
GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE KULLANILAN GENETİK ALGORİTMANIN PARAMETRELERİNİN İNCELENMESİ

Meryem PULAT¹

İpek DEVECİ KOCAKOÇ²

Öz

Gezgin satıcı problemi, optimizasyon alanında araştırmacı ve akademisyenler tarafından üzerinde uzun yıllardır yoğun olarak çalışılan çözümü zor (NP-hard) bir problemdir. Gezgin satıcı problemi tanımlanması kolay olmasına rağmen optimal çözümü elde etmek çok zordur. Aynı zamanda birçok problem gezgin satıcı problemi olarak modellenebilmektedir. Problemdaki değişken sayısı arttıkça olası çözüm sayısı da büyük oranda arttığından kesin yöntemlerle kısa sürede optimal çözüm elde etmek mümkün değildir. Bu yüzden bu probleme kısa sürede makul bir çözüm bulan yöntemler kullanılmaktadır. Bu yöntemlerden en sık kullanılan sezgisel yöntemlerden biri de genetik algoritmalarıdır. Genetik algoritmalar özellikle geleneksel optimizasyon yöntemlerinin daha az etkin olduğu zor optimizasyon problemlerini çözmek için uygundur. Genetik algoritmalar gezgin satıcı problemi gibi çeşitli NP-hard problemleri çözmek için kullanılan en iyi yöntemlerden biridir. Bu çalışmada genetik algoritmanın performansı üzerinde önemli etkiye sahip olan parametreleri üç farklı büyüklükteki problem ve üç farklı çaprazlama operatörü ile kıyaslayarak belirlemek amaçlanmıştır.

Anahtar Kelimeler: Gezgin Satıcı Problemi, Genetik Algoritmalar, Genetik Algoritma Parametreleri.
JEL Sınıflandırması: C61, C44, C49.

INVESTIGATION OF PARAMETERS OF GENETIC ALGORITHM IN THE SOLUTION OF TRAVELING SALESMAN PROBLEM

Abstract

The Traveling salesman problem is a difficult (NP-hard) problem that has been studied intensively by researchers and academics in the field of optimization for many years. Although it is easy to identify the traveling salesman problem, it is very difficult to obtain the optimal solution. At the same time, many problems can be modeled as traveling salesman problems. As the number of variables in the problem increases, the number of possible solutions increases greatly, so it is not possible to obtain an optimal solution in a short time with exact methods. Therefore, methods that find a reasonable solution in a short period of time are used for this problem. One of the most commonly used heuristic methods is genetic algorithms. Genetic algorithms are particularly suited to solve difficult optimization problems where traditional optimization methods are less effective. Genetic algorithms are one of the best methods used to solve various NP-hard problems such as the Traveling salesman problem. In this study, we aimed to determine parameters which have significant effect on the performance of genetic algorithm by comparing with three different size problem and three different crossover operators.

Keywords: Traveling Salesman Problem, Genetic Algorithms, Genetic Algorithm Parameters.
JEL Classification: C61, C44, C49.

¹ Arş. Gör., Dokuz Eylül Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Ekonometri Bölümü, meryem.pulat@deu.edu.tr

² Prof. Dr., Dokuz Eylül Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Ekonometri Bölümü, ipek.deveci@deu.edu.tr

1. Giriş

Gezgin satıcı probleminde aralarındaki uzaklıkları bilinen şehirlerin her birinden yalnız bir kez geçen en kısa yolu veya en az maliyetli turu bulmayı hedeflemektedir. Gezgin satıcı problemi tanımlanması kolay olmasına rağmen optimal çözümü elde etmek çok zordur ve literatürde NP-zor (Non polynomial-hard) problemler arasında yer almaktadır. Bu problemin temel zorluğu olası tur sayısının şehir sayısı arttıkça büyük oranda artmasıdır bu da problemi kesin yöntemlerle kısa sürede çözümünü imkânsız hale getirmektedir bu yüzden problemi kısa sürede çözebilen farklı yöntemler öne sürülmüştür. Bu yöntemlerden biri de genetik algoritmalarıdır.

Genetik algoritmalar rastlantısal arama tekniklerini kullanarak çözüm bulmaya çalışan parametre kodlama esasına dayanan sezgisel bir arama tekniğidir. Bu yöntem optimal çözümü garanti edemez ancak kısa sürede makul bir çözüm elde edebilmektedir. Genetik algoritmalar özellikle geleneksel optimizasyon yöntemlerinin daha az etkin olduğu zor optimizasyon problemlerini çözmek için uygundur.

Çalışmanın amacı, gezgin satıcı problemini çözmek için kullanılan genetik algoritmaların performansını önemli ölçüde etkileyen parametrelerin farklı değerlerini incelemek ve uygun değerleri belirlemektir.

İlk olarak genetik algoritma özellikleri, parametreleri ve operatörleri açıklanmıştır. Daha sonra gezgin satıcı problemi tanımı, özellikleri, çözüm yöntemleri açıklanmıştır ve gezgin satıcı probleminin genetik algoritmalarla çözümü ele alınmıştır. Son kısımda TSPLIB 'den alınan üç farklı boyuttaki problem kullanılarak MATLAB programında çözüm yapılmıştır ve uygun parametreler belirlenmiştir.

2. Genetik Algoritmalar

Genetik algoritmalar (GA), genetik ve evrim ilkelerine dayanan olasılıksal, sağlam (robust) ve sezgisel arama algoritmasıdır (Javidi vd., 2015:27). GA evrimsel hesaplamadaki en popüler tekniklerden biridir (Sivanandam ve Deepa, 2008:2). GA, John Holland tarafından 1960 yılında ortaya atılmış ve Holland öğrencileri ve arkadaşları tarafından 1960 ve 1970 'lerde Michigan üniversitesinde geliştirilmiştir (Melanie, 1998:2). 1975 yılında Holland'ın "Adaptation in natural and artificial system" isimli kitabında doğal evrim prensibinin optimizasyon problemlerine nasıl uygulanacağını açıklanmıştır.

2.1. Genetik Algoritmaların Tasarımı

Basit GA yapısı evrimsel programlama yapısına benzerdir. t iterasyonu boyunca potansiyel çözümlerin popülasyonunu sürdürmektedir. Çözüm uygunluk değerine göre değerlendirilir. Yeni popülasyon t+1 iterasyonda daha uygun bireylerin seçimi ile oluşmaktadır. Oluşturulan yeni popülasyondaki bireyler çaprazlama ve mutasyon ile değişime uğramaktadır ve yeni çözümler oluşmaktadır (Michalewicz, 1992:17).

2.1.1. Kodlama

Birey genlerinin temsil (gösterim) sürecidir. Bireyin nasıl kodlanacağı ilgilenilen probleme göre değişmektedir. İkili (Binary) kodlama, 16 tabanlı sayı sistemi (Heksadesimal) kodlama, yol (Path) kodlama, değer kodlama örnek olarak verilebilir.

Şekil 1: İkili Kodlama Yapısı

Kromozom A	1 1 0 1 0 0 0
	1
Kromozom B	0 1 0 1 1 0
	0 1

Şekil 2: Heksadesimal Kodlama Yapısı

Kromozom A	6 A B
7	
Kromozom B	2 C A
E	

Şekil 3: Yol Kodlama Yapısı

Kromozom A	1 2 5 4 3
6	
Kromozom B	6 5 2 1 3
4	

Şekil 4: Değer Kodlama Yapısı

Kromozom A	1,2324 5,3244 0,4556 2,3525
Kromozom B	A B J H E D J H A B G I U J F D A

2.1.2. Başlangıç Popülasyonun Oluşturulması

Gen havuzu olarak adlandırılan dizilerin başlangıç popülasyonu başlangıç popülasyonu çoğunlukla rasgele seçilir ama genetik algoritmaların performansını geliştirmek için farklı sezgisellerde kullanılmaktadır (Sivanandam ve Deepa, 2008:42).

2.1.3. Uygunluk Fonksiyonu

Genetik biliminde bireyin çevre şartına ne kadar dayanıklı olduğunu belirten uygunluk değeri GA ile çözümün kalitesini belirler. Başlangıç popülasyonu oluşturulduktan sonra yapılacak ilk işlem bu popülasyonda yer alan kromozomların uygunluk değerlerinin hesaplanmasıdır (Çolak, 2010:427). Hangi kromozomların (diziler) bir sonraki nesle taşınacağı ve hangi kromozomların yok olacağı uygunluk değerinin büyüklüğüne göre karar verilmektedir (Elmas, 2016:389).

2.1.4. Seçim

Seçme işlemi tabiatta çevre tarafından, yapay sistemlerde ise amaç fonksiyonu ve diğer kalite değerlendirme işlemleri tarafından kontrol edilir. Darwin 'in teorisine göre; en iyi olanlar çocuğu oluşturmak için hayatta kalır. Bireylerin bir jenerasyondan diğer kuşağa geçerken kalite değerlerine göre daha fazla kopya edilme şansları vardır. Böylece, daha kaliteli çözümler popülasyonda barınırken, kalitesiz olanların yavaş yavaş kaybolması sağlanmaktadır (Karaboğa, 2014:81). Hangi dizilerin bir sonraki nesle aktarılacağı farklı seçim mekanizmaları ile sağlanır. Bunlara örnek olarak orantılı rulet tekerleği, *sıraya dayalı rulet tekerleği* ve *turnuva seçimi* verilebilir.

2.1.5. Çaprazlama

İki ebeveyn çözümü olarak onlardan çocuk üretme sürecidir. Seçim işleminden sonra çaprazlama ile popülasyon daha iyi bireyler ile zenginleştirilmiştir. Farklı çözümler arasında bilgi alışverişi sağlar (Michalewicz, 1992:17). Bir önceki nesilden daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörleri kullanılmaktadır. GA Holland tarafından ortaya atıldıktan sonra probleme bağlı olarak birçok çaprazlama operatörü araştırmacılar tarafından ortaya atılmıştır. Çünkü GA performansı etkileyen en önemli operatörlerden biridir (Esmkhan ve Zamanifar, 2012:1). Çaprazlama operatörlerine örnek olarak *tek noktalı çaprazlama*,

çok noktalı çaprazlama, pozisyona dayalı çaprazlama, örnek olarak verilebilir. Bunlardan temel olan iki tanesi açıklanmıştır.

2.1.6. Mutasyon

Diğer önemli genetik algoritma operatörü mutasyon operatörüdür. Çaprazlamadan sonra diziler mutasyona tabi tutulur. Mutasyon geleneksel basit bir arama operatörü olarak kabul edilmektedir ve popülasyonda genetik çeşitliliği korumak için bir arka plan operatör olarak görülmektedir. Kromozomlarda rasgele değişiklikler yapılarak GAs arama uzayında yeni kısımlara ulaşılmasını sağlar. Aynı zamanda önemli özelliklerin erken kaybolmamasının ve böylece eşleşme havuzunda çeşitliliğin korunmasını sağlamaktadır (Ahmed, 2010:101). GA performansını etkileyen operatörlerden biridir. Problem türüne bağlı olarak farklı mutasyon operatörleri ortaya atılmıştır. Çevirme (Flipping), Ters Çevirme (Reversing), Karşılıklı Değişim (Interchanging) en temel olanlarıdır.

2.1.7. Genetik Algoritmaların Parametreleri

Popülasyon büyüklüğü, çaprazlama oranı ve mutasyon oranı genetik algoritmanın parametreleridir. GA parametreleri GA'nın performansı üzerinde çok önemli bir etkiye sahiptir. Bu parametrelerin değeri hakkında bir kesinlik mevcut değildir problemle bağlı olarak değişmektedir.

Popülasyon Büyüklüğü: En önemli kararlardan biri olan popülasyon büyüklüğü GA'nın genel performansını ve GA'nın etkinliğini iki şekilde etkilemektedir (Grefenstette, 1986:124). Popülasyon büyüklüğü *çok küçük* olduğunda; çoğu hiperdüzlem için yetersiz örneklem büyüklüğü sağlamaktadır bu da aramanın belirli bir alt optimal noktaya sürükleyecektir. Popülasyon büyüklüğü *çok büyük* olduğunda ise; çok sayıda hiperdüzlemin temsiliyi içermektedir. Aynı zamanda alt optimal çözüme yakınsamayı da engellemektedir. Ancak büyük popülasyon nesil gelişimi için daha fazla süreye ihtiyaç duymaktadır bu da istenmeyen bir durumdur.

Çaprazlama Oranı: Çaprazlamanın amacı mevcut iyi kromozomları birleştirerek daha iyi kromozom oluşturmaktır. Temel parametrelerden biri olan çaprazlama olasılığı (P_c), ne sıklıkla çaprazlamanın yapılacağını tanımlayan bir parametredir (Javidi vd., 2015:31). Bu oran büyükse; popülasyon değişkenliği hızlı bir şekilde gerçekleştirir. Bu oran düşükse; aramanın çok yavaş gerçekleşmesine sebep olmaktadır (Elmas, 2016:394).

Mutasyon Oranı: Mutasyon yapılmasının nedeni; birbirini izleyen daha uygun bireylerin atılmasından gelmektedir. Mutasyon oranı (P_m), ne sıklıkla mutasyonun yapılacağını tanımlayan bir parametredir. Bu oran büyükse; GA rasgele aramaya dönüşmektedir. Bu oran düşük ise yerel optimum tuzağına düşmesine sebep olmaktadır (Pham vd., 1997:157).

3. Gezgin Satıcı Problemi

Gezgin satıcı problemi (TSP-Traveling Salesman Problem) aralarındaki uzaklıkları bilinen n adet noktanın (şehir veya düğüm) her birinden yalnız bir kez geçen en kısa yolu veya en az maliyetli turu bulmayı hedeflemektedir (Cevre vd., 2007:1).

Gezgin satıcı problemi ilk olarak 1930'lu yılların başında Karl Menger tarafından matematiksel olarak tanımlanmış kombinyonel optimizasyon (combinatorial optimization) problemleri arasında yer almaktadır (Çolak, 2010: 424). TSP'nin araştırmacının ilgisini çekmesi ve aktif bir araştırma alanı olmaya devam etmesinin başlıca dört sebebi vardır (Allaoua ve Brahim, 2015:110; Gopal vd., 2015:151):

- I. Gezgin satıcı probleminin tanımlanması kolay olmasına rağmen çözümü zordur NP-zor (NP-hard) bir problemdir.
- II. NP-zor problemler problem boyutu büyük olduğunda problemi kısa sürede kesin olarak çözmek için etkin bir yol bulunamamıştır.
- III. Çok sayıda gerçek dünya problemi TSP ile modellenebilir.
- IV. Farklı sonuçları karşılaştırarak çözüm kalitesini ölçmeyi sağlayan TSPLIB kütüphanesi vardır.

TSP'nin Tam Sayılı Doğrusal Programlama Formülasyonu aşağıdaki gibidir (Laporte, 1992:233):

$$(1) \quad \text{Minimize:} \quad \sum_{i \neq j}^n c_{ij} x_{ij}$$

Kısıtlar:

$$(2) \quad \sum_{j=1}^n x_{ij} = 1 \quad i=1,2,\dots,n$$

$$(3) \quad \sum_{i=1}^n x_{ij} = 1 \quad j=1,2,\dots,n$$

$$(4) \quad \sum_{i,j \in S} x_{ij} \leq |S| - 1$$

$$S \subset V, 2 \leq |S| \leq n - 2$$

$$(5) \quad x_{ij} \in \{0,1\}$$

$$i, j = 1, \dots, n \quad i \neq j$$

Yukarıdaki eşitlikte x_{ij} karar değişkenleri, c_{ij} şehir i den şehir j ye seyahat maliyeti ve n şehir sayısını göstermektedir. (1) numaralı eşitlik amaç fonksiyonunu verir. Amaç fonksiyonu optimal turun maliyetini tanımlamaktadır. (2) numaralı kısıt her köşeye bir kez girileceğini; (3) numaralı kısıt her köşeden bir kez ayrılacağını (4) numaralı kısıt alt tur eleme kısıtıdır diğer bir deyişle oluşabilecek alt turlardan kurtulmaya yöneliktir. Uygun çözümleri tek turla sınırlandırmaktadır. (5) numaralı kısıt ikili (binary) koşuludur.

Gezgin satıcı probleminde maliyet matrisinin yapısı dikkate alındığında TSP simetrik gezgin satıcı problemi (STSP) ve asimetrik gezgin satıcı problemi (ATSP) olarak iki gruba ayrılmaktadır. STSP de tüm i,j için $c_{ij} = c_{ji}$ olur. Eğer $c_{ij} \neq c_{ji}$ olduğunda ATSP dir (Cevre vd., 2007:2). N şehirli STSP $(n - 1)!/2$ olası çözüm varken ; ATSP için $(n - 1)!$ olası çözüm vardır (Ahmed, 2010:96).

Tablo 1: STSP ve ATSP için Olası Tur Sayısı

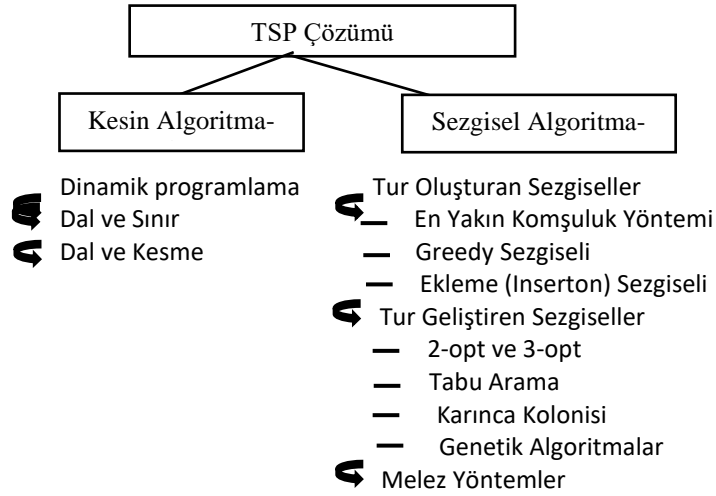
Şehir (Düğüm, Nokta) sayısı	ATSP için Olası tur sayısı $(n - 1)!$	STSP için Olası tur sayısı $(n - 1)!/2$
10	362880	181440
13	479001600	239500800
15	87178291200	43589145600
22	$5,11 \times 10^{19}$	$2,55 \times 10^{19}$

Tablo 1'de görüldüğü gibi şehir sayısı çok küçük olduğunda bile olası tur sayısı çok büyüktür ve şehir sayısı arttıkça olası tur sayı da üstel olarak artmaktadır.

3.1. Gezgin Satıcı Problemi Çözüm Yöntemleri

Gezgin satıcı problemi kombinasyonel optimizasyon problemidir ve problemin NP-zor olduğu bilinmektedir. Problem boyutu arttıkça olası tur sayısı arttığından kesin olarak kısa sürede çözmek zor hatta imkânsız hale gelmektedir. TSP çözmek için kesin (exact) ve sezgisel (heuristic) algoritmalar geliştirilmiştir (Potvin, 1996:339).

Şekil 5: Gezgin Satıcı Probleminin Çözüm Yöntemleri



3.2. Gezgin Satıcı Probleminin Genetik Algoritmalarla Çözümü

3.2.1. Kodlama

Gezgin satıcı problemini genetik algoritmalar kullanarak çözmek için pek çok farklı gösterim (temsil) kullanılmaktadır. Bunlar ikili(binary) gösterim, yol(path) gösterimi, komşuluk(adjacency) gösterimi, sıralı(ordinal) gösterim ve matris(matrix) gösterimi. Farklı gösterim türlerinin tercih edilmemesinin nedeni: ikili gösterimde çok küçük problem durumunda kullanılabilir ancak problem büyüdükçe her şehir $\lceil \log_2 n \rceil$ bit dizisi olarak kodlandığından gösterim turları yönetemeyecek büyüklükte olmaktadır. Aynı zamanda klasik operatörler geçerli çocuk turuyla sonuçlanamaz tamir algoritması kullanmayı gerektirir. Komşuluk gösteriminde bu gösterime özgü çok az sayıda çaprazlama operatörü geliştirilmiştir. Ancak bütün geliştirilen çaprazlama operatörleri düşük performans göstermiştir. Sıralı gösterimde klasik operatörler kullanılabilir tamir algoritmasının kullanımını gerektirmemektedir. ancak kötü sonuçlar verdiği için tercih edilmemektedir. Matris gösteriminde ise temel zorluk geçerli çocuğa yol açan operatörleri tanımlamaktadır. TSP'nin GA çözümünde gösterim olarak yol gösterimi kullanılmaktadır ve bu gösterime ilişkin çok sayıda operatör geliştirilmiştir (Larranaga vd., 1999:166).

Yol Gösterimi: Gezgin Satıcı Problemi bir sıralama problemidir. Yol gösteriminde; ziyaret edilmesi gereken n şehir n elemanlı listeye göre sıraya konur. i 'inci şehir listenin j 'inci elemanı ise şehir i ziyaret edilecek j 'inci şehirdir. Genetik algoritmalar ile TSP çözümü bu gösterim kullanılarak yapılmaktadır (Larranaga vd., 1999:134).

Şekil 6: Yol Kodlama Yapısı

Kromozom A	2 1 5 3 4 7
	6
Kromozom B	6 5 2 7 1 3
	4

Yedi şehirli bir örneği (olay/durum) ele alırsak düğümlerin etiketleri $\{1, 2, 3, 4, 5, 6, 7\}$ şeklindedir. Tur ise; $\{2 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 2\}$ olarak ya da $(2\ 1\ 5\ 3\ 4\ 7\ 6)$ olarak gösterilebilir.

3.2.2. Uygunluk Fonksiyonu

Kromozomun uygunluk değeri genetik algoritmalar yaklaşımına göre maksimize edilmelidir. TSP de amaç en düşük maliyetle turu bulmaktır bu en düşük maliyetli tur en kaliteli çözümü temsil

etmektedir. Bu yüzden dönüşümle uygunluk fonksiyonu maksimize edilmelidir. Dönüşüm şu şekilde gerçekleştirilebilir (Bhattacharyya ve Bandyopadhyay, 2008:7):

$$f(x) = \frac{1}{d}$$

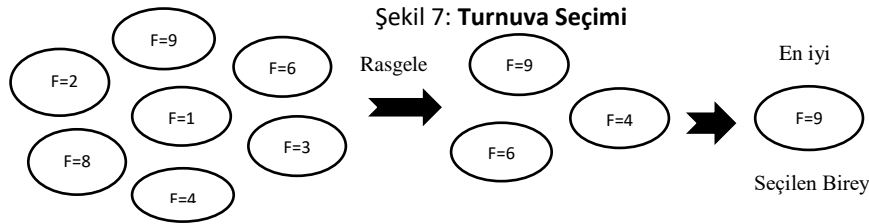
d: Kromozom ile temsil edilen turun maliyeti (uzaklık/uzunluk)

3.2.3. Seçim

Turnuva seçimi kolaylık ve uygunluğu açısından en yaygın olarak kullanılan seçim yöntemidir. Turnuva seçiminin aşağıdaki özellikleri onu özellikle de çok amaçlı optimizasyon durumunda kullanışlı hale getirir (Yu ve Gen, 2010:74):

- Turnuva seçimi sadece yerel bilgiyi kullanır.
- Turnuva seçiminin uygulaması kolaydır ve zaman karmaşıklığı azdır.
- Turnuva seçimi paralel olarak kolaylıkla uygulanabilir.

Sıralı ölçekleme ile kıyaslandığında sıralı ölçekleme tüm popülasyonu sıralamayı gerektirir bu da zaman alıcıdır (Melanie, 1998:127). Paralel olarak uygulama açısından bakıldığında ise; diğer yöntemleri paralelleştirmek daha zordur. Çünkü; bir miktar paralel bilgiye ihtiyaç duyarlar. Orantılı seçim, fonksiyon değerlerinin toplamın gerektirirken; sıralı seçim, global sıralamayı elde edebilmek için tüm bireylere ve bunların fonksiyon değerlerine erişmeyi gerektirir (Goldberg ve Deb, 1991:82).



Kaynak: Razali ve Geraghty, 2011: 3.

3.2.4. Çaprazlama

İki ebeveyn seçim yöntemi ile seçildikten sonra çaprazlamaya tabi tutulur. Çaprazlama iki ebeveynden iki çocuk (çözüm) oluşturmak için eşleyen bir operatördür. Gezgin satıcı problemine klasik GA kullanılan çaprazlama operatörleri uyguladığında geçerli olmayan turlar oluşturduğu görülmektedir.

TSP problemleri için araştırmacılar tarafından çok sayıda çaprazlama operatörü ortaya atılmıştır. PMX (Partially mapped crossover), CX (Cycle crossover), OX (Order crossover), ERX (Edge recombination crossover), EERX (Enhanced edge recombination crossover), PMX (Partially mapped crossover), OX2 (Order based crossover), POS (Position based crossover), MPX (Maximal preservative crossover), VR (Voting recombination crossover) TSP için geliştirilmiş operatörlerinden bir kısmıdır. Tamamı (Pulat ve Deveci Kocakoç, 2016:10-17) çalışmalarında ele alınmıştır. Uygulamada kullanılan CX (Dairesel çaprazlama-Cycle crossover), OX (Sıralı çaprazlama-order crossover) ve EERX (Gelişmiş kenar birleştirme çaprazlama-Enhanced edge recombination crossover) çaprazlama operatörleri aşağıda açıklanmıştır.

CX (Dairesel Çaprazlama-Cycle Crossover): Oliver ve arkadaşları (1987) tarafından önerilmiştir. Şehirlerin alt kümelerine odaklanır (Potvin, 1996:352).

Ebeveyn A : 1 2 3 4 5 6 7 8
Ebeveyn B : 2 4 6 8 7 5 3 1

Alt Kümeler:

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \end{pmatrix} \begin{pmatrix} 4 \\ 8 \end{pmatrix} \begin{pmatrix} 8 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Tekrardan 1-2 olduğundan yani başa döndüğünden bulunanlar alt kümeleri oluşturmaktadır. Alt kümeler aynı pozisyona kopyalanır kalan pozisyonlar diğer ebeveynden alınmaktadır.

Çaprazlama Sonrası:

Çocuk A: 1 2 6 4 7 5 3 8

Çocuk B: 2 4 3 8 5 6 7 1

Bu şekilde her bir şehrin konumu iki ebeveynden birinden gelmektedir. Her iki ebeveynde de elemanların ortalama yarısının pozisyonu korunmaktadır.

OX (Sıralı Çaprazlama-Order Crossover): Davis (1985) tarafından önerilmiştir. Şehirlerin konumu değil sırası önemlidir. İlk olarak kesim noktaları rasgele seçilir. Burada, ilk kesim noktasının 2. ve 3. bit arasında ve ikincisinin 5. ve 6. bit arasında seçildiğinde;

Ebeveyn A: 1 2 | 3 4 5 | 6 7 8

Ebeveyn B: 2 4 | 6 8 7 | 5 3 1

Kesim noktaları arasındaki kısım çocuğa kopyalanır.

Çocuk A: -- | 3 4 5 | ---

Çocuk B: -- | 6 8 7 | ---

Sonra ikinci kesim noktasından başlayarak zaten mevcut olan şehirleri atlayarak diğer ebeveynde görüldükleri sırada çocuğa kopyalanır dizin sonuna ulaşıldığında ilk pozisyondan devam edilmektedir.

Çaprazlama Sonrası:

Çocuk A: 7 1 | 3 4 5 | 2 6 8 (~~2~~ 4 6 8 ~~5~~ 3 1)

Çocuk B: 4 5 | 6 8 7 | 1 2 3 (1 2 3 4 ~~5~~ 6 ~~7~~ 8)

EERX (Gelişmiş kenar birleştirme çaprazlama-Enhanced edge recombination crossover): Starkweather, Mcdaniel, Mathias, ve Darrell (1991) tarafından önerilmiştir.

EERX operatörü şu şekilde çalışır (Starkweather vd., 1991:70):

Adım1: Turdaki her şehir için bir kenar listesi oluşturulur. Bir şehrin komşu şehirleri her iki ebeveynde mevcutsa bu negatif işaret (-) kullanılarak gösterilir. İki ebeveyn turundan başlangıç şehri rasgele seçilir bu mevcut şehir olarak alınır.

Adım2: Mevcut şehir tüm kenar listesinden kaldır.

Adım3: Bir sonraki şehri seçme işlemi şu şekilde yapılmaktadır: Bu operatörün mevcut şehrin kenar listesindeki negatif işaretli şehirler ilk önceliğe sahiptir ancak hiç negatif işaretli şehir yoksa kendi kenar listesinde en az bağlantıya sahip şehir ikinci önceliğe sahiptir. Eğer aynı sayıda bağlantılı şehir varsa bu durumda seçim rasgele yapılır. Şehri seç adım 2'ye git.

Ebeveyn A: (1 2 3 4 5 6)

Ebeveyn B: (2 4 3 1 5 6)

Yukarıda verilen ebeveyn turları için kenar haritası şu şekildedir:

Tablo 2: EERX Turlar için Kenar Haritası

Şehir	Kenar Listesi
1	2, 6, 3, 5
2	1, 3, 4, 6
3	2,- 4, 1
4	- 3, 5, 2
5	4,- 6, 1
6	1,- 5, 2

Başlangıç şehri olarak [2] seçildiğinde ve adımlar uygulandığında çaprazlama sonrası tur şu şekildedir: [2 6 5 1 3 4]

3.2.5. Mutasyon

Gezgin satıcı problemini çözmek için çok fazla sayıda çaprazlama operatörü geliştirilmesine rağmen mutasyon ile ilgili çok az sayıda operatör önerilmiştir. Bunun nedenleri arasında mutasyon oranının çok düşük alınması çaprazlama oranına göre yani çok küçük değişimler meydana getirmektedir. TSP özgü önerilen mutasyon operatörüne örnek olarak; **DM** (Displacement mutation), **EM** (Exchange mutation), **SIM** (Simple inversion mutation), **SM** (Scrabble mutation), **IVM** (Inversion mutation) verilebilir. Uygulamada kullanılan **yer değiştirme mutasyonu** (*Swap mutation*) aşağıda açıklanmıştır.

Yer Değiştirme Mutasyonu (Swap Mutation): İki şehir rasgele bir şekilde seçilir ve karşılıklı yer değiştirilir. Bu mutasyon operatörü orijinal mutasyon felsefesine çok yakındır. Çünkü; turda sadece çok küçük değişiklikler yapmaktadır (Potvin, 1996:361).

Mutasyon Öncesi: 2 6 5 1 3 4

Mutasyon Sonrası: 2 3 5 1 6 4

4. Uygulama

Bu çalışmanın amacı, gezgin satıcı problemini çözmek için kullanılan genetik algoritmaların performansını önemli ölçüde etkileyen parametrelerin farklı değerlerini incelemek ve uygun değerleri belirlemektir. Çalışmada kullanılan veriler **TSPLIB** 'den alınmıştır ve Matlab R2015b programında kodlanmıştır.

TSPLIB 1990'da Gernard Reinelt tarafından kurulmuş gezgin satıcı problemi (ve benzer problemler) ait 14 şehirden 85900 şehir boyutuna kadar 100 üzerinde örnek bulunmaktadır. Bu örneklerin optimal çözüm değerleri de yer almaktadır. Optimal çözüm değerlerinin var olması elde edilen değerlerin kıyaslanabilmesini ve optimal değere ne kadar yaklaşıldığının görülmesini sağlamaktadır. Çalışmada farklı boyuttaki problem üzerinde parametrelerin etkisini görebilmek için küçük, orta ve büyük olmak üzere üç farklı boyutta veri seçilmiştir. Çalışmada '**eil51**', '**eil76**' ve '**eil101**' verileri üzerinde inceleme yapılmıştır. Buradaki sayılar şehir sayısını göstermektedir. Bu veriler için optimal çözüm değerleri sırasıyla şu şekildedir: **426**, **538** ve **629**.

İlk olarak genetik algoritmanın performansını önemli ölçüde etkileyen parametrelerden biri olan uygun popülasyon büyüklüğü belirlenmektedir. Bu maksimum nesil sayısı ile birlikte değerlendirilmiştir. Daha sonra çaprazlama operatörü olarak **OX**, **CX** ve **EEAX** kullanılmıştır. Bu üç operatör sırasıyla şehirlerin sırasını, konumunu ve kenarlarını korumayı amaçlamaktadır. Bu üç operatörün seçilmesinin nedeni; hangi korumanın daha iyi sonuç verdiğini görebilmektir. Bu çaprazlama operatörleri arasından en iyi olan seçilmiştir. Mutasyon operatörü olarak yer değiştirme mutasyonu kullanılmıştır. Daha sonra en iyi popülasyon büyüklüğü ve nesil sayısı belirlendikten sonra bir diğer önemli parametrelerden biri olan mutasyon oranı için farklı değerler deneyerek uygun değer belirlenmeye çalışılmıştır.

Uygulamada kullanılan üç şehir için başlangıç popülasyonu rasgele bir şekilde üretilmiştir. Seçim yöntemi olarak **turnuva seçimi** kullanılmış, turnuvaya uğrayacak kromozom sayısı k farklı denemeler sonucunda 5 olarak belirlenmiştir. Tüm bireyler çaprazlamaya tabi tutulmuştur. Farklı değerlerin çözüm sonuca etkisine bakılarak uygun mutasyon oranı 0.01 alınmıştır.

4.1. 51 Şehir için İnceleme

51 şehirli problemde öncelikle uygun popülasyon büyüklüğünü belirlemek için [20-2000] arasında farklı popülasyon büyüklüğü ile [100-5000] arasında farklı nesil sayısı OX, CX ve EEX çaprazlama operatörü kullanılarak karşılaştırılmıştır. En iyi sonucu veren P_s ve N seçilmiştir. Eil51 için 15 tekrar sonucu elde edilen değerlerin ortalaması şu şekildedir:

Tablo 3'e bakıldığında belirli nesil sayısında popülasyon büyüklüğünü arttırıldığında sonuçlar iyileşmektedir. Popülasyon büyüklüğünün 2000 ile sınırlandırılmasının nedeni daha önce değinildiği gibi popülasyon büyüklüğü çok büyük olduğunda hesaplama zamanını arttırmaktadır bu da istenmeyen bir durumdur. 2000'den sonra algoritmanın çalışma zamanını arttırdığından 2000'den sonrası dikkate alınmamıştır aynı şekilde maksimum nesil sayısı 5000'den fazla incelenmesinin sebebi nesil sayısı da çalışma zamanı üzerinde olumsuz etkiye sahip olmaktadır.

Belirli bir nesil sayısında popülasyon büyüklüğünü her arttırıldığında sonuç tüm çaprazlama operatörlerinde daha da iyileşmiştir. Aynı zamanda nesil sayısını arttırıldığında da bir önceki nesil değerlerine bakıldığında nesil sayısını arttırılması olumlu etkiye sahiptir. Çözüm sonucu küçük bir değerden başlayıp popülasyon büyüklüğü arttıkça daha iyi yani daha düşük değer elde edilmiştir. $G=100$ ve $P_s=20$ bakıldığında çözüm değeri $1.295e+03$; $G=500$ ve $P_s=20$ olduğunda değeri $1.099e+03$ yani daha düşük bir değer elde edilmiştir böylece daha iyi bir değerle başlayıp popülasyon büyüklüğünün artırılmasıyla önemli oranda iyileşme göstermektedir. Çaprazlama operatörlerini kıyasladığında en iyi değerler OX ile elde edilmiştir daha sonra CX ve en kötüsü ise EERX.

Eil51 için; $G=500$, $P_s=2000$ ve OX operatörleri ile en iyi değer **470.5779** olarak elde edilmiştir. Bu problem için optimal çözüm değeri ise 426' dır.

Tablo 3: Eil51 için G, Ps ve Çaprazlama Operatörlerinin Sonuçları

Nesil sayısı (G)	Popülasyon büyüklüğü (P_s)	CX	Hesaplama Zamanı (saniye)	OX	Hesaplama Zamanı (saniye)	EERX	Hesaplama Zamanı (saniye)
G=100	$P_s=20$	562,5358	0,27	544,9796	0,60	565,6101	3,52
	$P_s=40$	553,1187	0,43	542,9809	0,69	560,5374	6,86
	$P_s=100$	560,8414	1,14	529,5886	1,18	555,1215	16,60
	$P_s=200$	546,8842	1,81	525,8089	2,01	545,4876	32,47
	$P_s=1000$	528,0577	11,82	494,1491	9,06	527,2362	158,41
	$P_s=2000$	519,4636	32,65	491,6080	19,46	493,2827	328,60
G=500	$P_s=20$	555,9036	0,90	535,3246	1,59	562,4611	17,38
	$P_s=40$	544,8516	2,64	503,8325	2,54	559,5374	33,89
	$P_s=100$	528,5809	6,58	509,1363	5,16	535,9904	81,44
	$P_s=200$	522,1803	9,88	502,9264	9,31	524,9827	162,59
	$P_s=1000$	493,9065	59,85	485,3694	45,73	497,3474	757,41
	$P_s=2000$	499,0217	177,39	479,5519	98,20	489,3468	1509,68

G=1000	P _s =20	540,6834	1,35	535,5380	2,89	538,3267	32,75
	P _s =40	532,9093	2,33	503,7230	4,88	530,9193	62,99
	P _s =100	518,3735	8,45	487,6658	10,22	519,9131	155,29
	P _s =200	511,2961	15,37	485,7942	18,73	514,9502	301,74
	P _s =1000	504,6697	116,46	479,8622	92,67	489,3468	1627,76
	P _s =2000	503,5949	414,42	479,5487	196,62	489,3468	3200,85
G=5000	P _s =20	529,4	9,66	505,3773	13,16	517,0883	173,13
	P _s =40	515,4277	16,01	491,3615	23,09	495,8959	340,46
	P _s =100	489,3468	43,57	488,9915	49,01	503,5949	816,07
	P _s =200	477,3942	83,93	475,9989	92,40	494,0631	1506,71
	P _s =1000	492,5889	508,81	475,9449	472,48	489,3408	9547,76
	P _s =2000	480,8778	1819,87	476,3189	976,13	482,6375	16268,11

4.2. 76 Şehir için İnceleme

Eil76 için de 15 tekrar sonucu elde edilen değerlerin ortalaması şu şekildedir:

Tablo 4'e bakıldığında belirli nesil sayısında popülasyon büyüklüğünü arttırdıkça sonuçlar iyileşmektedir. Belirli bir nesil sayısında popülasyon büyüklüğünü her arttırdığında sonuç tüm çaprazlama operatörlerinde daha da iyileşmiştir. Aynı zamanda nesil sayısını arttırıldığında bir önceki nesil değerlerine bakıldığında nesil sayısının arttırılması olumlu etkiye sahiptir. Daha küçük bir değerden başlayıp popülasyon büyüklüğü arttıkça daha iyiye gitmektedir. G=500 ve Ps=40 baktığımızda değeri 1.4087e+03; G=1000 ve Ps=40 olduğunda değeri 1.2166e+03 yani daha düşük bir değer elde edilmiştir. Çaprazlama operatörlerini kıyasladığında en iyi değerler genellikle OX ile elde edilmiştir daha sonra CX ve en kötüsü ise EEAX.

Eil76 için; **G= 5000, Ps=2000** ve **OX** operatörleri ile en iyi değer **581.0146** olarak elde edilmiştir. Bu problem için optimal çözüm değeri ise 538 'dir.

Tablo 4: Eil76 için G, Ps ve Çaprazlama Operatörlerinin Sonuçları

Nesil sayısı (G)	Popülasyon büyüklüğü (P _s)	CX	Hesaplama Zamanı (saniye)	OX	Hesaplama Zamanı (saniye)	EEAX	Hesaplama Zamanı (saniye)
G=1000	P _s =20	1.9789e+03	0,37	2.0092e+03	0,63	2.0954+03	7,60
	P _s =40	1.9242e+03	0,52	1.6187e+03	0,94	1.18439e+03	12,78
	P _s =100	1.6307e+03	1,13	1.3083e+03	1,68	1.7505e+03	29,05
	P _s =200	1.6287e+03	1,95	1.2715e+03	2,80	1.6139e+03	59,20
	P _s =1000	1.2027e+03	8,01	962,9679	21,51	1.4006e+03	287,73

	P _s =2000	1.1865e+0 3	19,80	926.2154	29,67	1.2974+03	553,38
	P _s =20	1.6272e+0 3	1,00	1.6857e+0 3	2,30	1.8924e+0 3	23,98
	P _s =40	1.4087e+0 3	1,67	1.3318e+0 3	3,38	1.5190e+0 3	46,20
	P _s =100	1.2476e+0 3	5,19	1.1996e+0 3	7,17	1.3662e+0 3	113,54
G=500	P _s =200	1.1073+03	10,25	1.0705e+0 3	10,43	1.2034e+0 3	229,18
	P _s =1000	931,5209	51,48	744,6683	52,18	926,7359	1177,89
	P _s =2000	792,56668	143,71	724,6413	117,21	888,6731	2469,11
	P _s =20	1,5909e+0 3	1,87	1.4780e+0. 3	3,44	1.4754e+0 3	54,40
G=1000	P _s =40	1.2166e+0 3	3,14	1.2303e+0 3	6,12	1.2805e+0 3	102,90
	P _s =100	1.1889e+0 3	11,17	940,5132	12,83	1,1481e+0 3	249,18
	P _s =200	934,3077	20,81	965,6038	24,74	976,9619	521,90
	P _s =1000	867,1732	125,45	747,2906	126,71	804,0556	2308,62
G=5000	P _s =2000	797,9674	407,24	702,16669	250,44	760,5126	4845,81
	P _s =20	1.1291e+0 3	8,35	1.1269e+0 3	16,08	1.0907e+0 3	251,98
	P _s =40	978,7012	16,29	891,5406	30,95	978,2829	496,59
	P _s =100	883,4997	56,10	841,3365	68,48	951,0521	1258,60
	P _s =200	812,6075	114,22	800,9275	128,01	887,4929	2453,33
	P _s =1000	733,3307	675,91	697,5895	592,36	781,0212	12071,29
	P _s =2000	714,2675	2301,14	581,0146	1189,92	751,0521	23164,77

4.3. 101 Şehir için İnceleme

101 şehirli problemde de aynı şekilde 15 tekrar sonucu elde edilen değerlerin ortalaması şu şekildedir:

Tablo 5: Eil101 için G, Ps ve Çaprazlama Operatörlerinin Sonuçları

Nesil sayısı (G)	Popülasyon büyüklüğü (P _s)	CX	Hesaplama Zamanı (saniye)	OX	Hesaplama Zamanı (saniye)	EERX	Hesaplama Zamanı (saniye)
1	P _s =20	2.9768e+0 3	0,39	2.5605e+0 3	0,57	2.7159+03	7,87

G=500	P _s =40	2.4448e+0 3	0,55	2.4596e+0 3	0,80	2.7038e+0 3	16,33
	P _s =100	2.2639e+0 3	1,17	2.2316e+0 3	1,42	2.5494e+0 3	36,27
	P _s =200	2.0526e+0 3	2,02	1.9552e+0 3	2,40	2.4341e+0 3	71,93
	P _s =1000	2.0526e+0 3	9,37	1.4870e+0 3	10,98	2.0464e+0 3	371,32
	P _s =2000	1.8496e+0 3	22,71	1.2927e+0 3	22,70	2.0412e+0 3	720,13
	P _s =20	2.1701e+0 3	1,09	2.4724e+0 3	1,59	2.4180e+0 3	42,09
	P _s =40	2.1349e+0 3	1,86	1.8850e+0 3	2,71	2.2954e+0 3	93,38
	P _s =100	1.6844e+0 3	5,34	1.6653e+0 3	5,63	1.8255e+0 3	216,96
	P _s =200	1.5566e+0 3	10,81	1.3852e+0 3	10,52	1.6542e+0 3	365,14
	P _s =1000	1.2085e+0 3	65,91	1.0708e+0 3	52,86	1.3701e+0 3	2052,38
G=1000	P _s =2000	1.2095e+0 3	139,41	2.2676e+0 3	116,07	1.19683+0 3	3426,66
	P _s =20	2.1044e+0 3	1,89	1.7992e+0 3	3,14	2.1916e+0 3	87,29
	P _s =40	1.9742e+0 3	3,57	1.2303e+0 3	5,42	1.9687e+0 3	165,38
	P _s =100	1.5542e+0 3	10,25	1.5053e+0 3	11,34	1.4972e+0 3	418,05
	P _s =200	1.3868e+0 3	21,38	1.2709e+0 3	21,72	1.4782e+0 3	763,97
	P _s =1000	1.2387e+0 3	122,74	967,372	118,34	1.2347e+0 3	3446,91
	P _s =2000	963,4249	373,80	938,6020	277,46	1.1160e+0 3	6398,00
	P _s =20	1.4877e+0 3	9,92	1.5279e+0 3	16,51	1.7115e+0 3	417,54
	P _s =40	1.1619e+0 3	17,61	1.3268e+0 3	30,79	1.3990e+0 3	896,96
	P _s =100	1.1762e+0 3	58,45	1.1392e+0 3	60,32	1.1687e+0 3	1972,71
G=5000	P _s =200	1.0875e+0 3	127,19	1.0674e+0 3	96,45	1.0695e+0 3	3364,765
	P _s =1000	987,6611	972,76	957,9026	498,33	986,7768	17705,66
	P _s =2000	898,7513	2613,41	705,6579	1301,51	935,6677	32358,59

Tablo 5'e bakıldığında diğer iki tabloda olduğu gibi yorumlanabilmektedir. Çaprazlama operatörleri kıyaslandığında en iyi değerler genellikle OX ile elde edilmiştir daha sonra CX ve en kötüsü ise Enhanced edge recombinationdır. Ancak bazı durumlarda EERX' in daha iyi olduğu görülmektedir.

Eil101 için; **G= 50000**, **Ps=2000** ve **OX** operatörleri ile en iyi değer **705,6579** olarak elde edilmiştir. Bu problem için optimal çözüm değeri ise 629 'dur.

5.Sonuç

Uygulamada üç farklı boyuttaki problem ele alınmış ve bunlar için uygun parametreler belirlenmeye çalışılmıştır. Başlangıç popülasyonu rasgele bir şekilde oluşturulmuş, seçim yöntemi olarak turnuva seçimi tercih edilmiş, OX, CX ve EEAX çaprazlama operatörü ve yer değiştirme mutasyon operatörü kullanılmıştır. Farklı boyutlardaki problemler için en iyi sonucu veren çaprazlama operatörü, PS, G ve Pm değerleri belirlenmiştir.

Tablo 6: Eil51, Eil76 ve Eil101 için Çözüm Sonuçları

Problem	G	Ps	Pm	Çaprazlama Operatörü	Çözüm Değeri	Optimal Değer
Eil51	500	2000	0.01	OX	470.5779	426
Eil76	5000	2000	0.01	OX	581.0146	538
Eil101	5000	2000	0.01	OX	705,6579	629

Her üç şehir boyutunda da aynı çaprazlama operatörü ve PS değeri iyi çözüm elde edilmiştir ancak nesil sayısında farklılık meydana gelmiştir Eil51 için 500 iken diğerlerinde 2000 ile çözüm elde edilmiştir. Mutasyon oranı olarak da 0.01 en iyi sonucu vermiştir.

6. Sonuç ve Öneriler

Gezgin satıcı probleminin şehir sayısı arttıkça olası çözüm sayısı arttığından bunu kısa sürede kesin yöntemlerle çözmek mümkün değildir. Bu yüzden çözümü zor problemlerin çözümünde kısa sürede kabul edilebilir çözümler veren genetik algoritmalar kullanılmaktadır. Genetik algoritmaların da performansını önemli ölçüde etkileyen parametrelerin doğru bir şekilde belirlenmesi, sonucun daha da iyileşmesini sağlamaktadır.

Çalışmanın uygulama kısmında ilk olarak şehir sayısına bağlı olarak yani problemin boyutuna bağlı olarak belirlenmesi gereken popülasyon büyüklüğü ve nesil sayısı incelenmiştir. Her problem boyutu için deneyerek uygun değerler belirlenmektedir. Popülasyon büyüklüğü artırıldığında sonuçlarda önemli iyileşmeler gözlenmiştir. Ancak bu hesaplama zamanının da arttırmaktadır, bu da istenmeyen bir durumdur. Amaç kısa sürede sonuç elde etmek olduğundan popülasyon büyüklüğünü arttırmak hesaplama zamanını da arttırdığından; algoritmanın çalışma zamanı bir yerde durdurulmalıdır. Aynı zamanda nesil sayısı da iyileştirme sağlamaktadır ancak popülasyon büyüklüğünü arttırmak daha fazla bir iyileşme sağlamaktadır. Popülasyonun sonuçlar üzerinde önemli etkiye sahip olduğunu görülmektedir. Sonuçları daha da iyileştirmek için başlangıç popülasyonunu farklı sezgisellerle veya farklı yöntemlerle oluşturduktan sonra GA kullanılması bir yaklaşım olabilir.

Bir diğer dikkat edilmesi gereken husus kullanılan çaprazlama operatörüdür. Uygulamada daha önce önerilmiş olan üç operatöre ilişkin çözüm sonuçlarına bakıldığında; çaprazlama operatörlerin çözüm sonucunu önemli ölçüde etkilediği ve çözüm sonucunu iyileştirdiği görülmektedir. Kullanılan operatörlerden OX ele alınan üç problem boyutunda en iyi sonucu vermiştir. OX operatöründe; şehirlerin sırası korunmaktadır. Bu yüzden şehirlerin sırasını koruyan operatörlerin geliştirilmesi çözümü daha da iyileştirebilir.

Diğer parametre mutasyon oranıdır. 0,01 mutasyon oranı kullanılmıştır. Farklı mutasyon oranının sonuca etkisine bakıldığında mutasyon kullanılmalıdır ve oranı dikkatle belirlenmelidir.

Genel olarak uygulamada elde edilen sonuçlar beklentileri karşılamaktadır. Bu operatörlerin etkisini iddia edildiği şekilde olduğu görülmüş ve 3 farklı problem boyutu için uygun parametreler belirlenmiştir.

Kaynakça

- Ahmed, Z. H. (2010). Genetic Algorithm for the Traveling Salesman Problem Using Sequential Constructive Crossover Operator. *International Journal of Biometrics ve Bioinformatics (IJBB)*, 3(6), 96–105.
- Allaoua, H. ve Brahim, B. (2015). A Mono Crossover Genetic Algorithm for TSP. *Global Journal on Technology*, (7), 109–115.
- Bhattacharyya, M. ve Bandyopadhyay, A. K. (2008). Comparative Study of Some Solution Methods for Traveling Salesman Problem Using Genetic Algorithms. *Cybernetics and Systems*, 40(1), 1–24.
- Cevre, U., Özkan, B. ve Uğur, A. (2007). Gezgin Satıcı Probleminin Genetik Algoritmalarla Eniyilemesi ve Etkileşimli Olarak İnternet Üzerinde Görselleştirilmesi. *XI I. "Türkiye'de İnternet "Konferansı ,Ankara*.
- Çolak, S. (2010). Genetik Algoritmalar Yardımı ile Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama. *Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi*, 19(3), 423–438.
- Deep, K. ve Thakur, M. (2007). A New Mutation Operator for Real Coded Genetic Algorithms. *Applied Mathematics and Computation*, 193(1), 211–230.
- Elmas, Ç. (2016). *Yapay Zeka Uygulamaları*. Ankara: Seçkin Yayıncılık.
- Esmkhan, H. I. ve Zamanifar, K. (2012). Developing Improved Greedy Crossover to Solve Symmetric Traveling Salesman Problem. *arXiv Preprint arXiv:1209.5339*.
- Goldberg, D. E. ve Deb, K. (1991). A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. *Foundations of Genetic Algorithms*, 1, 69–93.
- Gopal, G., Kumar, R., Jawa, I. ve Kumar, N. (2015). Enhanced Order Crossover for Permutation Problems. *International Journal of Innovative Research in Science, Engineering and Technology*, 4(2), 151–157.
- Grefenstette, J. J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), 122–128.
- Javidi, M. M., Fard, R. H. ve Jampour, M. (2015). Research in Random Parameters of Genetic Algorithm and Its Application on TSP and Optimization Problems. *Walailak Journal of Science and Techonology*, 12(1), 27–34.
- Karaboğa, D. (2014). *Yapay Zeka Optimizasyon Algoritmaları*. İstanbul: Nobel Yayın Dağıtım.
- Laporte, G. (1992). The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms. *European Journal of Operational Research*, 59, 231–247.
- Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I. ve Dızdarevic, S. (1999). Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artificial Intelligence Review*, 13(2), 129–170.
- Melanie, M. (1998). *An Introduction to Genetic Algorithms*. Boston:MIT Press.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer-Vergal.
- Pham, D. T. ve Karaboga, D. (1997). Genetic Algorithms with Variable Mutation Rates : Application to Fuzzy Logic Controller Design. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 211(2), 157–167.
- Potvin, J.-Y. (1996). Genetic Algorithms for the Traveling Salesman Problem. *Annals of Operations Research*, 63(3), 337–370.

- Pulat, M. ve Deveci Kocakoç, İ. (2016). Gezin Satıcı Probleminin Genetik Algoritmalar Kullanarak Çözümünde Çaprazlama Operatörlerinin Örnek Olaylar Bazlı İncelenmesi. In *Yöneylem Araştırması Endüstri Mühendisliği 36.Ulusal Kongresi* (pp. 1–27). İzmir.
- Razali, N. M. ve Geraghty, J. (2011). Genetic Algorithm Performance with Different Selection Strategies in Solving TSP.
- Sivanandam, S. N. ve Deepa, S. N. (2008). *Introduction to Genetic Algorithms. Introduction to Genetic Algorithms*. Berlin: Springer Science & Business Media.
- Starkweather, T., Mcdaniel, S., Mathias, K. ve Darrell, W. (1991). A Comparison of Genetic Sequencing Operators. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 69–76.
- Yu, X. ve Gen, M. (2010). *Introduction to Evolutionary Algorithms*. New York: Springer Science & Business Media.