

**GEZGİN SATICI PROBLEMİNİN  
İYİLEŞTİRİLMİŞ GENETİK ALGORİTMA İLE ÇÖZÜMÜ**

**İçerik**

Özet .....	2
Gezgin Satıcı Problemi.....	3
Genetik Algoritma.....	3
Seçim Operatörleri.....	6
Çaprazlama Operatörleri.....	9
Mutasyon Operatörleri.....	13
Uygulama.....	15
Sonuç.....	24
Kaynakça.....	26

## ÖZET

Günümüzde yaşadığımız problemler her geçen gün artmakta ve farklılaşmaktadır. Gelişen teknoloji ve bilim de bununla birlikte farklılaşarak problemleri optimize eder. Optimizasyon bir problemin olabilecek farklı çözümleri arasından en iyi sonucu verenin bulunmasıdır. Ele alınan problemin büyüklüğüne göre bu zorlaşabilir. O yüzden metasezgisel çözümlerin kullanılması yaygınlaşmıştır. Optimizasyon algoritmaları genel olarak sezgisel optimizasyon algoritmaları ve matematiksel optimizasyon algoritmaları olarak ikiye ayrılır. Sezgisel algoritma türlerinden biri olan evrimsel yapıdaki algoritmalar “doğal seleksiyon” (seçim) prensibi ile açıklamaktadır. Yapay zeka gibi insanların daha iyi yaptığı işleri bilgisayarlara nasıl yaptırılacağı ile ilgili bir çalışma alanı olarak tanımlanabilir. “Genetik Algoritma(GA)” evrimsel yapıdaki algoritmaya bir örnektir.

Bu çalışmadaki hedef gerçek dünya problemlerine uygulanabilen Gezgin Satıcı Problemini(GSP) metasezgisel yöntemlerden biri olan Genetik Algoritma ile çözmektir. GSP’de data arttıkça çözümü zorlaşır. Bu sebepten GA gidişat ve yapı bakımından GSP için minimum sürede en uygun çözümü elden algoritmalarından biridir. Fakat GA’nın genel en iyi çözümün bulunduğunu garanti edememelerinden dolayı aynı evrimsel süreç gibi iyileştirebilmek , literatürdeki çaprazlama ve mutasyon yaklaşımları arasından en optimum olanı seçerek çözüme erişmektir.

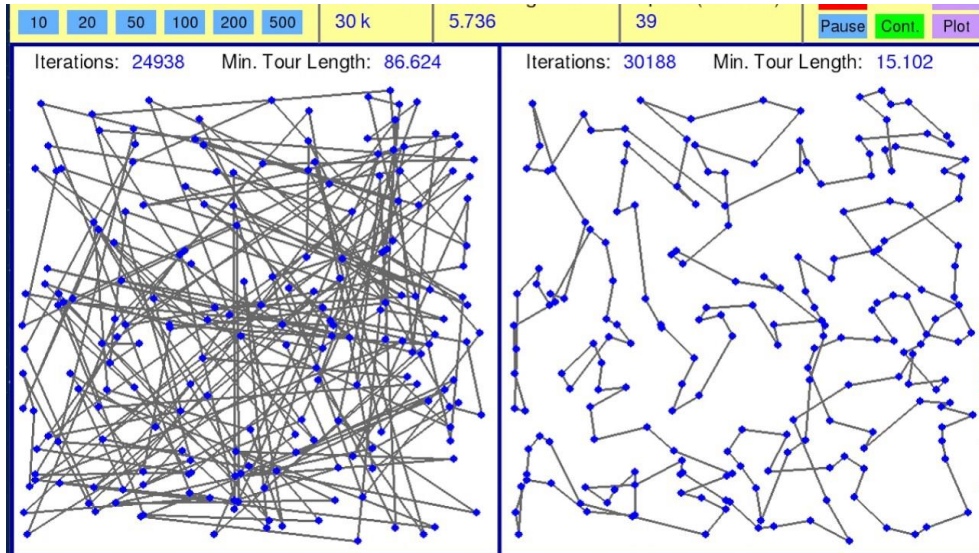
Bu rapordaki çalışmamın amacı doğrultusunda, GSP’yi iyileştirilmiş GA ile çözmek, olan iyileştirme yöntemlerine değineceğim. GA’nın çoğu adımına yönelik farklı operatörler kullanarak ilerleyebiliriz. Fakat GA’nın genel en iyi çözümün bulunduğunu garanti edememelerinden dolayı aynı evrimsel süreç gibi iyileştirebilmek , literatürdeki seçim ,çaprazlama ve mutasyon yaklaşımları arasından en optimum olanı seçerek çözüme erişmektir. GA metasezgisel süreçlerden esinlenen bir algoritma olduğu için gerek içgüdüsel gerek araştırmalar ile operatörleri yorumlayacağım.

### Gezgin Satıcı Problemi Nedir?

GSP , koordinatlar üzerindeki  $n$  adet şehir arasında , şehirlerin her birine yalnız bir kez uğrayarak , başlangıç noktasına geri dönmek şartıyla, tur boyunca kat edilen toplam yolun en kısa olduğu rotanın bulunmasını amaçlayan bir problemdir. Şehir sayısı azken çözüme ulaşmak kolay olacakken , şehir sayısı fazlalaştığında en uygun çözümü kısa sürede bulmak olanağı azalır. Bu sebepten dolayı uzun yıllardır üzerinde çalışmalar yapılan NP-ZOR(çözümü zor) sınıfında yer alan bir problemdir.

GSP'nin genel orijinal formu, 1920'lerde matematikçi ve iktisatçı olan Karl Menger tarafından incelenmiştir. Dantzig, Fulkerson ve Johnson'ın 1954 yılındaki "büyük ölçekli gezgin satıcı problemlerinin çözümü" isimli makaleleri kombinatoriyal optimizasyona ait temel akademik çalışmalardan biri olarak kabul görmektedir.

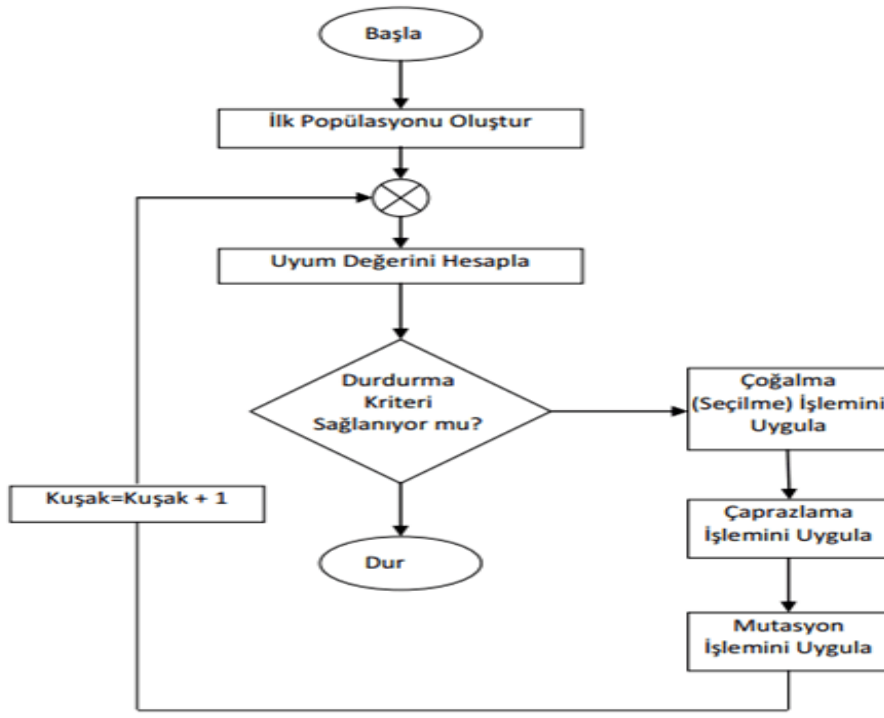
Çözümü zor olsa da rotalama, dağıtım, lojistik , kuruluş yeri belirleme gibi kullanıldığı yaygın alanları vardır. Örneğin elektronik devre tasarımlarında, kargo şirketlerinin dağıtım planlanmasında , baz istasyonlarının yerlerinin belirlenmesine vs. birçok avantaj sağlar.



GSP çözümüne bir örnek. Şekil 1

### Genetik Algoritma Nedir ?

GA, biyolojik evrime dayanılarak geliştirilmiş 1970 yılında John Holland tarafından ortaya atılmıştır. Karmaşık problemlerin çözümünde kullanılan en uygun ve en hızlı çözümü sağlayan popülasyon tabanlı algoritmalarından biridir. Tüm evrimsel algoritmalar popülasyon temellidir. Popülasyondaki bireyler arasında etkileşim ve bilgi değişimi mevcuttur. Tipik bir evrimsel algoritmada bir başlangıç popülasyonu oluşturulur, değerlendirme, seçim, yeniden üretim ve mutasyon gibi tekrarlanan uygulamalarla nesilden nesile dönüştürülür. . Popülasyondaki her bir birey çevreye olan uygunluğu ölçüsünde bir değer alır. Seçim, mevcut uygunluk bilgisini kullanarak yüksek uygunluğu olan bireyler üzerinde odaklanır. Yeniden birleştirme ve mutasyon, yeni keşifler için sezgisel yöntemler sağlayarak bu bireyleri farklılaştırır. Çocuklar ya da yeni bireyler daha sonra yeniden değerlendirilir ve son olarak popülasyonda hangisinin kalacağına karar verilir. Kabaca akış şeması Şekil 2 deki gibidir.



Şekil 2

Genetik Algoritmanın Temel Adımları:

- 1) Başlangıç Popülasyonunun seçilmesi
- 2) Küme içerisindeki her bir kromozomun incelenmesi ve uygunluk değerinin belirlenmesi
- 3) Elitizm ve seçim
- 4) Çaprazlama
- 5) Mutasyon
- 6) Yeni kromozomların incelenmesi ve kümeye dahil edilmesi.
- 7) Durdurma kriteri sağlanana kadar adım 2'ye dönülmesi
- 8) Algoritmanın sonuçlanması

GA'da popülasyondaki her bir birey kromozomdan oluşmaktadır. Kromozomlar problem için oluşturulan çözümleri temsil eder ve probleme ait bilgileri içeren genlerden oluşur. Her bir gen problemlerin çözümlerine göre farklı şekilde ayrı ayrı kodlanır. Problemin çözümüne uygun kodlama belirlendikten sonra GA süreçlerini başlatmak için başlangıç popülasyonu oluşturulur. Popülasyon boyutu doğru sonuca ulaşmamız için önemli parametrelerden biridir. Başlangıç popülasyonu, her biri olası problemin çözümleri olabilecek bireylerden oluşmaktadır. Popülasyon boyutu arttıkça; popülasyondaki çeşitlilik ve en uygun çözüme ulaşma ihtimali artarken çözümü elde etme süresi de artacaktır.

Başlangıç popülasyonu oluşturulduktan sonra her bir bireyin en uygun çözüme yakınlığının belirlenmesi gerekmektedir. Bireylerin en uygun çözüme yakınlığını hesaplayan fonksiyon uygunluk fonksiyonudur. Optimizasyon problemlerinden biri olan GSP için uygunluk fonksiyonu, oluşturulan rotadaki mesafelerin toplamı şeklinde tanımlanmaktadır. Bir noktadan başlayarak her noktadan yalnızca bir kez geçerek yine aynı noktaya ulaşma

amacıyla oluşturulan rotalar kromozom olarak kodlanır. Kromozomda kodlanan noktalar arasındaki mesafelerin toplamı GSP uygunluk fonksiyonudur. Popülasyondaki en iyi birey en düşük toplam mesafeye sahip kromozomdur. Uygunluk fonksiyonu değeri en düşük birey çözüme en uygun birey anlamına gelmektedir.

Seçim süreci, elimizdeki popülasyondaki her bir bireyin uygunluk değeri hesaplandıktan sonra yeni bir popülasyon oluşturmak için üretilebilecek en iyi bireylerin seçilmesidir. Seçim süreçlerinde rastgele seçimler olduğu için en iyi uygunluk değerine sahip bireylerin kaybolma olasılığı bulunmaktadır. Bu nedenle bazı GA’larda elitizm seçimi kullanılmaktadır. Elitizm ile en iyi bireyler nesiller boyunca korunabilmektedir. Elitizm, kullanıcı tarafından belirlenen oran kadar en iyi uygunluk değerine sahip o sürece kadar elde edilen en bireylerin korunması ve diğer nesillere aktarılmasıdır. Elitizm ile seçilen bireylerinin oranının doğru belirlenmesi gerekmektedir. Eğer oran çok yüksek olursa popülasyon aynı uygunluk değerine sahip birçok bireyden oluşacaktır. Bu yüzden popülasyondaki çeşitlilik azalır ve en uygun değeri bulmak zorlaşır. GA belli süreçlerin yinelenerek devam ettiği bir algoritma olduğu için durdurma kriterine ihtiyaç duyar. Farklı durdurma yöntemleri vardır. Algoritma içerisindeki tüm süreçlerin sonunda yani yapılan her iterasyon sonucunda yeni bir nesil oluşmaktadır. Kullanıcıdan istenilen veya program içerisinde tanımlanan nesil sayısına ulaşıncaya kadar algoritmanın durdurulması istenebilir. Başka bir sonlandırma yaklaşımı olarak, önceden belirlenmiş bir süre tamamlandıktan sonra algoritma durdurulabilir.

Yeni bireyler oluşturmak için çaprazlama operatörü kullanılmaktadır. Bu süreçte, her iki ebeveyn den alınan genler bir araya getirilerek yeni bireyler oluşturulmaktadır. Uygunluk değeri iyi olan ebeveynlerden daha iyi uygunlukta bireylerin elde edilmesi amaçlanmaktadır. Popülasyonda çaprazlamaya uğrayacak birey sayısı çaprazlama oranıyla belirlenmektedir. Permütasyon temelli problemlerden biri olan GSP için önerilmiş üç özel çaprazlama operatörü bulunmaktadır. Bu çaprazlama operatörleri; Sıralı Çaprazlama (Ordered Crossover), Döngüsel Çaprazlama (Cycle Crossover) ve Kısmi Eşleşen Çaprazlama (Partially Matched Crossover) olarak adlandırılmaktadır (Senapati, 2017).

Mutasyon ise aynı bireyin kromozom yapısı içerisinde sıklıkla seçilmiş olan iki nokta arasındaki parçanın ters çevrilerek yeniden dizilmesi şeklinde gerçekleşmektedir ve GSP çözümünden bu işlemlerin uygulanışı Şekil 3’de verilmiştir.

Tip	Ebeveyn	Çocuklar
Çaprazlama	1 – 2 / 3 – 4 – 5 – 6	1 – 2 – 4 – 3 – 6 – 5
	2 – 1 / 3 – 4 – 5 – 6	2 – 1 – 3 – 4 – 5 – 6
Mutasyon	2 – 4 – 3 – 1 – 5 – 6	6 – 4 – 3 – 1 – 5 – 2
Değiştirme	2 – 1 – 3 – 4 – 5 – 6	2 – 4 – 3 – 1 – 5 – 6

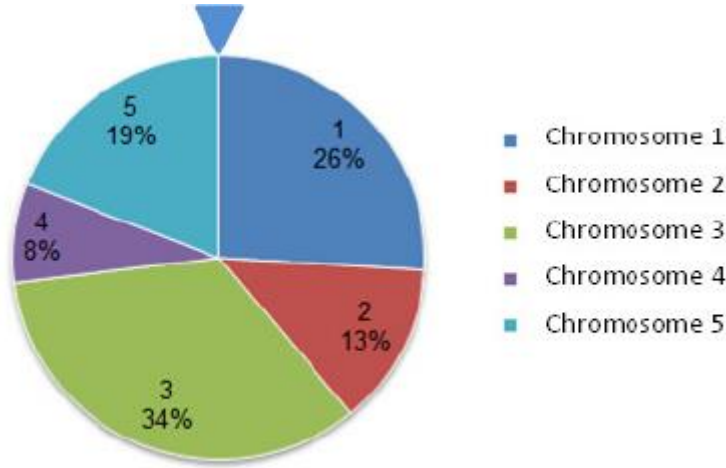
Çaprazlama , mutasyon ve değişim örnekleri. Şekil 3

Sonuç olarak bu temel adımları yerine GA ‘yı GSP’ye uygulamaktır. GA’da belirtilen seçim ve mutasyon operatörlerinde iyileştirilme yapıp çözüme doğru şekilde ulaşarak birçok alanda güvenilir bir şekilde kullanılabilir.

## SEÇİM OPERATÖRLERİ

### 1- Rulet Tekerleği Seçim Operatörü:

GA' nın her döngüsünde oluşan yeni kromozomun uygunluk(fitness) değeri oluyor. Bu değer her bireyin aldığı yola göre hesaplanır. Bu uygunluk değerlerinin büyüklüğüne göre bir rulet çarkı çevresinde sıralandığını hayal edin. Yani fitness değeri ne kadar uygunsa rulet çarkında o kadar geniş yer kaplar. Topun dilimi en geniş yere düşme ihtimali daha fazla olduğundan doğru orantılı olarak fitness değeri uygun olan kromozomun seçilme ihtimali yüksek demektir. Bunu *Şekil 1*'de resmedecek olursak:



Şekil 1

Bir bireyin seçilme olasılığı da formüldeki gibidir:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

i: birey (individual)

N: neslin boyutu

f(i) : i bireyinin uygunluk değeri

Avantajı : Her zaman en uygun bireyi seçer

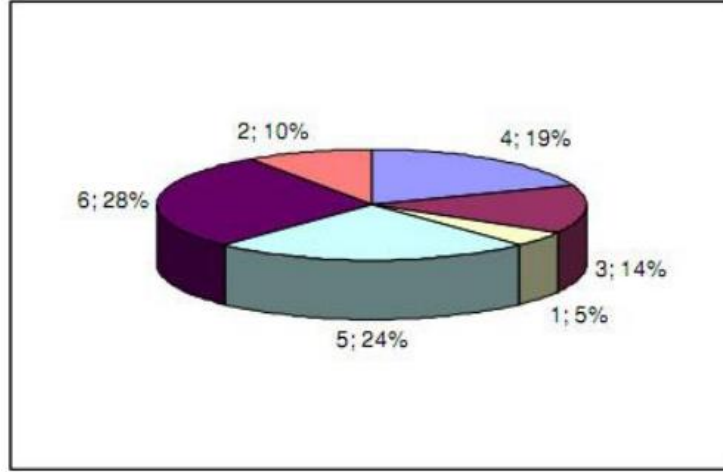
Dezavantajı: Seçilme ihtimali çok yüksek olan bireyden dolayı başka bireyler kaybolabilir.

Zaman Karmaşıklığı : O(n<sup>2</sup>)

### 2- Rank Seçim Operatörü:

Rulet seçim methoduna kıyasla uygunluk değerlerinin ağırlıklarına göre değil , sıralamaya göre dilimlendirme yapılır. Örnek verecek olursak :

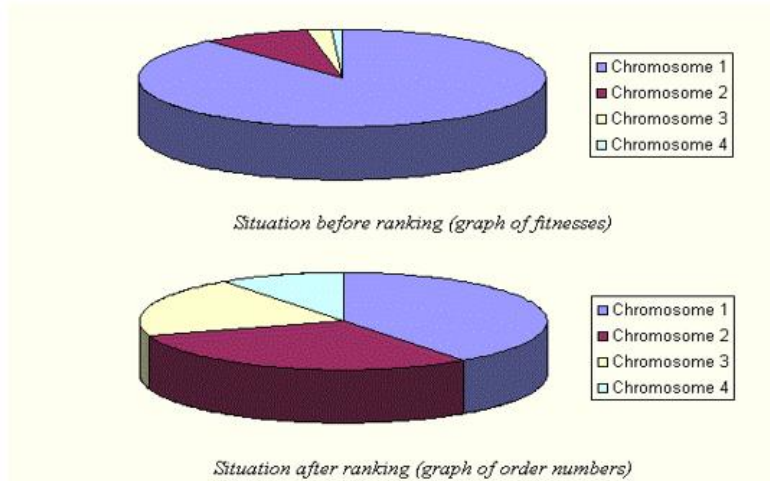
- Bunlar bireylerin uygunluk değerleri olsun :10, 9, 3, 15, 85, 7.
- Bu değerler en kötünden en iyiye doğru sıralanır: 1. : **3**, 2.: **7**, 3.: **9**, 4.: **10**, 5. : **15**, 6.: **85**
- Tüm sıraların toplamı  $1+2+3+4+5+6=21$
- Her bireyin olasılığı bu şekilde hesaplanır :  $1/21, 2/21, 3/21, 4/21, 5/21, 6/21$  ve yüzdelik değeri alınır.
- Sonuç *Şekil 2*'deki gibidir:



*Şekil 2*

Uygunluk değeri yüksek olanın seçilme ihtimali yine fazladır fakat rulet methodu kadar orantısız bir şekilde büyük bir yer kaplamaz.

*Şekil 3*'te rulet ile rank seçim methodlarının arasındaki farkı anlayabiliriz.



*Şekil 3*

**Avantajı :** Populasyondaki bireylerin seçilme ihtimalini eşitlemeye çalışır. Böylece uygunluk değeri çok yüksek olan bireyin ihtimali, diğer bireylerin seçilme ihtimalini fazlaca bir şekilde kapatamaz.

Dezavantajı: Uygunluk değeri düşük olan bireyin seçilme ihtimalini de arttırır. Populasyon her yenilendiğinde sıralanmalıdır.

Zaman Karmaşıklığı :  $O(n \ln n)$

### 3- Steady State Selection

Her nesilde yeni bir yavru oluşturmak için iyi ve en iyi uygunluk değeri olan ebeveynlerden belli sayıda kromozom seçilir. Daha sonra uygunluk değeri kötü olan ebeveynlerden aynı sayıda çıkarılır ve seçtiğimiz iyi ve en iyi ebeveynlerin yavrularını bir sonraki nesile koymak yerine bu çıkardığımız kötü uygunluk değeri olan kromozomların yerine konur.

Avantajı : Populasyon büyüklüğü sabit kalır. Uygunluk değeri önceden hesaplanmış bireylerin bir sonraki nesillerde hesaplanmayabilir. Bu programın çalışma süresine olumlu bir etkidir.

Dezavantajı: Aynı bireylerden oluştuğu için populasyon çeşitliliği azalır. Çeşitlilik en önemli etkenlerden biridir.

Zaman Karmaşıklığı :  $O(n)$

### 4- Tournament Selection

Turnuva boyutu rastgele seçilir. Örneğin  $n$  olsun. Populasyondan rastgele  $n$  tane ebeveyn seçilir ve bunların arasında turnuva düzenlenir. Turnuva galibi en iyi uygunluk değerine göre belirlenir bu da çaprazlamaya(crossover) katılır yani çiftleşme havuzuna(mating pool) atılır. Bir turnuvadan sadece bir birey çıkar ve çiftleşme havuzundaki ebeveyn sayısı  $n$  olana kadar turnuva devam eder.

Bir ebeveynin bir turnuvaya katılma olasılığının olasılıksal bir ölçüsü olan seçim baskısı adı verilen bir parametreye de sahiptir. Turnuva boyutu daha büyükse, zayıf adayların daha güçlü bir adayla rekabet etmesi gerektiğinden seçilme şansları daha azdır. Bu da çeşitliliği azaltır. Turnuva boyutu küçükse tüm bireylere seçilme şansı verir ve bu nedenle çeşitliliği korur, ancak çeşitliliği korumak yakınsama hızını düşürebilir.

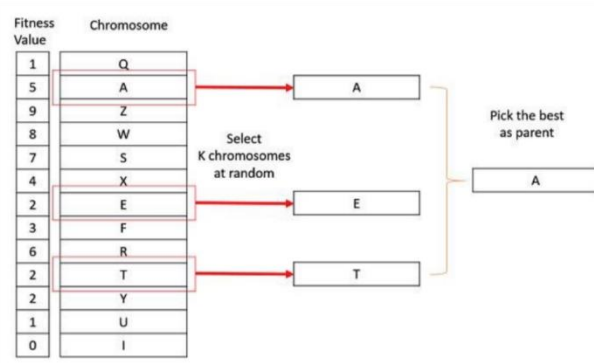
Avantajı: Seçim baskısını kendimiz ayarlayabiliriz.

Dezavantajı: En iyi bireyin çoğalmasına izin vermez.

Zaman Karmaşıklığı :  $O(n)$

Örnek olarak *Şekil 4* verilmiştir:





Şekil 4

## 5- Elitism Selection

Her nesilde en iyi bireyin ya da bireylerin saklanarak mutasyona uğramadan bir sonraki nesile direkt aktarılmasıyla bir sonraki neslin kötüye gitmesi engellenmelidir. Elitleri ebeveyn olarak kullanılmak. Elitizm kullanmanın ana noktaları şunlardır:

Çeşitliliği korumak için popülasyondaki seçkinlerin sayısı toplam nüfusun %10'unu geçmemelidir.

Bunun %5'i gelecek neslin doğrudan bir parçası olabilir ve geri kalanı diğer elit olmayan nüfusla çaprazlama ve mutasyona uğramalı.

Avantajı: En iyilerle beraber çeşitliliği korumaya devam edebiliriz. İyi bir nesil elde eder.

Dezavantajı : Bu seçim operatörünü kullanışlı hale getirmek için parametreleri iyi bir şekilde ayarlamalıyız.

Zaman Karmaşıklığı :  $O(MN^2)$

## ÇAPRAZLAMA OPERATÖRLERİ

GA'nın 4. Adımı olan çaprazlama, seçim aşamasında başarılı seçilen bireyler, ebeveyn kabul edilerek bu ebeveynlerden yeni bireyler üretmek için çaprazlamaya tabi tutulur. Bu adımda çaprazlama operatörlerinden bazıları olan ve öngörülen MCC, PMX, OX operatörlerini inceleyeceğiz. VPR(Araç Rotalama Problemi)'nde en iyi OX seçildi. GSP için ise bu inceleyeceğimiz 3 operatör öneriliyor. Bunlar kıyaslanınca MCC en iyi uygunluk değerini oluşturan operatördür fakat süre performansı açısından yavaştır. Hem uygunluk değerleri hem süre performans açısından ikinci sırada PMX yer alıyor. Üçüncü sırada OX ise daha düşük performansa sahipken avantajı daha büyük verilerde daha iyi potansiyele sahip olmasıdır.

## 1- Modified-Cycle Crossover (MCC)

Değiştirilmiş-Döngüsel Çaprazlama (MCC); ebeveynler üzerinde belirli çaprazlama

noktalarının kullanılmadığı, OX ve PMX'den farklı bir operatördür.

Örnek üzerinden gidecek olursak :

P1 = (3 4 8 2 7 1 6 5) and

P2 = (4 2 5 1 6 8 3 7)

O1= first offspring (ilk yavru)

- İlk olarak bu teknikte P2'nin ilk ögesi ilk yavrunun ilk ögesi olacak şekilde alınır:

$$O1 = (4 \times \times \times \times \times \times)$$

- Seçilen öge 4 tür. Bu öge P1'de hangi konumda olduğuna bakılır. P1'de 2. konumda olduğunu görüyoruz. Daha sonra bu konumun P2'de karşılık gelen ögesine bakıyoruz. Ve bunun 2 olduğunu görüyoruz. Bu öge O1'in sonraki ögesi olmak üzere yerleştirilir:

$$O1 = (4 \ 2 \times \times \times \times \times)$$

- Seçilen öge P1'de hangi konumda olduğuna tekrar bakıyoruz. Ve bu 4. konumdadır. Tekrar P2'de bu konumda hangi öge varsa , örnekte bu öge 1 dir, tekrar bunu O1'in sıradaki ögesi olarak yerleştiriyoruz:

$$O1 = (4 \ 2 \ 1 \times \times \times \times)$$

- Tekrar aynı adımları kopyalanmayan öge kalana kadar uygulayarak ve O1'i tamamlanmış olur:

$$O1 = (4 \ 2 \ 1 \ 8 \ 5 \ 7 \ 6 \ 3)$$

- Benzer şekilde ikinci yavru için P1=P2 , P2=P1 olacak şekilde yerleri değiştirilir:

P2 = (4 2 5 1 6 8 3 7)

P1 = (3 4 8 2 7 1 6 5)

- Ve O2 elde edilir:

$$O2 = (3 \ 6 \ 7 \ 5 \ 8 \ 1 \ 2 \ 4)$$

Eğer yavrunun son ögesi ilk ebeveynin ilk ögesi olursa konum biter ve ebeveynlerde kopyalayamadığımız öğeler kalır. Örneğin ;

P1 = (1 2 3 4 5 6 7 8)

P2 = (6 8 4 3 1 2 5 7)

- Aynı adımlardan tekrar geçirerek Q1 bireyi oluşturulmaya çalışılır :

$$O1 = (6\ 2\ 8\ 7\ 5\ 1\ \times\ \times)$$

- Ebeveynlerin kopyalanan ve kopyalanmayan öğelerini işaretleriz:

$$P1 = (\bullet\bullet\ 3\ 4\ \bullet\bullet\bullet\bullet)$$

$$P2 = (\bullet\bullet\ 4\ 3\ \bullet\bullet\bullet\bullet)$$

- Yeni yavruları kopyalanmayan öğelerin kombinasyonunu yerleştiririz:

$$O1 = (6\ 2\ 8\ 7\ 5\ 1\ 4\ 3)$$

$$O2 = (1\ 5\ 7\ 8\ 2\ 6\ 3\ 4)$$

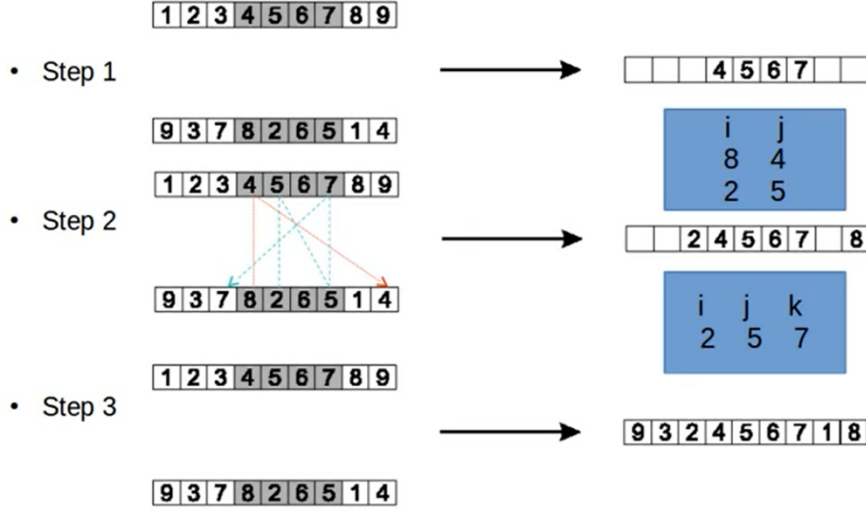
## 2- Partially Mapped Crossover (PMX)

Kısmi Eşleşen Çaprazlama (PMX) iki noktalı bir çaprazlama operatörüdür.

- P1(Parent1) ve P2(Parent2) olan iki ebeveynin genlerinden iki çaprazlama noktası seçilir.
- Başlangıçta P1 ebeveynin geninde rastgele bir segmenti, oluşacak yavrunun C(children) genine konumunu koruyarak kopyalanır.
- Daha sonra aynı konumdan P2'nin geninden kopyalanmamış öğeleri bulunur.
- P2 de kopyalanmayan öğelerin P1 konumundakileri ile eşleştirip haritalandırma yapılır.
- Haritalandırma sonucunda i:P2'nin kopyalanmamış öğeleri ve j:P1 de i'nin eşleşmesi olmak üzere, P2'nin j konumunda C'ye i konur.
- Eğer j konumu C'de dolu ise dolu olan kısmın ögesini k olarak haritaya ekleriz. Ve P2'de k ögesinin olduğu konumu alıp i'yi C'de o konuma koyarız.
- En sonunda C'de boş kalan yerlere P2'deki öğeleri doğrudan koyarız.

Yukardaki adımların örneği *Şekil 5*'teki gibidir:

## PMX example



Şekil 5

### 3- Order Crossover (OX)

Farklı bir çaprazlama tekniği olan Sıralı Çaprazlama(OX) , PMX'in bir varyasyonudur diyebiliriz.

Örnek üzerinden adımlar şu şekildedir :

$$P1 = (3\ 4\ 8\ 2\ 7\ 1\ 6\ 5)$$

$$P2 = (4\ 2\ 5\ 1\ 6\ 8\ 3\ 7)$$

- P1 ve P2 olan ebeveynlerimizde iki çaprazlama noktası belirleriz.

$$P1 = (3\ 4\ 8\ | 2\ 7\ 1\ | 6\ 5)$$

$$P2 = (4\ 2\ 5\ | 1\ 6\ 8\ | 3\ 7)$$

- İki ebeveynlerden rastgele bir segmentini aynı konumda çocuklara ekleriz.

$$O1 = (\times\ \times\ \times\ | 2\ 7\ 1\ | \times\ \times)$$

$$O2 = (\times\ \times\ \times\ | 1\ 6\ 8\ | \times\ \times)$$

- Ebeveynlerin ikinci çaprazlama noktasından itibaren sıralandırmasını yaparız. P2'den örnek gösterecek olursak :

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

- İlk yavruda bulunan segmenti bu ebeveynlerden çıkararak tekrar sıralarız :

$$3 - 4 - 5 - 6 - 8$$

- Ve ilk yavrunun ikinci kesim noktasından itibaren bu sıralamayı yerleştiririz:

$$O1 = (5\ 6\ 8\ |\ 2\ 7\ 1\ |\ 3\ 4)$$

- İkinci yavruya da diğer ebeveyn sırası üzerinden aynı işlemi yaparız :

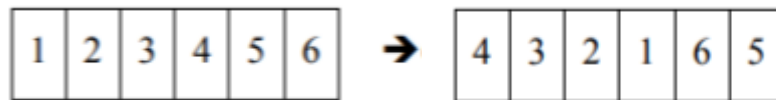
$$O2 = (4\ 2\ 7\ |\ 1\ 6\ 8\ |\ 5\ 3)$$

## MUTASYON OPERATÖRLERİ

Mutasyon operatörü, bir geni ortaya çıkaran kromozomların değiştirilerek popülasyondaki çeşitliliğin artırılması ve korunması için kullanılmaktadır. Çaprazlama sonrasında algoritmanın yerel en iyiye takılıp çeşitliliğin kaybolduğu durumlar olabileceğinden gerekli bir adımdır. Biyolojik mutasyona benzer. Populasyondaki tüm bireyler mutasyona tabi tutulmaz. Mutasyon Operatörü tekli bir operatördür ve üzerinde çalışmak için yalnızca bir ebeveyne ihtiyaç duyar. Belirlediğimiz oran kadar birey mutasyona uğrar. Çeşitli mutasyon operatörleri içinde GSP için öngörülen Centre inverse mutation (CIM), Reverse Sequence Mutation (RSM), Tuvors Mutation, Scramble Mutation, Bit Flip Mutation operatörlerini inceleyeceğiz.

### 1- Centre Inverse Mutation (CIM)

Kromozom iki bölüme ayrılmıştır. Her bölümdeki tüm genler kopyalanır ve sonra ters olarak aynı bölümüne yerleştirilir.

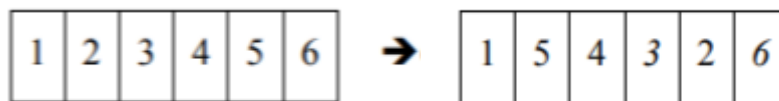


Şekil 6

Şekil 6'daki örnekte 4 ile 5 arasında bölünme yapılmıştır.

### 2- Reverse Sequence Mutation (RSM)

Ters dizi mutasyon operatöründe, gen üzerinde rastgele iki tane i ve j noktası seçilir ve arasında kalan S bölmesini alırız. İki pozisyonu i ve j'yi (n gen uzunluğu olmak üzere) " $1 \leq i \leq j \leq n$ " alınır. Bu dizilimdeki gen düzeni, önceki operasyonda ele alınanlarla aynı şekilde tersinlenir.

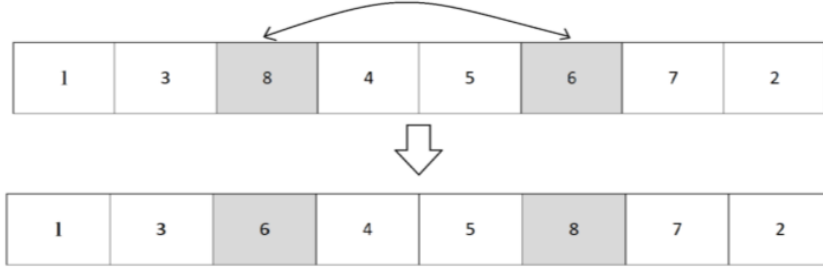


Şekil 7

Şekil 7'deki örnekte 1 ile 6 arasındaki bölme seçilmiştir.

### 3- Twors Mutation (Swap Mutation)

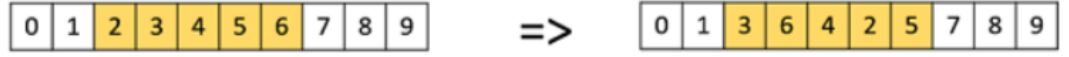
Twors mutasyonu, rastgele seçilen iki genin pozisyonunun değiştirilmesine izin verir. Şekil 8'de Twors mutasyonuna uğramış birey gösterilmiştir.



Şekil 8

### 4- Scramble Mutation

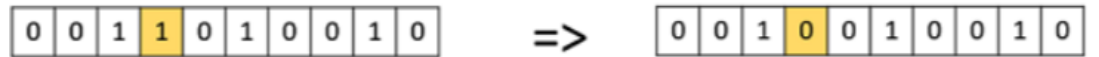
Scramble mutasyonunda kromozomdan bir gen alt kümesi seçilir ve değerleri rastgele karıştırılır. Şekil 9'da Twors mutasyonuna uğramış birey gösterilmiştir.



Şekil 9

### 5- Bit Flip Mutation

Bit çevirme mutasyonunda, bir veya daha fazla gen (dizi indeksleri) seçeriz ve değerlerini çeviririz, yani 1'leri 0'lara ve 0'lari 1'lere çeviririz. Şekil 10'da Twors mutasyonuna uğramış birey gösterilmiştir.



Şekil 10

## UYGULAMA

Çalışma boyunca sabit olacak parametreler şu şekildedir:

Şehir Sayısı	25
Populasyon Sayısı	1000
Jenerasyon Sayısı	2000

### Başlangıç Popülasyonunun Belirlenmesi

Başlangıç popülasyonunun oluşturulması GA'nın ilk adımı olarak belirlenmiştir. Popülasyon boyutu doğru sonuca ulaşmamız için önemli parametrelerden biridir. Başlangıç popülasyonu, her biri olası problemin çözümleri olabilecek bireylerden oluşmaktadır. Popülasyon boyutu arttıkça; popülasyondaki çeşitlilik ve en uygun çözüme ulaşma ihtimali artarken çözümü elde etme süresi de artacaktır. Populasyon sayısını şehir sayısı ile orantılı ve her operatörde sabit kalacak şekilde 1000 olarak belirlendi.

### Uygunluk Değeri

Poşasyondaki her bireyin bir uygunluk değeri vardır. Bu bireyin rota üzerindeki izlediği yolun toplam mesafesidir. Her uğradığı şehrin bir önceki şehirle olan arasındaki uzaklığı öklid mesafesi formülü kullanılarak hesaplanmış ve tüm rota boyunca oluşannbu mesafelerin toplamı bize bireyin uygunluk(fitness) değerini verir.

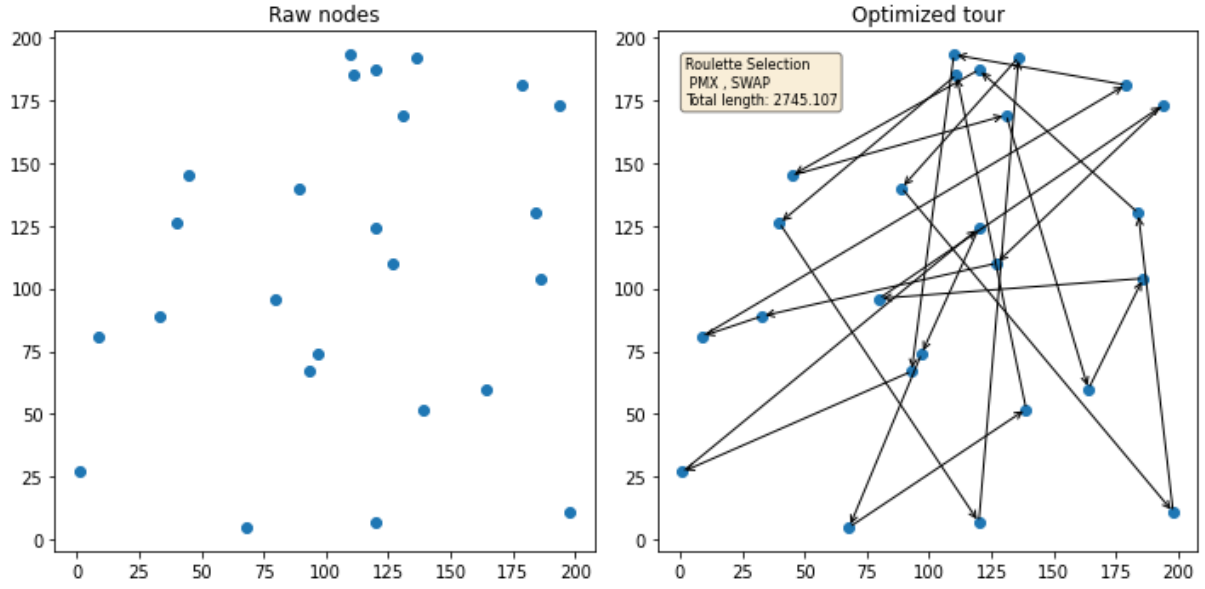
### Seçim Operatörlerinin Karşılaştırılması

İlk operatör adımı olan seçim operatörlerinden “Rank” , “Roulette Whell” , “Tournament”, “Elitism” operatörlerini karşılaştıracamız. Rastgele bireylerden oluşan bu popülasyonda çaprazlama işlemi için seçilecek bireylerde bu seçim operatörleri kullanılır. Seçim operatörlerini işleme sürecinde çaprazlama ve mutasyon operatörleri sabit tutulacak ve seçim operatörlerini değiştirerek sonuçlar gözlemlenecektir.

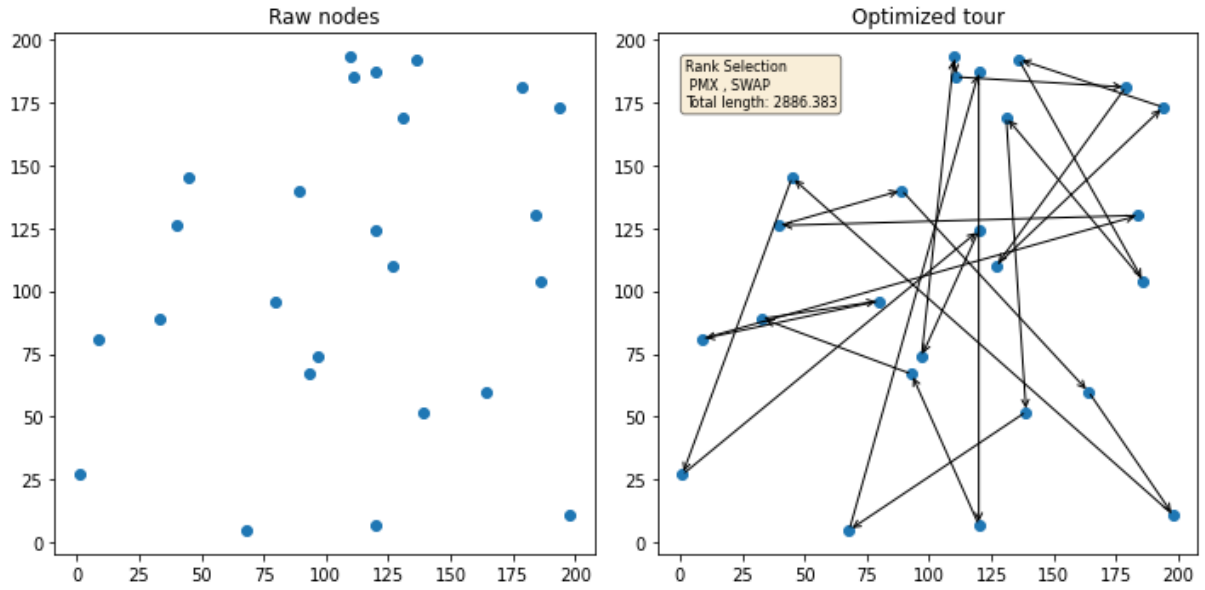
Seçim Operatörleri	Çaprazlama Operatörleri	Mutasyon Operatörleri	Mesafe
Roulette Whell	PMX	SWAP	2745
Rank	PMX	SWAP	2886
Tournament	PMX	SWAP	3026
Elitism	PMX	SWAP	980

Aşağıda Şekil ... deki tablodaki operatörlerin her birinin izlediği yol ve sonuçları grafik üzerinde gösterilmiştir.

#### -Roulette Wheel Seçim Operatörü

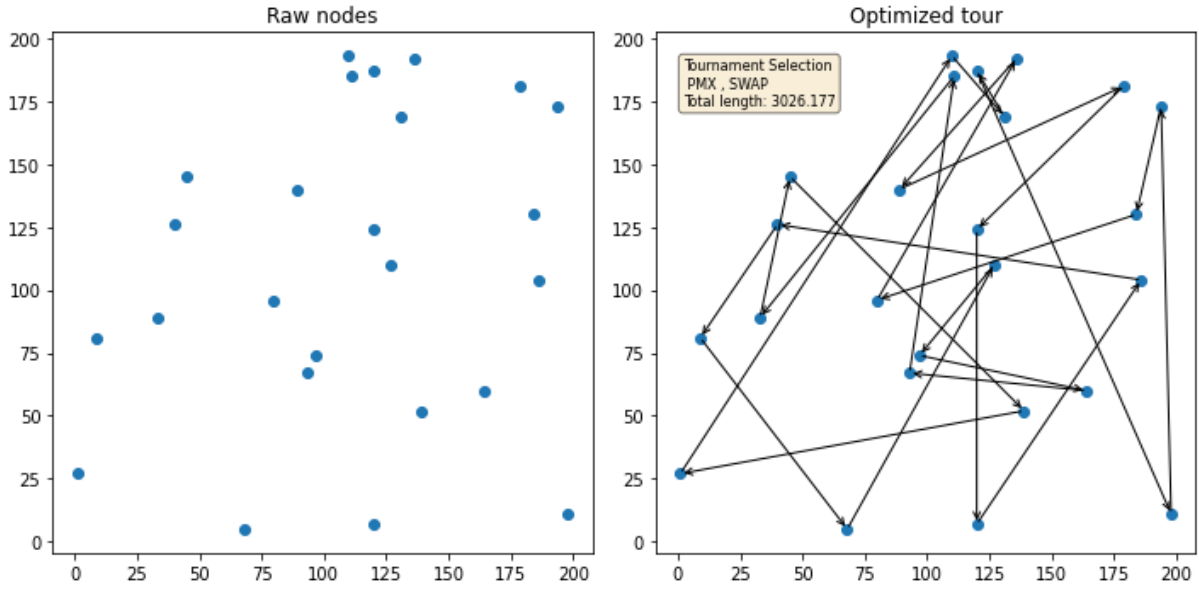


#### -Rank Seçim Operatörü

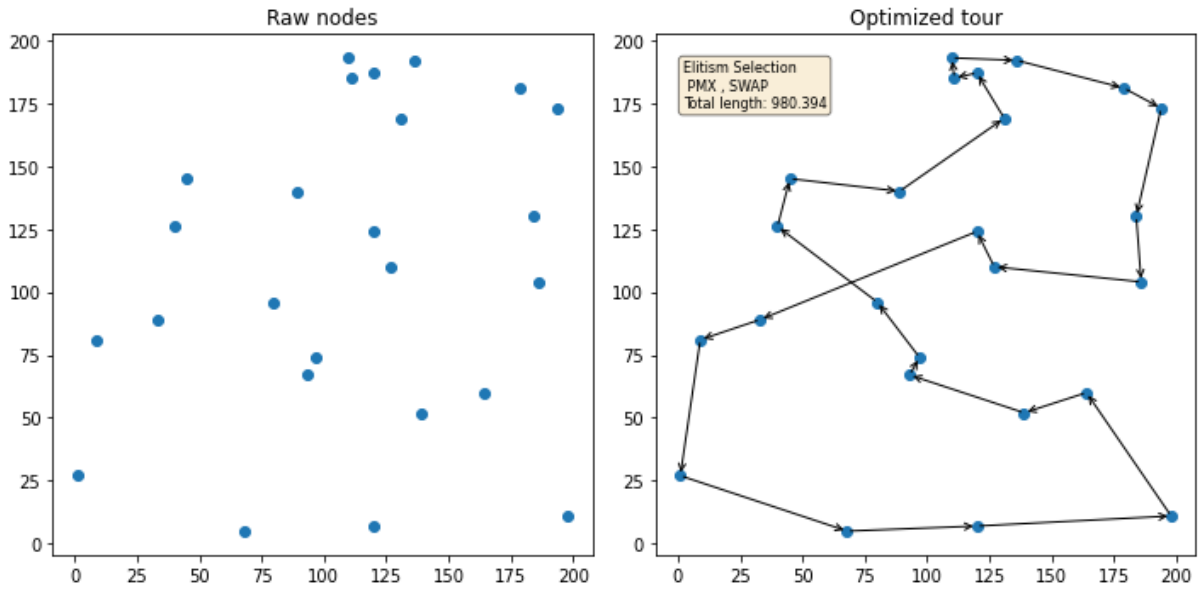




### -Tournament Seçim Operatörü



### -Elitism Seçim Operatörü



### Seçim Operatörlerinin Değerlendirilmesi

Yukardaki test sonuçlarına göre “Elitism” seçim operatörü diğerlerine göre oldukça büyük bir fark atmıştır. “Tournament” en kötü sonucu vermiştir. “Rank” ve “Roulette” arasında pek bir fark yoktur.

Elitism’in bu kadar iyi sonuç vermesinin nedeni sürekli en iyi nesilleri gelecek jenerasyona aktarmasındandır. Bu seçim operatörünün dezavantajı olan “çeşitlilik azlığı”nı da popülasyon sayısı yüksek tutulduğundan dengeleyebilmiştir. Tabi bu sefer run time süresi artmıştır fakat diğerlerine nazaran aralarında en hızlı işlem yapan yine elitismdir.

Uygulama sürecinin geri kalanında elitism seçim operatörü sabit tutularak çaprazlama ve mutasyon operatörleri test edilecektir.

## Çaprazlama Operatörlerinin Karşılaştırılması

Çaprazlama operatörlerinin görevi seçim operatöründen gelen bireyleri ikili bir şekilde alıp çaprazlama sonucunda yeni birey üretmektir. Bu oluşan yeni birey gelecek nesile aktarılır.

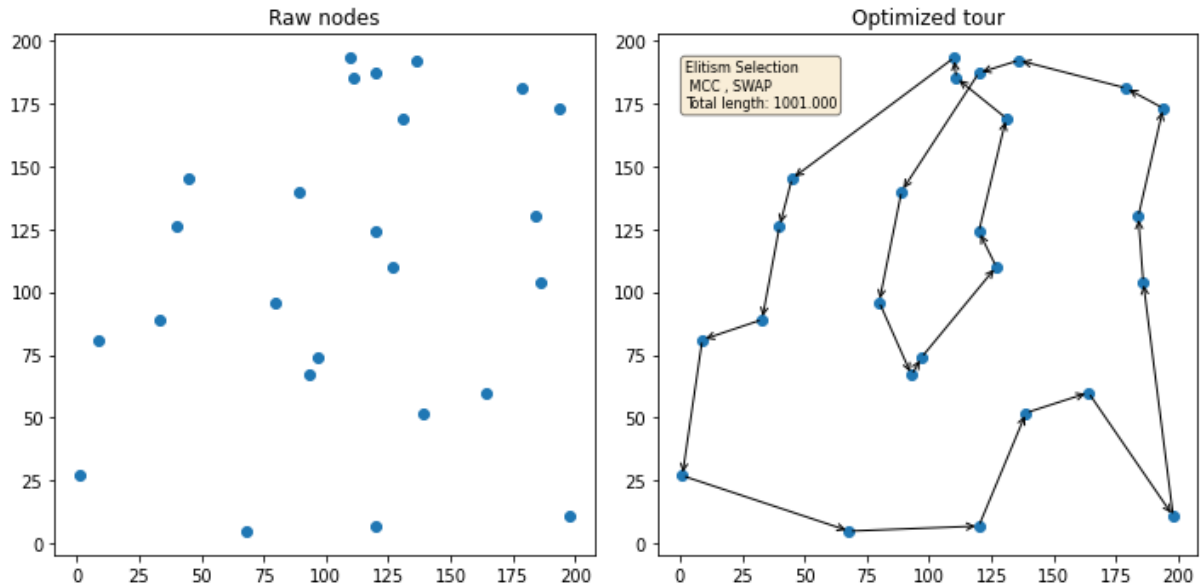
Modified-Cycle Crossover (MCC) ve Partially Mapped Crossover (PMX) çaprazlama operatörleri karşılaştırılacaktır. Order Crossover (OX) ise uygulamamıza katılmasında çeşitli sorunlar görüldüğü için teste tabii tutulmamıştır. Bu sorun iki partnerin çaprazlama sonucu oluşturduğu bireyde aynı şehrin birden fazla kez bulundurulabilmesi ihtimalinden kaynaklanmaktadır. Test edilirken seçim operatörü ve murasyon operatörü sabit tutulmuştur.

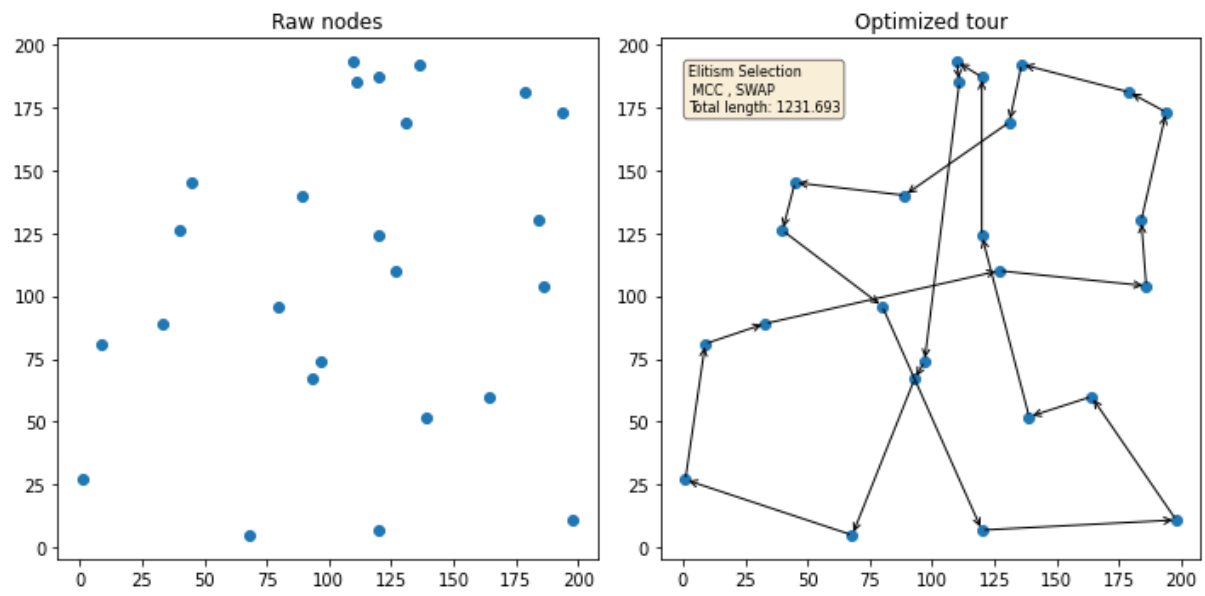
Çaprazlama Operatörleri	Seçim Operatörleri	Mutasyon Operatörleri	Mesafe
MCC	Elitism	SWAP	1001(1) 1231(2)
PMX	Elitism	SWAP	980(1) 1243(2)

Değerler yakın olduğundan 2 kere test edilip sonuçları paylaşılmıştır.

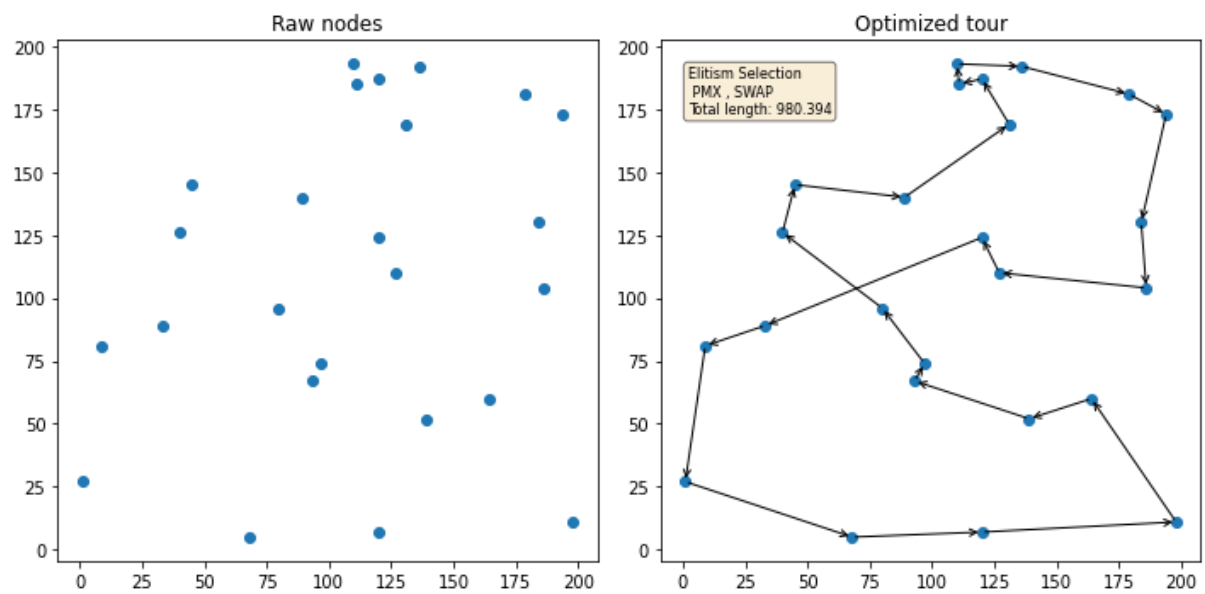
Aşağıda Şekil ... deki tablodaki operatörlerin her birinin izlediği yol ve sonuçları grafik üzerinde gösterilmiştir.

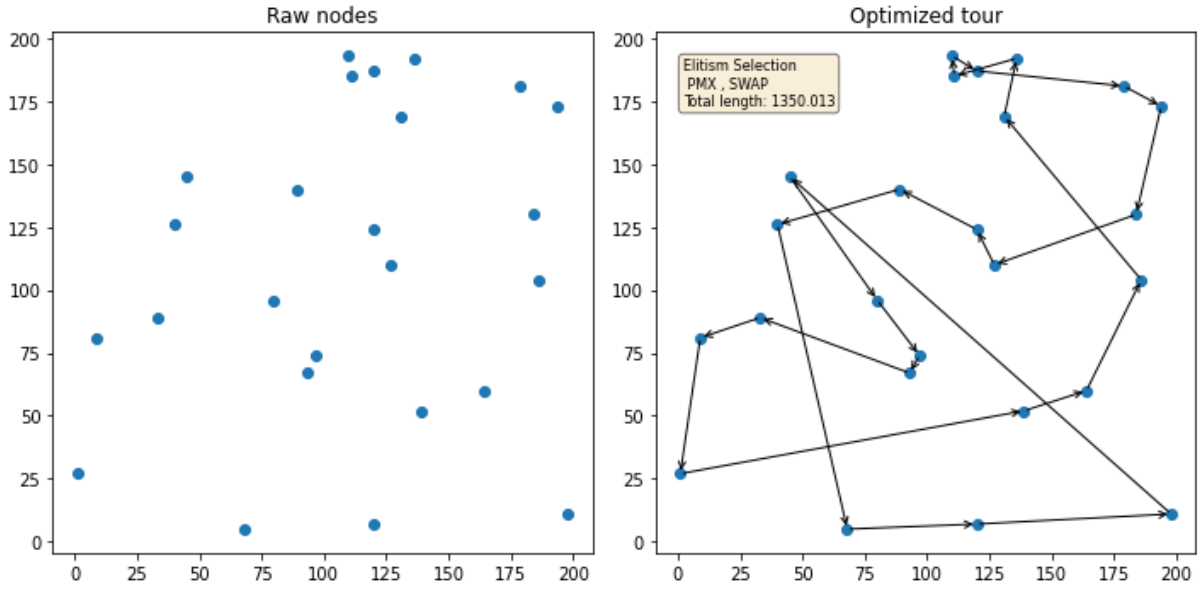
### - Modified-Cycle Crossover (MCC)





- Partially Mapped Crossover (PMX)





### Çaprazlama Operatörlerinin Değerlendirilmesi

En kısa mesafeleri baz alırsak sonuç olarak arada çok fark olmasa da PMX operatörünün kazandığını söyleyebiliriz. Bunun sebebi MCC operatörünün bir nebze de olsa bireyleri daha karmaşık şekilde karar. Bu da her bireyi literatür olarak çaprazlama tabi tuttuğumuz için fazla çeşitliliğe yol açar.

Kıvrıkcık kumral saçlı ve uzun boylu bir anne ile siyah düz saçlı kısa boylu bir babadan oluşan bir çift var. Bu verilerle PMX in oluşturduğu birey kıvrıkcık kumral saçlı ve kısa boylu bir çocukken , MCC nin oluşturduğu birey kumral düz saçlı ve kısaboyludur.

Uygulamanın geri kalan testinde çaprazlama operatörlerinden PMX sabit kalacak şekilde devam edilecektir.

### Mutasyon Operatörlerinin Karşılaştırılması

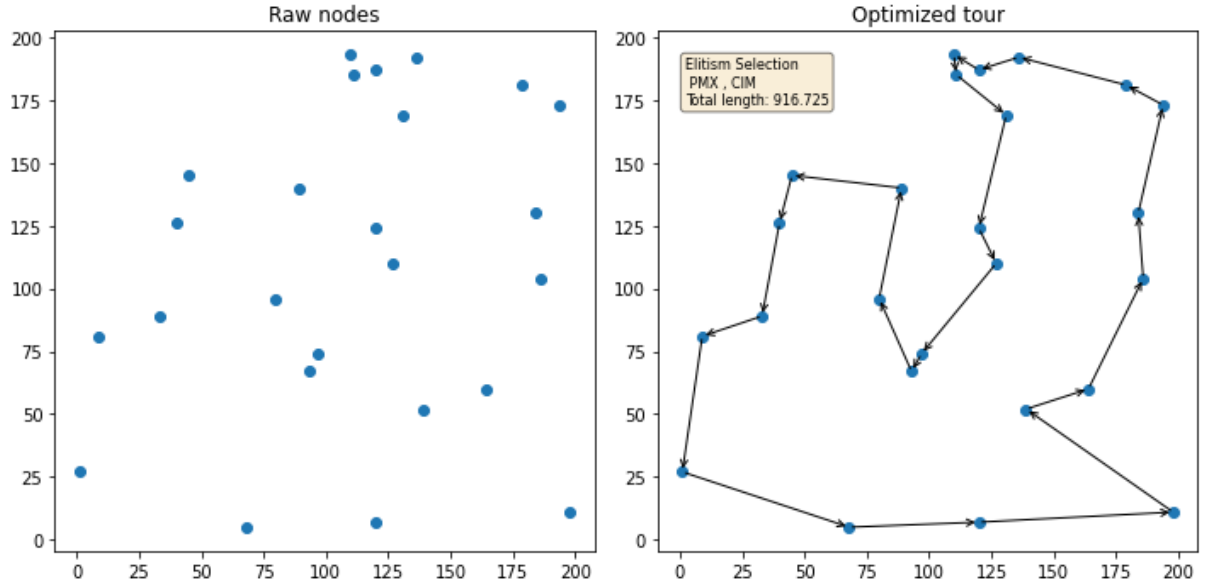
Mutasyon operatörleri bize ihtiyacımız olan çeşitliliği vermede ve korumada yardımcı olan bir adımdır. Çaprazlama sonucundaki popülasyonu gelecek nesile aktarmadan hemen önce, yani gelecek nesili oluşturmada son adımdır. Bu operatör popülasyondan uygun mutasyon oranı kadar bireyi mutasyona uğratır.

Centre Inverse Mutation (CIM) , Reverse Sequence Mutation (RSM) , Scramble , Swap(Twors) operatörlerini karşılaştıracğız.

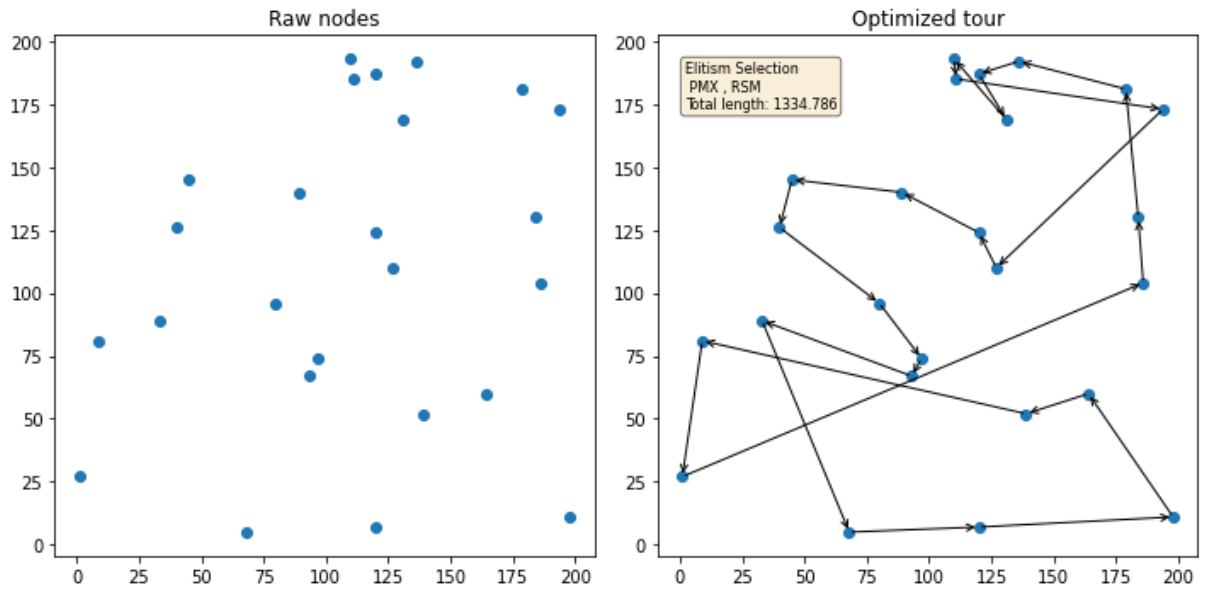
Mutasyon Operatörleri	Seçim Operatörleri	Çaprazlama Operatörleri	Mesafe
CIM	Elitism	PMX	916
RSM	Eltisim	PMX	1334
Scramble	Eltisim	PMX	1356
Swap	Eltisim	PMX	980

Aşağıda Şekil ... deki tablodaki operatörlerin her birinin izlediği yol ve sonuçları grafik üzerinde gösterilmiştir.

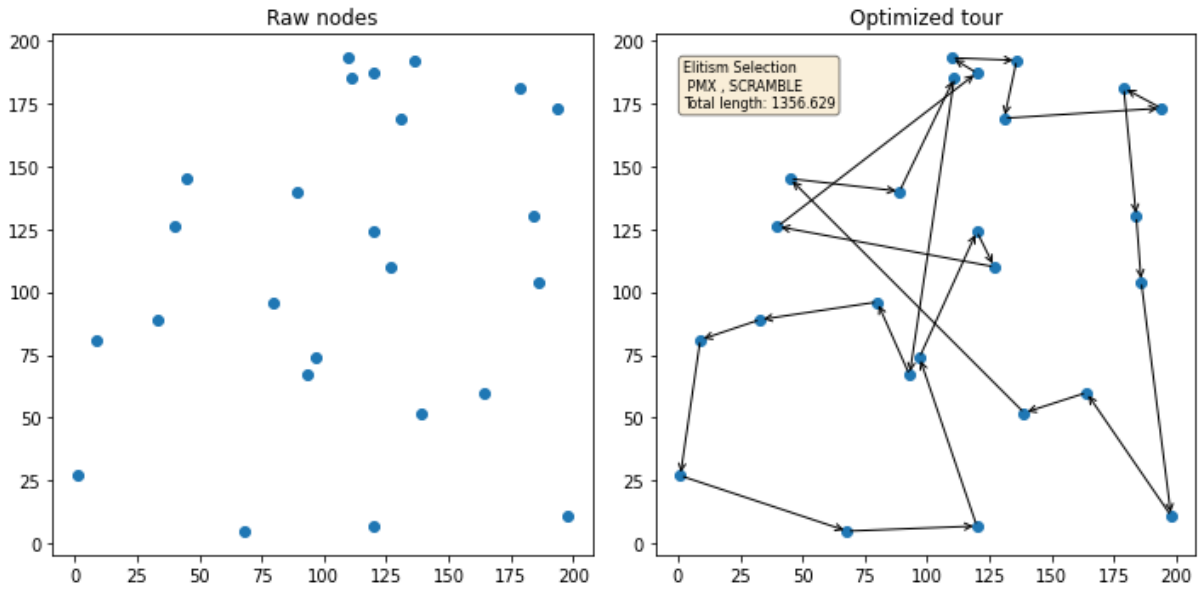
#### -Centre Inverse Mutation(CIM)



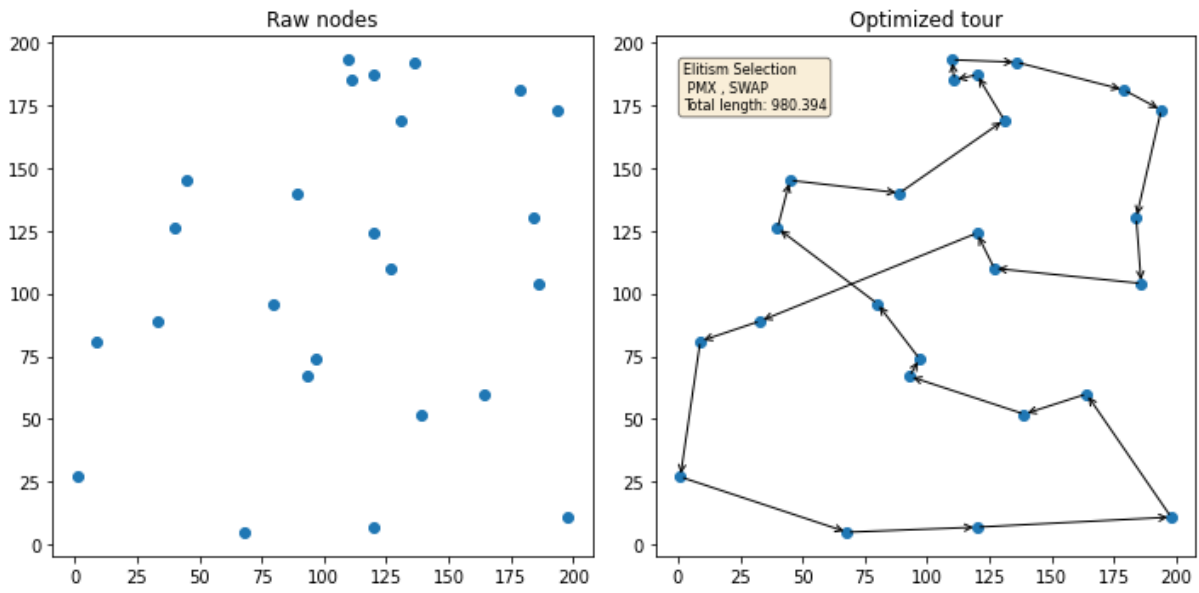
#### -Reverse Sequence Mutation(RSM)



### -Scramble Mutation



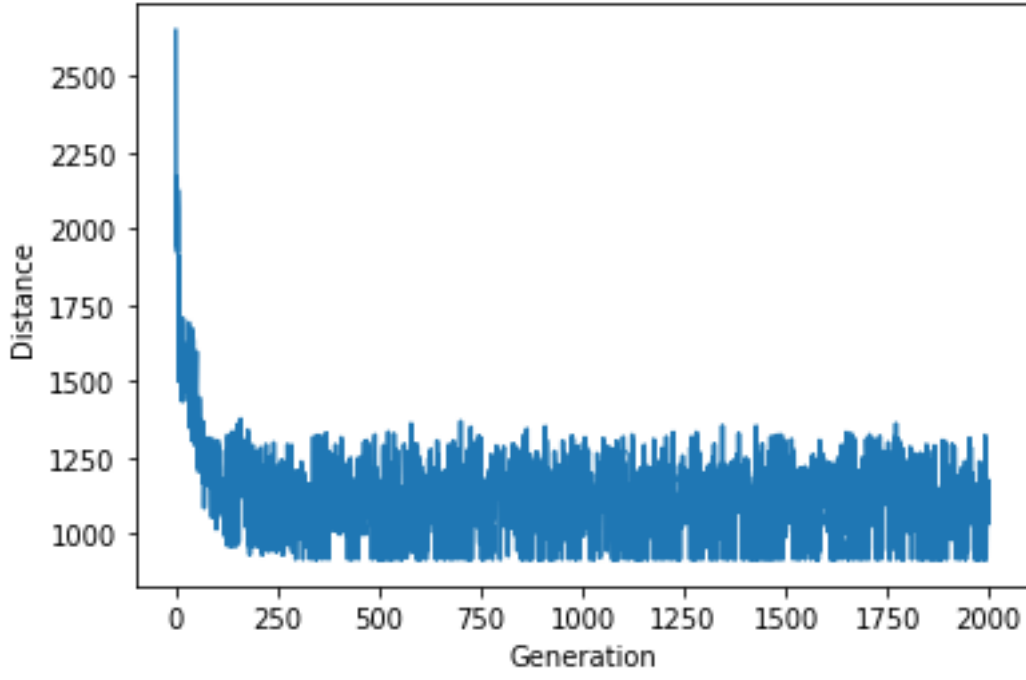
### -Swap(Twos) Mutation



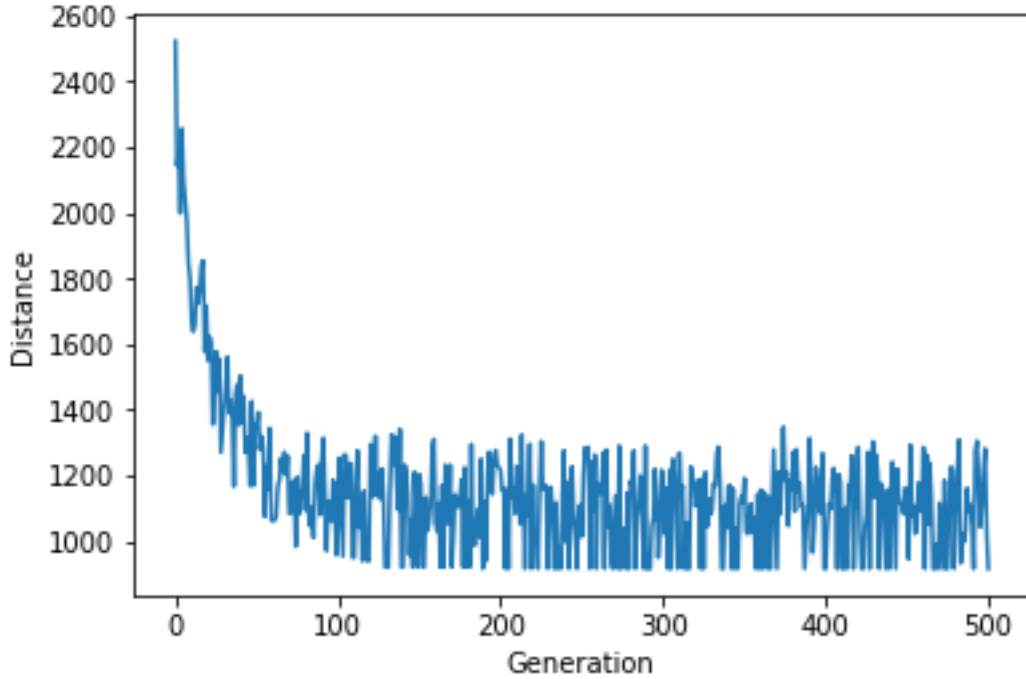
## Mutasyon Operatörlerinin Değerlendirilmesi

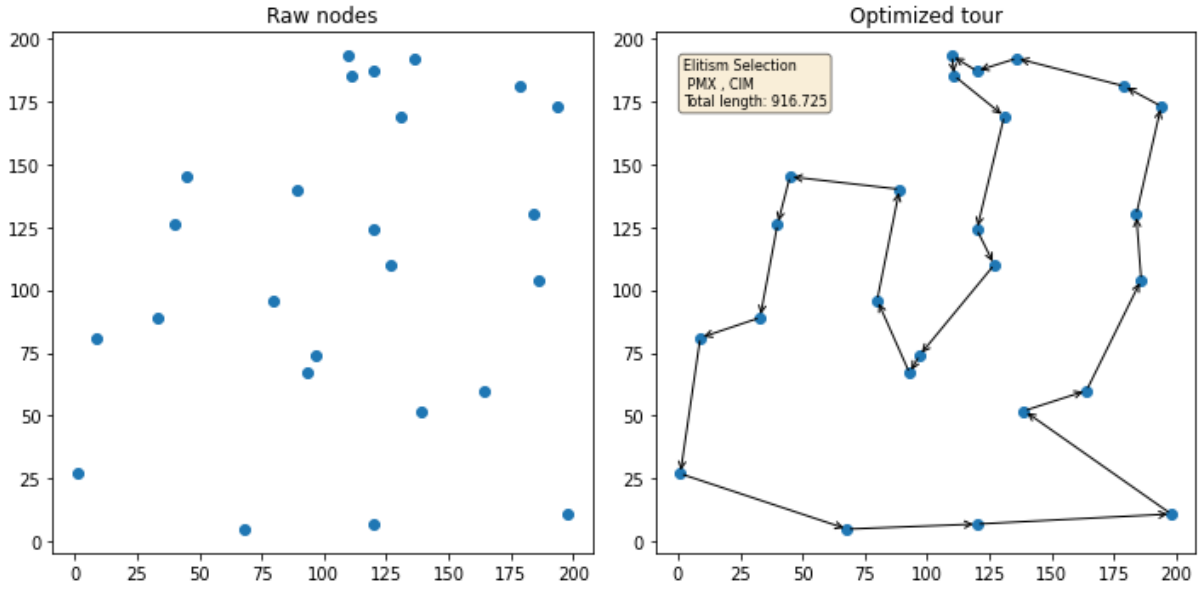
RSM ve Scramble operatörlerinin sonuçları birbirine yakın ve diğerlerine göre büyük denilebilecek bir farkla daha kötü sonuç verdikleri için elendiler. CIM ve Swap operatörleri birbirine yakın olsa da burda en kısa mesafeyi veren iyileştirme yolundaki asıl husustur. Sonuç olarak CIM 63.669 fark ile en iyi seçilmiştir.

Operatörleri belirlemiş bulunmaktayız. Yukarda gösterilen en iyi performansın jenerasyona sayısına bağlı olarak mesafesini gösteren bir grafik Şekil .. deki gibidir.



Görünen o ki 0 ile 250. Jenerasyonlar arası mesafe bir hayli düşüp daha sonra sabit kalmıştır. Bu nedenle 2000 jenerasyon sayısı sadece çalışma süresini arttırıyor. 2000 sabit değişken olan jenerasyon sayısını en fazla 500 belirleyerek kodun çalışma süresinden tasarruf edilebilir.





Yukardaki Şekil..’da görüldüğü üzere 500. jenerasyon da aynı en iyi sonucu verebilmektedir.

### Sonuç

2 boyutlu XY düzlemi üzerinde 25 noktadan oluşan ve belirlediğimiz popülasyon sayısı kadar birey bu 25 noktanın her birine uğrayıp başladığı yere dönme kuralıyla ilk popülasyon oluştu. Bu başlangıç popülasyonu her run time edildiğinde rastgele oluşur.

Seçim operatörüne ilk döngüde bu başlangıç popülasyonu girer ve daha sonra seçim operatörlerinin seçtiği bireyler bunlar çaprazlama operatörüne uğrayarak yeni bireyler oluştururlar. Yani başlangıç popülasyonundan oluşan yeni bireyler topluluğu denilebilir. Bu yeni jenerasyonun çeşitliliğini korumak için popülasyondan belirlenen mutasyon oranı kadar birey mutasyona uğrar. Çalışmamızda mutasyon oranı 0.01 ve popülasyon sayısı 1000 ‘dir. Bu da 10 birey mutasyona uğrar demektir. Belirlenen jenerasyon sayısı kadar bu döngü devam eder. Durdurma kriteri olarak jenerasyon sayısının sonuna gelindiğinde eldeki son popülasyonda en kısa mesafeye sahip bireyin rotasyonu sonuç olarak belirlenir ve grafiğe çizilir.

Opsiyonel olan seçim , çaprazlama , mutasyon adım operatörlerini test sonuçlarına göre alternatif olan seçilmiştir. Bunlara ek olarak popülasyon sayısı , jenerasyon sayısı ve kodun genel işleyiş düzeni de sonuçlara etki eden diğer etmenlerdir. Bu yolda karşılaştırmalar değerlendirilip son haline gelebilmiştir.

Belirlenen operatörler ve sabit değişkenler şu şekildedir :

- Popülasyon Sayısı : 1000
- Jenerasyon Sayısı :500
- Şehir Sayısı:25
- Elitism Seçim Operatörü
- Partially Mapped Crossover (PMX)
- Centre Inverse Mutation(CIM)



Süreçler rastgele temelli olduğundan diğer algoritmalarla karşılaştırıldığında GA'nın hem avantajları hem de dezavantajları bulunmaktadır. Rastgele oluşturulan çözüm alternatifleri ile optimum sonucu elde etme ihtimali yüksek olurken diğer yandan hiç elde edilememe durumu da oluşmaktadır. Bu nedenle, en kesin sonucu bu çalışmada gereken bir kural olarak başlangıç popülasyonu rastgele bireyden oluşturmak yerine sabit kalmasının vereceğini düşünüyorum.

Travelling Salesman Problemini iyileştirilmiş genetik algoritma ile amaç doğrultusunda çözülmüştür.

#### **Kullanılan Dil ve Kütüphaneler :**

Python Programlama Dili

Numpy , Random , Operator , Matplotlib Kütüphaneleri

## KAYNAKÇA

<https://www.geeksforgeeks.org/traveling-salesman-problem-using-genetic-algorithm/>

<https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35>

<https://dergipark.org.tr/en/download/article-file/1043003>

<https://setu677.medium.com/how-to-perform-roulette-wheel-and-rank-based-selection-in-a-genetic-algorithm-d0829a37a189>

<https://jaketae.github.io/study/genetic-algorithm/>

<https://www.hindawi.com/journals/cin/2017/7430125/>

<https://dergipark.org.tr/tr/download/article-file/534290>

<https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html>

<https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012131/pdf>

<https://www.mdpi.com/2078-2489/10/12/390>

<https://setu677.medium.com/how-to-perform-roulette-wheel-and-rank-based-selection-in-a-genetic-algorithm-d0829a37a189>

<https://www.zditect.com/blog/59401757.html>

[https://www.academia.edu/39020666/Solving\\_TSP\\_Using\\_Genetic\\_Algorithm\\_with\\_Tournament\\_Selection\\_Variants](https://www.academia.edu/39020666/Solving_TSP_Using_Genetic_Algorithm_with_Tournament_Selection_Variants)

<https://towardsdatascience.com/use-genetic-algorithms-and-evolutionary-computing-for-solving-the-travelling-salesman-problem-b9623ccc9427>