

2017-2018 Güz Yarıyılı

EMM4131

# Popülasyon Temelli Algoritmalar

(Population-based Algorithms)

4

**Genetik Algoritma – Örnek Uygulamalar**  
(Sırt Çantası Problemi, Sınav Programı Çizelgeleme)

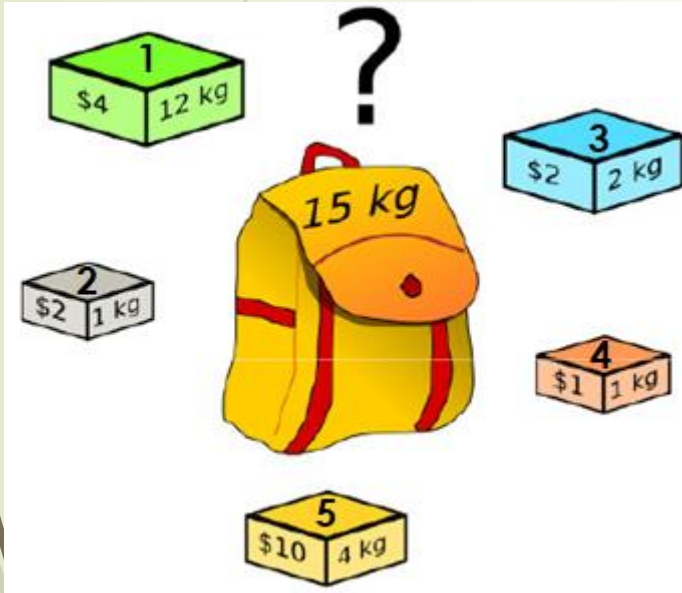
Yrd. Doç. Dr. İbrahim KÜÇÜKKOÇ

Web: <http://ikucukkoc.baun.edu.tr>

Email: [ikucukkoc@balikesir.edu.tr](mailto:ikucukkoc@balikesir.edu.tr)

# Sırt Çantası Problemi – Örnek 1

Aşağıda verilen sırt çantası (knapsack) problemini, kapasite kısıtını sağlarken faydayı maksimize etmek amacıyla çözelim:



$n$ : eşya sayısı ( $i = 1, 2, \dots, n$ )

$w_i$ :  $i$  eşyasının ağırlığı (kg)

$v_i$ :  $i$  eşyasının faydası (\$)

$W$ : Çantanın kapasitesi (kg)

$X_i$ : 1, eğer  $i$  eşyası alınmışsa;  
0, alınmamışsa

$$\max \sum_i v_i * X_i, \quad i = 1, 2, \dots, n$$

Kapasite Kısıtı: 
$$\sum_i w_i * X_i \leq W$$

İşaret Kısıtı: 
$$X_i \in \{0, 1\}$$

## Parametreler:

- Popülasyon büyüklüğü: 10
- Çaprazlama oranı (Cr): 0.2
- Mutasyon oranı (Mr): 0.1
- Seçilim stratejisi: Turnuva
- Turnuva büyüklüğü (t): 3
- Yeni popülasyonu oluşturma stratejisi: En kötüyü değiştir.
- Not: Herhangi bir aşamada oluşan uygunsuz çözümler (başlangıç popülasyonu dahil) tamir edilmektedir.
- 1 iterasyon için GA'nın aşamaları, takip eden fotoğraflarda sunulmuştur.

Popülasyon:

Pop. Büy. = 10

	1	0	0	0	1	1	:	Fitness	Toplam Ağ.
1	0	0	0	1	1		:	11	5 ✓
2	0	0	1	0	1		:	12	6 ✓
3	0	1	0	1	0		:	3	2 ✓
4	1	1	1	0	0		:	8	15 ✓
5	0	1	1	0	1		:	14	7 ✓
6	1	1	0	0	<del>0</del>		:	6	<del>13</del> ✓
7	0	1	0	1	1		:	13	6 ✓
8	1	0	0	0	0		:	4	12 ✓
9	1	0	1	0	<del>0</del>		:	6	<del>14</del> ✓
10	0	0	1	1	1		:	13	7 ✓

t. turnuva büyüklüğü = 3

pop. büyüklüğü  $\times$   $C_r = 0.2 \times 10 = 2$   $\rightarrow$  2 tane crossover uygulanacak

Seçilim  $\rightarrow$  Turnuva 1  
 K6: 6  
 K7: 13  
 K8: 4

Elbanyla

K7	0	1	0	1	1
K5	0	1	1	0	1

Turnuva 2  
 K2: 12  
 K3: 3  
 K5: 14

2. crossover  $\rightarrow$  K5: 01101  
 K7: 01011

Turnuva 2  
 K2: 12  
 K7: 13  
 K9: 6

Crossover oranı 20%

1. crossover bitti

2. crossover bitti



	Fitness	Toplam Ağ
1 1 :	11	5 ✓
0 1 :	12	6 ✓
1 0 :	3	2 ✓
0 0 :	8	15 ✓
0 1 :	14	7 ✓
0 <del>1</del> :	6	<del>13</del> ✓
1 1 :	13	6 ✓
0 0 :	4	12 ✓
0 <del>0</del> :	6	<del>14</del> ✓
1 :	13	7 ✓

t. turnuva büyüklüğü = 5  
 Gapsızlama oranı %20 (Cr)

pop büyü x Cr = 0.2 x 10 = 2 → 2 best gapsızlama yapılıcak  
 Ebünyölü Gapsızlama

Seçilim →

Turnuva 1

- K6: 6
- K7: 13\*
- K8: 4

Turnuva 2

- K2: 12
- K3: 3
- K5: 14\*

	1	2	3	4
K7:	0	1	0	1
K5:	0	1	1	0

	Fitness	Toplam Ağ
C1: 01001	12	5
C2: 01111	15	8

2. Gapsızlama → K5 011101  
 Turnuva 1: K2: 12, K5: 14\*, K7: 13  
 Turnuva 2: K2: 12, K7: 13\*, K9: 6

C1: 01111 15 8  
 C2: 01001 12 5

1. Gapsızlama bitt.

2. Gapsızlama bitt.

Handwritten notes on the right side of the board, including fitness values and tournament results.

Mutasyon Oranı: %10  
(Mr)

Mutasyon Oranı X Pop. Büy. = 1 adet mutasyon

Turnuva. K1\*  
K6  
K9

K1: 00011

r<sub>1</sub>: 3

r<sub>2</sub>: 1

M1: 00111      Fitness    Top. Ağ.  
                         13            7

↓ mutasyon

Mutasyon bitti

Yeni Biraylar:

		<u>Fitness</u>	<u>Toplam Ağ.</u>
Y1*	0 1 0 0 1	12	5
Y2	0 1 1 1 1	15	8
Y3	0 0 1 1 1	13	7

En kötüyü değiştir:

Y1 → pop. en kötü K3: 3 → Karar: değiştir

Y2 → pop. en kötü K8: 4 → Karar: değiştir.

Y3 → popülasyonda aynı bireyden olduğu için değiştirme yapılmıyor.

Popülasyon:

Pop. Büy. = 10

						Fitness	Toplam Ağ
1	0	0	0	1	1	11	5
2	0	0	1	0	1	12	6
3	0	1	0	0	1	12	5
4	1	1	1	0	0	8	15
5	0	1	1	0	1	14	7
6	1	1	0	0	0	6	13
7	0	1	0	1	1	13	6
8	0	1	1	1	1	15	8
9	1	0	1	0	0	6	14
10	0	0	1	1	1	13	7

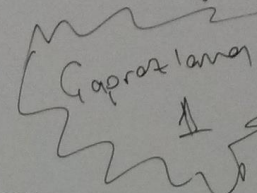
- 1 iterasyon sonrasında:
- Popülasyonda 2 birey değişti.
- En iyi fitness değeri 14'ten 15'e yükseldi.
- Ortalama fitness değeri 9'dan 11'e yükseldi.
- Bu iterasyonlar devam ettirilerek aşağıdaki durdurma kriterlerinden birisi sağlanınca algoritma sonlandırılabilir:
  - Maksimum iterasyon sayısı
  - İyileşme olmaksızın maksimum iterasyon sayısı

**SON...**

**Aynı problem için bir başka başlangıç popülasyonu ve bir iterasyonluk ilerleme**  
**(Not: İlerleme sonrası yeni popülasyon başlangıç popülasyonunun üzerinde düzeltilmiştir!)**

Başlangıç popülasyonu	Fitness	Top. Ağ.
K1. 0 0 1 0 1	12	6
K2. 1 0 1 1 0	7	15
K3. 1 0 0 1 0	5	13
K4. 1 1 0 0 0	6	13
K5. 0 0 0 1 1	11	5
K6. 0 1 1 0 1	14	7
K7. 1 1 1 0 0	8	15
K12 0 1 0 1 1	13	6
K11 0 1 1 0 0	4	3
K10 1 0 1 0 0	6	14

Turnuva Büyüklüğü (+) = 3



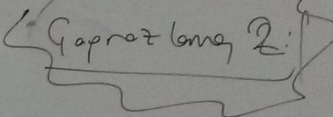
Ab1: 1 1 1 0 0  
 Eb2: 0 1 1 0 1

Turnuva 1:

K7: 8\*  
 K2: 7  
 K4: 6

Turnuva 2:

K8: 3  
 K5: 11  
 K6: 14\*



Turnuva 1:  
 K3: 5  
 K6: 14\*  
 K10: 6

Turnuva 2:

K1: 12\*  
 K5: 11  
 K10: 6

Çaprazlama Oranı: %20

Çaprazlama sayısı: %20 · Pop Büy = 2 defa çap.

	Fitness	Top. Ağ.
C1: 1 1 1 0 0	8	15
C2: 0 1 1 0 0	4	3

Eb1: 0 1 1 0 1  
 Eb2: 0 0 1 0 1

Çaprazlama noktası = 1

	Fitness	Top. Ağ.
C1: 0 0 1 0 1	12	6
C2: 0 1 1 0 1	14	7

1. iterasyon sonunda En iyi fitness: 14 (aynı kaldı)

Fort: 7.3 den 8.6'ya çıktı.

0/20

0. Pop Büy = 2 defa c. aprazlama.

Fitness	Top. Ag.
8	15
4	3

1101  
0101

ma. nolitesi = 1

Fitness	Top. Ag.
12	6
14	7

1101

Mutasyon Oranı: 0/10

Mutasyon Sayısı =  $0.10 \times 10 = 1$   
Kaz mutasyon yapacaktır

(t=3)

Turnuva:  $\rightarrow 00011$   
 K5: 11  $r_1 = 1$  (0'ını 1'ini!)  
 K3: 5  $r_2 = 2$  (hangisi yan?)  
 K7: 8

M1: 01011  $\frac{Fitness}{13}$   $\frac{Top. Ag.}{6}$

Yeni Pop. Oluştur

Yeni Bireyler:	Fitness
Y1: 11100	8
Y2: 01100	4
Y3: 00101	12
Y4: 01101	14
Y5: 01011	13

En kötüyi değiştir.

Y1: pop. aynı olduğu için değiştirilmedi!  
 Y2: 4  $\rightarrow$  pop. en kötü K9: 4  $\rightarrow$  Karar: Değiştir.  
 Y3: pop. aynı var  
 Y4: pop. aynı var.  
 Y5: 13  $\rightarrow$  pop. en kötü K8: 3  $\rightarrow$  Karar: Değiştir

Durdurma Kriteri: 1) Max iterasyon sayısı

2) İyileşme olmadan max iterasyon sayısı

- 1 iterasyon sonrasında popülasyonda 2 birey değişti; en iyi fitness değeri 14'te kaldı, değişmedi; ortalama fitness değeri 7.3'ten 8.6'ya yükseldi.



## Sirt Çantası Problemi – Örnek 2

**Item:**        1 2 3 4 5 6 7

**Benefit:**    5 8 3 2 7 9 4

**Weight:**    7 8 4 10 4 6 4

**Knapsack holds a maximum of 22 pounds**

**Fill it to get the maximum benefit**

The problem description:

- Maximize  $\sum_i v_i$
- While  $\sum_i w_i \leq W$

## Sirt Çantası Problemi – Örnek 2

- Solutions take the form of a string of 1's and 0's

Where

0 = don't take the item in a given positions

1 = take the item in a given positions

- Example chromosomes:
- 1100100 items {1,2,5} included in sack
- 0010000 items {3} included in sack
- 0001100 items {4,5} included in sack
- 0100001 items {2,7} included in sack

## Sırt Çantası Problemi – Örnek 2

Kromozom: 1 1 0 0 1 0 0

Item	1	2	3	4	5	6	7
<b>Solution</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
Benefit	5	8	3	2	7	9	4
Weight	7	8	4	10	4	6	4

Fitness:  $5 + 8 + 7 = 20$

Weight:  $7 + 8 + 4 = 19 \leq 22$

Kromozom: 0 1 0 1 0 1 0

Item	1	2	3	4	5	6	7
<b>Solution</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
Benefit	5	8	3	2	7	9	4
Weight	7	8	4	10	4	6	4

Fitness:  $8 + 2 + 9 = 20$

Weight:  $8 + 10 + 6 = 24 \geq 22$

## Sırt Çantası Problemi – Örnek 2

Penalize:

$$Fitness = \begin{cases} \sum_i v_i : \left( \sum_i w_i \leq W \right) \\ W - \sum_i w_i : (otherwise) \end{cases}$$

Item	1	2	3	4	5	6	7
<b>Solution</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
Benefit	5	8	3	2	7	9	4
Weight	7	8	4	10	4	6	4

Fitness (Benefit):  $22 - (8 + 10 + 6) = -2$

Weight:  $8 + 10 + 6 = 24 > 22$

## Sırt Çantası Problemi – Örnek 2

Uygun olmayan (infeasible) kromozomların tamiri:

- Kromozomu soldan sağa doğru okurken:
- Eşyayı alıp almamanın kapasiteye yapacağı etki dikkate alınarak:
- Eğer mevcut kapasite kısıtını **aşmıyorsa** eşya dahil edilir.
- Eğer mevcut kapasite sınırını **aşıyorsa** eşya dahil edilmez.

Item	1	2	3	4	5	6	7
Benefit	5	8	3	2	7	9	4
Weight	7	8	4	10	4	6	4

Kromozom: 0 1 0 1 0 1 0

**Ağırlık:**  $8 + 10 + 6 = 24 > 22$



Tamir

Kromozom: 0 1 0 1 0 0 0

**Ağırlık :**  $8 + 10 = 18 < 22$

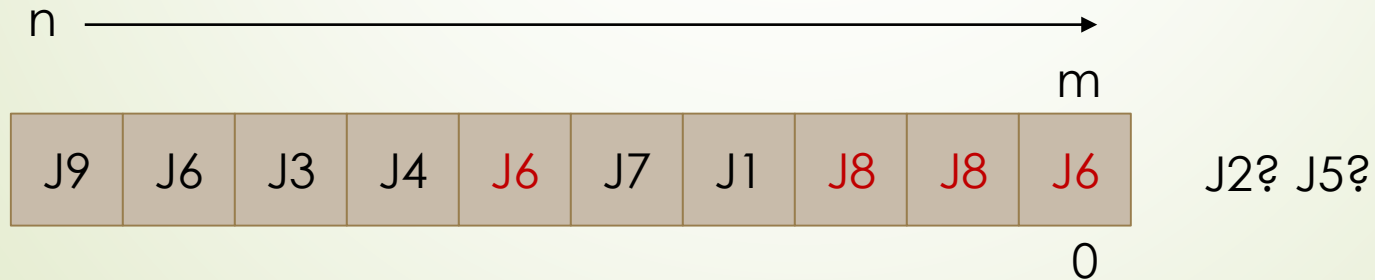
Fitness: 10

# İş Atama Problemi

- $n$  işçi,  $m$  iş: ne tür kromozom gösterimi (kodlama) kullanırsınız?
- Her işe tam olarak bir işçi atanmak zorundadır.
- Her işçi birden fazla işi yapabilir. Her işin yapılması için belirli bir süre gerekli.

## ■ Kodlama 1

- 1- $m$  arasında değerler alabilen  $n$  adet genden oluşan bir kromozom
  - + her işçiye en az bir iş atanmış durumda
  - fakat her işe bir işçi atanmayabilir
  - her işe tam olarak bir işçi atanmak zorundadır kısıtı sağlanmayabilir
  - iki veya daha fazla işçi aynı işe atanmamalıdır

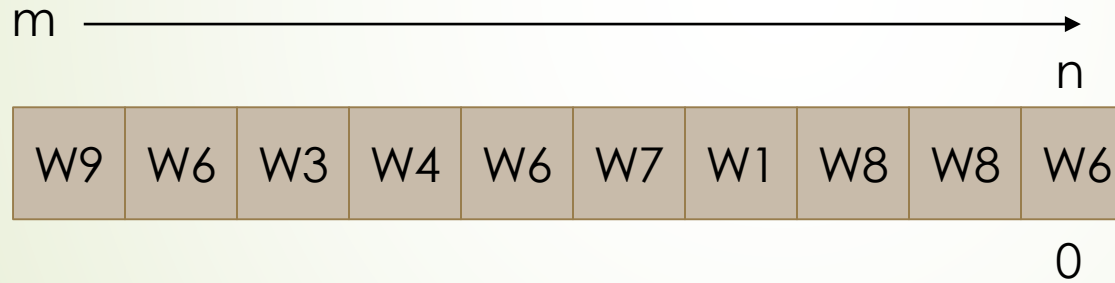


# İş Atama Problemi

## ■ Kodlama 2

■  $1-n$  arasında değerler alabilen  $m$  adet genden oluşan bir kromozom:

- + her işe tam olarak bir işçi atanmak zorundadır kısıtı sağlanır
- - fakat bir işçiye çok fazla iş atanabilir.



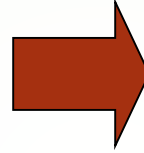
# Sınav Programı Çizelgeleme - Doğrudan Kodlama

Sınav1 (E1), 4. slotta

4, 5, 13, 1, 1, 7, 13, 2

Sınav2 (E2), 5. slotta

...



	ptesi	salı	çar	perş
9:00	E4, E5	E2		E3, E7
11:00	E8			
2:00		E6		
4:00	E1			

- 8 genden oluşan bir kromozom.
- Her gen 1-16 arasında rastgele bir değer alabilir
- Fitness fonksiyonu çakışma sayısı olabilir. Amaç, min fitness.

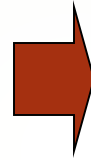
*Fitness fonksiyonu ve gösterime karar verdikten sonra popülasyon oluşturulup çözüm aranabilir.*



# Mutasyon

## Mutasyon Öncesi

4, 5, 13, 1, 1, 7, 13, 2



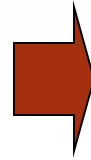
	mon	tue	wed	thur
9:00	E4, E5	E2		E3, E7
11:00	E8			
2:00		E6		
4:00	E1			

- Doğrudan tek noktalı mutasyon uygulanabilir
- Rastgele bir gen seç

# Mutasyon

## Mutasyon Sonrası

4, 5, 6, 1, 1, 7, 13, 2



	mon	tue	wed	thur
9:00	E4, E5	E2		E7
11:00	E8	E3		
2:00		E6		
4:00	E1			

- Bir mutasyon, bir sınavın yerini deęiřtirir.

# Sınav Programı Çizelgeleme - Dolaylı Kodlama

4, 5, 10, 1, 1, 7, 15, 2

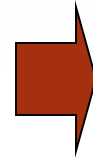


...

E3'ü, 10. uygun (çakışma olmayan) konuma

E2'yi, 5. uygun (çakışma olmayan) konuma

E1'i, 4. uygun (çakışma olmayan) konuma



	mon	tue	wed	thur
9:00	E4 E5	E2		
11:00	E8		E3	
2:00		E6		E7
4:00	E1			

Aşağıdaki grupların çakışma oluşturduğunu varsayalım (aynı saate gelmemeli)

{E1, E2}, {E1, E3}, {E2, E6}, {E2, E7}, {E2, E8}, {E3, E5}, {E3, E6},  
{E4, E6}, {E4, E7}, {E5, E7}, {E5, E8}, {E6, E8}

## Mutasyon (Öncesi)

4, 5, 10, 1, 1, 7, 15, 2

E3'ü, 10. uygun (çakışma olmayan) konuma  
E2'yi, 5. uygun (çakışma olmayan) konuma  
E1'i, 4. uygun (çakışma olmayan) konuma

	mon	tue	wed	thur
9:00	E4 (E5)	E2		
11:00	E8		E3	
2:00		E6		E7
4:00	E1			

Aşağıdaki grupların çakışma oluşturduğunu varsayalım (aynı saate gelmemeli)

{E1, E2}, {E1, E3}, {E2, E6}, {E2, E7}, {E2, E8}, {E3, E5}, {E3, E6},  
{E4, E6}, {E4, E7}, {E5, E7}, {E5, E8}, {E6, E8}

## Mutasyon (Sonrası)

4, 5, 10, 1, 14, 7, 15, 2

E3'ü, 10. uygun (çakışma olmayan) konuma  
E2'yi, 5. uygun (çakışma olmayan) konuma  
E1'i, 4. uygun (çakışma olmayan) konuma

	mon	tue	wed	thur
9:00	E4	E2		
11:00	E8		E3	E5
2:00		E6		E7
4:00	E1			

Aşağıdaki grupların çakışma oluşturduğunu varsayalım (aynı saate gelmemeli)

{E1, E2}, {E1, E3}, {E2, E6}, {E2, E7}, {E2, E8}, {E3, E5}, {E3, E6},  
{E4, E6}, {E4, E7}, {E5, E7}, {E5, E8}, {E6, E8}

# Doğrudan Kodlama vs. Dolaylı Kodlama

Doğrudan:

- Kolaylıkla kodlama → çözüme → kodlama yapılabilir
- Mutasyonun etkisi kolaylıkla gözlenebilir
- Kromozomların kolaylıkla çözülmesi sebebiyle hızlı fitness hesaplama

Dolaylı:

- Problem bilgisi çözüme aktarılabilir
- Uygun çözümler üretildiği için uygunsuzluk problemi olmaz
- Arama uzayını çok etkili bir şekilde dağıtır
- Fakat yorumlanması (kod çözümü) daha zordur

# References

- ▶ Eiben and Smith. Introduction to Evolutionary Computing, Springer-Verlag, New York, 2003.
- ▶ E.C.Keedwell, Nature Inspired Computing (Lecture Notes), University of Exeter, 2015. Eiben, A.E., Smith, James E. Introduction to Evolutionary Computing, Springer, 2003.
- ▶ J. Drezo A. Petrowski, P. Siarry E. Taillard, Metaheuristics for Hard Optimization, Springer-Verlag, 2006.
- ▶ R.J. Moraga, G.W. DePuy, G.E. Whitehouse, Metaheuristics: A Solution Methodology for Optimization Problems, Handbook of Industrial and Systems Engineering, A.B. Badiru (Ed.), 2006.
- ▶ M. Yaghini, What is a Metaheuristic?  
[http://webpages.iust.ac.ir/yaghini/Courses/AOR\\_872/What%20is%20a%20Metaheuristic.pdf](http://webpages.iust.ac.ir/yaghini/Courses/AOR_872/What%20is%20a%20Metaheuristic.pdf),  
Accessed 18 Sept 2017.