



**YZM 3217 – YAPAY ZEKA**  
**DERs#13: GENETİK ALGORİTMA**

# Sınıflandırma Yöntemleri:

- Karar Ağaçları (Decision Trees)
- Örnek Tabanlı Yöntemler (Instance Based Methods): k en yakın komşu (k nearest neighbor)
- Bayes Sınıflandırıcı (Bayes Classifier)
- Yapay Sinir Ağları (Artificial Neural Networks)
- Genetik Algoritmalar (Genetic Algorithms)
- ...

# Genetik Algoritmalar

- Yapay zekanın bir araştırma alanıdır ve birçok alanda kullanılmaktadır.
- Darwin'in doğal seçim ve evrim teorisi ilkelerine dayanan bir **arama** ve **optimizasyon** yöntemidir.
- Genetik algoritmaların temelleri 1970'li yıllarda John HOLLAND tarafından ortaya atılmıştır.

# Dođal Seęim

- Canlılar, dođada yařamlarını sŸrdŸrmek ięin savařırlar ve gŸęlŸ olanlar savařı kazanır.
- Yeni bir nesil ięin **iyi bireyler** seęilir ve bunlardan ęocuklar Ÿretilir.
- ęocuklar anne ve babalarına benzer ama aynı deęillerdir. Bu nedenle anne ve babalarından **daha kŸtŸ** veya **daha iyi** olabilirler.
- Amaę **iyi bir kuřak** oluřturmaktır.

# Neden Genetik Algoritmalar?

- Bazı problemlerde sonsuz sayıda çözüm olabilir ama **en uygun** ve **hızlı çözümü** bulmaya ihtiyacımız vardır.

# Biyolojik bilgiler

- **Gen:** Canlıların karakterlerini belirleyen ve anlamlı bilgi içeren en küçük birimlerdir.
- **Kromozom (Birey):** Birden fazla genin bir araya gelerek oluşturduğu diziye denir.
- **Popülasyon:** Kromozomlardan oluşan topluluktur.

# Genetik Algoritma (GA)

## Algoritmanın Adımları:

1. Uygunluk (**fitness**) fonksiyonunun tanımlanması
2. Kodlama (**genetik kodlama**)
3. Rastgele bireylerden oluşacak şekilde **başlangıç popülasyonunun** seçilmesi

# Genetik Algoritma (GA)

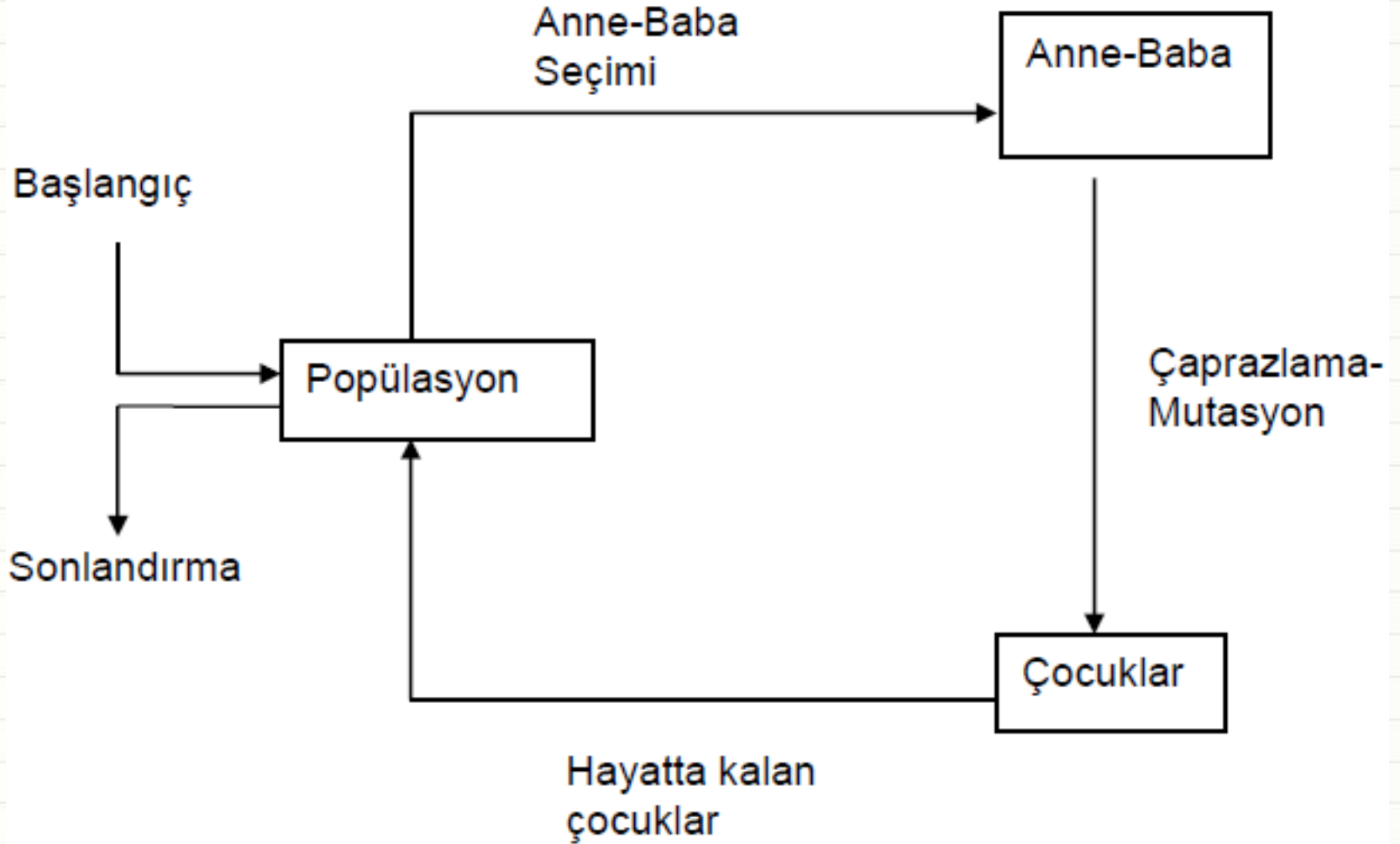
4. Tekrarla (yeterince iyi bir çözüm bulunana kadar)

- Popülasyondaki bütün bireylerin uygunluk fonksiyonunun hesaplanması
- Yeni nesil için **en iyi bireylerin** seçilmesi
- **Çaprazlama** ve **mutasyon** ile yeni neslin oluşturulması
- Yeni nesli (kromozomları) popülasyona ekle

5. En iyi çözümü döndür.



# GA Şematik Gösterimi



# Başlangıç Popülasyonun Oluşturulması

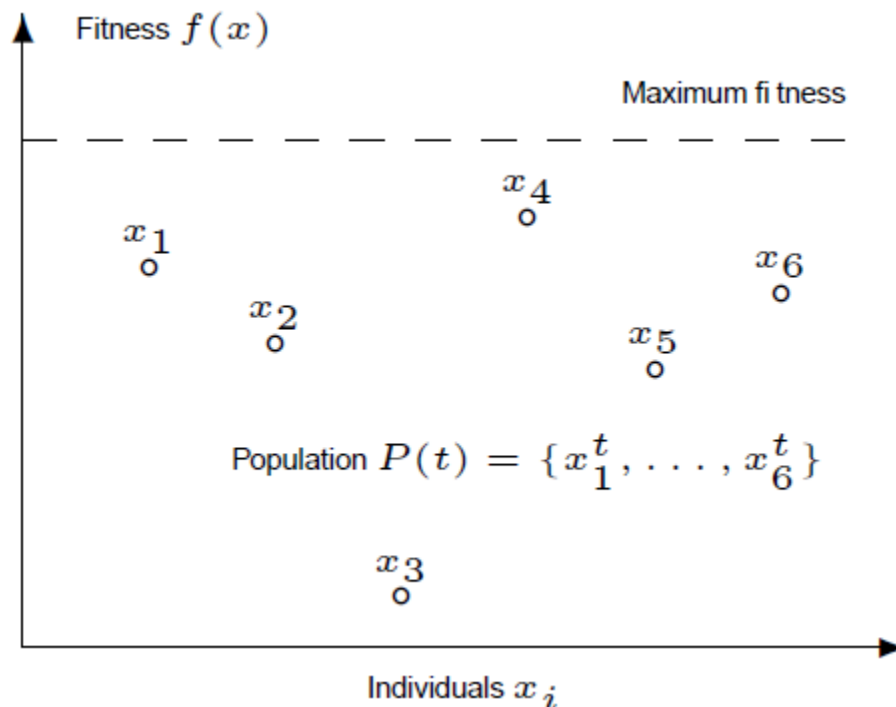
- Başlangıçta belirli kriterlere göre rastgele bireylerden oluşan bir popülasyon oluşturulur.
  - Popülasyondaki bireylerin sayısı **küçük** seçildiğinde iterasyonlar daha hızlı olur ancak algoritmanın yerel optimuma takılma şansı artar.
  - Birey sayısının çok **fazla** olması ise çözüm kalitesini arttırır ancak algoritmanın adımları daha uzun zaman alır.
  - $n$ : kromozom uzunluğu  $\Rightarrow [n, 2n]$

# Uygunluk(Fitness) Fonksiyonu

- Farklı çözümlerin (**kromozomların**) karşılaştırılması ve iyi olanların seçimi için gereklidir.
- $f(X1) > f(X2)$  ise  
X1, X2'den **daha iyidir.**

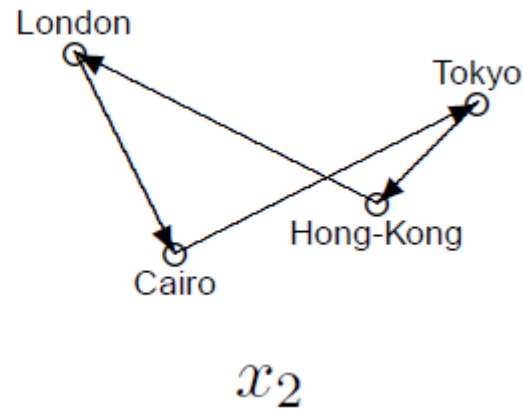
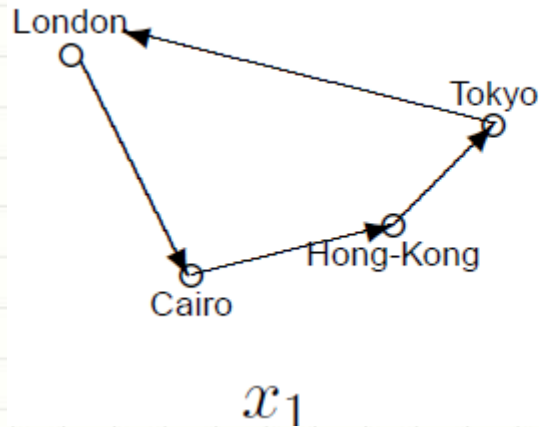
# Uygunluk(Fitness) Fonksiyonu

- Bir popülasyon oluşturulduktan sonra, popülasyondaki her bireyin uygunluk (fitness) değeri hesaplanır.



# Örnek: Gezgin Satıcı Problemi

- Gezi için birçok yol seçeneği vardır:
  - Uygunluk fonksiyonu:  $f(x) = \min(\text{uzunluk})$
  - $f(X1) < f(X2)$  olduğu için  $X1$ ,  $X2$ 'den daha iyi bir çözümdür. Bu nedenle  $X1$  seçilir.

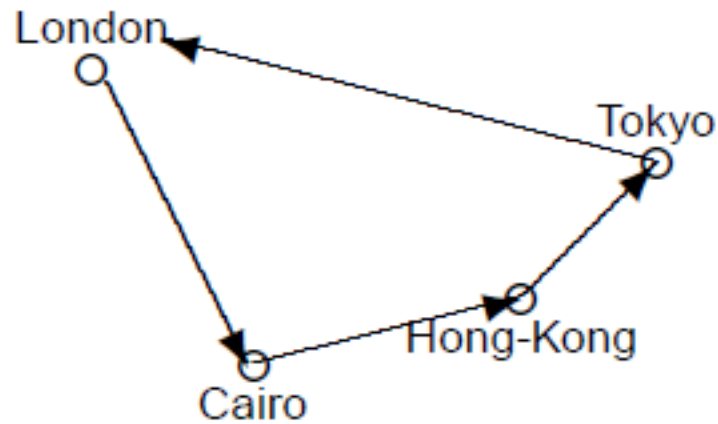


# Genetik Kodlama

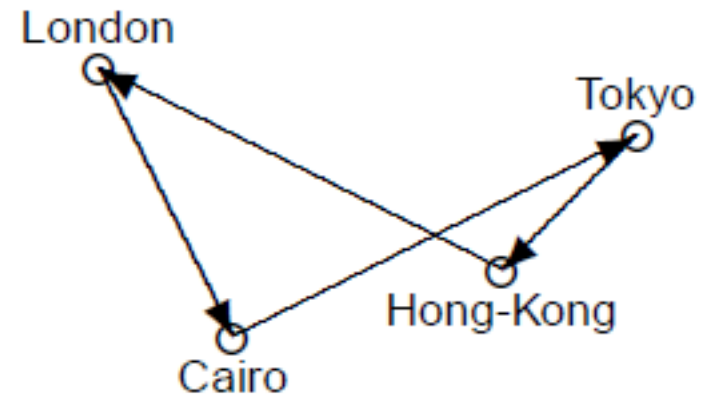
- Kodlama genetik algoritmanın önemli bir aşamasıdır.
- GA uygulamasına başlamadan önce, verilerin uygun şekilde kodlanması gerekir.
- Geliştirilen modelin hızlı ve güvenilir çalışması için bu kodlamanın doğru yapılması gerekir.

# Örnek kodlama

- Gezgin satıcı problemi için  
L: London, C: Cairo, H: Hong-Kong, T: Tokyo



$$x_1 = LCHTL$$



$$x_2 = LCTHL$$

# Kodlama Çeşitleri

1. İkili (binary) kodlama
2. Permütasyon kodlama
3. Değer kodlama
4. Ağaç kodlama



# İkili (Binary) kodlama

- En çok kullanılan kodlama yöntemidir. Basit, kolay ve hızlı işlemler için uygundur.
- Kromozomların tanımlanması genellikle ikili sayı sistemi ile (0,1) yapılıır.
- **Örnek:**
  - Kromozom1: 1101100100110110
  - Kromozom2: 1101111000011110

# Permütasyon kodlama

- Sıralamaya veya düzenlemeye yönelik problemler için uygundur.
- Gezgin satıcı problemi (TSP) veya görev sıralama gibi problemlerin çözümü için kullanılır.
- **Örnek:**
  - Kromozom-A: 1 5 3 2 6 4 7 9 8
  - Kromozom-B: 8 5 6 7 2 3 1 4 9

# Değer kodlama

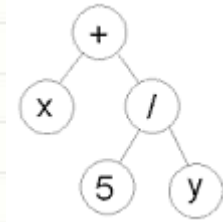
- Gerçek sayılar gibi karmaşık verilerin kullanıldığı problemlerde kullanılır.
- Değerler; gerçek sayılar, karakterler veya nesnelere olabilir.
- Bu tip kodlamada probleme özgü yeni çaprazlama ve mutasyon yöntemleri geliştirmek gereklidir.

Kromozom A	1.2324 5.3243 0.4556 2.3293 2.4545
Kromozom B	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom C	(geri), (geri), (sağ), (ileri), (sol)

# Ağaç kodlama

- İfadesele çözümler içeren problemlerde kullanılır, her kromozom bir nesnenin ağacıdır.

- Örneğin, işlevler veya programlama dilindeki komutlar gibi.



- Evrimleşen programlar veya ağaç şeklinde kodlanabilecek herhangi diğere yapılar için uygundur.
  - Örnek: **LISP**

# Seçim

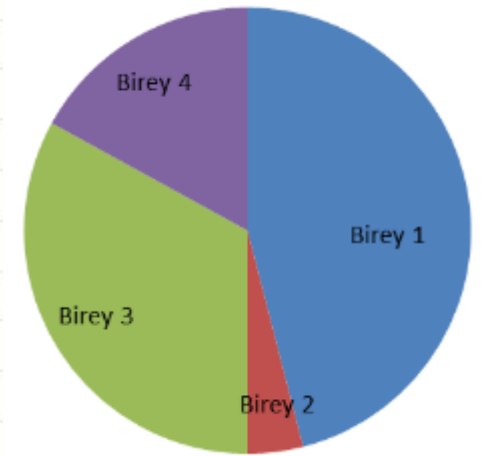
- Popülasyondaki bütün bireylerin uygunluk fonksiyonu hesaplandıktan sonra, bunlardan bazıları yeni kuşaklar üretmek için seçilirler
- Seçim işlemi için bazı yöntemler:
  - Rulet yöntemi
  - Sabit durum yöntemi
  - Turnuva yöntemi
- Genelde, popülasyonun büyüklüğünün değişmesine (**artmasına**) izin verilmez.

# Rulet yöntemi

- Kromozomlar uygunluklarına göre seçilirler.
- Daha iyi kromozomlar, daha fazla seçilme şansına sahip olanlardır.
  - Popülasyondaki tüm kromozomların yerleştirildiği bir rulet tekerini hayal edelim.
  - Rulet tekeri üzerindeki kromozomun yerinin boyutu kromozomun uygunluğuyla orantılıdır.
  - Daha uygun olan kromozom daha geniş bir kısma sahip olur.

# Örnek: Rulet yöntemi

- Uygunluk fonksiyonu:  $f(x)=x^2$ 
  - Birey 1: 1101,  $x = 13$ ,  $x^2 = 169$
  - Birey 2: 0100,  $x = 4$ ,  $x^2 = 16$
  - Birey 3: 1011,  $x = 11$ ,  $x^2 = 121$
  - Birey 4: 1000,  $x = 8$ ,  $x^2 = 64$



- Toplam =  $169+16+121+64 = 370$ 
  - Birey 1:  $169/370 = \%46$
  - Birey 2:  $16/370 = \%4$
  - Birey 3:  $121/370 = \%33$
  - Birey 4:  $64/370 = \%17$

Rulet 1 defa çevrildiğinde gelme ihtimalleri

# Rulet yöntemi

- Eğer uygunluk değerleri arasında büyük farklar varsa problemler ortaya çıkacaktır.
- Örneğin, eğer en iyi kromozomun uygunluğu diğer tüm kromozomların toplamının %90'ı ise diğer kromozomların seçilme şansı çok azalacaktır.
- Bunu önlemek için sıralı seçim kullanılabilir.



# Sıralı seçim

- Sıralı seçimde en kötü uyumlulukta olan kromozoma 1 değeri, sonrakine 2 değeri, ..., sonuncuya N (birey sayısı) verilir
- Böylelikle seçilmede bunlara öncelik tanınmış olur. Bu şekilde onların da seçilme şansı artar
- Ancak bu yöntem, çözümün daha geç yakınsamasına neden olabilir.

# Sabit durum yöntemi

- Her yeni nesilde yüksek uygunluk değerine sahip kromozomlar yeni çocukları oluşturmak için seçilir.
- Düşük uygunluk değerine sahip çocuklar kaldırılarak yerlerine bu yeni oluşturulan çocuklar konur.
- Toplumun geri kalan kısmı da aynen yeni nesle aktarılır.

# Çaprazlama

- Anne ve babadaki bazı genlerin yer değiştirmesi ile yeni çocukların oluşturulma işlemidir.
- Örnek: Tek nokta çaprazlama
  - P: Parents (anne-baba)
  - O: Offspring (çocuk)

$$\begin{array}{l} P_1 = \quad 101 \mid 0010 \\ P_2 = \quad 011 \mid 1001 \end{array} \Rightarrow \begin{array}{l} O_1 = \quad 101 \mid 1001 \\ O_2 = \quad 011 \mid 0010 \end{array}$$

# 2 noktalı çaprazlama

1101100100110110

1101111000011110

1101111000010110

1101100100111110

# Mutasyon

- Rastgele seçilen bir genin değiştirilmesi:  
Örneğin, bit değerini diğer bit değerine dönüştürme (0 veya 1).
- Sürekli yeni nesil üretimi sonucunda, belli bir süre sonra nesildeki kromozomlar tekrar edebilir ve farklı kromozom üretimi azalabilir.
- Nesildeki kromozom çeşitliliğini artırmak için kromozomlardan bazılarına mutasyona işlemi uygulanır.

# Mutasyon

- Örneğin  $O_1$ 'deki 3 ve 5 nolu bitlerdeki mutasyon işlemi şu şekildedir.

$$O_1 = 10\underline{1}1\underline{0}01 \Rightarrow O_1 = 10\underline{0}1\underline{1}01$$

- Çaprazlama ve mutasyon işlemlerinden sonra, yeni nesil bireyler, anne ve babalarından daha iyi olabilirler.

# Diğer Mutasyon Yöntemleri

- **Öteleme Yöntemi:** Dizi içerisinde rassal olarak belirlenen bir blok genin yine aynı dizi içerisinde rastgele bir konuma yerleştirilmesidir.

– P1 = AKDEMZIF

– P1' = AEMZIKDF

# Diğer Mutasyon Yöntemleri

- **Yerleştirme Yöntemi:** Dizi içinde rastgele bir genin seçilerek, yine aynı dizi içinde rastgele başka bir konuma yerleştirilmesidir.

– P1 = AKDEMZIF

– P1' = ADEMZKIF



# Diğer Mutasyon Yöntemleri

- **Karşılıklı Değişim Yöntemi:** Dizi içerisinde rassal olarak iki gen seçilmekte ve bu iki genler birbirleriyle yer değiştirmektedir.

– P1 = AKDEMZIF

– P1' = AZDEMKIF

# Genetik Algoritma Parametreleri

- Çaprazlama Olasılığı
  - Çaprazlamanın ne sıklıkla yapılacağını belirtir.
- Mutasyon Olasılığı
  - Kromozom parçalarının ne sıklıkla mutasyon geçireceğini belirtir.
- Popülasyon Büyüklüğü
  - Toplumdaki birey sayısını belirtir.
- Uygunluk Fonksiyonu

# Genetik Algoritma Özellikleri

- **Avantajları**

- Çok amaçlı optimizasyon yöntemleri ile kullanılabilmesi
- Çok karmaşık ortamlara uyarlanması
- Kısa sürelerde iyi sonuçlar verebilmesi

- **Dezavantajları**

- Son kullanıcının modeli anlaması güç
- Problemi GA ile çözmeye uygun hale getirmek zor
- Uygunluk fonksiyonunu belirlemek zor
- Çaprazlama ve mutasyon tekniklerini belirlemek zor

# Genetik Algoritma Örneği

- Elimizde 6 genden oluşan kromozomlar var:
  - $x_i = abcdef$
  - Her gen 0 ile 9 arasında numaralandırılmıştır.
- 4 kromozomdan oluşan popülasyonumuz var:

$$x_1 = 435216$$

$$x_2 = 173965$$

$$x_3 = 248012$$

$$x_4 = 908123$$

# Genetik Algoritma Örneği

- Uygunluk (fitness) fonksiyonu aşağıda verilmiştir:

$$f(x) = (a + c + e) - (b + d + f)$$

# Genetik Algoritma Örneği

- İstenecekler:
  - Kromozomları uygunluk fonksiyonuna göre sıralayınız.
  - 3. ve 4. kromozomları tam orta noktalarından ayırarak çaprazlama yapınız.
  - 1. ve 2. kromozomların 2. ve 4. bitlerinden ayırarak 2 noktalı çaprazlama yapınız.
  - Yeni neslin uygunluk fonksiyonunu hesaplayınız.

# Genetik Algoritma Örneği

- Uygunluk fonksiyonunu uygulayalım:

$$f(x) = (a + c + e) - (b + d + f):$$

$$f(x_1) = (4 + 5 + 1) - (3 + 2 + 6) = -1$$

$$f(x_2) = (1 + 3 + 6) - (7 + 9 + 5) = -11$$

$$f(x_3) = (2 + 8 + 1) - (4 + 0 + 2) = 5$$

$$f(x_4) = (9 + 8 + 2) - (0 + 1 + 3) = 15$$

Uygunluk fonksiyonuna göre sıralama **X4**, **X3**, **X1** ve **X2**

# Genetik Algoritma Örneği

- 3. ve 4. kromozomları tam orta noktalarından ayırarak çaprazlama:

$$\begin{array}{l} x_4 = \mathbf{908} \mid \mathbf{123} \\ x_3 = \mathbf{248} \mid \mathbf{012} \end{array} \Rightarrow \begin{array}{l} O_1 = \mathbf{908012} \\ O_2 = \mathbf{248123} \end{array}$$

- 1. ve 2. kromozomların 2. ve 4. bitlerinden ayırarak 2 noktalı çaprazlama:

$$\begin{array}{l} x_3 = \mathbf{24} \mid \mathbf{80} \mid \mathbf{12} \\ x_1 = \mathbf{43} \mid \mathbf{52} \mid \mathbf{16} \end{array} \Rightarrow \begin{array}{l} O_3 = \mathbf{245212} \\ O_4 = \mathbf{438016} \end{array}$$



# Genetik Algoritma Örneği

- Yeni neslin uygunluk fonksiyonunun hesaplanması:

$$f(x) = (a + c + e) - (b + d + f):$$

$$f(O_1) = (9 + 8 + 1) - (0 + 0 + 2) = 16$$

$$f(O_2) = (2 + 8 + 2) - (4 + 1 + 3) = 4$$

$$f(O_3) = (2 + 5 + 1) - (4 + 2 + 2) = 0$$

$$f(O_4) = (4 + 8 + 1) - (3 + 0 + 6) = 4$$