



# Optimizasyon Teknikleri

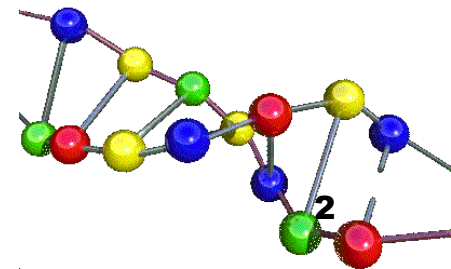
Prof. Dr. Bilal ALATAŞ

Genetik Algoritmalar ve Parçacık Sürü  
Optimizasyonu

Ders Notları - 8

# GENETİK ALGORİTMALAR (GA)

- Genetik algoritmalar, canlıların **çiftleşmesi** sonucunda kendi özelliklerini bir sonraki nesle **aktarma** yeteneklerini, **güçlü** olanın yaşama şansının yüksek olması ve bazı durumlarda önceden nasıl olacağı **bilinmeyen değişimlerin** bir hesaplama yöntemi içerisinde bir arada kullanılması sonucunda ortaya çıkmış bir yöntemdir.



# GA

- GA yöntemi ilk olarak, John Holland ve arkadaşlarının yaptığı çalışmalarda 1970'li yıllarda ortaya çıkmıştır (SGA).
- GA'nın temel amacı, fazla sayıda sınırlama içeren ve karmaşık optimizasyon sorunlarının çözümlerini, yazılımlar yardımıyla araştırmaktır.
- Konuyla ilgili Goldberg, 1989 yılında "*Genetic Algorithms in Search, Optimization, and Machine Learning*" adlı klasik sayılacak eseri yayınlamıştır.
- Bu eserle birlikte, konu üzerine yapılan çalışmalar büyük hız kazanmıştır.

# GA nedir?

- GA, rastsal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan bir arama tekniğidir.
- Bir veri grubu içinde özel bir veriyi bulmak için kullanılır.

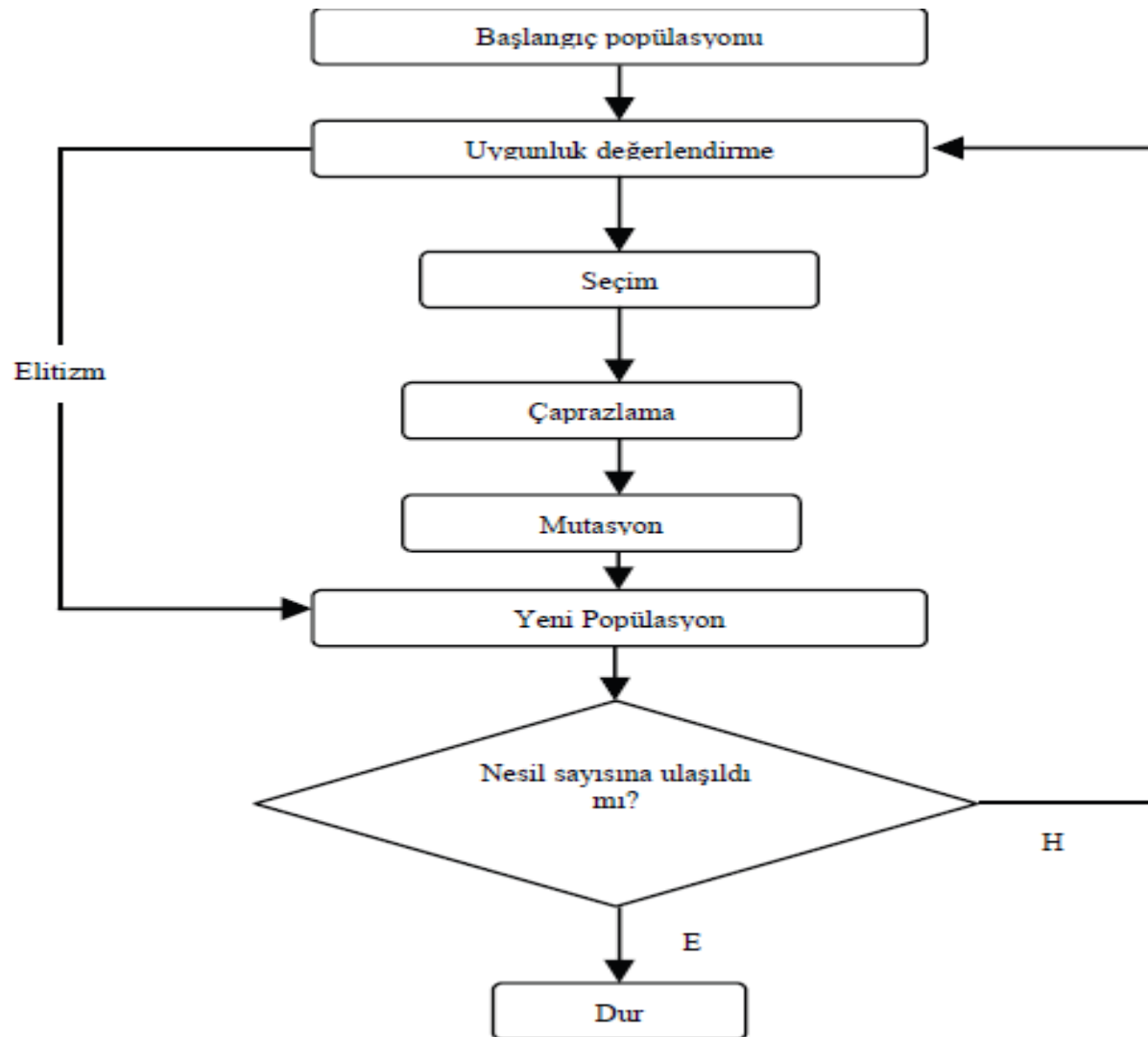
# GA – Klasik Optimizasyon


- GA parametrelerin kendileri ile değil, parametre takımının kodlanmış bir haliyle uğraşırlar.
- GA amaç fonksiyonunun türevlerini ve bir takım ek bilgileri değil, doğrudan amaç fonksiyonunun kendisini kullanırlar.
- Genetik algoritmalar aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalmazlar.
- GA'da deterministlik değil rastlantısal geçiş kuralları kullanılır.

# GA Adımları

1. Başlangıç popülasyonunu rastlantısal olarak üret.
2. Popülasyon içindeki tüm kromozomların amaç fonksiyonu değerlerini hesapla.
3. Tekrar üreme, çaprazlama ve mutasyon operatörlerini uygula.
4. Oluşturulan her yeni kromozomun amaç fonksiyonu değerlerini bul.
5. Amaç fonksiyonu değerleri kötü olan kromozomlar popülasyondan çıkar.
6. 3-5 arasındaki adımlar tekrar et.

# Akış Diyagramı



- 
- Kodlama Türleri
  - Çaprazlama Türleri
  - Mutasyon Türleri



# GA terimleri

## ■ Gen

- Kalıtsal molekülde bulunan ve organizmanın karakterlerinin tayininde rol oynayan kalıtsal birimlere denir.
- Yapay sistemlerde gen, kendi başına anlamlı bilgi taşıyan en küçük birim olarak tanımlanır.

## ■ Kromozom (Birey)

- Birden fazla genin bir araya gelerek oluşturduğu diziye denir.
- Kromozomlar, alternatif aday çözümleri gösterirler.

# GA terimleri

## ■ Popülasyon

- Kromozomlardan oluşan topluluğa denir.
- Popülasyon, geçerli alternatif çözüm kümesidir.
- Popülasyondaki birey sayısı (kromozom) genelde sabit tutulur.
- GA'da popülasyondaki birey sayısı ile ilgili genel bir kural yoktur.
- Popülasyondaki kromozom sayısı arttıkça çözüme ulaşma süresi (iterasyon sayısı) azalır.

# GA terimleri

## ■ Kodlama

- Kodlama GA'nın çok önemli bir kısmını oluşturmaktadır.
- Probleme GA uygulanmadan önce, verinin uygun şekilde kodlanması gerekmektedir.
- Kurulan genetik modelin hızlı ve güvenilir çalışması için bu kodlamanın doğru yapılması gerekmektedir.

# GA kodlama türleri

## □ İkili Kodlama

- Her kromozom ikili diziye sahiptir  $\{ 0, 1 \}$
- Bu dizideki her bit, çözümün belli karakteristiğini temsil eder veya tüm dizi bir sayıyı temsil eder.
- Kodlamada en sık kullanılan yöntemdir.
- Örnek  $\{ 10101001 \}$

# GA kodlama türleri

## □ Permütasyon Kodlama

- Düzenleme problemlerinde kullanılır.
- Burada her kromozom, sayıları bir sırada temsil etmektedir.
- Permütasyon kodlama, gezgin satıcı ve çizelgeleme problemleri için kullanışlıdır.

Kromozom A	7 8 9 4 1
Kromozom B	8 7 9 1 4

# GA kodlama türleri

## □ Değer Kodlama

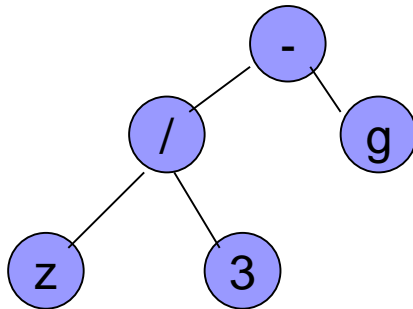
- Gerçek sayılar gibi karmaşık değerlerin kullanıldığı problemlerde, ikili kodlama zor olduğu için doğrudan değer kodlanması kullanılabilir

Kromozom A	1.2324 3.5354 4.6465 3.5556
Kromozom B	Doğu, Batı, Güney, Kuzey

# GA kodlama türleri

## □ Ağaç Kodlama

- Bu yöntem **gelişen, değişen programlar** veya ifâdeler için kullanılır.
- Örneğin GA. Ağaç kodlamada her kromozom, bâzi nesnelerin (örneğin fonksiyonlar ya da programlama dilindeki komutlar gibi) ağacıdır.



$(z/3)-g$

# Seçim türleri

## ■ Rulet-Çember Seçimi

- En basit seçim yöntemi olarak bilinmektedir.

### Uygulanışı :

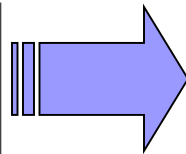
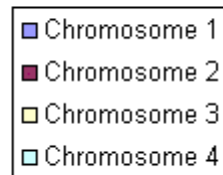
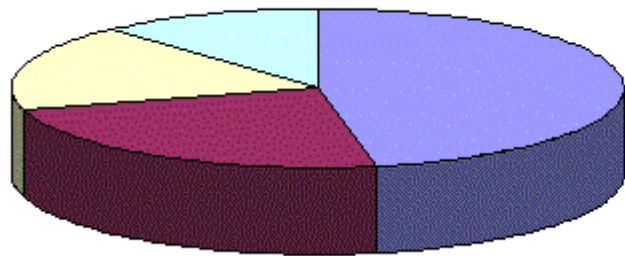
- Tüm fertler birbirine bitişik bir şekilde düz bir çizgi üzerine dizilirler.
- Her bir ferde ilişkin bölümün uzunluğu, onun uygunluk değeri kadar olur.
- Rasgele sayı üretilir ve rasgele sayı hangi bölüm içerisine gelirse, o bölümün ait olduğu fert seçilir.
- İşlem ulaşılabilecek popülasyonun gerekli adedi elde edilene kadar devam eder.



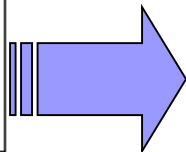
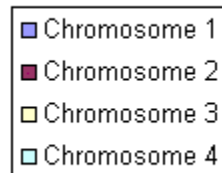
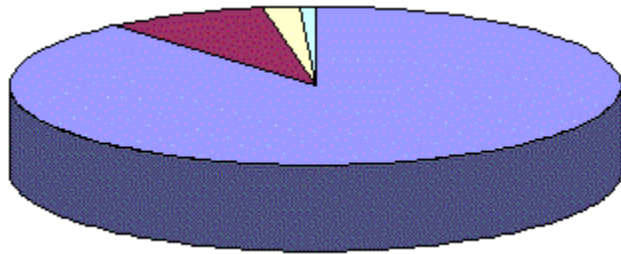
# Seçim türleri

## ■ Rank Seçimi

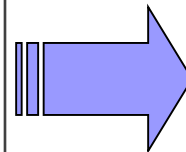
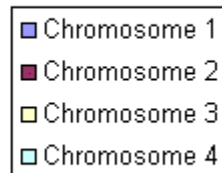
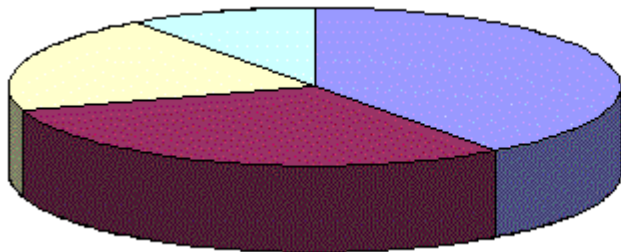
- En iyi kromozomun uygunluk değeri çok yüksek ise, Rulet-Çemberi seçim yöntemi problem yaratabilir(Sürekli yüksek olasılığa sahip kromozom seçilecek). Bu nedenle, Rank seçim yöntemi uygulanabilir.
- Popülasyon uygunluk değerine göre tersten sıralanır. Yani en iyi kromozom  $N$  adetlik bir popülasyonda  $N$  değerini alır
- Seçim bu değerlere göre yapılır.



**Rulet-Çember  
Seçimi**



**Rank Seçimi  
Öncesi**



**Rank Seçimi  
Sonrası**

# Seçim türleri

## ■ Kararlı Hal Seçimi (Yerine Geçme)

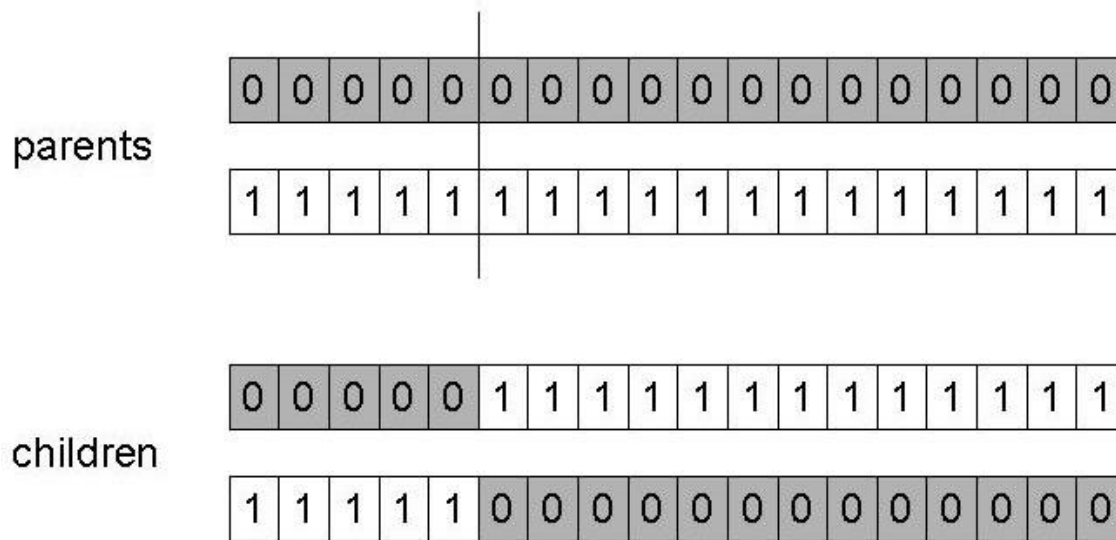
- Bu seçimin ana düşüncesi, kromozomların büyük kısmının bir sonraki nesilde hayatta kalmak zorunda olmasıdır.
- Yeni çocuklar oluşturmak için her nesilde UD yüksek birkaç kromozom seçilir.
- UD düşük bazı kromozomlar atılır ve yeni çocuk onun yerine yerleştirilir.
- Popülasyonun geri kalan kısmı yeni nesilde hayattadır.
- Bu yöntemde alt popülasyon oluşturulduktan sonra uygunluklar hesaplanır, en kötü kromozomlar yerlerini başlangıç popülasyonundaki en iyi kromozomlara bırakır.

# Çaprazlama

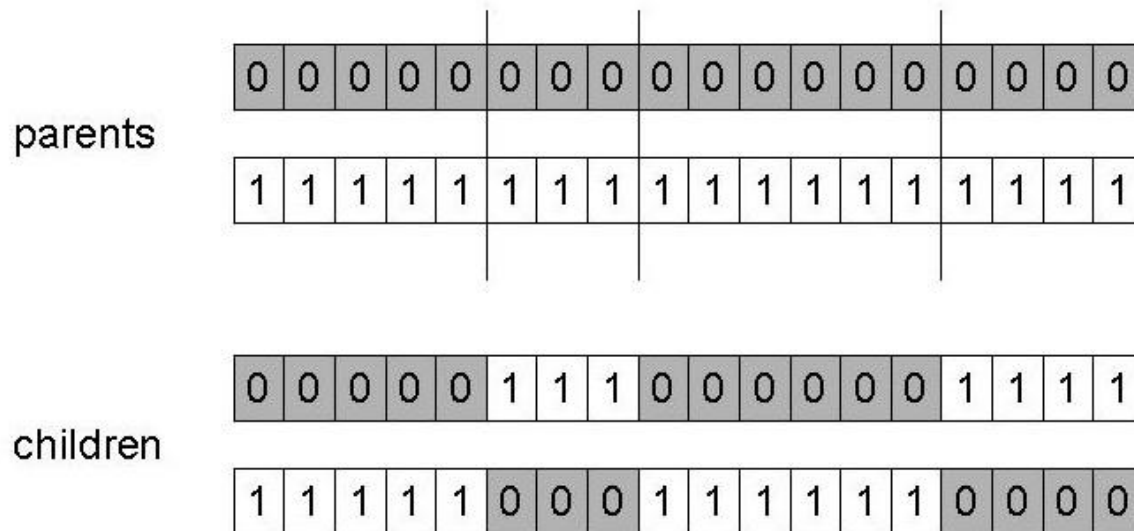
- Çaprazlama, iki kromozomun (çözümün) birbirleri arasında gen alışverişinde bulunup iki yeni kromozom oluşturmaktır.
- Genetik algorithmadaki en önemli parametrelerden bir tanesidir.
- İkili yöntemle kodlanmış değişkenlerin yaptıkları üreme faaliyeti, kromozomların çaprazlamasına benzemesi dolayısıyla böyle adlandırılmaktadır.
- Eğer kodlamada gerçek değerler kullanılıyorsa, klasik çaprazlama yöntemi yerine daha farklı yöntemler kullanılmaktadır.
- Seçim yöntemi ile yapay seçim sonucunda elde edilen yeni popülasyon dizisinden rastsal olarak iki kromozom seçilir ve karşılıklı çaprazlama işlemine tâbi tutulur.

# 1-Noktalı Çaprazlama (İkili Kod)

- $P_c$  (0.6, 0.9) aralığındadır



# N-Noktalı Çaprazlama (İkili Kod)



# Düzenli Çaprazlama (İkili Kod)

parents

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

children

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

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

# Tek Aritmetik Çaprazlama (Değer Kod)

- Atalar  $\langle x_1, \dots, x_n \rangle$  ve  $\langle y_1, \dots, y_n \rangle$
- $(k)$  gelişigüzel seçilir
- child1 :  $\langle x_1, \dots, x_k, \alpha \cdot y_k + (1 - \alpha) \cdot x_k, \dots, x_n \rangle$
- Diğer child için tersidir.  $\alpha = 0.5$  için

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.5	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----



0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

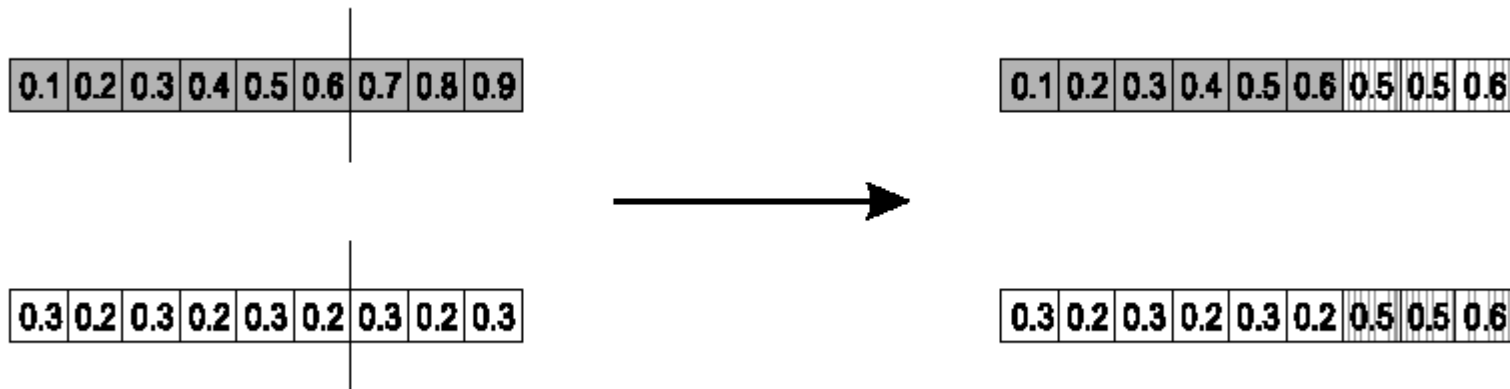
0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.5	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----



# Basit Aritmetik Çaprazlama (Değer Kod)

$$\langle x_1, \dots, x_k, \alpha \cdot y_{k+1} + (1-\alpha) \cdot x_{k+1}, \dots, \alpha \cdot y_n + (1-\alpha) \cdot x_n \rangle$$

$\alpha = 0.5$  için



# Bütün Aritmetik Çaprazlama (Değer Kod)

$$a \cdot \bar{x} + (1 - a) \cdot \bar{y}$$

$\alpha = 0.5$  için

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----



0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----

# Sıralama çaprazlama (OX) (Permutasyon Kod)

- Çaprazlama noktası seç, aynısını kopyala

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



			4	5	6	7		
--	--	--	---	---	---	---	--	--

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

- Gerisini 1,9,3,8,2 sırasıyla ikinciden al

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



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

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

# Parçasal eşleşmeli çaprazlama (PMX) (Permutasyon Kod)

■ Adım 1

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



			4	5	6	7		
--	--	--	---	---	---	---	--	--

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

■ Adım 2

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



		2	4	5	6	7		8
--	--	---	---	---	---	---	--	---

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

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

■ Adım 3

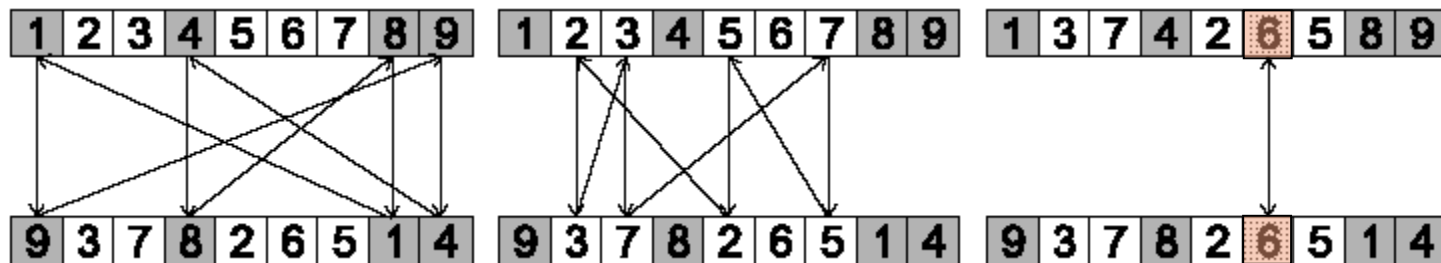


9	3	2	4	5	6	7	1	8
---	---	---	---	---	---	---	---	---

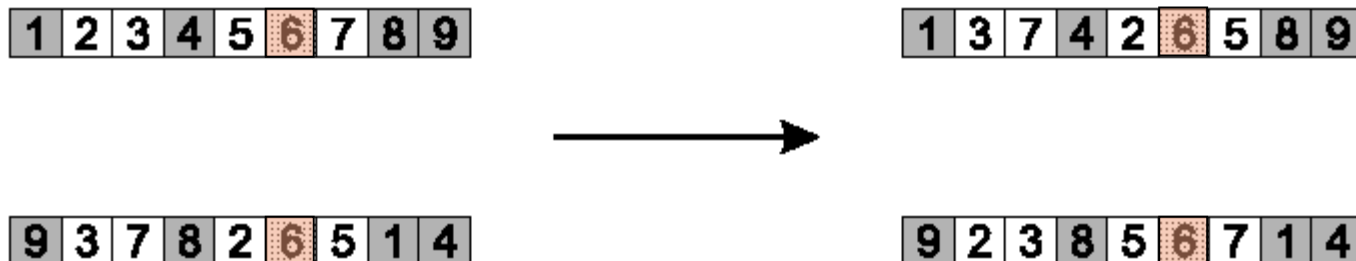
9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

# Çevrim Çaprazlama (CX) (Permutasyon Kod)

- Adım 1: Çevrimleri bul



- Adım 2: Değişimli çevrimleri çocuğa kopyala



# Mutasyon

- $\mu_m$  = mutasyon oranı (1/pop. büyük and 1/ Kromozom uzunluğu)
  - Mutasyon Oranı, mutasyon olasılığını gösteren orandır. Amaç mevcut kromozomların genlerinin bir ya da birkaçını değiştirerek yeni kromozomlar elde etmektir.
  - Genelde mutasyon olasılığı (0.01 gibi) düşük tutulmaktadır.
  - Bu nedenle mutasyon etkileri kromozomlarda az görülmektedir.
  - Mutasyon sırasında kromozomdaki gen sayısı değişmez, sabit kalır.
  - Mutasyon yapılmasının bir başka amacı da, popülasyondaki çeşitliliğin azalmamasını sağlamaktır

# İkili kodda mutasyon

parent

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

child

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

# Değer kodda mutasyon

$$\bar{x} = \langle x_1, \dots, x_l \rangle \rightarrow \bar{x}' = \langle x'_1, \dots, x'_l \rangle$$

$$x_i, x'_i \in [LB_i, UB_i]$$



# Permutasyon kodda mutasyon

Ekleme mutasyon

1 2 3 4 5 6 7 8 9



1 2 5 3 4 6 7 8 9

Yer değiştirme mutasyon

1 2 3 4 5 6 7 8 9



1 5 3 4 2 6 7 8 9

Ters mutasyon

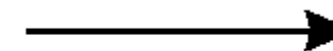
1 2 3 4 5 6 7 8 9



1 5 4 3 2 6 7 8 9

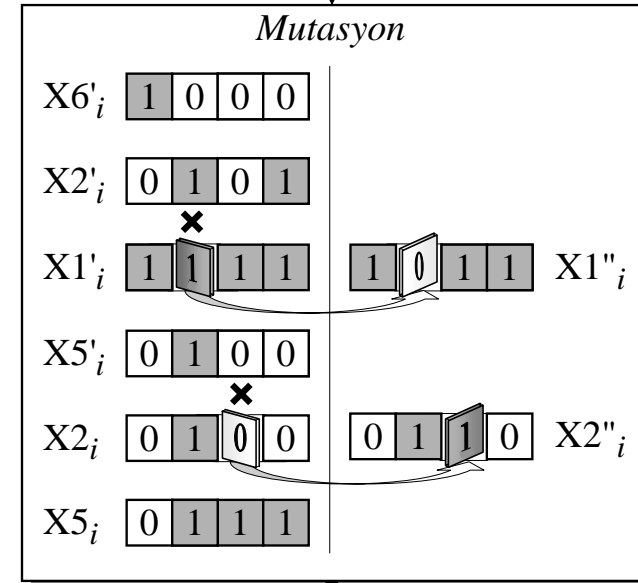
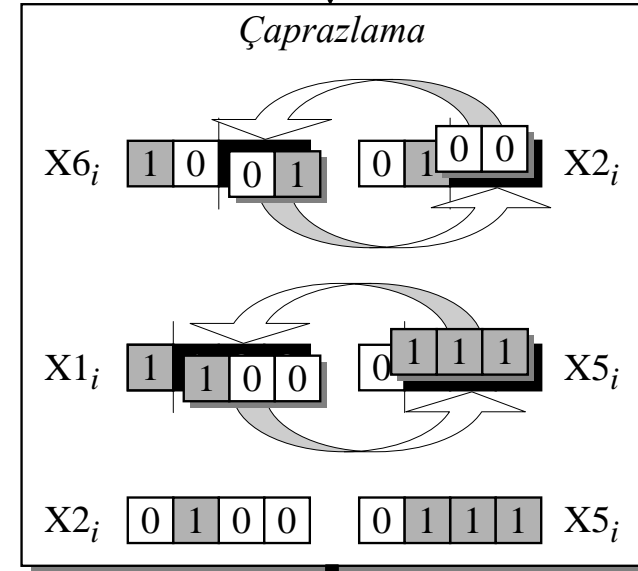
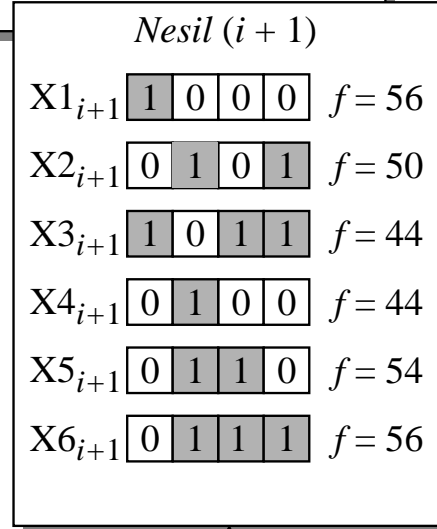
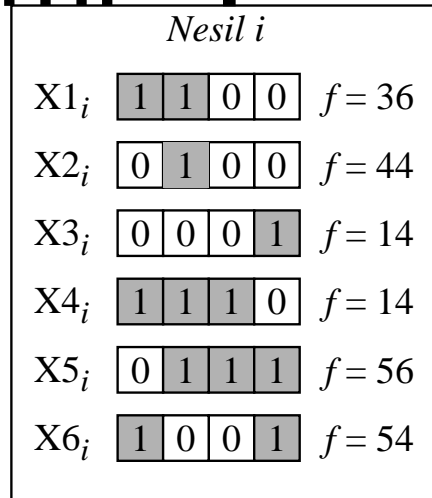
Karışık mutasyon

1 2 3 4 5 6 7 8 9



1 3 5 4 2 6 7 8 9

# GA Çevrimi



# GA Örnek

■  $\max x^2 \ x=\{0,1,\dots,31\}$

String no.	Initial population	$x$ Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

# Çaprazlama

String no.	Mating pool	Crossover point	Offspring after xover	$x$ Value	Fitness $f(x) = x^2$
1	0 1 1 0   1	4	0 1 1 0 0	12	144
2	1 1 0 0   0	4	1 1 0 0 1	25	625
2	1 1   0 0 0	2	1 1 0 1 1	27	729
4	1 0   0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

# Mutasyon

String no.	Offspring after xover	Offspring after mutation	$x$ Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

# İÇERİK

- Optimizasyon
- Genetik Algoritmalar
- **Parçacık Sürü Optimizasyonu**

# Parçacık Sürü Optimizasyonu (PSO)

- Kuş, balık ve hayvan sürülerinin bir “bilgi paylaşma” yaklaşımı uygulayarak çevrelerine adapte olabilme, zengin yiyecek kaynağı bulabilme ve avcılardan kaçabilme yeteneklerinden esinlenmiştir.



# PSO



## ■ Parçacıklar

- Mevcut pozisyon, mevcut hız, bilgiyi değiştirme, bir önceki pozisyonu hatırlama (kişisel en iyi pozisyon), bilgiyi kullanma

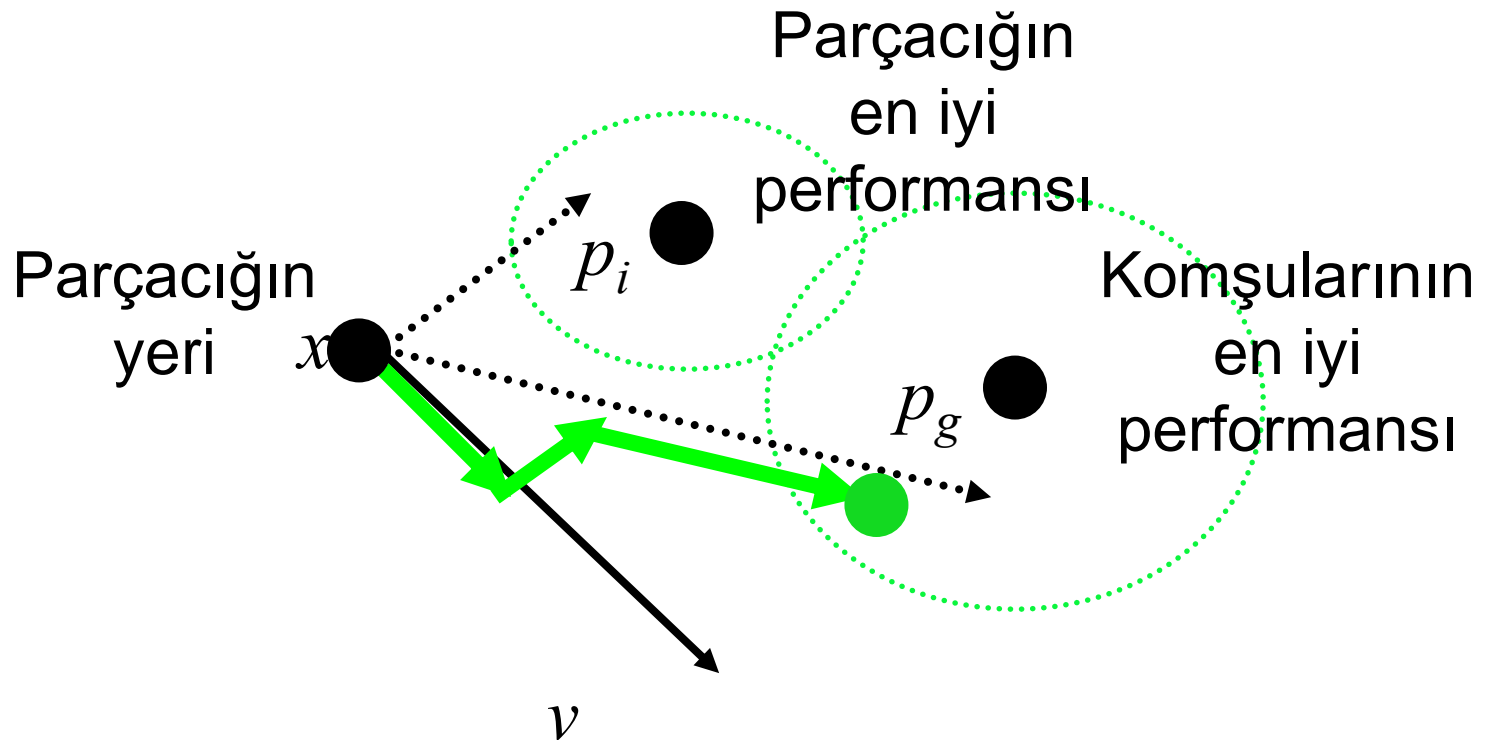
## ■ Sürü

## ■ Bireyler arasındaki bilgi paylaşımı

- Sürüdeki en iyi pozisyon + bir önceki tecrübe



# Uzlaşma



# Orijinal Algoritma

$x_{i,j}$  ve  $v_{i,j}$ 'ye gelişigüzel başlangıç değerleri vererek sürüyü oluştur

**Do**

**For**  $i = 1$  **to** Parçacık sayısı

**if**  $f(x_i) < f(y_i)$  **then**  $y_i = x_i$  // lokal (kişisel) en iyiyi güncelle

$\hat{y}_i = \min(x_{komşular})$  // global en iyiyi güncelle

**For**  $j = 1$  **to** Optimize edilen boyut sayısı

$v_{i,j} = v_{i,j} + c_1 r_{1,j} [y_{i,j} - x_{i,j}] + c_2 r_{2,j} [\hat{y}_j - x_{i,j}]$  // hız vektörünü güncelle

$x_{i,j} = x_{i,j} + v_{i,j}$  // pozisyon vektörünü güncelle

**Next**  $j$

**Next**  $i$

**Until** Sonlandırma kriteri

**Döngüdeki  
Rasgelelik**

# PSO'ya Modifikasyonlar

- *global – lokal* modeller (hızlı sonuç - iyileştirme)
- **İKİLİ PSO**
  - Melez sürü (ikili-sürekli sürü)
- **YAKINSAMA ORANI GELİŞTİRMELERİ**
  - Atalet Ağırlığı
  - Bulanık Atalet Ağırlığı
  - Sınırlama Faktörü
  - Seçim
  - Zamanla Değişen Hızlanma Katsayıları
  - Yetiştirme
  - Yakınsama Garantili PSO (GCPSO)
    - Global en iyi parçacık için yeni bir hız güncelleme denklemi

# PSO'ya Modifikasyonlar

## ■ ÇEŞİTLİLİK ARTTIRMA GELİŞTİRMELERİ

- ☐ Uzaysal Komşuluklar
- ☐ Komşuluk Topolojileri
- ☐ Sosyal Tespit
- ☐ Alt Populasyonlar
- ☐ Çoklu Başlatmalı PSO
- ☐ Mutasyon
- ☐ Çekici ve İtici PSO
- ☐ Dağıtan PSO
- ☐ Diferansiyel Gelişimli PSO
- ☐ Yaşam Çevrimi Modeli
- ☐ Öz-örgütlenmiş Tehlikelilik
- ☐ Uygunluk-Uzaklık Oran Tabanlı PSO

## ■ PARALEL PSO

# SORULAR?

