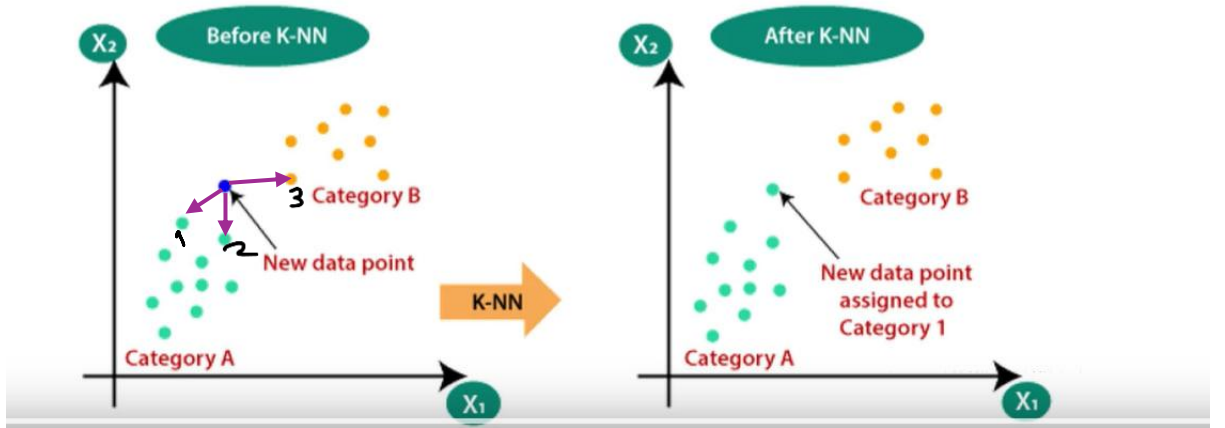


Yapay Zeka Final Tekrar

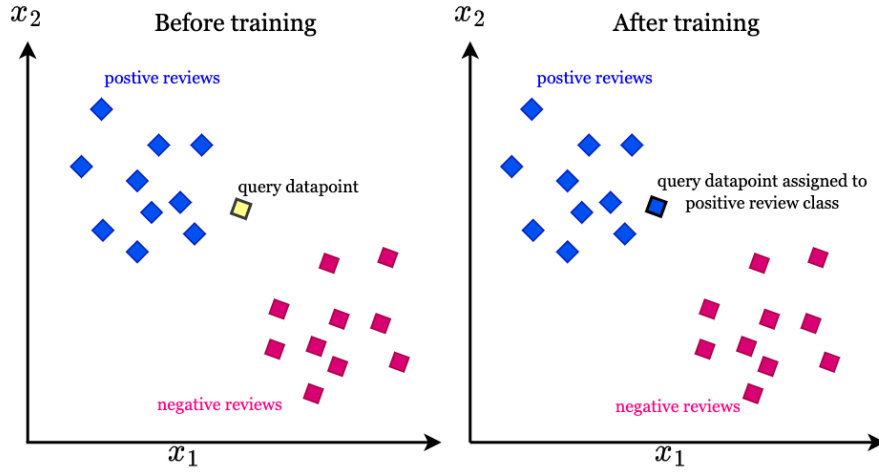
Makine Öğrenmesi Algoritmaları

1) KNN Algoritması:

- k değeri en yakın komşu sayısını ifade eder.
- k değeri genelde 5 seçilir. Çok fazla seçilmesi overfittinge yol açabilir.

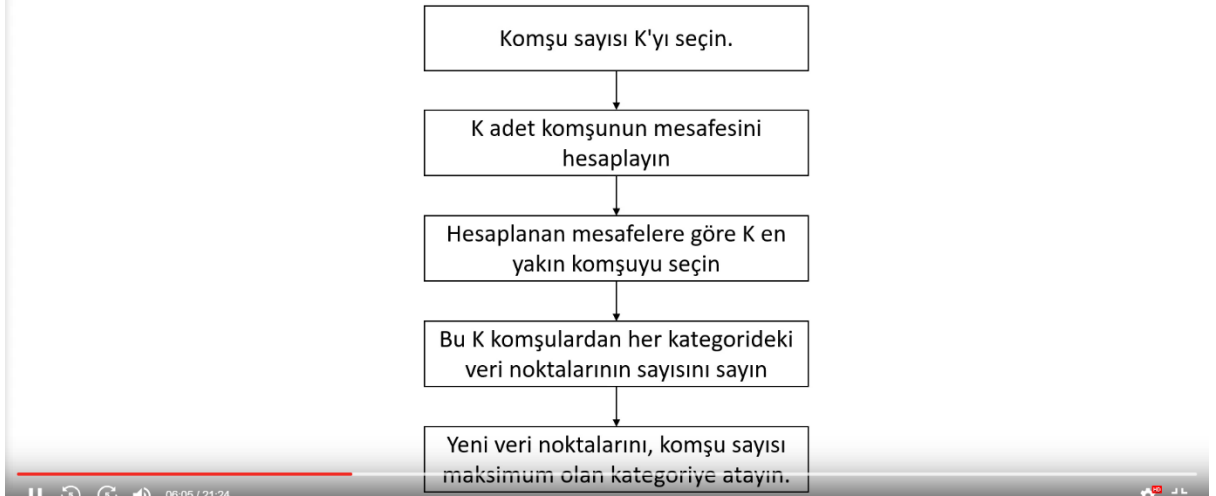


Örneğimizde k değerini 3 seçtiğimizi varsayalım. New data pointimize uygun kategoriyi bulmak istiyoruz. k değerini 3 seçtik. Bu yüzden veri noktasına en yakın 3 veriyi buluyoruz. Bu veri noktaları 1,2 ve 3 ile numaralandırılan veri noktalarıdır. Şimdi ise veri noktalarının hangi kategoriye ait olduklarını inceleyelim. 1 ve 2 numaralı veri noktası görüldüğü üzere A kategorisine, 3 numaralı veri noktası ise B kategorisine aittir. A kategorisine ait veri noktaları daha fazla olduğu için new data point ile gösterilen verimiz için A kategorisine ait diyebiliriz.



- En yakın komşuların belirlenmesi için mesafe ölçümünde genelde Öklid mesafesi formülü kullanılır.

KNN Akış Şeması



Yukarıdaki görselde KNN akış şemasını görmekteyiz.

- Küçük veri setlerinde k değeri genelde veri sayısının karekökü olarak seçilebilir. k değerinin fazla seçilmesi fazla öğrenmeye eksik seçilmesi ise eksik öğrenmeye neden olabilir.

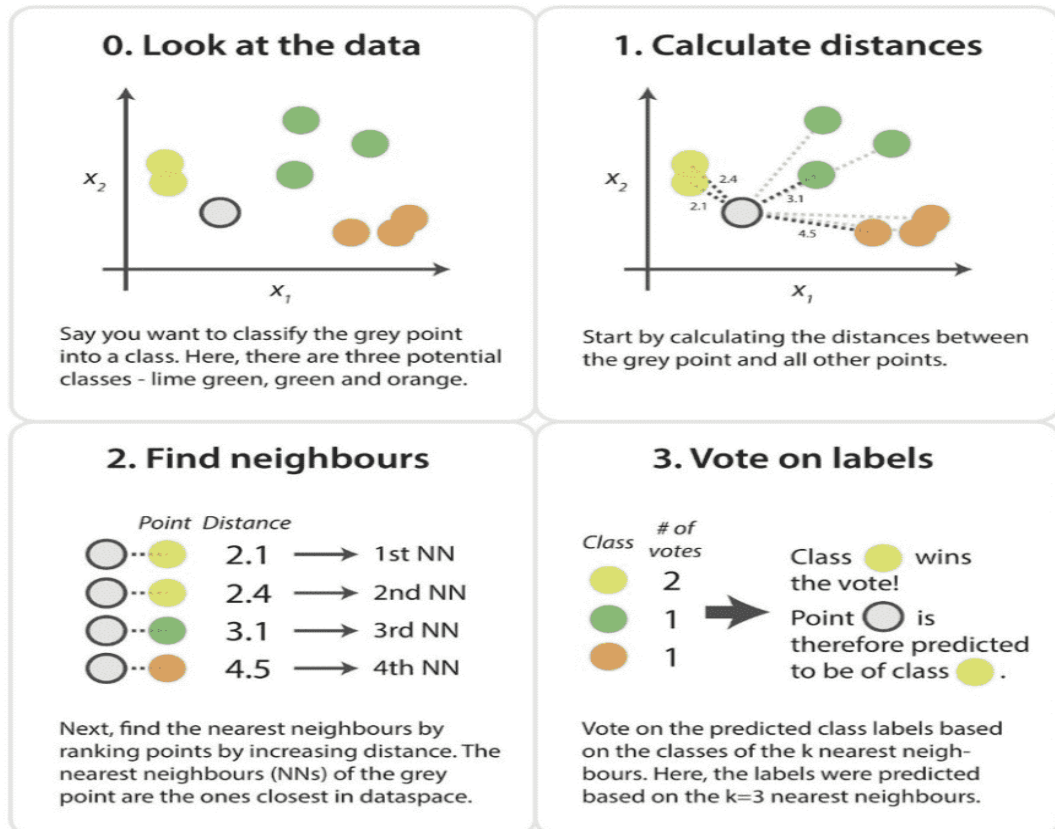
KNN Algoritmasının Avantajları:

- Temelde 2 parametresi (k ve mesafe) olduğu için uygulaması basittir.
- Algoritma çok yönlüdür.(Sınıflandırma , Regresyon)
- Eğitim aşamasında bir şey öğrenmez bu yüzden hızlıdır.

KNN Algoritmasının Dezavantajları:

- Büyük veri gruplarında algoritma hem zaman alıcıdır, hemde çok fazla kaynak tüketir. Daha fazla bellek ve depolama birimine ihtiyaç duyar.
- En uygun k değerini seçmek zordur.
- Anlaşılması ve uygulanması kolaydır. Fakat büyük veri gruplarında problem çıkardığı için fazla tercih edilmez.

K-Nearest Neighbors (KNN), hem sınıflandırma hem de regresyon problemlerinde kullanılan, basit ve etkili bir makine öğrenimi algoritmasıdır. Model, verileri eğitmez; bunun yerine, test sırasında en yakın komşulara dayalı tahminler yapar. Model, test verisi ile tüm eğitim verileri arasındaki mesafeleri hesaplayarak en yakın komşuyu seçer ve bu komşuların oy çokluğuna (sınıflandırma) veya ortalamasına (regresyon) dayalı tahmin yapar. En yaygın kullanılan mesafe ölçümü Öklid mesafesi olmakla birlikte, Manhattan, Minkowski veya Kosinüs Benzerliği gibi alternatifler de kullanılabilir. KNN, eğitim süresi gerektirmeyen bir "lazy learner" olup, basitliği ve açıklanabilirliği ile avantaj sağlarken, yüksek boyutlu veri setlerinde hesaplama maliyetinin artması ve ölçek duyarlılığı gibi dezavantajlara sahiptir. k değeri, genellikle çapraz doğrulama ile belirlenir ve model performansını doğrudan etkiler. Algoritma; görüntü işleme, sahtecilik tespiti ve medikal teşhis gibi birçok alanda yaygın olarak kullanılmaktadır.



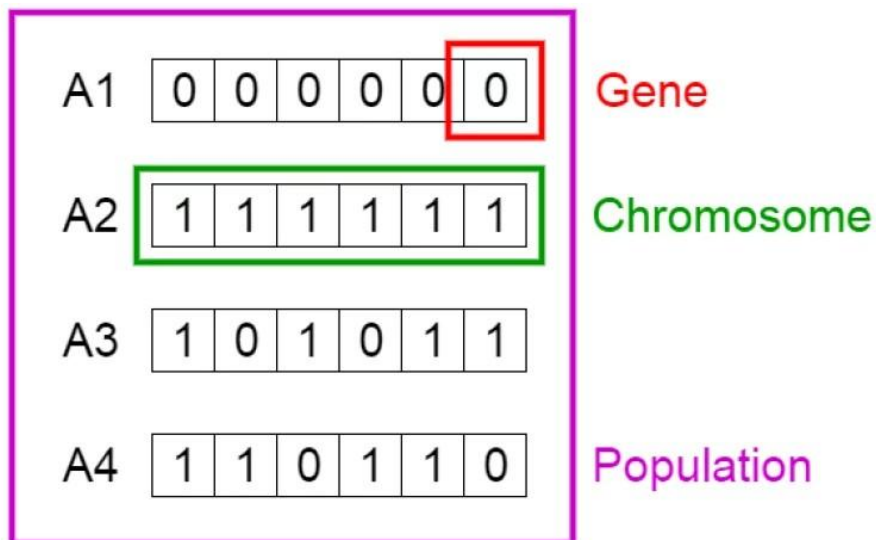
2) Karar Ağacı Algoritması:

Genetik Algoritma

- Darwinin doğal seçim ve evrim teorisine dayanan bir arama ve optimizasyon yöntemidir.
- Bazı problemler binlerce çözüme sahip olabilir fakat en hızlı ve en uygununu bulmak zorundayız.
- Gerçek hayatta karşılaştığımız bir problemi (kuru yük gemisine konteynerler nasıl yerleştirilmeli, bir noktadan başka bir noktaya nasıl gidilebilir ya da en uygun teslimat rotası oluşturma gibi), biz bir arama problemine dönüştürebilsek, bu problemi Genetik Algoritma ile çözebiliriz.

-

-



-

Genetik Algoritmanın Adımları

1) Uygunluk Fonksiyonunun Tanımlanması

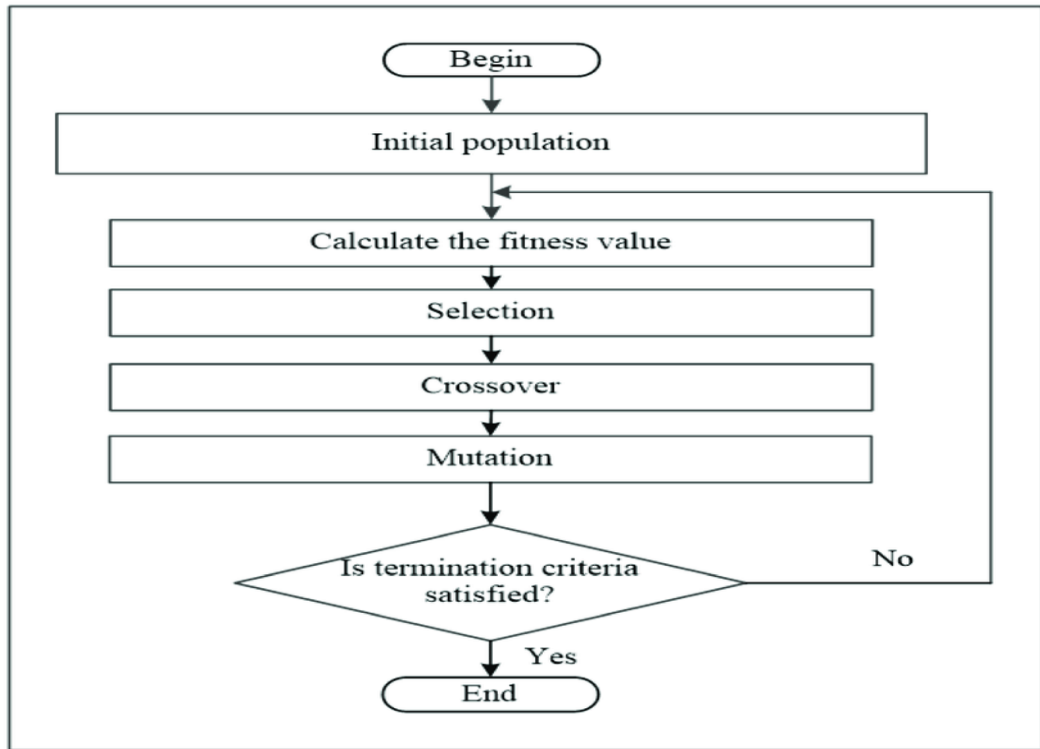
2) Kodlama(Genetik Kodlama)

3) Rastgele Bireylerden Oluşacak Şekilde Bir Başlangıç Popülasyonunun Seçilmesi

4) Tekrarlama(Yeterince İyi Bir Çözüm Bulana Kadar Tekrarlama)

- Popülasyondaki bütün bireylerin uygunluk fonksiyonunun hesaplanması.
- Yeni Nesil için en iyi bireyleri n seçilmesi
- Çaprazlama ve mutasyon ile yeni neslin oluşturulması
- Yeni Nesli(Kromozomları) Popülasyona ekle

5)En İyi Çözümü Döndür



Başlangıç Popülasyonunun Oluşturulması:

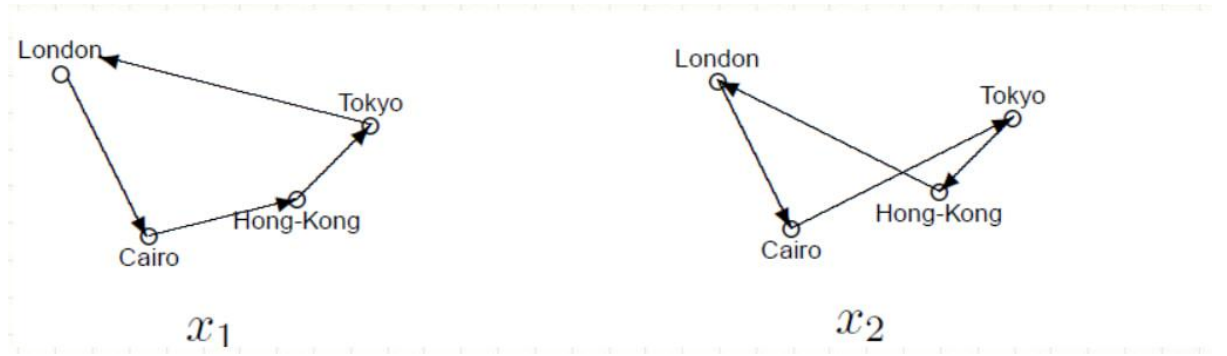
- Başlangıçta belirli kriterlere göre rastgele bireylerden bir popülasyon oluşturulur.
- Popülasyondaki bireylerin sayısı küçük seçildiğinde iterasyon daha hızlı olur fakat algoritmanın yerel optimuma takılma şansı artar.
- Birey sayısının çok olması çözüm kalitesini artırır fakat algoritmanın adımları daha uzun zaman alır.

Uygunluk(Fitness) Fonksiyonu

- Farklı çözümlerin karşılaştırılması ve iyi olanların seçimi için gereklidir.
- $F(X1) > F(X2)$ ise $X1$ $X2$ 'den daha iyidir.
- Bir popülasyon oluşturulduktan sonra popülasyondaki her bireyin uygunluk fonksiyonu belirlenen kurala göre hesaplanır.

Örnek Gezgin Satıcı Problemi

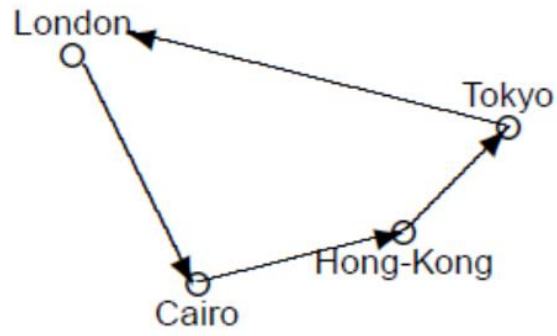
- Gezi için birkaç yol seçeneği vardır.
- Uygunluk Fonksiyonu: $F(x) = \min(\text{uzunluk})$
- $F(X1) < F(X2)$ olduğu için $X1$ yolu $X2$ yolundan daha iyi bir çözüm olabilir.



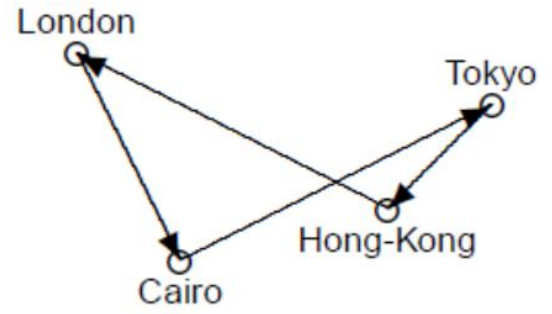
Genetik Kodlama

- Kodlama genetik algoritmanın önemli bir aşamasıdır.
- Genetik Algoritma uygulamasına başlamadan önce verilerin uygun şekilde kodlanması gerekir.

Örnek Kodlama



$$x_1 = LCHTL$$



$$x_2 = LCTHL$$

Kodlama Çeşitleri

- İkili Binary Kodlama

- En Çok kullanılan kodlama yöntemidir.
- Kromozomların tanımlanması genelde 2 li (0,1) sayı sistemi ile yapılır.
- Örneğin Kromozom1:110100010010110

- Permütasyon Kodlama

- Sıralamaya veya düzenlemeye yönelik problemler için uygundur.
- Örneğin Kromozom1: 1 5 3 2 6 4 7 9 8

- Değer Kodlama

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

Seçim

- Popülasyondaki bütün bireylerin uygunluk fonksiyonu hesaplandıktan sonra bunlardan bazıları yeni kuşaklar üretmek için seçilirler.
- Seçim işlemi:
 - Rulet Yöntemi
 - Sabit Durum Yöntemi
 - Turnuva Yöntemi gibi yöntemlerle belirlenebilir.

Rulet Yöntemi

- Kromozomlar uygunluklarına göre seçilirler. Daha uygun kromozomlara daha büyük alan tanınır. Bu yüzden rulette gelme olasılıkları fazladır.

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

23

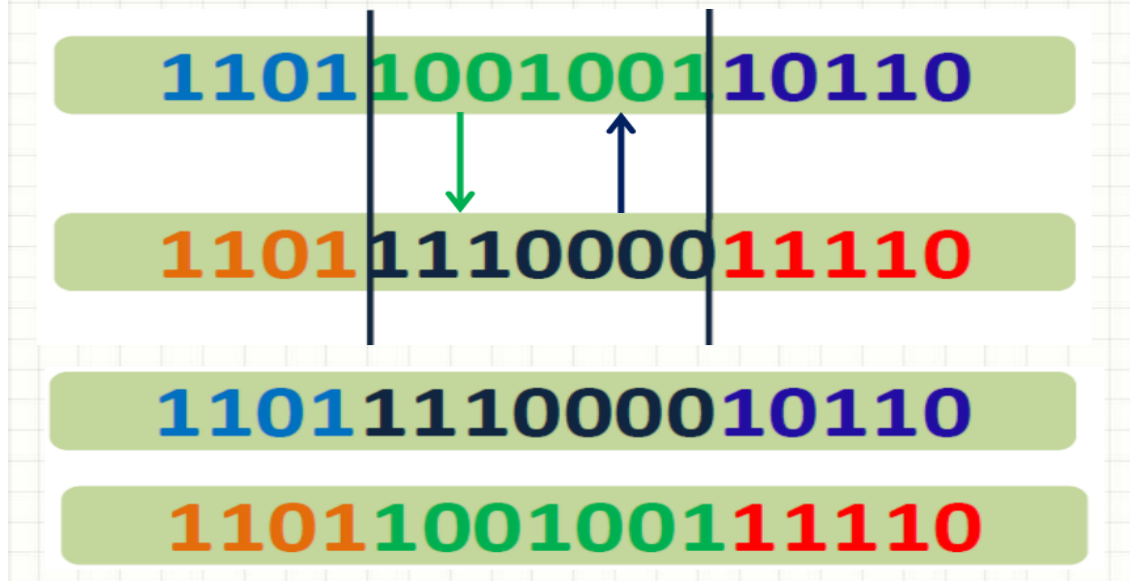
Eğer uygunluk değerleri arasında çok fazla fark varsa diğer kromozomların seçilme şansı azdır. Bunun önüne geçmek için sıralı seçim yöntemi kullanılabilir. En düşük uygunluğa sahip kromozoma 1 değeri verilir. En uygun kromozoma ise N değeri (N birey Sayısı) verilebilir.

Çaprazlama

Tek Nokta Çaprazlama

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

2 noktalı çaprazlama



Mutasyon

- Rastgele seçilen bir genin değiştirilmesidir.
- Belirli bir süre sonra kromozomlar tekrar edebilir bunun için mutasyon işlemi uygulanır.

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

Çaprazlama ve mutasyon işlemlerinden sonra bireyler anne ve babalarından daha iyi genlere sahip olabilirler.

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

– P1 = AKDEMZIF

– P1' = AEMZIKDF

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

– P1 = AKDEMZIF

– P1' = ADEMZKIF

- **Karşılıklı Değişim Yöntemi:** Dizi içerisinde rassal olarak iki gen seçilmekte ve bu iki genler birbiriyle 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 Avantajları:

- Çok amaçlı optimizasyon yöntemleri ile kullanılabilir.
- Çok karmaşık ortamlara uyarlabilir.
- Kısa sürede iyi sonuçlar verebilir.

Dezavantajları

- Son kullanıcının modeli anlaması güçtür.
- Problemi GA ile çözülebilir duruma getirmek zordur.
- Uygunluk fonksiyonunu belirlemek zordur.

Genetik Algoritma Örneği

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

$$x_1 = 435216$$

$$x_2 = 173965$$

$$x_3 = 248012$$

$$x_4 = 908123$$

- Uygunluk (fitness) fonksiyonu aşağıda verilmiştir:

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

Genetik Algoritma Örneği

- İstenenler:
 - 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.

- 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**

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

$$\begin{array}{lcl} x_4 = & \mathbf{908} & \left| \mathbf{123} \right. \\ x_3 = & \mathbf{248} & \left| \mathbf{012} \right. \end{array} \Rightarrow \begin{array}{lcl} O_1 = & \mathbf{908012} \\ O_2 = & \mathbf{248123} \end{array}$$

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

$$\begin{array}{lcl} x_3 = & \mathbf{24} & \left| \mathbf{80} \right| \mathbf{12} \\ x_1 = & \mathbf{43} & \left| \mathbf{52} \right| \mathbf{16} \end{array} \Rightarrow \begin{array}{lcl} O_3 = & \mathbf{245212} \\ O_4 = & \mathbf{438016} \end{array}$$

- Yeni neslin uygunluk fonksiyonunun hesaplanması:

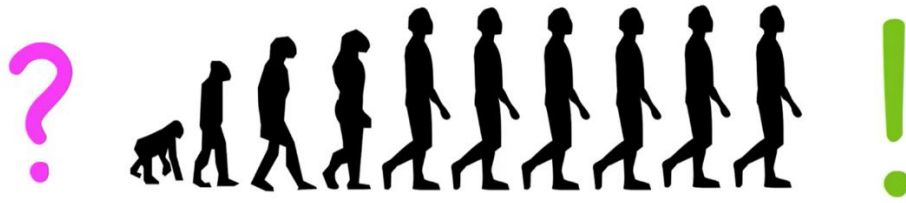
$$f(x) = (a + c + e) - (b + d + f):$$

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

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

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

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



1. Initial population
2. Fitness function
3. Selection
4. Crossover & mutation
5. Repeat steps 3-4