

8. Genetik Algoritmalar

Genetik algoritma (GA), belirli bir soruna iyi çözümler bulma umuduyla yeni genotipler üretmek için mutasyon ve çaprazlama gibi yöntemleri kullanarak doğal seleksiyon sürecini taklit eden bir arama algoritması ve sezgisel tekniktir. Makine öğrenmesinde, 1980'ler ve 1990'larda genetik algoritmalar kullanılmıştır. Tersine, genetik ve evrimsel algoritmaların performansını arttırmak için makine öğrenme teknikleri kullanılmıştır.

Genetik Algoritma yaklaşımının ortaya çıkışı 1970'lerin başında olmuştur. 1975'te John Holland'ın makine öğrenmesi üzerine yaptığı çalışmalarda canlılardaki evrimden ve değişimden etkilenerek, bu genetik evrim sürecini bilgisayar ortamına aktarması ve böylece bir tek mekanik yapının öğrenme yeteneğini geliştirmek yerine, çok sayıdaki böyle yapıların tamamını "çoğalma, değişim" gibi genetik süreçler sonunda üstün yeni bireylerin elde edilebileceğini gösteren çalışmasından çıkan sonuçların yayınlanmasından sonra geliştirdiği yöntemin adı "Genetik Algoritmalar" olarak tanınmıştır.

Genetik Algoritmalar, süreçleri model olarak kullanan problem çözme teknikleridir. Öğrenen Makina Algoritmaları ise tecrübe ile ortaya çıkan yeni duruma ya da belirsizliğe ilişkin parameterlerin otomatik olarak belirlendiği matematiksel modellerdir.

Genetik Algoritmalar mümkün olan çözümlerin bir popülasyonu üzerinde işlem yapan olasılıklı (stochastic) arama algoritmalarıdır. Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri, bunların yardımıyla daha kolay ve hızlı olarak çözüme ulaştırılmaktadır. Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için "iyi"nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama, değiştirme gibi operatörleri kullanır. Görevler için programın ölçülen performansı tecrübe ile artıyor ise bu program tecrübe ile öğreniyor denilebilir. Genetik algoritmalar robot kontrolü için kural kümelerinin öğrenilmesi ve yapay sinir ağları için topoloji ve öğrenme parametrelerinin optimize edilmesi için kullanılmaktadır.

Göçmen kuşlar, her yıl çıktıkları yolculukta adreslerini hiç şaşırmadan, hiçbir duraklarını atlamadan, yüzyıllardır sürekli hareket halindedir; üstelik pusulasız ve haritasız... Dünyadaki on binlerce kuş türünün sadece sekiz bin kadarı her yıl binlerce kilometre kat ederek nesillerini sürdürmeye çalışır. Kimileri sürüler halinde, kimileri küçük gruplar halinde ömürlerinin sonuna kadar kendilerini bu uzun yolculuğa adarlar. Bu kuşların tek bir ortak amaçları vardır: Üremek ve beslenmek. Göçmen kuşların uzun uçuşlarına iç güdüsel olarak hazırlandıklarını da söyleyebiliriz. Öyle ki, genlerinde uzak diyarlara göç etme nitelikleri bulunan hayvanat bahçelerinde, ya da kafeste beslenen kuşlar göç vakti geldiklerinde kıpır kıpır oldukları gözlenir.

Göçmen kuşların her yıl binlerce kilometre süren yolculukları sırasında yönlerini kaybolmadan nasıl tayin edebildikleri üzerine yapılan araştırmalarda ortaya sadece mantıksal tahminler

atılmış durumda. Örneğin, kuşların tanıdık kara parçalarını ezberledikleri, sürekli sahil şeridini takip ettikleri gibi var sayımlar bir yere kadar doğru ama uzun deniz yolculukları yapanlar bu varsayımları yerle bir ediyor. Diğer taraftan gece uçuşu yapanlar içinse yıldızların onlara yol gösterdiği düşünülüyor. Bir diğer var sayım ise göçmen kuşların, dünyanın manyetik alan çizgilerini takip ederek kaybolmadan yönlerini bulabildikleri şeklinde. Modern çağın nimetlerinden biri olan GPS sayesinde işaretlenen kimi göçmen kuşların izledikleri rotalar her yıl tıpa tıp aynı. Hatta inanılması güç ama, her yıl mola verdikleri yerler bile şaşmıyor.

Göçmen kuşların hayranlıkla izlenebilecek bir başka özellikleri de havada çizdikleri V şeklindeki düzen olsa gerek. Öyle ya, ancak bu şekilde onları diğer sürülerden net bir şekilde ayırt edebiliyoruz. Bu uçuş düzeninin iki önemli avantajı var. İlki, en öndeki lider kuşu görebilmek. İkincisi ve tabi ki daha önemli olanı ise hemen öndeki kuşun yarattığı hava akımından yararlanarak daha az enerji harcayarak uçmak. Kuşların bu şekilde uçmaları sayesinde tahminen yüzde 20'lik bir enerji tasarrufu sağladıklarına inanılıyor. Eğer havada bu tür bir V oluşumu yakalarsanız, kuşların yerlerini sürekli değiştirdiğine de tanık olacaksınız. Özellikle de en önde uçarak en fazla yorulan lider kuşa eşlik edenlerin onun yerini nasıl almaya çalıştıklarını göreceksiniz.

Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri, bunların yardımıyla daha kolay ve hızlı olarak çözüme ulaştırılmaktadır. Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için "iyi"nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama, değiştirme gibi operatörleri kullanır. Görevler için programın ölçülen performansı tecrübe ile artıyor ise bu program tecrübe ile öğreniyor denilebilir. Genetik algoritmalar robot kontrolü için kural kümelerinin öğrenilmesi ve yapay sinir ağları için topoloji ve öğrenme parametrelerinin optimize edilmesi için kullanılmaktadır.

Genetik algoritmalar, evrim teorisinin dayandığı temel prensiplerinden olan doğal seçim ile en iyi bireylerin hayatta kalması ilkesini taklit eden bir tekniktir. Burada yapılan, en iyi çözümün pek çok çözüm seçeneği içinden arama yapılarak belirlenmesidir. Rassal arama teknikleri ile eldeki mevcut çözümlerden hareketle en iyi çözüme ulaşmaya çalışılmaktadır. Basit bir genetik algoritmanın işlem adımları; problemin olası çözümlerinin dizilere (kromozomlar) kodlanarak çözüm yığınının oluşturulması, kromozomların çözüme yaklaşma başarısının uygunluk fonksiyonu ile değerlendirilmesi, genetik parametrelerin belirlenmesi, seçim stratejisi ve mekanizmaları, genetik operatörler ve durdurma kriteri olarak sıralanabilir.

Genetik algoritmalar, bilinen yöntemlerle çözilemeyen veya çözüm süresi problemin büyüklüğüne göre oldukça fazla olan problemlerde, kesin sonuca çok yakın sonuçlar verebilen bir yöntemdir. Bu özelliği ile, NP (Nonpolynomially-Polinomal olmayan) problemler yanında gezgin satıcı, karesel atama, yerleşim, atölye çizelgeleme, mekanik öğrenme, üretim planlama, elektronik, finansman ve hücrel üretim gibi konularda uygulanmaktadır.

GEN: Kendi başına anlamı olan ve genetik bilgi taşıyan en küçük genetik birimdir.

- Bir gen A, B gibi bir karakter olabileceği gibi 0 veya 1 ile ifade edilen bir bit veya bit dizisi olabilir. Örneğin bir cismin x koordinatındaki yerini gösteren bir gen 101 şeklinde ifade edilebilir

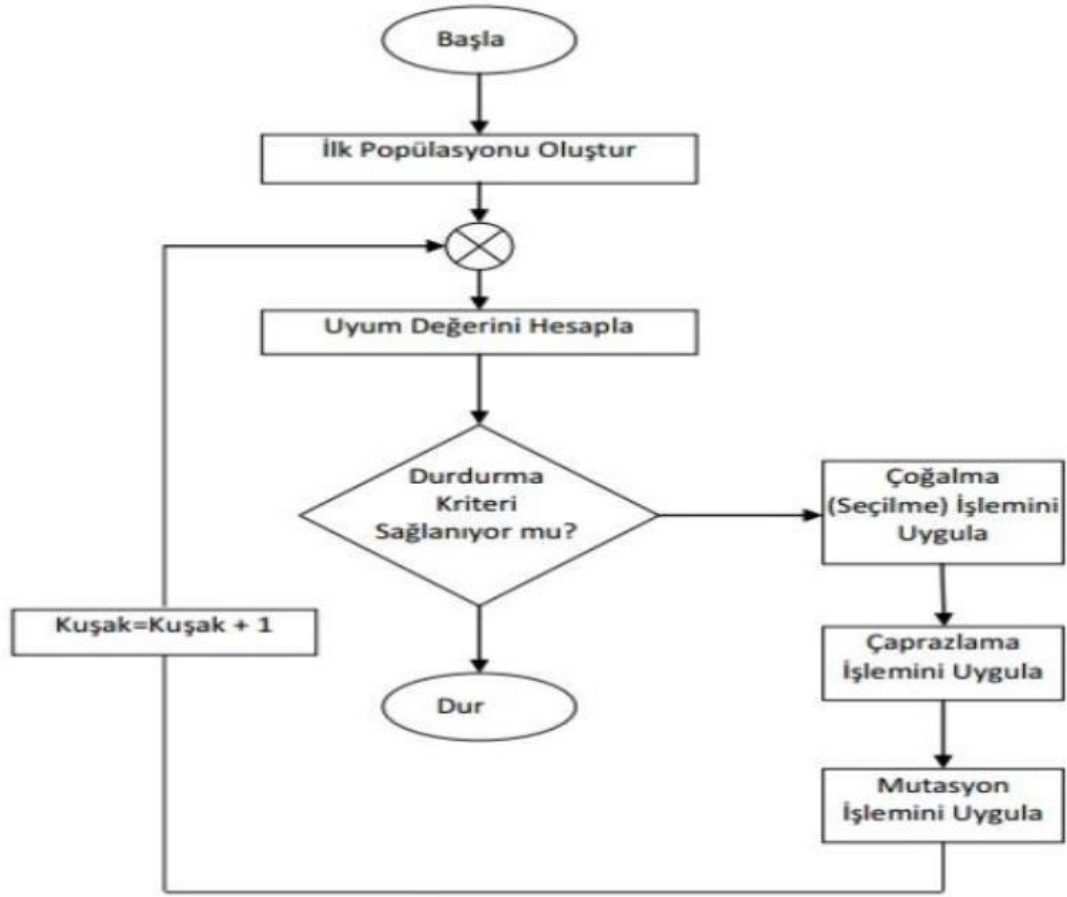
KROMOZOM: Bir ya da birden fazla genin bir araya gelmesiyle oluşurlar Probleme ait tüm bilgileri içerirler. Kromozomlar toplumdaki bireyler yada üyelere karşılık gelirler. Ele alınan problemde alternatif çözüm adaydır.

Örneğin, kromozom bir problemde açı, boyut ve koordinat değişkenlerinden veya bir dikdörtgen prizmasının ölçülerinden (genişlik, derinlik) oluşabilir 001 101 111 1 5 7 değerleri kromozomu oluşturan genlerdir. Genetik algoritma işlemlerinde kromozomları kullandığı için kromozom tanımları çok iyi ifade edilmelidir.

POPÜLASYON: Kromozomlar veya bireyler topluluğudur. Popülasyon üzerinde durulan problem için alternatif çözümler kümesidir. Aynı anda bir popülasyonda ki birey sayısı sabit ve probleme göre kullanıcı tarafından belirlenir. Zayıf olan bireylerin yerini kuvvetli yeniler almaktadır.

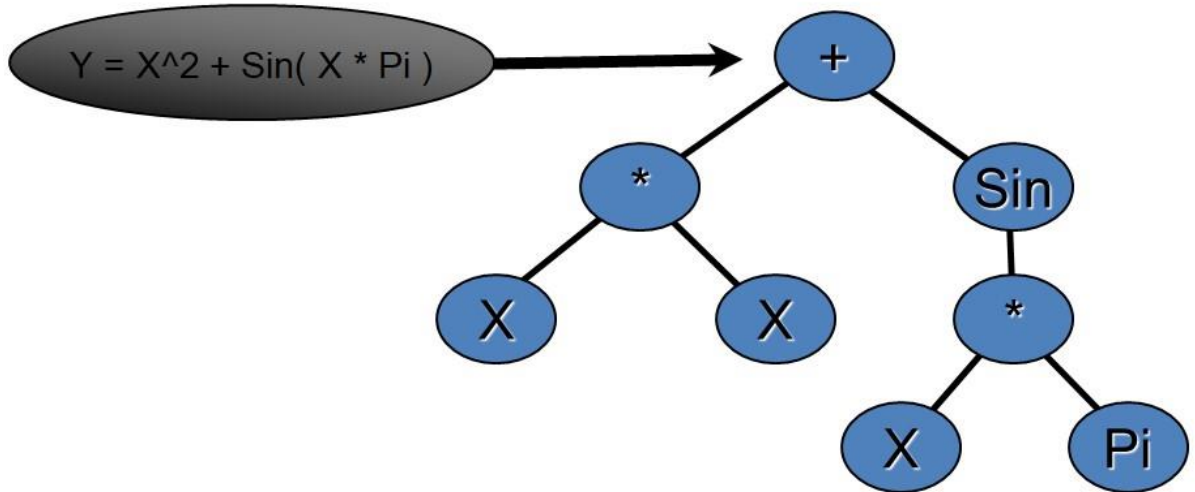
Genetik Algoritmalar Nasıl Çalışır?

1. Adım: Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Çözüm grubuna biyolojideki benzerliği nedeniyle populasyon çözümlerin kodları da kromozom olarak adlandırılır. Bu adıma populasyonda bulunan birey sayısı belirleyerek başlanır. Bu sayı için bir standart yoktur. Genel olarak önerilen 100 - 300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Populasyon bu işlemde sonra rasgele oluşturulur.
2. Adım: Her kromozomun ne kadar iyi olduğu bulunur Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu denir Bu fonksiyon işletilerek kromozomların uygunluklarının bulunmasına ise hesaplama(evalution adı verilir Bu fonksiyon genetik algoritmanın beynini oluşturmaktadır GA da probleme özel çalışan tek kısım bu fonksiyondır. Çoğu zaman GA nın başarısı bu fonksiyonun verimli ve hassas olmasına bağlı olmaktadır



Genetik algoritmanın akış diyagramı.

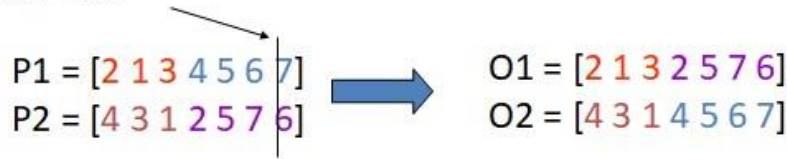
Genetic Programming Background:



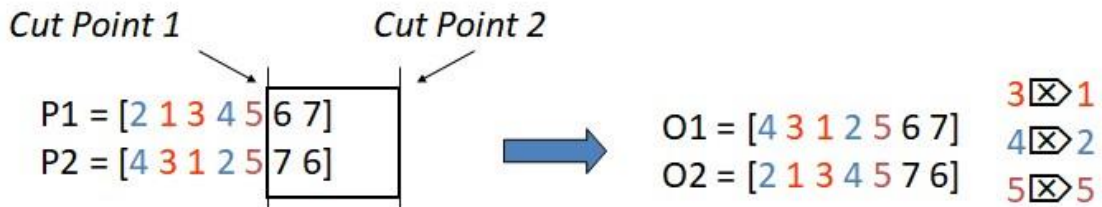
Crossover: bir üst programdaki bir makinedeki işlem sırasını başka bir ana makinedeki başka bir makinedeki işlem dizisiyle birleştirin.

Example 1.

Cut Point



Example 2. Partially Mapped Crossover



Örnek: P1'den alınan işlerin mutlak konumlarını ve P2'den alınanların göreceli konumlarını korur.

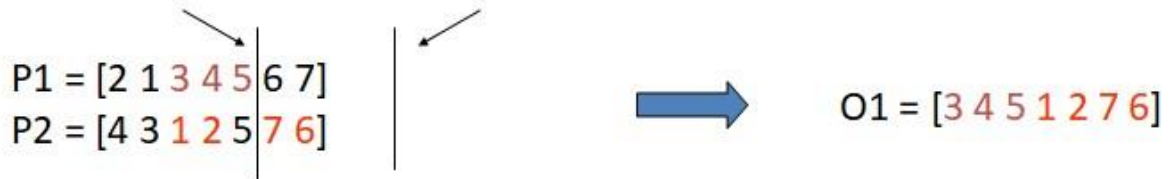
Cut Point 1



Örnek: Örnek 3'e benzer ancak 2 geçiş noktası vardır.

Cut Point 1

Cut Point 2



Örnek:

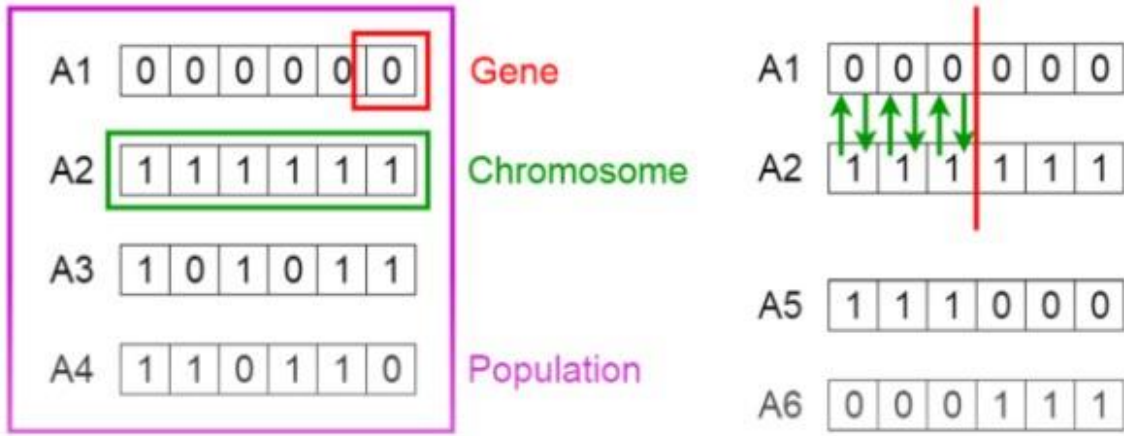
Aşağıdaki ifadeyi karşılayan GA'yı kullanarak a ve b'nin optimal değerlerini tahmin edelim.

$$2a^2 + b = 57$$

Herhangi bir optimizasyon problemi, bir amaç fonksiyonuyla başlar. Yukarıdaki denklem şu şekilde yazılabilir: $f(a,b) = 2a^2 + b - 57$

Fonksiyonun değerinin 0 olduğu anlaşılmaktadır. Bu fonksiyon bizim amaç fonksiyonumuzdur ve amaç a ve b değerlerini tahmin etmektir, öyle ki objektif fonksiyonun değeri sıfıra indirilir.

Örnek:



Örnek:

Adil bir jetonu 60 kez atıyoruz ve aşağıdaki ilk popülasyonu elde ediyoruz:

$s_1 = 1111010101$	$f(s_1) = 7$
$s_2 = 0111000101$	$f(s_2) = 5$
$s_3 = 1110110101$	$f(s_3) = 7$
$s_4 = 0100010011$	$f(s_4) = 4$
$s_5 = 1110111101$	$f(s_5) = 8$
$s_6 = 0100110000$	$f(s_6) = 3$

Seçimi yaptıktan sonra aşağıdaki popülasyonu elde ettiğimizi varsayalım

$$s_1' = 1111010101 \quad (s_1)$$

$$s_2' = 1110110101 \quad (s_3)$$

$$s_3' = 1110111101 \quad (s_5) \quad s_4' =$$

$$0111000101 \quad (s_2) \quad s_5' =$$

$$0100010011 \quad (s_4) \quad s_6' =$$

$$1110111101 \quad (s_5)$$

Sonra crossover için dizeleri eşleştiriyoruz. Her çift için crossover olasılığına (örneğin 0.6) göre crossover yapıp yapmamaya karar veririz. Sadece çiftler için (s_1' , s_2') ve (s_5' , s_6') crossover yapmaya karar verdiğimiz varsayalım. Her çift için rastgele bir geçiş noktası çıkarırız, örneğin birinci için 2 ve ikinci için 5.

Before crossover: s_1'
 $= 1111010101$ $s_2' =$
 1110110101

$s_5' = 0100010011$ $s_6' =$
 1110111101

After crossover: s_1''
 $= 1110110101$ s_2''
 $= 1111010101$

$s_5'' = 0100011101$ s_6''
 $= 1110110011$

Son adım, rastgele mutasyon uygulamaktır: yeni popülasyona kopyalayacağımız her bit için küçük bir hata olasılığına izin veriyoruz (örneğin 0.1) Mutasyonu uygulamadan önce:

$$s_1'' = 1110110101$$

$$s_2'' = 1111010101$$

$$s_3'' = 1110111101$$

$$s_4'' = 0111000101$$

$$s_5'' = 0100011101$$

$$s_6'' = 1110110011$$

After applying mutation:

$$s_1''' = 1110100101 \quad f(s_1''') = 6$$

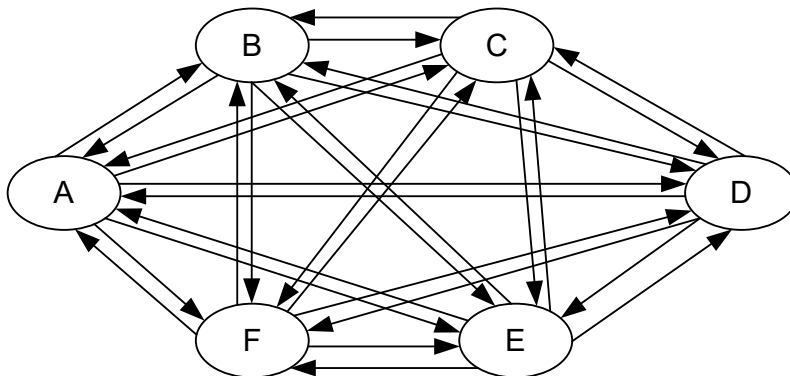
$$s_2''' = 1111110100 \quad f(s_2''') = 7$$

$$\begin{aligned}
s_3''' &= 1110101111 & f(s_3''') &= 8 \\
s_4''' &= 0111000101 & f(s_4''') &= 5 \\
s_5''' &= 0100011101 & f(s_5''') &= 5 \\
s_6''' &= 1110110001 & f(s_6''') &= 6
\end{aligned}$$

Bir nesilde, toplam popülasyon uygunluğu 34'ten 37'ye değişti, böylece ~% 9 arttı. Bu noktada, bir durdurma kriteri karşılanana kadar aynı süreci yeniden yaşıyoruz.

Örnek:

A, B, C, D, E, F olmak üzere 6 durumum olsun. Her bir durumdan(genden) diğer bütün durumları (genlere) geçiş mümkün olsun. Tüm durumlar bir defa muhakkak ziyaret edilecek. O halde $5!=120$ tur mümkündür. Tur, her zaman A'dan başlayacak bütün durumlar dolaşılacak sonunda yeniden A'ya dönülecek. Durumlar arasında rasgele bir rota oluşturalım. ABDFECA.



Bu problem Genetik Algoritma ile çözülürken,

- Uygunluk fonksiyonu seçilir. Uygunluk fonksiyonu mesafeler olacak, mesafeyi en aza indirilmesi gerekiyor.
- Popülasyon büyüklüğü, kaç çözümden oluşacak, mesela 10 seçilebilir.
- Ebeveyn seçimi, çözüm uzayındaki farklı durumları kullanarak yeni bir rota oluşturulacak. (Rulet tekerleği, sıralı yöntem)
- Çaprazlama, iki turdan yeni bir tur oluşturulurken çaprazlama iki noktalı mı yoksa tek noktalı mı?
- Mutasyon oranı, yeni tur oluşturulduktan sonra bir veya iki durumun (gen) rasgele değiştirilmesi
- Bitirme Şartı, iterasyon sayısının belirli bir sayıya ulaşması mı, uygunluk fonksiyonun belirli bir değerinin altında veya üstünde olması mı karar verilir.

Öncelikle durumlar (genler) arasında rasgele 2 çözüm turu oluşturalım. 6 durum arasında oluşturulacak tur sayısı=5!=120 farklı tur oluşturulabilir.

Tur-1: A B D F E C A

Tur-2: A C E D F B A

Bu iki tur kendi aralarında çaprazlamaya tabi tutulur. Çaprazlama oranı %50 olsun ve tek noktalı olarak yapılsın.

Tur-1: A B D **F E C A**

Tur-2: A C E **D F B A**

Yeni tur oluşturulurken birinci turdan ilk kısım ikinci turdan ise ikinci kısım alınır. İkinci kısımda olanlar birinci kısımda var ise alınmaz, yerine ikinci turundan başından itibaren olmayanlar alınır.

Tur-3: A B D ~~F B A~~ = A B D F C E A olur.

Mutasyonu yaparken tek noktalı bir durumun (genin) değiştirilsin. Yeni oluşan Turda rasgele bir durum (gen) değiştirilsin. Sözelimi B'yi değiştirelim ve E olsun. Yeni tur (kromozom),

Tur-4: A E D **F C B A** olur. E durumu iki adet olamayacağı için sağ taraftaki E B ile değiştirilir. Böylece mutasyondan sonraki turda elde edilmiş olur.

Bu aşamadan sonra uygunluk fonksiyonu hesaplanır. Popölasyon içerisindeki diğer çözümlerden daha kısa mesafe ise diğer çözümlerden daha iyi çözüm olarak kabul edilir. Eğer değilse öldürölür.

Adımları sıralarsak,

- 10 tane başlangıç çözümlünü oluşturulur.
- Herbir çözüm için uygunluk fonksiyonu değeri hesaplanır.
- Başlangıç çözümlerinden iki tane ebeveyn seçilir.
- Ebeveynler üzerinde çaprazlama yapılır.
- Mutasyona tabi tutulur.
- Yeni bireyin uygunluk fonksiyon değeri hesaplanır.
- Yeni birey öldürölülecek mi yoksa popölasyona katılacak mı karar vermek için seleksiyon işlemi yapılır.
- Bitirme şartı sağlandı mı diye kontrol edilir, sağlandıysa en iyi çözüm al ve çık.