} \\ \begin{bmatrix} 0{,}70 & 0 \\ 0{,}50 & 0{,}50 \end{bmatrix} \end{array} \qquad I = \begin{array}{c} \\ 1 \\ 4 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

- Örnek problemin temel matrisi;

$$T = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0{,}30 \\ 0 & 0 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 1 & -0{,}30 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{array}{c} \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0{,}30 \\ 0 & 1 \end{bmatrix} \end{array}$$

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

$$T = \begin{array}{c} \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0{,}30 \\ 0 & 1 \end{bmatrix} \end{array}$$

- SORU:Müşteri 2 durumunda, yani borcunu bir hafta geciktirmiş ise; borcunu ödemeden veya ceza almadan önce, ortalama olarak kaç defa 3 durumuna geçecek ya da ortalama olarak ne kadar süre 3 durumunda kalacaktır?
  CEVAP:0.30, yani temel matrisin (2,3) elemanı. o halde birinci hafta tamamlandıktan sonra müşteri 0.30 hafta daha (ikinci haftanın %30'u) borcunu ödemeyecektir.
- SORU:2 durumundaki bir müşteri (borcunu ödemeden veya ceza almadan önce) ortalama olarak ne kadar süre borçlu olarak kalacaktır?
  CEVAP:1+0.30=1.30, yani temel matrisin 2 durumuna ait satırın toplamı. buna göre borcunu bir hafta geciktiren bir müşteri ortalama olarak 1.30 hafta borcunu ödemez.

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

$$T = \begin{array}{c c} & \begin{array}{cc} 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0{,}30 \\ 0 & 1 \end{bmatrix} \end{array}$$

- SORU:2 durumundaki bir müşterinin en sonunda borcunu ödeme olasılığı veya ceza alma olasılığı nedir?

$$T \cdot R = (I - Q)^{-1} \cdot R = \begin{array}{c c} & \begin{array}{cc} 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0{,}30 \\ 0 & 1 \end{bmatrix} \end{array} \cdot \begin{array}{c c} & \begin{array}{cc} 1 & 4 \end{array} \\ \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} 0{,}70 & 0 \\ 0{,}50 & 0{,}50 \end{bmatrix} \end{array} = \begin{array}{c c} & \begin{array}{cc} 1 & 4 \end{array} \\ \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} 0{,}85 & 0{,}15 \\ 0{,}50 & 0{,}50 \end{bmatrix} \end{array}$$

- Buna göre borcunu bir hafta geciktiren bir müşteri 0.85 olasılıkla iki hafta tamamlanmadan borcunu ödeyecek, 0.15 olasılıkla borcunu ödemeyip ceza alacaktır.

Yutan Zincirler

ÖRNEK UYGULAMA (ALACAKLARIN DEĞERLENDİRİLMESİ)

$$T = \begin{array}{c c} & \begin{array}{cc} 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0{,}30 \\ 0 & 1 \end{bmatrix} \end{array}$$

- SORU:Mağazanın bir hafta gecikmeli 400 birim ve iki hafta gecikmeli 600 birim olmak üzere toplam 1000 birim alacağı olduğuna göre bu alacağın ne kadarı ödenecek, ne kadarı ise cezalandırılacaktır?
  CEVAP:Bu soruya cevap verebilmek için bu alacak miktarlarından oluşan bir matris ileTxR matrisi çarpılır.

$$\begin{array}{cc} 2 & 3 \end{array} \quad \begin{array}{cc} 1 & 4 \end{array} \quad \begin{array}{cc} 1 & 4 \end{array}$$

$$\begin{bmatrix} 400 & 600 \end{bmatrix} \cdot \begin{array}{c c} \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} 0{,}85 & 0{,}15 \\ 0{,}50 & 0{,}50 \end{bmatrix} \end{array} = \begin{bmatrix} 640 & 360 \end{bmatrix}$$

Buna göre mağaza, 1000 birim alacağının 640 birimini tahsil edecek, 360 birimini ise tahsil edemeyip cezalandıracaktır.

Yutan Zincirler

ÖRNEK UYGULAMA İŞGÜCÜ PLANLAMA

- Özel bir şirket teknisyen veya şef olarak çalışan personelin durumunu her yıl gözden geçirmekte ve buna göre personel alımına karar vermektedir. Personel başlangıçta teknisyen veya şef olarak işe alınmaktadır. Önceki yıllara ait veriye göre teknisyen olarak çalışan bir personel ertesi yıl *0.70* olasılıkla teknisyen olarak kalmakta, *0.20* olasılıkla şefliğe terfi etmekte ve *0.10* olasılıkla işten ayrılmaktadır. Şef olarak çalışan birisi ise ertesi yıl *0.95* olasılıkla şef olarak kalmakta ve *0.05* olasılıkla işten ayrılmaktadır. Ayrılan personelin yeniden işe alınması sözkonusu değildir. Buna göre;
  1. Teknisyen olarak işe yeni alınan bir personel ortalama olarak ne kadar süre şirkette çalışacaktır?
  2. Teknisyen olarak şirkette çalışan birisinin şef olarak işten ayrılma olasılığı nedir?
  3. Şef olarak işe başlayan bir personel ortalama olarak ne kadar süre şirkette şef olarak çalışacaktır?

Yutan Zincirler

ÖRNEK UYGULAMA İŞGÜCÜ PLANLAMA

- Personelin herhangi bir yıldaki konumu problemin durumlarını gösterir;
  Durum 1 teknisyen olarak çalışıyor
  Durum 2 şef olarak çalışıyor
  Durum 3 teknisyen olarak işten ayrılmış
  Durum 4 şef olarak işten ayrılmış

Geçiş matrisi;

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cc|cc} 0{,}70 & 0{,}20 & 0{,}10 & 0 \\ 0 & 0{,}95 & 0 & 0{,}05 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

$$Q = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 0{,}70 & 0{,}20 \\ 0 & 0{,}95 \end{bmatrix} \end{array} \qquad R = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 4 \end{array} \\ \begin{bmatrix} 0{,}10 & 0 \\ 0 & 0{,}05 \end{bmatrix} \end{array}$$

$$T = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0{,}70 & 0{,}20 \\ 0 & 0{,}95 \end{bmatrix} \right)^{-1} = \begin{bmatrix} 0{,}30 & -0{,}20 \\ 0 & 0{,}05 \end{bmatrix}^{-1} = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 3{,}33 & 13{,}33 \\ 0 & 20 \end{bmatrix} \end{array}$$

$$T \cdot R = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 3{,}33 & 13{,}33 \\ 0 & 20 \end{bmatrix} \end{array} \cdot \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 4 \end{array} \\ \begin{bmatrix} 0{,}10 & 0 \\ 0 & 0{,}05 \end{bmatrix} \end{array} = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 4 \end{array} \\ \begin{bmatrix} 0{,}333 & 0{,}67 \\ 0 & 1 \end{bmatrix} \end{array}$$

Yutan Zincirler

ÖRNEK UYGULAMA (İŞGÜCÜ PLANLAMA)

- CEVAP 1

$$\begin{Bmatrix}\text{Teknisyenin}\\ \text{Ortalama}\\ \text{Çalışma Süresi}\end{Bmatrix} = \begin{Bmatrix}\text{Teknisyen Olarak}\\ \text{Çalışacağı}\\ \text{Ortalama Süre}\end{Bmatrix} + \begin{Bmatrix}\text{Şef Olarak}\\ \text{Çalışacağı}\\ \text{Ortalama Süre}\end{Bmatrix}$$

$$= \begin{Bmatrix}\text{Temel Matrisin}\\ (1,1)\\ \text{Elemanı}\end{Bmatrix} + \begin{Bmatrix}\text{Temel Matrisin}\\ (1,2)\\ \text{Elemanı}\end{Bmatrix}$$

$$\boxed{= 3.33 + 13.33 = 16.66 \text{ yıl}}$$

$$T = \begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 3{,}33 & 13{,}33 \\ 2 & 0 & 20 \end{array}$$

Yutan Zincirler

ÖRNEK UYGULAMA (İŞGÜCÜ PLANLAMA)

- CEVAP 2

$$\begin{Bmatrix}\text{Teknisyenin Şef}\\ \text{Olarak İşten}\\ \text{Ayrılma Olasılığı}\end{Bmatrix} = \begin{Bmatrix}\text{T} \bullet \text{R Matrisini}\\ (1,2)\\ \text{Elemanı}\end{Bmatrix}$$

$$\boxed{= 0.67}$$

$$T \bullet R = \begin{array}{c|cc} & 3 & 4 \\ \hline 1 & 0{,}333 & 0{,}67 \\ 2 & 0 & 1 \end{array}$$

Yutan Zincirler

ÖRNEK UYGULAMA (İŞGÜCÜ PLANLAMA)

- CEVAP 3

$$\left\{\begin{matrix}\text{Şef Olarak İşe Başlayan}\\ \text{Personelin}\\ \text{Şef Olarak Çalışma Süresi}\end{matrix}\right\} = \left\{\begin{matrix}\text{Temel Matrisin}\\ (2,2)\\ \text{Elemanı}\end{matrix}\right\}$$

$$\boxed{= 20 \text{ yıl}}$$

$$T = \begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 3{,}33 & 13{,}33 \\ 0 & 20 \end{bmatrix} \end{array}$$

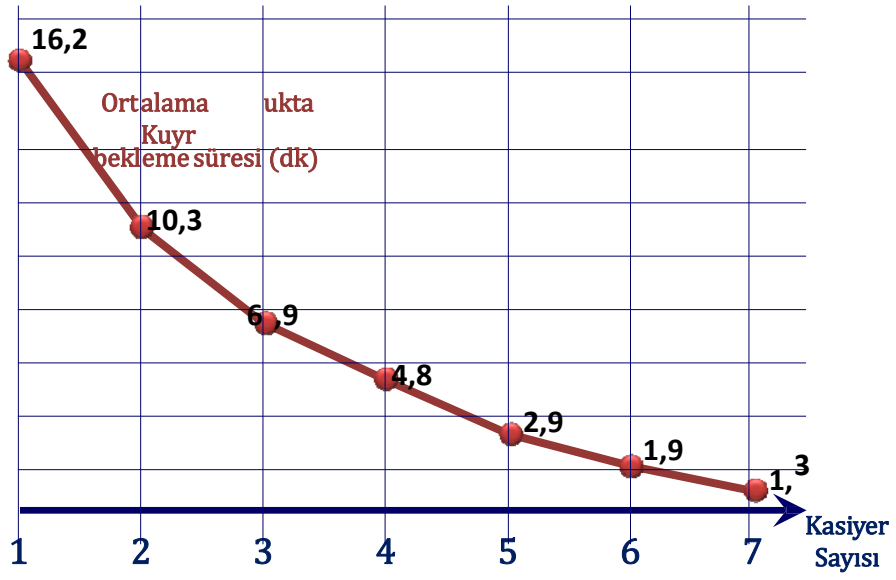
KUYRUK TEORİSİ

YÖNEYLEM ARAŞTIRMASI - III

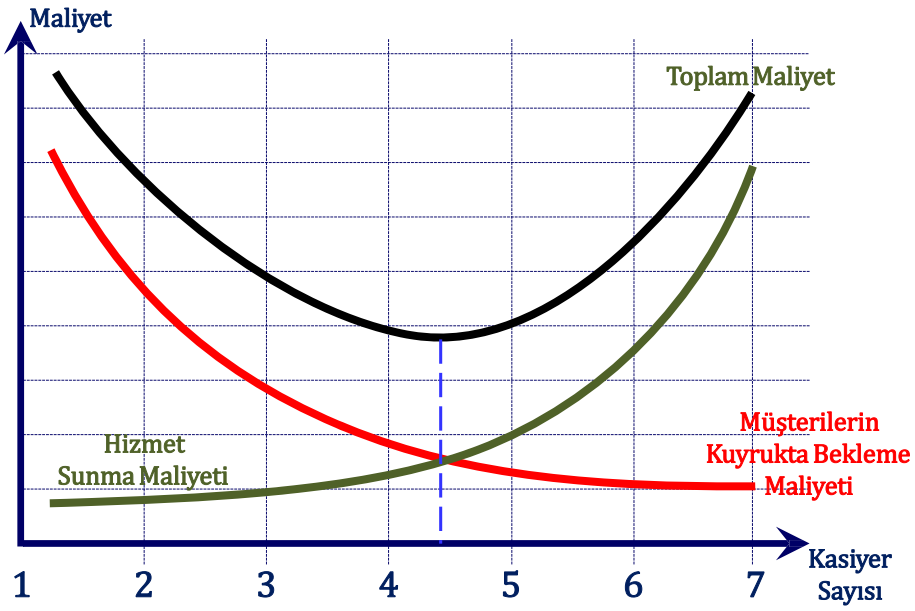
Kuyruk Sistemleri

- Günlük hayatımızın bir parçasıdır.
- Sadece insanlara özgün bir deneyim değildir.
- Kuyrukta beklemenin bir **maliyeti** vardır.

Kuyruk Sistemleri

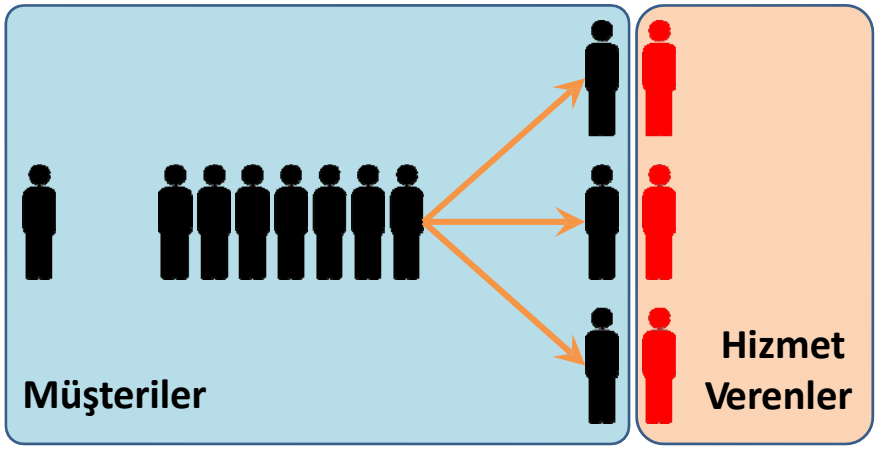


Kuyruk Sistemleri



Kuyruk Modelinin Bileşenleri

- Kuyruk modelinin temel aktörleri;

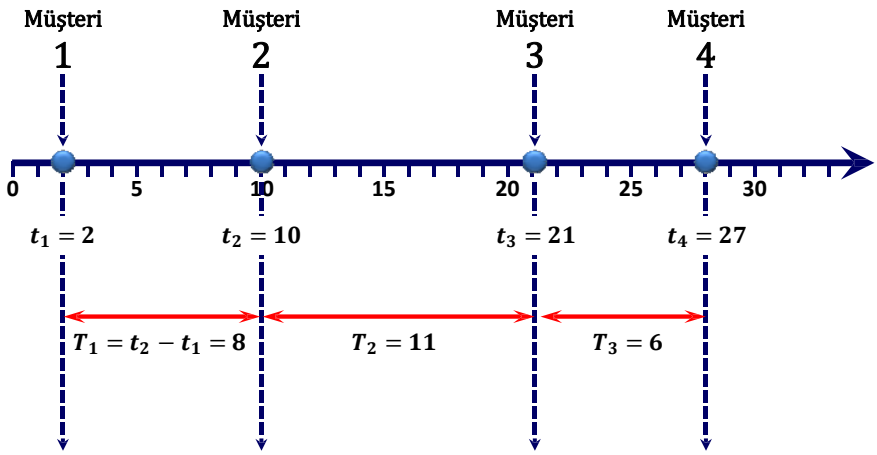


Kuyruk Modelinin Bileşenleri

- Müşteriler;
  - Belirli bir kaynaktan türetilir,
  - hizmet yerine vardığında
    - Hizmet bekleyen kimse yoksa hizmet görür,
    - Tüm hizmet verenler dolu ise kuyruğa girer,
    - Ya da kuyrukta bekleyenlerin sayısı fazla ise kuyruğa girmeden hizmetten vazgeçer
- Hizmet verenler;
  - hizmeti bitirdikçe kuyruktan müşteri çeker
  - kuyrukta bekleyen yok ise boş kalır.

Kuyruk Modelinin Bileşenleri

- Gelişler ve gelişler arası süre;

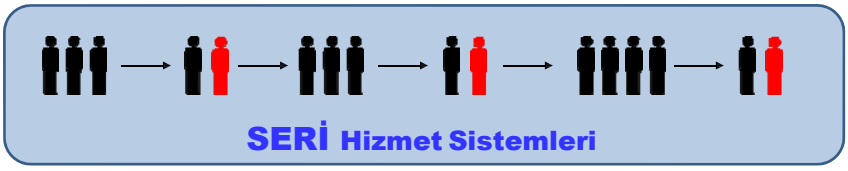


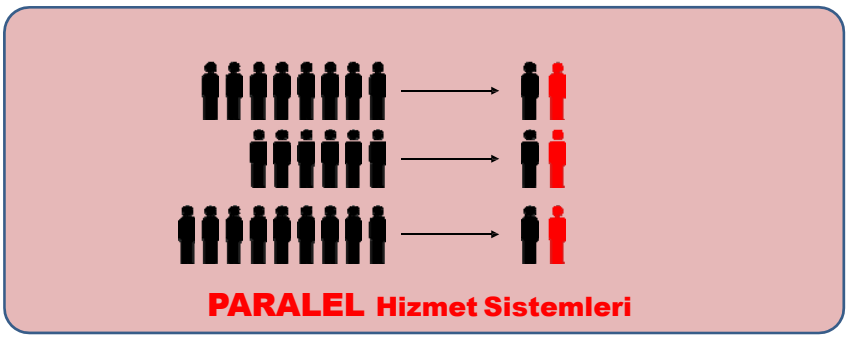
Kuyruk Modelinin Bileşenleri

- Kuyruk Disiplini
  - **İGİHG**: **İ**lk **G**elen **İ**lk **H**izmet **G**örür (FCFS:**F**irst **C**ome **F**irst **S**erved)
    - Banka, Süpermarket, Yemek sırası, vb.
  - **SGİHG**: **S**on **G**elen **İ**lk **H**izmet **G**örür (LCFS:**L**ast **C**ome **F**irst **S**erved)
    - Asansör
  - **RSHV**: **R**astgele **S**ırada **H**izmet **V**erme (SIRO:**S**erved **I**n **R**andom **O**rder)
  - **ÖSHV** : **Ö**ncelik **S**ırasında **H**izmet **V**erme (Priority Rules)
    - Hastaneler

Kuyruk Modelinin Bileşenleri

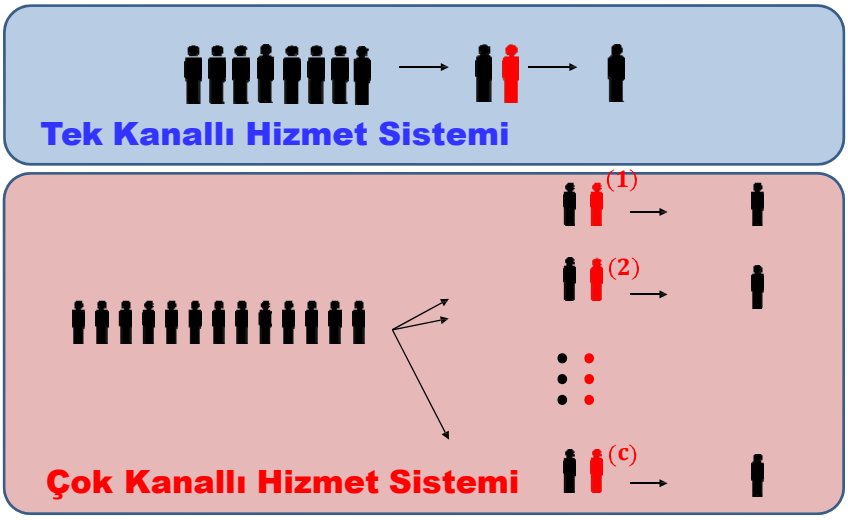
- Hizmet Yeri Tasarımı





Kuyruk Modelinin Bileşenleri

- Hizmet Yeri Tasarımı



Üstel Dağılımın Rolü

- Kuyruk sistemlerinde **gelişler arası sürenin** ve **hizmet sürelerinin** belirli bir olasılık dağılımına uyduğu varsayılır.
- Bu olasılık dağılımı genelde **üstel dağılım** olarak ifade edilir.
- Üstel Dağılımın parametresi olan λ müşterilerin geliş sıklığı olarak tanımlanır ve birim zamanda gelen müşteri sayısı olarak ifade edilir
- Örneğin $\lambda = 5$ müşteri / saat olan bir sisteme
  - **1 saat** içinde ortalama **5 müşteri** gelmekte,
  - müşterilerin gelişleri arasındaki ortalama süre $\frac{1}{5} = 0,2$ **saat** olarak tanımlanır.

Üstel Dağılımın Rolü

- X rassal değişkeni λ parametresi ile üstel dağılıma uyuyorsa;
  - $P(X = t) = f(t) = \lambda e^{-\lambda t}$
  - $P(X > t) = e^{-\lambda t}$
  - $\mathrm{E}[X] = \frac{1}{\lambda}$
  - $\mathrm{var}[X] = \frac{1}{\lambda^2}$

Üstel Dağılımın Rolü

- Üstel dağılımının **unutkanlık** özelliği vardır.
- Yani;

$$\underbrace{P(X > t + s \grave{a} X > t)}_{\substack{X \text{ rassal değişkeninin değerinin} \\ t\text{'den büyük olduğu biliniyor ise} \\ t+s\text{'den de büyük olma olasılığı}}} = \underbrace{P(X > s) = e^{-\lambda s}}_{\substack{X \text{ rassal değişkeninin} \\ \text{değerinin } s\text{'den büyük} \\ \text{olma olasılığı}}}$$

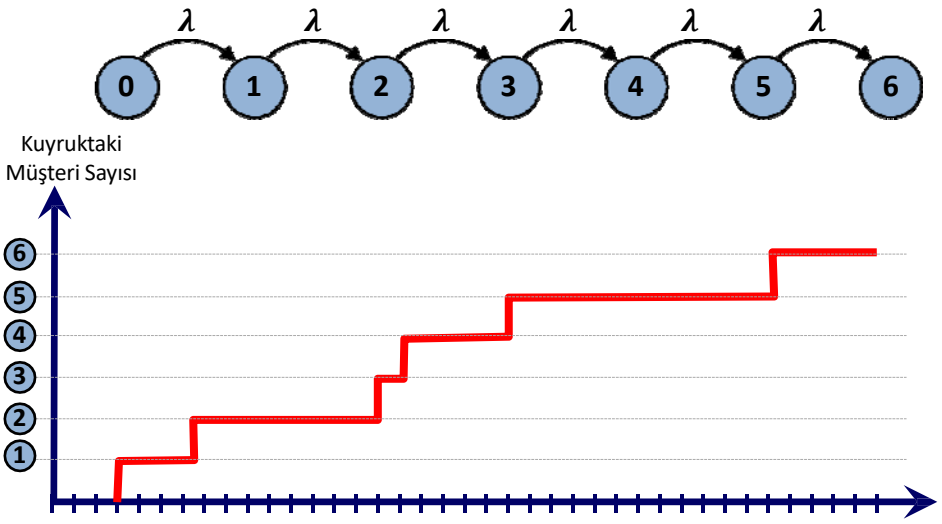
Saf Doğum ve Ölüm Modelleri

- **Saf Doğum Modeli**
  - Hiç hizmetin olmadığı bir kuyruğa, devamlı olarak müşterilerin geldiğini ve kuyruğa katıldıklarını düşünün…
  - Geliş sıklığı λ ile belirtilir
  - Belirli bir t periyodunda
    - gelişlerin ortalaması λt'dir
    - n adet geliş olma olasılığı ise;

$$p_n(t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

Saf Doğum ve Ölüm Modelleri

- **Saf Doğum Modeli**



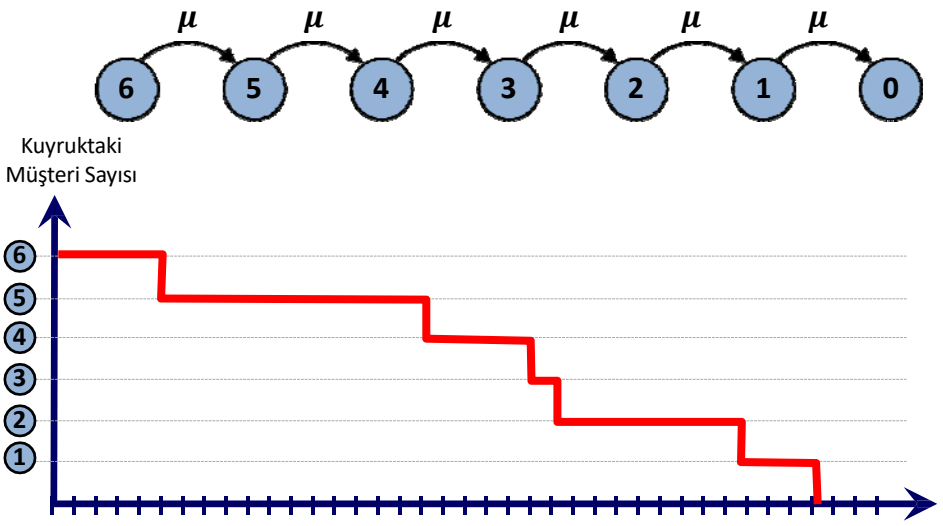
Saf Doğum ve Ölüm Modelleri

- **Saf Ölüm Modeli**
  - Hiç müşterinin gelmediği bir kuyrukta, kuyrukta bekleyen müşterilerin hizmet aldıkça kuyruktan ayrıldıklarını düşünün…
  - Hizmet sıklığı μ ile belirtilir
  - Belirli bir t periyodunda kuyruktaki N adet müşteriden,
    - hizmet alan müşteri ortalaması μt'dir
    - Kuyrukta n adet müşterinin kalma olasılığı ise;

$$p_n(t) = \frac{(\mu t)^{N-n} e^{-\mu t}}{(N-n)!}$$

Saf Doğum ve Ölüm Modelleri

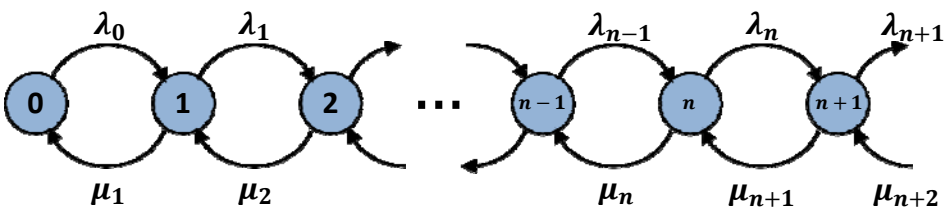
- **Saf Ölüm Modeli**



Genelleştirilmiş Poisson Kuyruk Modeli

- Gelişler arası ve hizmet süreleri **üstel dağılıma** uyar.
- Sistemin **kararlılık durumuna** ulaştığı varsayılır.
- Geliş ve gidiş hızlarının **durum bağımlı** olduğu varsayılır.
- Tanımlamalar;
  - $n =$ sistemdeki müşteri sayısı
  - $\lambda_n =$ sistemdeki n müşterinin geliş hızı
  - $\mu_n =$ sistemdeki n müşterinin gidiş hızı
  - $p_n =$ sistemdeki n müşterinin kararlılık durumu olasılığı
- $\boldsymbol{p_n}$, $\boldsymbol{\lambda_n}$ ve $\boldsymbol{\mu_n}$'nin bir fonksiyonu olarak türetilir.

Genelleştirilmiş Poisson Kuyruk Modeli



DURUM	GELİŞLER = GİDİŞLER
0	$\lambda_0 p_0 = \mu_1 p_1$
1	$\lambda_0 p_0 + \mu_2 p_2 = \lambda_1 p_1 + \mu_1 p_1$
⋮	⋮
n	$\lambda_{n-1} p_{n-1} + \mu_{n+1} p_{n+1} = \lambda_n p_n + \mu_n p_n$

Genelleştirilmiş Poisson Kuyruk Modeli

- Durum **0** için $\lambda_0 p_0 = \mu_1 p_1 \Rightarrow p_1 = \left(\frac{\lambda_0}{\mu}\right) p_0$
- Durum **1** için $\lambda_0 p_0 + \mu_2 p_2 = (\lambda_1 + \mu_1) p_1$

$$\lambda_0 p_0 + \mu_2 p_2 = (\lambda_1 + \mu_1)\left(\frac{\lambda_0}{\mu}\right) p_0$$

$$\lambda_0 p_0 + \mu_2 p_2 = \left(\frac{\lambda_0 \lambda_1}{\mu} + \lambda_0\right) p_0$$

$$\mu_2 p_2 = \left(\frac{\lambda_0 \lambda_1}{\mu_1} + \lambda_0\right) p_0 - \lambda_0 p_0$$

$$p_2 = \left(\frac{\lambda_0 \lambda_1}{\mu_1 \mu_2}\right) p_0$$

- Durum **n** için $p_n = \left(\frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n}\right) p_0$

∞ ∞ ∞

$\boldsymbol{p_n = p_0 +}$ $\boldsymbol{p_n = 1 \Rightarrow p_0 = 1 -}$

$\boldsymbol{p_n}$

Özel Poisson Kuyruk Modelleri

- Kendall – Lee Notasyonu

(a / b / c) : (d / e / f)

Geliş Dağılımlarını Tanımlar

- ***M***: Gelişler arası süre üssel dağılıma uymaktadır.
- ***D*** : Gelişler arası süre sabit süre aralıklarındadır.
- $\boldsymbol{E_k}$: Gelişler arası süre Erlang veya Gamma dağılımına uymaktadır
- ***GI*** : Gelişler arası süre genel bir dağılıma uymaktadır.

Özel Poisson Kuyruk Modelleri

- Kendall – Lee Notasyonu

$$(a / b / c) : (d / e / f)$$

Gidiş (Hizmet Süresi) Dağılımlarını Tanımlar

- ***M***: Hizmet süresi üssel dağılıma uymaktadır.
- ***D*** : Hizmet süresi sabittir.
- E_k : Hizmet süresi Erlang veya Gamma dağılımına uymaktadır
- ***G*** : Hizmet süresi genel bir dağılıma uymaktadır.

Özel Poisson Kuyruk Modelleri

- Kendall – Lee Notasyonu

$$(a / b / c) : (d / e / f)$$

Paralel hizmet verenlerin sayısı

- $c = 1,2,\ldots,\infty$

Özel Poisson Kuyruk Modelleri

- Kendall – Lee Notasyonu

(a / b / c) : (d / e / f)

Kuyruk Disiplini
- **İGİHG (FIFO)**
- **SGİHG (LIFO)**
- **RSHG (SIRO)**
- **GD**

Özel Poisson Kuyruk Modelleri

- Kendall – Lee Notasyonu

(a / b / c) : (d / e / f)

Sistemde (kuyruk+hizmet) izin verilen maksimum sayı

Özel Poisson Kuyruk Modelleri

- Kendall – Lee Notasyonu

$$(a/b/c):(d/e/f)$$

İstek kaynağının büyüklüğü

Özel Poisson Kuyruk Modelleri

$$(M/D/10):(GD/N/\infty)$$

- Gelişler arası sürenin **üstel**,
- Hizmet süresinin **sabit**,
- **10** paralel hizmet verenin olduğu,
- Kuyrukta **genel disiplinin** uygulandığı
- Sistem içinde **en fazla** N müşteriye müsaade edilen,
- Müşterilerin **sınırsız kaynaktan** geldiği bilinen bir sistem tarif edilmiştir.

Kararlılık Durumu Performans Ölçütleri

L_s	Sistemdeki Beklenen Müşteri Sayısı
L_q	Kuyruktaki Beklenen Müşteri Sayısı
W_s	Sistemdeki Beklenen Bekleme Zamanı
W_q	Kuyrukta Beklenen Bekleme Zamanı
c	Meşgul Durumdaki Hizmet Verenlerin Beklenen Sayısı
p_n	Sistemde n müşteri olma olasılığı
$\boldsymbol{\rho}$	Sistem Kullanım Faktörü

Kararlılık Durumu Performans Ölçütleri

Sistemdeki Beklenen Müşteri Sayısı

$$L_s = \sum_{n=1}^{\infty} n p_n$$

$$L_s = \lambda_{eff} W_s$$

Kuyruktaki Beklenen Müşteri Sayısı

$$L_q = \sum_{n=c+1}^{\infty} (n-c) p_n$$

$$L_q = \lambda_{eff} W_q$$

Little'ın Formülü

Kararlılık Durumu Performans Ölçütleri

$$\begin{pmatrix}\text{Sistemdeki}\\ \text{Beklenen}\\ \text{Bekleme Süresi}\end{pmatrix} = \begin{pmatrix}\textbf{Kuyruktaki}\\ \text{Beklenen}\\ \text{Bekleme Süresi}\end{pmatrix} + \begin{pmatrix}\text{Beklenen}\\ \textbf{Hizmet}\\ \text{Süresi}\end{pmatrix}$$

$$W_s = W_q + \frac{1}{\mu}$$

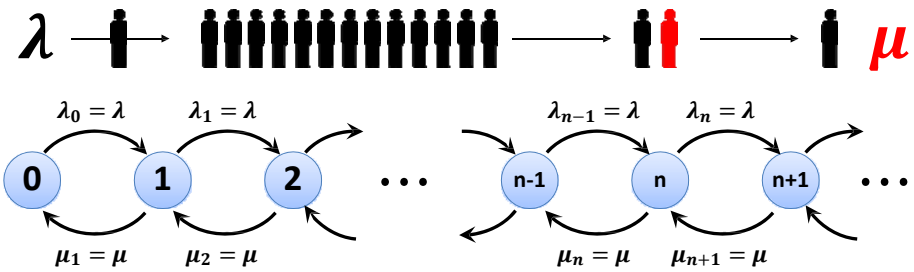
$$W_s\lambda_{eff} = W_q\lambda_{eff} + \frac{\lambda_{eff}}{\mu}$$

$$L_s = L_q + \frac{\lambda_{eff}}{\mu}$$

$$c(0) = L_s - L_q \quad \frac{\lambda_{eff}}{\mu}$$

$$= \text{Hizmet verenlerin kullanımı} = \frac{c}{(0)}$$

Tek Kanallı Modeller (M/M/1):(GD/∞/∞)



$$p_n = \left(\frac{\lambda_0\lambda_1\cdots\lambda_{n-1}}{\mu_1\mu_2\cdots\mu_n}\right)p_0 \quad \Rightarrow \quad p_n = \left(\frac{\lambda}{\mu}\right)^n p_0$$

$$\rho = \frac{\lambda}{\mu} \quad \Rightarrow \quad p_n = \rho^n p_0$$

$$p_0\left(1+\rho+\rho^2+\cdots\right) = 1 \Rightarrow \quad p_0 = 1-\rho$$

$$\boxed{p_n = (1-\rho)\rho^n}$$

Tek Kanallı Modeller
(M/M/1):(GD/∞/∞)

$$L_s = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}$$

$$W_s = \frac{L_s}{\lambda} = \frac{1}{\mu(1-\rho)}$$

$$L_q = \frac{\lambda}{\mu-\lambda} - \frac{\lambda}{\mu} = \frac{\lambda^2}{\mu(\mu-\lambda)} = \frac{\rho^2}{1-\rho}$$
$$= L_s - \rho = \lambda W_q$$

$$c\text{п}\text{ъ} = L_s \quad - L_q \quad = \rho$$

Tek Kanallı Modeller
(M/M/1):(FCFS/∞/∞) için bekleme süresi dağılımı

$\boldsymbol{\tau}$, yeni gelen bir kişinin sistemde beklemek zorunda olduğu süre olarak tanımlanırsa;

(gelen müşterinin önünde n müşteri varsa)

$$\boldsymbol{\tau} = \boldsymbol{t_1'} + \boldsymbol{t_2} + \cdots + \boldsymbol{t_n} + \boldsymbol{t_{n+1}}$$

$\boldsymbol{w}(\boldsymbol{\tau}\grave{\grave{a}}\boldsymbol{n}+\mathbf{1}) =$ gelen müşterinin önünde n müşteri olması durumunda, τ'nun koşullu yoğunluk fonksiyonu

$$\boldsymbol{w}(\boldsymbol{\tau}) = \sigma_{n=0}^{\infty} \boldsymbol{w}(\boldsymbol{\tau}\grave{\grave{a}}\boldsymbol{n}+\mathbf{1})p_n = \sigma_{n=0}^{\infty} \frac{\mu(\mu t)^n e^{-\mu t}}{n!}(1-\rho)\rho^n = \mu(1-\rho)e^{-\mu(1-\rho)\tau}$$

$$\mathrm{E}[\boldsymbol{w}(\boldsymbol{\tau})] = \frac{1}{\mu(1-\rho)}$$

Tek Kanallı Modeller (M/M/1):(GD/∞/∞)

- Tek kanallı bir sisteme her 6 dakikada 1 müşteri hizmet istemi ile gelmektedir. Sistemde müşteri başına ortalama servis zamanı 5 dakika olduğuna göre belirtilen varsayımların geçerliliği durumunda
-
- $\lambda = \frac{1}{6} = 0{,}1666 \quad dakikada\ geliş\ oranı$ (müşteri/dk)
- ya da $\lambda = \frac{1}{6}60 = 10 \quad saatte\ müşteri\ geliş\ oranı$ (müşteri/saat)
-
- $\mu = \frac{1}{5} = 0{,}2 \quad dakikada\ servis\ oranı$ (müşteri/dk)
- ya da $\mu = \frac{1}{5}60 = 12 \quad saate\ servis\ oranı$ (müşteri/saat)

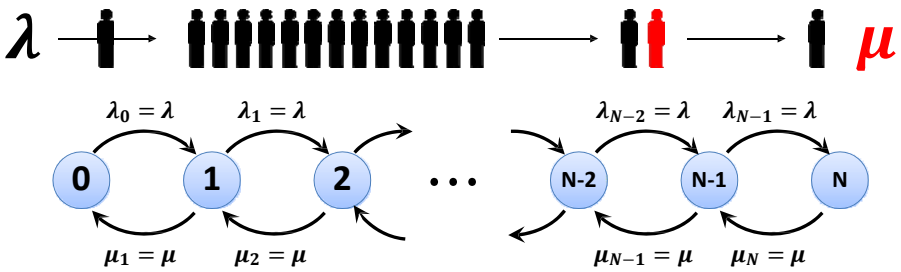
Tek Kanallı Modeller (M/M/1):(GD/∞/∞)

- Sistemde olması beklenen müşteri sayısı (L_s) :
- $L_s = \frac{\lambda}{\mu-\lambda} = \frac{0{,}1666}{0{,}2-0{,}1666} = 5\ müşteri\ olur$
- ya da
- $L_s = \frac{}{\mu-\lambda} = \frac{10}{12-10} = 5\ müşteri\ olur$
- Sırada (Kuyrukta) olması beklenen müşteri sayısı (Lq) :
-
- $L = \frac{\lambda^2}{\mu(\mu-\lambda)} = \frac{10^2}{12(12-10)} = 4 \quad müşteri$
-

Tek Kanallı Modeller (M/M/1):(GD/∞/∞)

- Müşteri başına sistemde geçen ortalama zaman (W):
- $W_s = \frac{1}{\mu -} = \frac{1}{12-10} = 0{,}5\ saat\ yani\ 30dak.\ dır$ ya da
- $W_s = \frac{1}{\mu -} = \frac{1}{0{,}2-0{,}1666} = 30\ dak.\ dır$
-
- Müşteri başına sırada (kuyrukta) geçen ortalama zaman (Wq), (yani hizmet edilmeden önce kuyrukta beklediği zaman):
- $W = \frac{\lambda}{\mu\,(\mu - \lambda} = \frac{10}{12(12-10)} = 0{,}4166\ saat$ yada,
- $W = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{0{,}1666}{0{,}2(0{,}2-0{,}1666)} = 25\ dak$

Tek Kanallı Modeller (M/M/1):(GD/N/∞)



$$p_n = \begin{cases} \rho^n p_0 & n \le N \\ 0 & n > N \end{cases}$$

$$p_0(1 + \rho + \cdots + \rho^n) = 1$$

$$p_n = \begin{cases} \dfrac{(1-\rho)\rho^n}{1-\rho^{N+1}} & \rho \neq 1 \\ \dfrac{1}{N+1} & \rho = 1 \end{cases}$$

Tek Kanallı Modeller (M/M/1):(GD/N/∞)

$$P_0 = \begin{cases} \left(\dfrac{1-\rho}{1-\rho^{N+1}}\right) & \rho \neq 1 \; veya \; \lambda \neq \mu \\ \left(\dfrac{1}{N+1}\right) & \rho = 1 \end{cases}$$

Tek Kanallı Modeller (M/M/1):(GD/N/∞)

- **Örnek:**
- Bir tek-kişilik bir berber dükkânında 10 koltuk bulunuyor. Varışlar arası süreler üstel dağılıma uymakta ve ortalama olarak her saat 20 potansiyel müşteri berber dükkânına varmaktadır. Dükkânı dolu bulan müşteriler geri dönmektedir. Berberin her müşterinin saçını kesmesi ortalama 12 dakika almaktadır. Saç kesme süreleri üstel dağılıma uymaktadır.
  - Ortalama olarak berber saatte kaç adet saç kesimi yapabilmektedir?
  - Ortalama olarak dükkâna giren bir müşterinin dükkânda harcayacağı süre ne kadardır?

Tek Kanallı Modeller (M/M/1):(GD/N/∞)

- Tüm gelenlerin P_{10} oranı dükkânı dolu bulacaktır. Böylece, her saat ortalama $\lambda(1-P_{10})$ oranında kişi dükkâna girecektir. Tüm giren müşteriler saçlarını kestirecek, böylece berber saat başına ortalama $\lambda(1-P_{10})$ saç kesimi hizmeti verecektir. Problemimizde, N=10, λ =saatte 20 ortalama müşteri ve μ=saat başına 5 müşteri olduğunu görüyoruz. Buradan ρ=20/5= 4

$$P_0 = \left(\frac{1-\rho}{1-\rho^{N+1}}\right) = \frac{1-4}{1-4^{11}}$$

$$P_{10} = \left(\frac{1-\rho}{1-\rho^{N+1}}\right)P^n = 4^{10}\left(\frac{1-4}{1-4^{11}}\right) = \frac{-3*(4^{10})}{1-4^{11}} = 0{,}75$$

Böylece, saatte ortalama 20*(1-3/4) =5 müşteri saç kesimi hizmeti alacaktır. Bu durum saatte ortalama 20-5=15 potansiyel müşterinin dükkâna girmeyeceği anlamına gelmektedir.

Tek Kanallı Modeller (M/M/1):(GD/N/∞)

- Dükkana giren bir müşterinin dükkanda harcayacağı süre Ws'nin belirlenmesi için Ls formülünden yararlanılır:

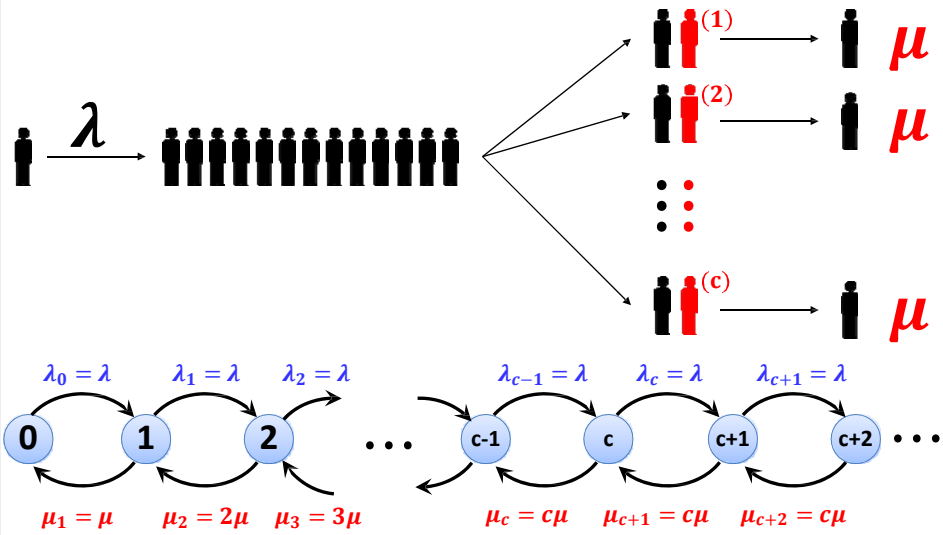
$$L_s = \frac{4\left[1-11(4^{10})+10(4^{11})\right]}{(1-4)(1-4^{11})} = 9{,}67$$ Müşteri hesaplanır

- Ws formülünde Ls yerine konulursa;:

$$W_s = \frac{L_S}{\lambda(1-P_N)} = \frac{9{,}67}{20(1-0{,}75)} = 1{,}93$$ Saat olarak bulunur

- Çalışma sonunda; bu berber dükkanının kalabalık olduğu görülmekte ve en az bir berber daha işe alınması tavsiye edilmektedir.

Çok Kanallı Modeller (M/M/c):(GD/ ∞ /∞)



Çok Kanallı Modeller (M/M/c):(GD/ ∞ /∞)

$$p_n = \begin{cases} \dfrac{\lambda^n}{\mu(2\mu)(3\mu)\cdots(n\mu)} p_0 = \dfrac{\lambda^n}{n!\,\mu^n} p_0 & n \le c \\ \dfrac{\lambda^n}{\mu(2\mu)(3\mu)\cdots\big((c-1)\mu\big)(c\mu)^{n-c+1}} p_0 = \dfrac{\lambda^n}{c!\,c^{n-c}\mu^n} p_0 & n > c \end{cases}$$

$$p_0 = \left\{ \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} \sum_{n=c}^{\infty} \left(\frac{\rho}{c}\right)^{n-} \right\}^{-1} = \left\{ \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} \left(\frac{1}{1-\frac{\rho}{c}} \right) \right\}^{-1}$$

Çok Kanallı Modeller (M/M/c):(GD/ ∞ /∞)

Kuyruk sisteminde n sayıda müşteri bulunma olasılığı

$$P_n = \begin{cases} (c\rho)^n \left(\dfrac{P_0}{n!}\right) & n < c \\ (c\rho)^n \left(\dfrac{P_0}{c!\, c^{n-c}}\right) & n \geq c \end{cases}$$

ρ≥1 ise servis oranı'dir. Bu durumda tüm servisler meşgul olacaktır. Tüm servislerin doluluğu;

$$P(n \geq c) = \frac{(c\rho)^c P_0}{c!\,(1-\rho)}$$

Aşağıdaki tabloda n≥c değerleri için durgun durum olasılıkları verilmektedir.
Tablo: M/M/c/GD/∞/∞/ Kuyruk Sistemi için P(n≥ c) değerleri

ρ	c=2	c=3	c=4	c=5	c=6	c=7
.10	.02	.00	.00	.00	.00	.00
.20	.07	.02	.00	.00	.00	.00
.30	.14	.07	.04	.02	.01	.00
.40	.23	.14	.09	.06	.04	.03
.50	.33	.24	.17	.13	.10	.08
.55	.39	.29	.23	.18	.14	.11
.60	.45	.35	.29	.24	.20	.17
.65	.51	.42	.35	.30	.26	.21
.70	.57	.51	.43	.38	.34	.30
.75	.64	.57	.51	.46	.42	.39
.80	.71	.65	.60	.55	.52	.49
.85	.78	.73	.69	.65	.62	.60
.90	.85	.83	.79	.76	.74	.72
.95	.92	.91	.89	.88	.87	.85

Çok Kanallı Modeller (M/M/c):(GD/ ∞ /∞)

$$L_q = \sum_{n=c}^{\infty} (n-c)p_n = \frac{\rho^{c+1}}{(c-1)(c-p)^2}p_0 \quad \Rightarrow \quad L_s = L_q + \rho$$

Neden toplam c'den başlar?

L_q ve L_s değerler, bulunduktan sonra $W_q = \frac{L_q}{\lambda}$ ve $W_s = \frac{L_s}{\lambda}$ olur.

NEDEN ?

- Bekleme hattındaki ortalama müşteri sayısı (ortalama kuyruk uzunluğu)

$$L_q = \frac{P(n \geq c)\rho}{1-\rho} = \frac{\frac{(c\rho)^c P_0}{c!(1-\rho}}{1-\rho} = \frac{(c\rho)^c P_0}{c!(1-\rho)^2}$$

- Sistemdeki ortalama müşteri sayısı;

$$L_S = L_q + \frac{\lambda}{}$$

- Sisteme gelen herhangi bir müşterinin harcadığı ortalama süre

$$W_q = \frac{L}{\lambda}$$

- Kuyruk sistemine gelen müşterilerin harcadığı ortalama süre

$$W_s = \frac{L_s}{\lambda} = \frac{L_q}{\lambda} + \frac{1}{\mu} = W_q + \frac{1}{\mu} = \frac{P(n \geq c)}{c\mu - \lambda} + \frac{1}{\mu}$$

- **Örnek:**
- İki veznedarın olduğu bir bankaya saatte ortalama 80 müşteri geliş yapmakta ve veznelerden birinin boşalması için tek bir hatta beklemektedir. Bir müşterinin hizmet alması için gereken ortalama süre 1,2 dakikadır. Varışlar arası süreler ve servis sürelerin üstel dağılıma uyduğunu varsayalım. Buna göre;
  - Bankada mevcut müşterilerin beklenen sayısı nedir?
  - Bir müşterinin bankada harcadığı zamanın beklenen uzunluğu nedir?
  - Belirli bir veznenin boş olduğu zamanın oranı nedir?

Elimizdeki sistem saatte ortalama λ=80 müşteriye ve saatte ortalama μ=50 müşteriye sahip bir M/M/2/GD/∞/∞/ kuyruk modelidir. Böylece

$\rho = \frac{80}{2(50)} = 0.80 \langle 1$, olduğundan durağan bir denge durumu mevcuttur. (λ=100 için bir denge durumu mevcut değildir.) Olasılık tablosundan P(j≥ 2)= 0.71 olarak belirlenip

$L_q = \frac{\rho^{c+1}}{(c-1)(c-\rho)^2} p_0 = \frac{P(j \geq c)\rho}{1-\rho}$ formülünde yerine konursa;

$L_q = \frac{P(j \geq 2)\rho}{1-\rho} = \frac{0.71 * 0.80}{1-0.80} = 2.84$ müşteri kuyrukta bekleyen müşteri sayısıdır.

Buradan da $L_s = L_q + \frac{\lambda}{\mu}$ formülünde yerine değerleri koyduğumuzda;

$L_s = 2.84 + \frac{80}{50} = 4.44$ müşteri bankada mevcut müşteri sayısını vermektedir.

Ws bankada bekleyen müşteri sayısını verdiğinden

$W_s = \frac{L_s}{\lambda} = \frac{4.44}{80} = 0.055$ saat =3.3 dak. beklendiği görülmektedir.

Bir veznenin boş olmasının olasılığı $P_0+0.5P_1$ olarak verilmektedir. P(j≥2)=0,71 olması durumunu kullanarak

$P(j \geq c) = \frac{(c\rho)^c p_0}{c!(1-\rho)}$ özelliğinden P_0'ı elde ederiz.

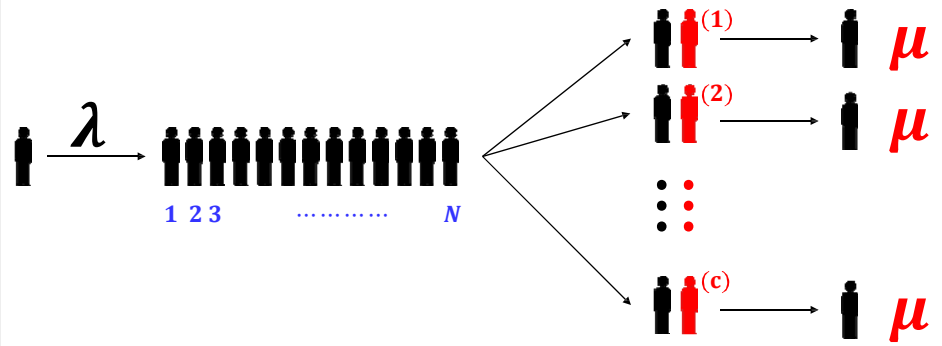
$$p_0 = \frac{P(j \geq c)c!(1-\rho)}{(c\rho)^c} = \frac{(0.71)2!(1-0.80)}{(1.6)^2} = 0.11$$

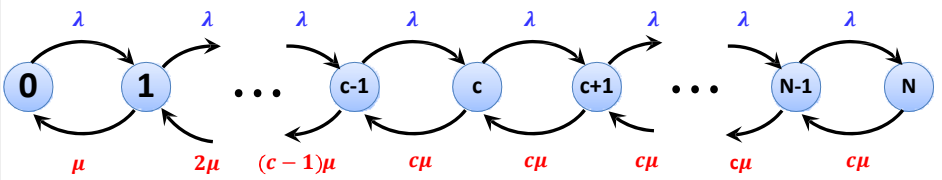
$p_j = \frac{(c\rho)^j p_0}{j!}$ formülündeki ilişkiden yararlanarak;

$$p_1 = \frac{(2*0.8)^1 p_0}{1!} = 0.176$$

Bu ilişkilerden elde edilen sonuçlardan, bir veznenin boş kalma oranı $P_0+0.5P_1$ =0.11+0.5(0.176)= 0.198 olarak bulunmuştur.

Çok Kanallı Modeller
(M/M/c):(GD/ ∞ /N) $c \leq N$





Çok Kanallı Modeller
(M/M/c):(GD/ ∞ /N) $c \leq N$

$$P_0 = \frac{1}{\sigma_{n=0}^{c-1} \frac{N!}{(N-n)!\,n!} (\rho)^n + \sigma_{n=c}^{n=N} \frac{N!}{(N-n)!\,c!\,c^{n-c}} (\rho)^n}$$

$$P_n = \begin{cases} \frac{P_0 N!}{(N-n)!\,n!} (\rho)^n & 0 \leq n \leq c \\ \frac{P_0 N!}{(N-n)!\,c!\,c^{n-c}} (\rho)^n & c \leq n \leq N \end{cases}$$

$$L_s = \sum^{c-1} nP_n + \sum^{N} (\quad)n - c \quad P_n + c(1 - \sum P_n)$$

- **Örnek:**
-
- Ekonometri bölümü öğrencileri gruplar halinde iki terminali bulunan bilgi işlem merkezinde proje hesaplarını yaptırmaktadır. Hesaplama işi ortalama olarak terminal zamanının 30 dakikasını almaktadır. Her öğrencinin hesapları için terminali kullanması 90 dakikada bir olmakta, yani her bir öğrencinin ortalama servislere gelişleri arası 90 dakikadır. Servise geliş süresi ve servis süresi üstel dağılımlıdır. Bir grupta 3 öğrenci vardır. Buna göre:
  - Her iki terminalin aylak kalma olasılığını
  - Üç öğrencinin tümünün sistemde olma olasılığını
  - Her iki terminalin kullanma olasılığını bulunuz.

μ = 60/30 = 2 iş/saat λ = 60/90 = 2/3 N=3 c=2 ρ=λ/μ=1/3

a)$P_0 = \frac{1}{\sigma_{n=0}^{1} \frac{3!}{(3-n)!n!}(1/3)^n + \sigma_{n=2}^{3} \frac{3!}{(3-n)!2!2^{n-2}} (1/3)^n}$

P_o=0,42

b)$P_3 = P_0 \left(\frac{3!}{0!2!2}\right)\left(\frac{1}{3}\right)^3 = 0.02$

c)Her iki terminalin kullanımda olma olasılığı (P_2 + P_3) dir.

$P_2 = P_0 \left(\frac{3!}{1!2!}\right)\left(\frac{1}{3}\right)^2 = 0.14$

P_2 + P_3 = 0.14 + 0.02 = 0.16