

EMM4131

Popülasyon Temelli Algoritmalar

(Population-based Algorithms)

3

Genetik Algoritma

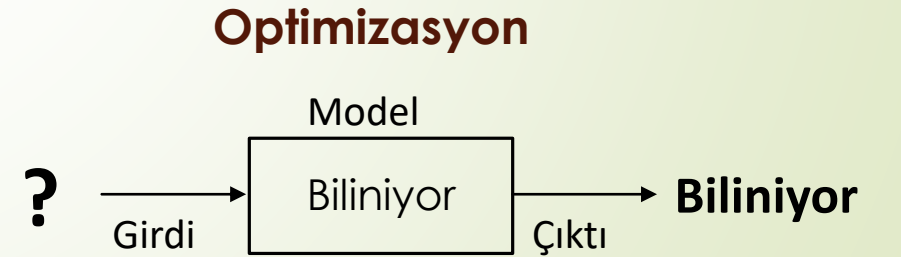
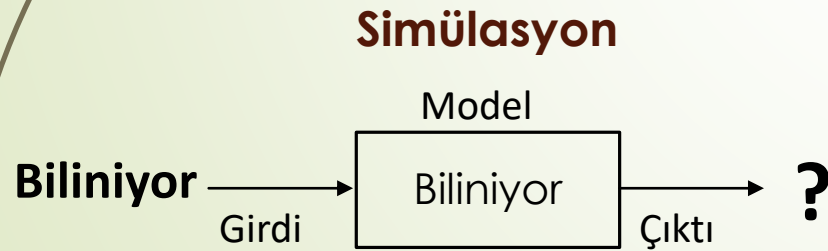
Yrd. Doç. Dr. İbrahim KÜÇÜKKOÇ

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

Email: ikucukkoc@balikesir.edu.tr

Onceki Dersten Tekrar

- Endüstri Mühendisliği'nde çalışılan Yöneylem Araştırması problemlerinin büyük çoğunluğu kesikli ve kombinatoriyal optimizasyon problemleridir.
- Kesin çözüm yöntemleri optimum çözümü garanti eder fakat çözüme ulaşması çok uzun sürdüğü için genellikle pratik değildir.
- Bu yüzden yaklaşık çözüm yöntemleri kullanılır. Bunlar da sezgisel ve metasezgisel olarak iki gruba ayrılırlar.
- Genetik Algoritma (GA) ve Karınca Koloni Optimizasyonu (ACO) algoritmaları, popülasyon temelli metasezgisel algoritmalar"dır. Stokastik arama



Onceki Dersten Tekrar

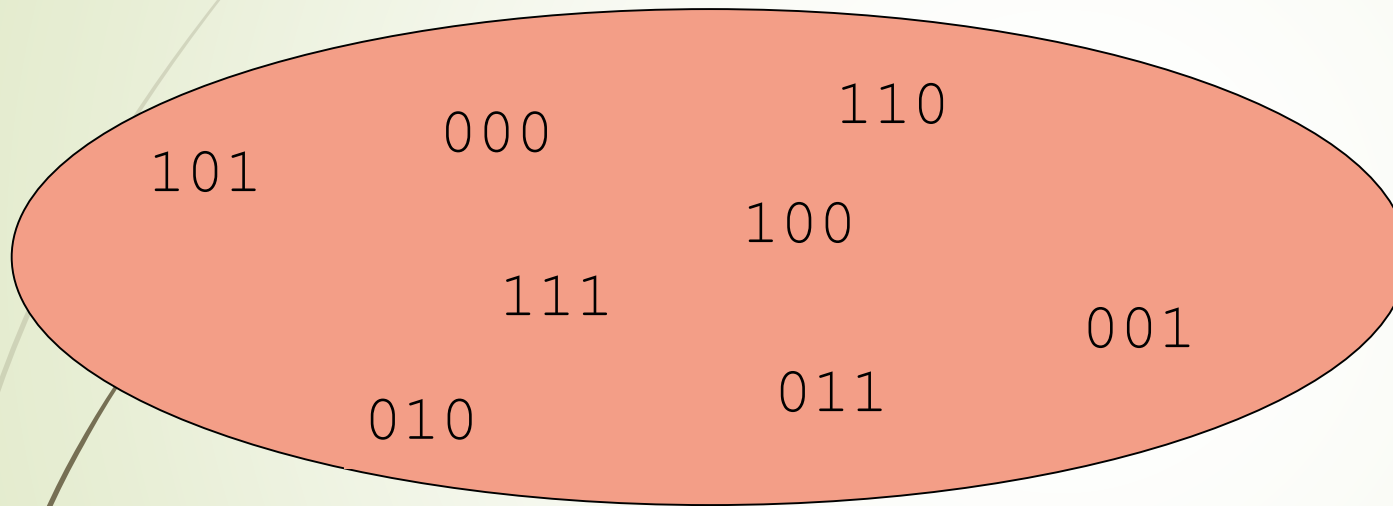
- Bireylerden oluşan bir popülasyon sınırlı kaynaklardan oluşan bir çevrede bulunmaktadır.
- Sınırlı kaynaklar için yapılan **rekabet** daha iyi olanın (çevreye **daha iyi adapte olanın**) seçilimine sebep olur.
- Bu bireyler çaprazlama ve mutasyon ile yeni jenerasyonun bireylerini oluştururlar.
- Yeni bireylerin fitness'ı hesaplanır ve hayatta kalmak için mücadele ederler.
- Zamanla, doğal seçim, popülasyonun fitness'ında iyileşmeye sebep olur.

Bugunku Dersin Planı

- GA'nın Çalışma Prensibi
- GA'nın Aşamaları
 - Çözümün Temsili
 - Başlangıç Popülasyonu
 - Seleksiyon
 - Genetik Operatörler (Çaprazlama ve Mutasyon)
 - Yeni Popülasyonun Oluşturulması
- GA'nın sonlandırılması

Arama ve Optimizasyon

20kg, 75kg ve 60kg ağırlıklarında üç adet nesne var. 100kg'a en yakın olacak şekilde hangi ürünleri almaliyiz?

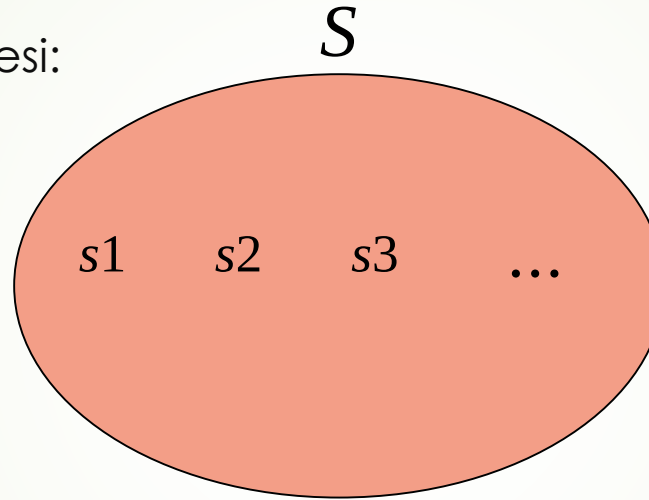


Which is the best?

Amac Fonksiyonu: $|\text{toplam ağırlık} - 100|$

Arama ve Optimizasyon

- Optimizasyon genellikle, verilen bir probleme bulunabilecek en uygun çözümün bulunması işlemidir.
- S ile verilen olası çözümler kümesi:

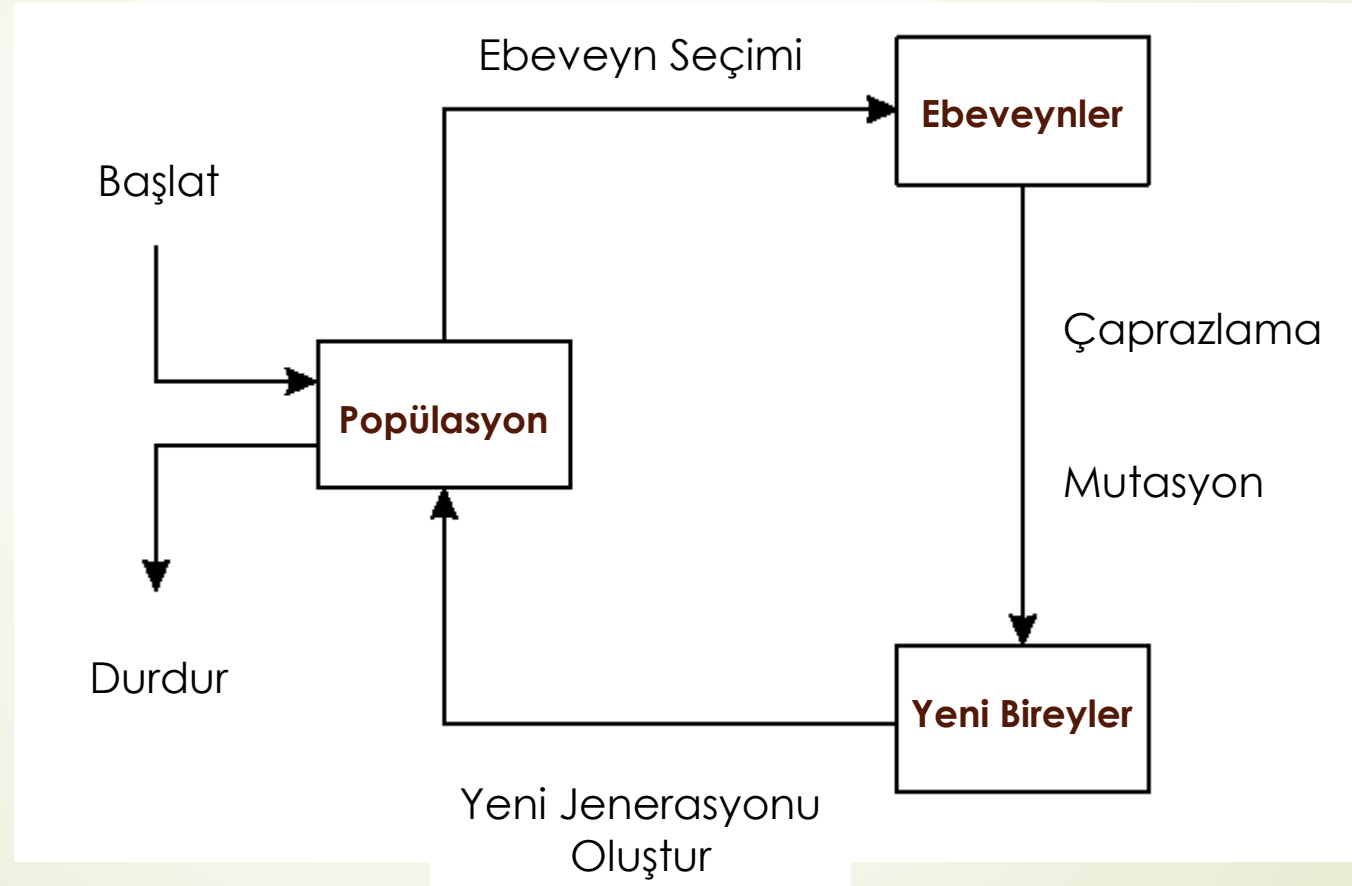


- S küçük olabilir (görüldüğü gibi)
- S çok, çok, çok büyük olabilir (örneğin 500 sınavın 3 haftalık bir döneme çizelgelenmesi problemi). Tipik bir gerçek hayat probleminin büyüklüğü 10^{30} olası çözüm içerebilir.
- S sonsuz çözüm içerecek derecede büyük olabilir (örneğin tüm gerçel sayılar).

GA Çalışma Mekanizması

- GA, üret ve test et mekanizmasına sahiptir.

GA çalışma mekanizması



GA Çalışma Mekanizması

Name	F(S)
S1	0.1
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S10	0.1

Ebeveyn seç

S5	0.9
S9	0.4

Genetik operatörleri
uygula

S11	1.0
S12	0.2

Bazı eski bireyleri

Bazı yeni bireylerle yer değiştir

GA Akış Şeması

GA Akış Şeması

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Genetik Algoritma (Jenerasyonel-Generational)

- Yeni popülasyon oluşturulana kadar, seçim ve genetik operatörler uygulanır.

Başlangıç Pop.

Name	F(S)
S1	0.1
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S10	0.1

Seçim ve genetik operatörleri yeni popülasyon oluşturulana kadar uygula (x10)

Jenerasyon 1

Name	F(S)
S11	0.5
S12	0.3
S13	0.3
S14	0.7
S15	0.7
S16	0.9
S17	0.4
S18	0.9
S19	0.9
S20	0.3

Seçim ve genetik operatörleri yeni popülasyon oluşturulana kadar uygula (x10)

Jenerasyon 2

Name	F(S)
S21	0.7
S22	0.8
S23	0.9
S24	0.9
S25	0.7
S26	0.8
S27	0.5
S28	0.7
S29	0.6
S30	0.7

...

Genetik Algoritma (Kararlı Hal-Steady State)

- Genetik operatörler N defa uygulanır ve popülasyondaki kötü çözümler yeni oluşturulan iyi çözümlerle değiştirilir.

Başlangıç Pop.

Name	F(S)
S1	0.1
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S10	0.1

Seçilim ve genetik operatörleri yeni bireyler oluşturmak için N defa uygula

Name	F(S)
S11	0.1
S12	0.5

Popülasyondaki kötü (zayıf) bireyleri değiştir

Jenerasyon 1

Name	F(S)
S12	0.5
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S11	0.1

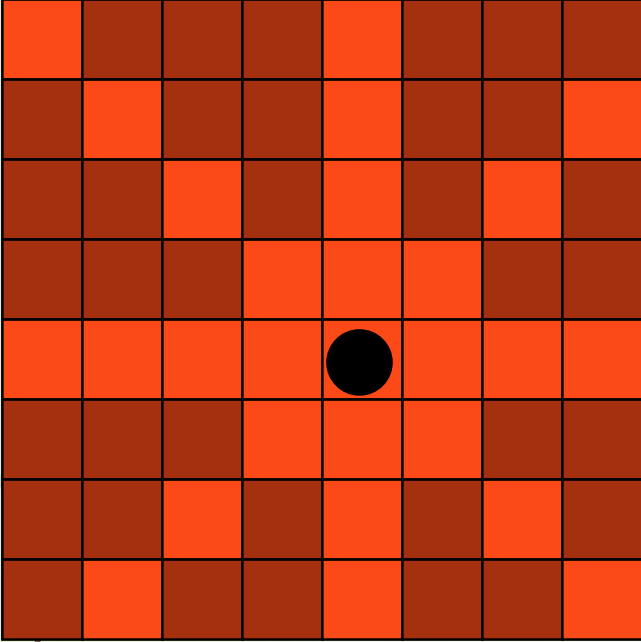
Başlangıç Popülasyonu

- Belirli sayıda (**popülasyon büyüklüğü kadar**) olası çözümleri tutar.
- Uygun bireyler-kromozomlar oluşturacak bir mekanizma ile popülasyon büyüklüğü kadar kromozomla/uygun çözümler oluşturulur.
- Bu mekanizma sadece gerekli kısıtları sağlayan rassal bir prosedür olabilir veya nispeten iyi çözümler üretecek belirli bir sezgisel yöntemin uyarlaması olabilir.
- Üretilen çözümlerdeki önemli nokta:
 - Arama uzayının farklı noktalarını temsil edecek şekilde kromozomlar oluşturulmalıdır.
 - Çeşitlilik korunmalıdır.
 - Aynı kromozomlardan oluşturulmamasına dikkat edilmelidir.

Kromozomun Yapısı / Çözümün Temsili?

- Muhtemelen GA'daki en temel meseledir.
- Verilen bir optimizasyon probleminin çözümünün temsiline yönelik uygun bir yol bulmalıyız.
- Aynı problem için pek çok farklı kodlama olabilir.
- Her farklı kodlama, arama uzayının şeklini ve arama stratejisini değiştirir.
- Bir kromozom aşağıdakilerden oluşabilir:
 - Tam sayılar
 - Gerçek değerler
 - İkili değerler (0-1)
- Kodlama türü, başarılı bir optimizasyon için gerekli genetik operatörleri etkiler.
- Bazı kodlamalar özel genetik operatörler gerektirir (Örn. Permütasyon tabanlı kodlama)

Kromozomun Yapısı / Çözümün Temsili?

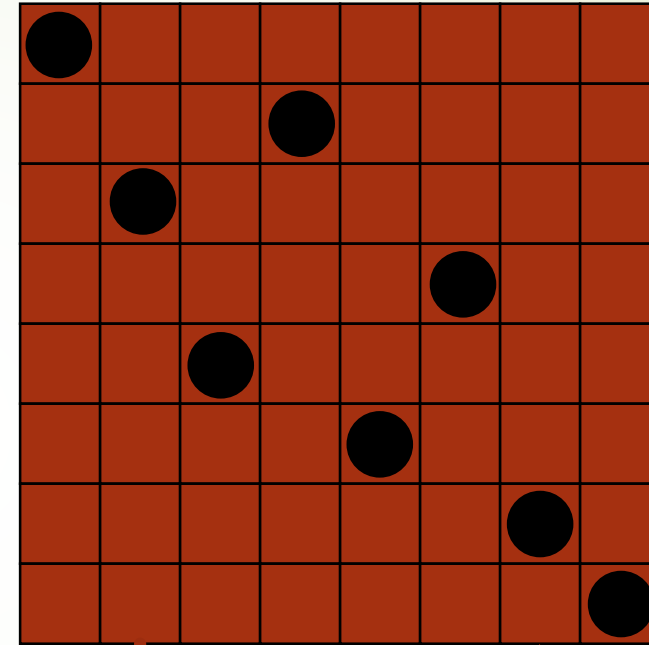


Satranç tahtası üzerine
birbirini tehdit etmeyecek şekilde
8 tane vezir yerleştirelim.

Kromozomun Yapısı / Çözümün Temsili?

Fenotip:
konfigürasyon

Genotip:
1-8 numaraların
permütasyonu



Kodlama:
(fenotip => genotip)

Kod Çözme:
genotip => fenotip

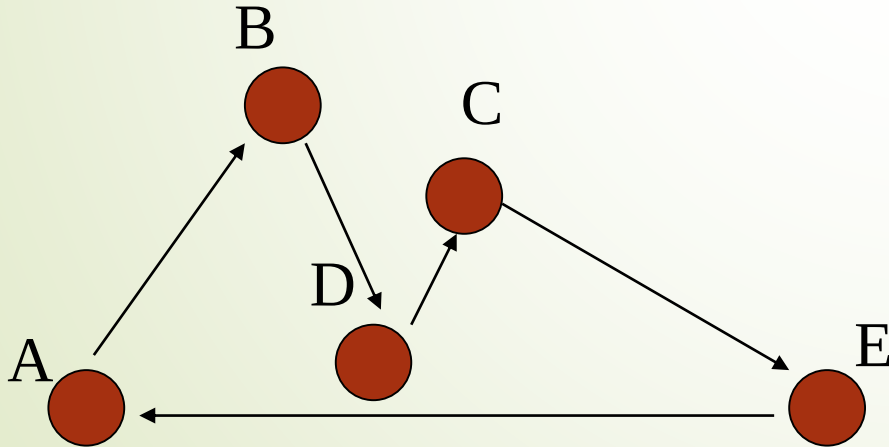
1 3 5 2 6 4 7 8

Kromozomun Yapısı / Çözümün Temsili?

Gezgin Satıcı Problemi

En kısa rotayı bul:

- Her şehir bir kez ziyaret edilmeli
- Başlanılan yere dönülmeli



Bu rotanın uzunluğu: 33 birimdir.

	A	B	C	D	E
A	-	5	7	4	15
B	5	-	3	4	10
C	7	3	-	2	7
D	4	4	2	-	9
E	15	10	7	9	-

Seilim (Selection)

Ruleť emberi, Turnuva

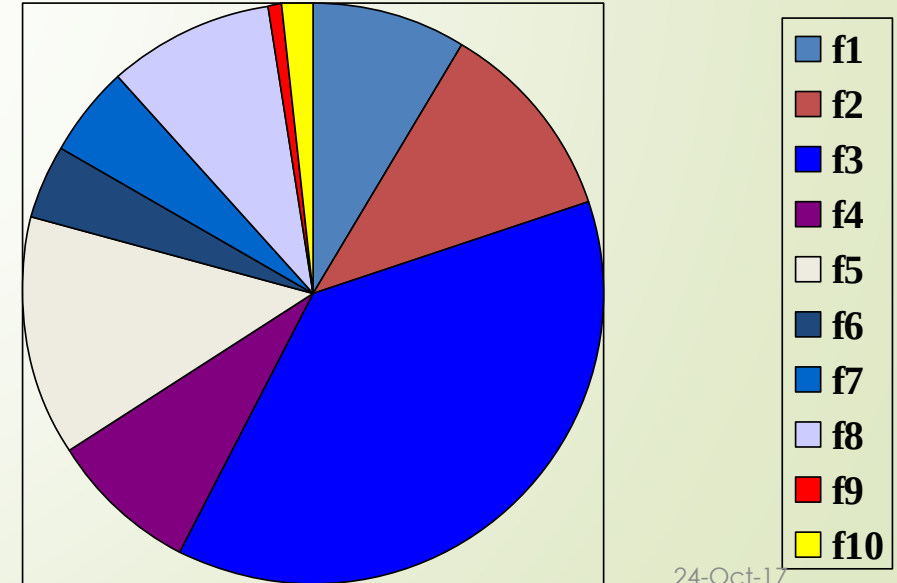
Seilim Mekanizmaları (Seleksiyon)

Rulet emberi

- Kromozom ve mutasyon işlemleri uygulanmak üzere ebeveyn seimleri yapılması gerekmektedir.
- Her bir kromozom (aday özüm), uygunluk deėerleri oranında seim şansına tabiidir. Uygunluk deėeri daha iyi aday, daha ok seilme şansına sahiptir.
- P adet bireyin fitness deėerlerinin $f_1, f_2, \dots, f_P$ olduğunu varsayalım.

- Birey i 'nin seilme olasılığı:
$$\frac{f_i}{\sum_{k=1}^P f_k}$$

- Bu durum, fitness deėerleri oranında bölünmüş bir rulet tekerleėinin evrilmesi gibidir.



Seçilim Mekanizmaları (Seleksiyon)

Rulet Çemberi

► Problemler:

- Amaç, minimizasyon ise ?
- Negatif fitness değerleri olursa?

► Amacımızın maksimizasyon olduğunu ve şu fitness değerlerine sahip olduğumuzu düşünelim:

100, 0.4, 0.3, 0.2, 0.1

-- en iyisi, en kötüye göre 100 kat daha fazla seçilme olasılığına sahiptir.

► Şu şekilde modifiye edilebilirler:

200, 100.4, 100.3, 100.2, 100.1

– çok daha kabul edilebilir bir durumdur.

Seilim Mekanizmaları (Seleksiyon)

Turnuva Seimi

Parametre: Turnuva büyüklüğü (t)

Bir ebeveyn seçmek için:

Popülasyondan rassal olarak t adet birey seç

Seilen t adet birey içinden en iyi fitness değerine sahip olanı al.

- + Ayarlanabilir
- + Çok iyi ve çok kötü fitness'lı kromozomlardan kaçınılabilir
- + Geliştirmesi çok kolaydır
- Ekstra parametre gerektirir (t -turnuva büyüklüğü)

Popsiz=10000 iken $t=10$ olursa ne olur?

Seilim Mekanizmaları (Seleksiyon)

- **Seilim**, özüm uzayının hangi bölgelerine yoğunlaşacağımızla ilgili stratejimizi belirler.
- Genetik operatörler (aprazlama ve mutasyon), yeni aday özümleri oluşturmak için bir araçtır.
 - Genetik operatörlerin yeni iyi özümlere yavaşa ve adım adım gitmelerini isteriz (küük deėişikliklerle ve/veya iyi olduğunu düşündüğümüz özümlerin bazı kısımlarını kullanarak)
 - Aynı zamanda özüm uzayının tamamını taramak isteriz.

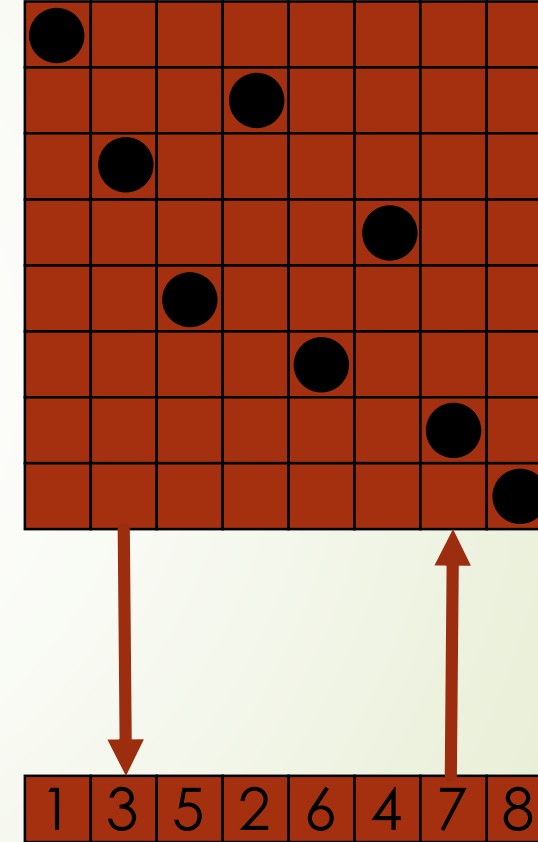
Mutasyon

Tek Noktalı, M-noktalı, Insert, Swap, Scramble

Mutasyon (k'lı Kodlama İçin)

- Bir aday çözümün 0'dan $k-1$ 'e kadar (veya 1'den k 'ya kadar) değerler aldığı gösterim şekline **k'lı kodlama** deriz.

Örn. 8 vezir problemi:



Mutasyon (k'lı Kodlama İçin)

Tek noktalı mutasyon:

Rastgele bir gen seç ve değerini rastgele ata.

Örn. 352872 → 312872

M-noktalı mutasyon:

Tek noktalı mutasyonun M defa tekrarıdır.

Swap (değiştirme) mutasyon:

Rastgele iki gen seç ve karşılıklı olarak yer değiştir.

Örn. 352872 → 372852

Bu yöntem neden çok iyi çözümler üretemez?

Mutasyon (Gerçek Değerli Kodlama İçin)

Bir aday çözüm, genellikle L uzunluğunda ve gerçek sayılardan oluşan bir vektördür.

Tek noktalı mutasyon:

Rastgele bir gen seç ve ona rastgele bir değişiklik ekle. Değişim genellikle Gaussian dağılımından elde edilir.

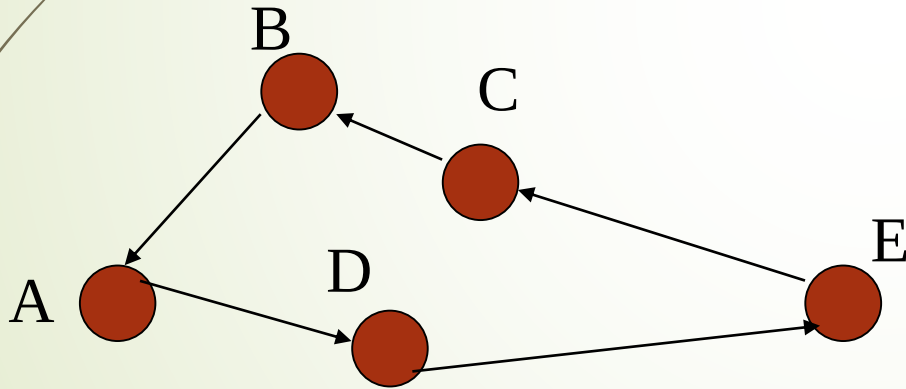
Vektör mutasyon:

L uzunluğunda rastgele bir vektör oluştur ve tamamen mevcut ebeveyne ekle.

Mutasyon (Permütasyon Tabanlı Kodlama İçin)

Gezgin satıcı problemini hatırlayalım.

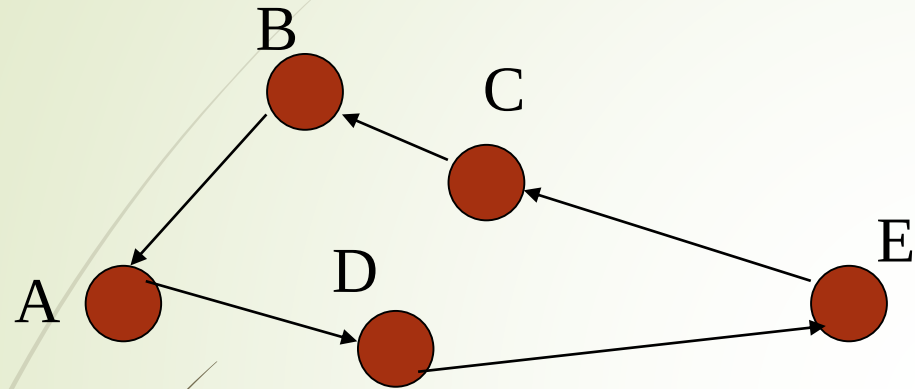
5 uzunluğunda bir permütasyon tabanlı kodlama:



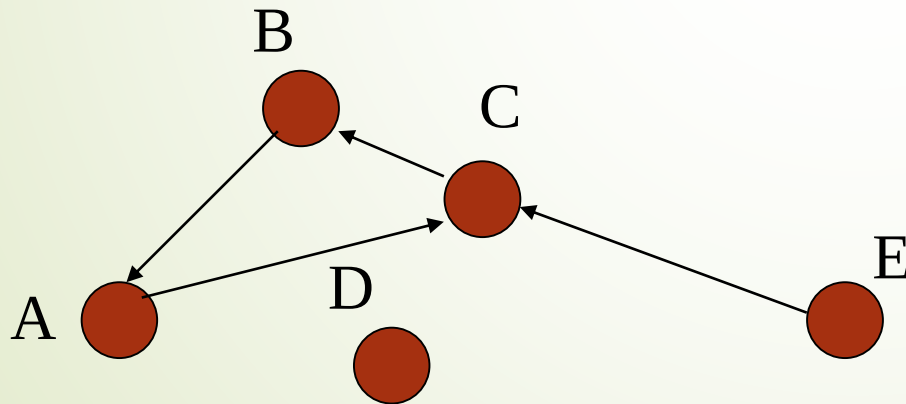
	A	B	C	D	E
A	-	5	7	4	15
B	5	-	3	4	10
C	7	3	-	2	7
D	4	4	2	-	9
E	15	10	7	9	-

- Tek noktalı mutasyon yapılabilir mi?
- Swap mutasyon kullanılabilir mi?

Mutasyon (Permütasyon Tabanlı Kodlama İçin)



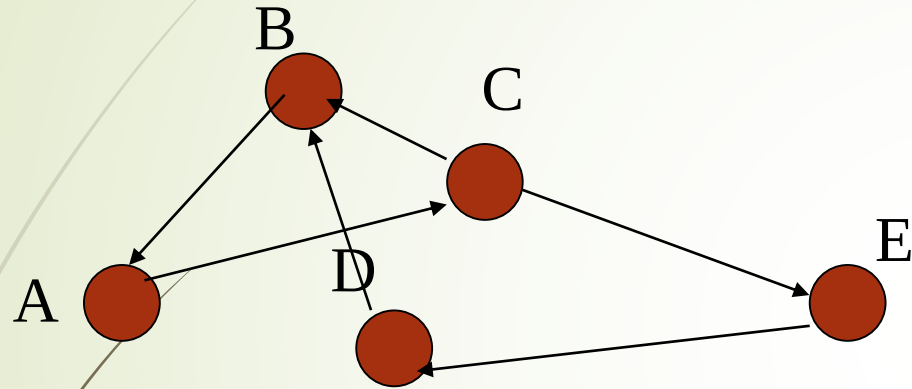
- Standart Mutasyon (kromozomdaki bir degeri rastgele degistir):



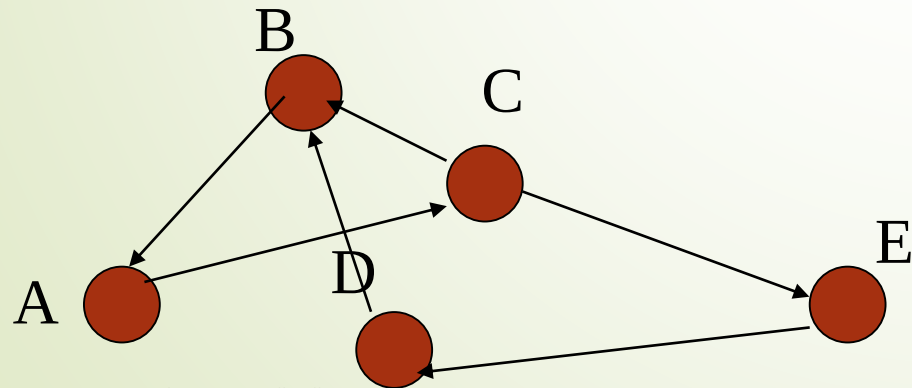
Kromozomda iki defa **C** var
ve **D** hic yok!

Mutasyon (Permütasyon Tabanlı Kodlama İçin)

- Burada **insert mutasyonu** kullanılabilir:



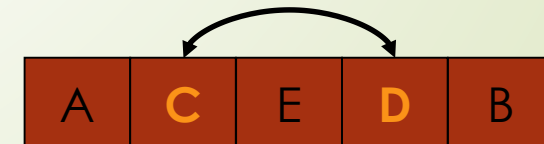
- Burada **swap mutasyonu** kullanılabilir:



↓ Mutasyon



↓ Mutasyon



Scramble Mutasyon (Permutasyon Kodlama)

- Genlerin rastgele bir bölümünü al.
- Rastgele düzenle ve aynı yere tekrar yerleştir.

Örnekler:

1 2 3 4 5 6 7 8 9



1 3 5 4 2 6 7 8 9

B D A E H G F C



B D F H G E F C

Not: Seçilen bölüm ardışık olmak zorunda değildir.

Mutasyon-Gezgin Satıcı Örnek

	A	B	C	D	E
A		5	7	4	15
B	5		3	4	10
C	7	3		2	7
D	4	4	2		9
E	15	10	7	9	

5 kromozomdan oluşan başlangıç popülasyonu:

	ACEBD	DACBE	BACED	CDAEB	ABCED
Fitness:	32	33	32	31	28

Seçilen kromozom mutasyona uğrarsa: CDAEB → ADCEB

Mutantın fitness değeri: 26

Mutant, en kötünün yerine geçerek popülasyona dahil edilir.

	A	B	C	D	E
A		5	7	4	15
B	5		3	4	10
C	7	3		2	7
D	4	4	2		9
E	15	10	7	9	

Generation 2

	ACEBD	ADCEB	BACED	CDAEB	ABCED
Evaluation	32	26	32	31	28

Mutant of selected parent BACED → BDCEA

Evaluation of mutant: 33

Mutant discarded— worse than current worst

Mutasyon-Gezgin Satıcı Örnek

	A	B	C	D	E
A		5	7	4	15
B	5		3	4	10
C	7	3		2	7
D	4	4	2		9
E	15	10	7	9	

Generation 3:

	ACEBD	ADCEB	BACED	CDAEB	ABCED
Evaluation	32	26	32	31	28

Mutant of selected parent ABCED → ABECD

Evaluation of mutant: 28

Mutant enters population, replacing worst

	A	B	C	D	E
A		5	7	4	15
B	5		3	4	10
C	7	3		2	7
D	4	4	2		9
E	15	10	7	9	

Generation 4:

	ABECD	ADCEB	BACED	CDAEB	ABCED
Evaluation	28	26	32	31	28

Mutant of selected parent ADCEB → BDCEA

Evaluation of mutant: 33

Mutant is discarded

Mutasyon-Gezgin Satıcı Örnek

	A	B	C	D	E
A		5	7	4	15
B	5		3	4	10
C	7	3		2	7
D	4	4	2		9
E	15	10	7	9	

Generation 5:

	ABECD	ADCEB	BACED	CDAEB	ABCED
Evaluation	28	26	32	31	28

Mutant of selected parent ABCED → ABECD

Evaluation of mutant: 28

Mutant enters population, replacing worst

Mutasyon-Gezgin Satıcı Örnek

	A	B	C	D	E
A		5	7	4	15
B	5		3	4	10
C	7	3		2	7
D	4	4	2		9
E	15	10	7	9	

Generation 6:

	ABECD	ADCEB	ABECD	CDAEB	ABCED
Evaluation	28	26	28	31	28

And so on.

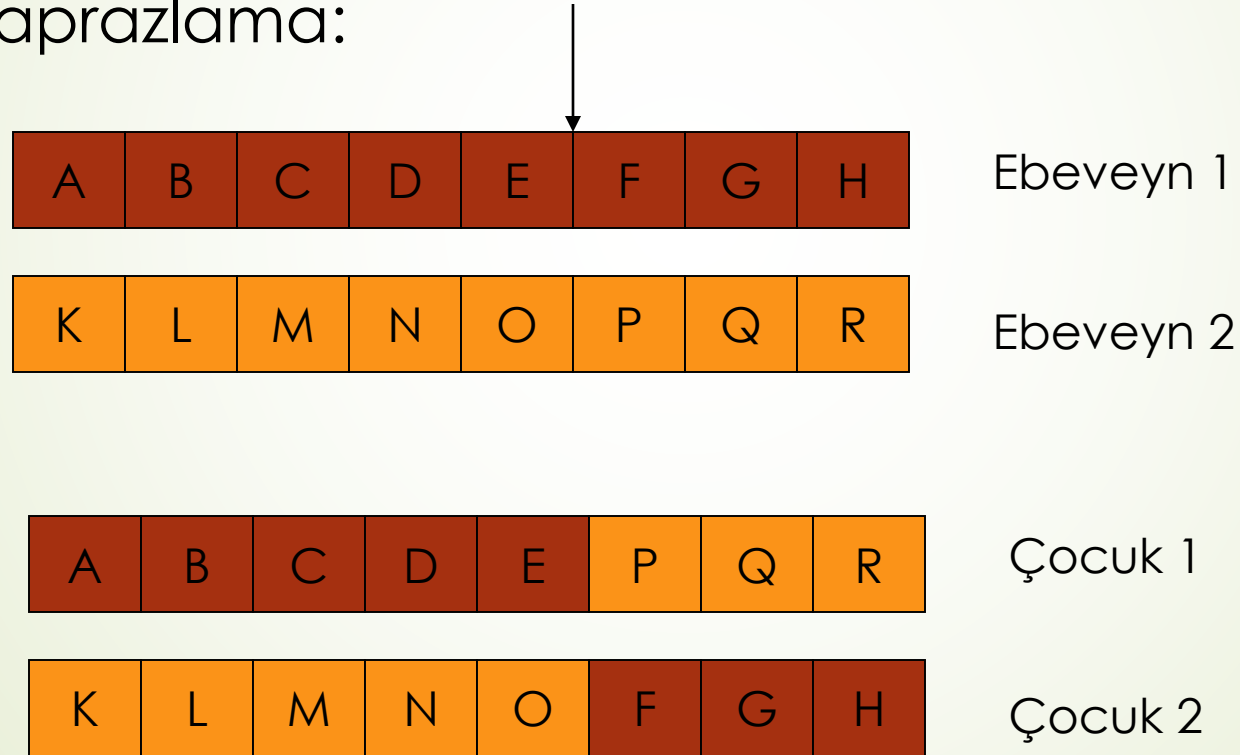
Note: population starting to converge, genotypically and phenotypically

Çaprazlama

Çaprazlama (k'lı Kodlama İçin)

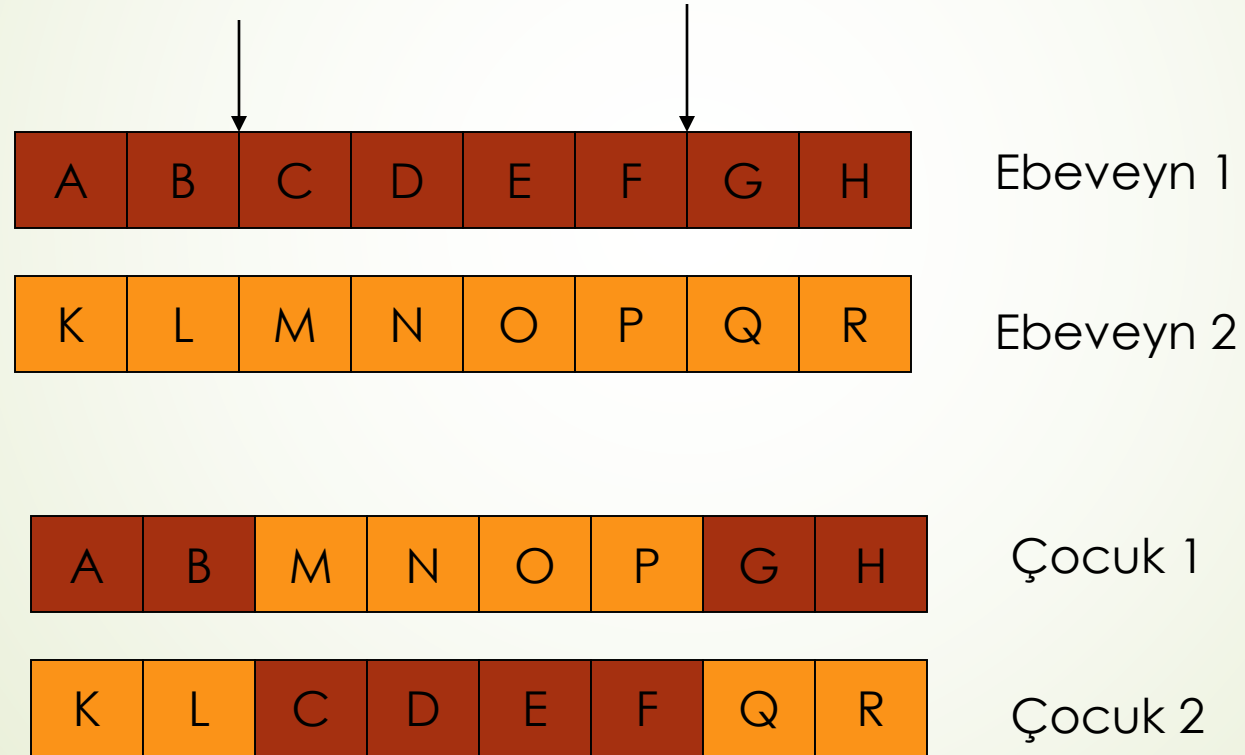
Standart çaprazlama yöntemleri:

Tek noktalı çaprazlama:



Çaprazlama (k'lı Kodlama İçin)

2-noktalı çaprazlama:



Uniform Çaprazlama

Rastgele "ikili kalıp" oluştur (kromozom uzunluğunda).
Hangi genlerin değiştirileceğine, hangilerinin aynen kalacağına karar vermek için bu kalıbı kullan.

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---

Ebeveyn 1

K	L	M	N	O	P	Q	R
---	---	---	---	---	---	---	---

Ebeveyn 2

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

İkili Kalıp

A	L	C	D	O	P	G	R
---	---	---	---	---	---	---	---

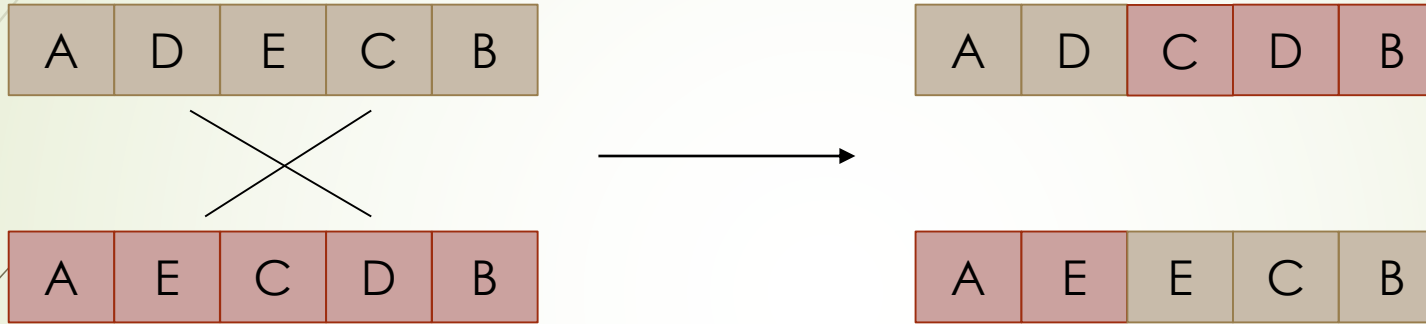
Çocuk 1

K	B	M	N	E	F	Q	H
---	---	---	---	---	---	---	---

Çocuk 2

Çaprazlama (Permütasyon Tabanlı Kodlama İçin)

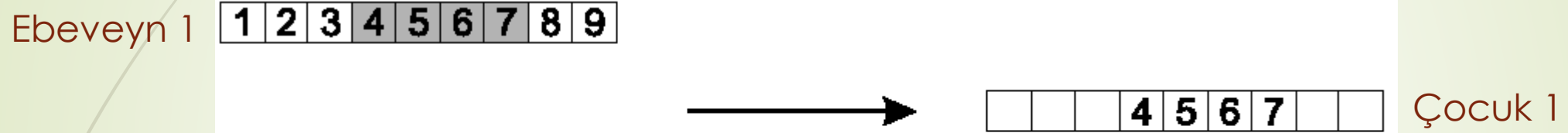
- Standart Çaprazlama ile aşağıdaki gibi 2 adet uygunsuz çocuk oluşur:



- 2 strateji kullanılır:
 - Çaprazlama yaparken uygun çocuk elde etmek.
 - Çaprazlama sonrasında elde edilen uygunsuz çözümü tamir ederek uygun çözüme dönüştürmek.

Çaprazlama (Permütasyon Tabanlı Kodlama İçin)

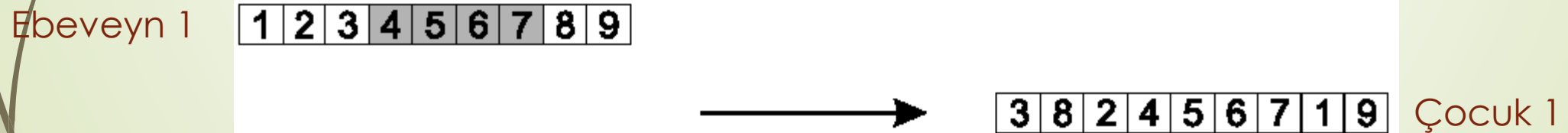
- İlk ebeveynden rastgele seçilen bir bölümü kopyala:



Ebeveyn 2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

- Geri kalan genleri, ikinci ebeveynde 'çaprazlama noktasından sonra' bulunduğu sırada al: 1,9,3,8,2

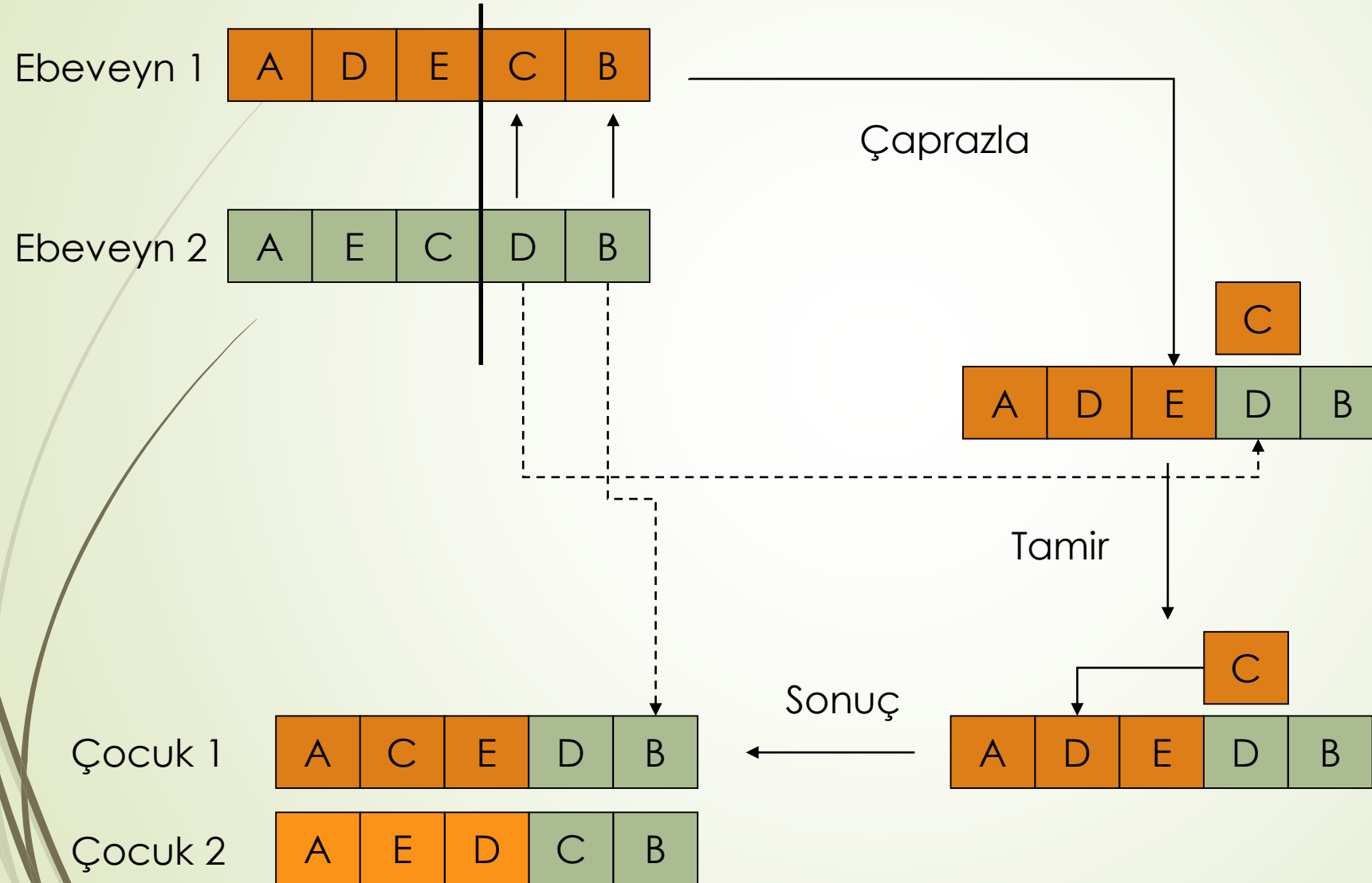


Ebeveyn 2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

Not: Çocuk 2 için aynı prosedür Ebeveyn 1 ve Ebeveyn 2 yer değiştirilerek işletilir.

PMX Crossover



Yer Değiştirme I

- Kararlı hal algoritmalarında kullanılan değişik yer değiştirme stratejileri vardır.
- En kötüyü yer değiştir:

Başlangıç Pop.

Name	F(S)
S1	0.1
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S10	0.1

Seçilim ve genetik operatörleri yeni bireyler oluşturmak için N defa uygula

Name	F(S)
S11	0.1
S12	0.5

Popülasyondaki en kötü (zayıf) bireyleri değiştir

Jenerasyon 1

Name	F(S)
S12	0.5
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S11	0.1

Yer Değiştirme II

► İlk Kötüyü Değiştir:

Başlangıç Pop.

Name	F(S)
S1	0.3
S2	0.5
S3	0.3
S4	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S10	0.1

Yeni Çözümler

Name	F(S)
S11	0.2
S12	0.5

Yeni çözümden daha kötü olan ilk çözümü bul

Jenerasyon 1

Name	F(S)
S12	0.5
S2	0.5
S3	0.3
S11	0.2
S5	0.9
S6	0.7
S7	0.3
S8	0.4
S9	0.4
S10	0.1

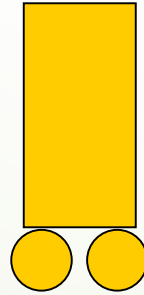
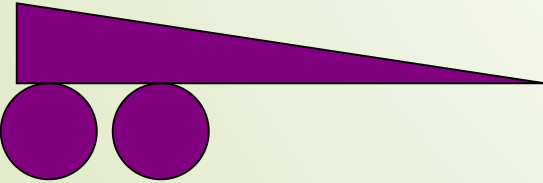
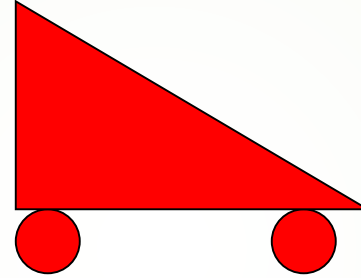
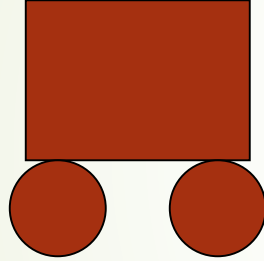
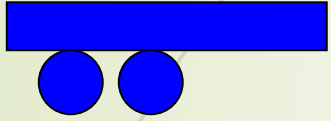
Ödev-1

8-vezir problemini GA ile el yordamıyla çözmeye çalışalım.

- Amaç: *Min Toplam Ceza*
- Tüm vezirlerin cezaları toplamı, toplam cezayı oluşturur.
- Bir vezir için ceza, tehdit ettiği vezir sayısıdır.
- Popülasyon Büyüklüğü: 10
- Çaprazlama Oranı: %60 (Permütasyon)
- Mutasyon Oranı: %1 (Insert veya Swap)
- Strateji: Popülasyondaki en kötüyü yer değiştir.
- Iterasyon Sayısı: 10

Ödev-2

- Aşağıdaki şekilleri bir başlangıç popülasyonu olarak düşünerek çaprazlama ve mutasyona tabii tutarsak nasıl şekiller elde ederiz? (5 iterasyon yapabiliriz).



References

- 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. Dreco 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,
Accessed 18 Sept 2017.