

## Gezgin Satıcı Probleminin Genetik Algoritmalar Kullanarak Çözümünde Çaprazlama Operatörlerinin Örnek Olaylar Bazlı İncelenmesi

Meryem PULAT<sup>1</sup>, İpek DEVECİ KOCAKOÇ<sup>2</sup>

### Özet

*Gezgin satıcı problemi, optimizasyon alanında araştırmacı ve akademisyenler tarafından üzerinde uzun yıllardır yoğun olarak çalışılan çözümü zor (NP-hard) bir problemdir. Genetik algoritmalar GSP (gezgin satıcı problemi) gibi çeşitli NP-hard problemleri çözmek için kullanılan en iyi yöntemlerden biridir. GSP problemi için çok sayıda çaprazlama operatörü önerilmiştir ve her çalışmada yenileri önerilmeye devam etmektedir. Bu çalışmanın amacı GSP çözümünü araştıran çalışmalarda kullanılan TSPLIB örnek olaylarının ve incelenen çaprazlama operatörlerinin detaylı bir envanterini çıkarmak ve bu konuda çalışmak isteyen araştırmacılara yön göstermektir. Literatürdeki çalışmalar geniş bir kapsamda (anahtar kelime ve yıl bazında) incelenerek ortak kullanılan örnek olayların ve bulunan sonuçların analizi yapılarak tablolaştırılmıştır.*

**Anahtar kelimeler:** Gezgin Satıcı Problemi, Genetik Algoritmalar, Çaprazlama operatörü, TSPLIB.

**Jel Kodu:** C61, C44.

## Investigation Of Crossover Operators Using Genetic Algorithms In The Solution Of The Traveling Salesman Problem Based On Case Studies

### 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. Genetic algorithms are one of the best methods used to solve various NP-hard problems, such as TSP (traveling salesman problem). Many crossover operators have been proposed for the tsp problem and new ones have been proposed in each study. Our aim in this study is to guide the researchers who want to work on this subject by taking out a detailed inventory of the TSPLIB case studies and the crossover operators that are used in the studies that are investigating the tsp solution. Studies in the literature have been examined in a wide range of contexts (based on keyword and year), and the sample events and the results have been analyzed.*

**Keywords:** Traveling Salesman Problem, Genetic Algorithms, Crossover operator, TSPLIB.

**Jel Codes:** C61, C44.

## 1 GİRİŞ

Gezgin Satıcı Problemi (GSP, Traveling Salesman Problem-TSP) bir kombinasyonel (combinatorial) optimizasyon problemidir. GSP'ni çözmek için farklı yöntemler kullanılmıştır. Bu çalışmada bu yöntemlerden biri olan Genetik Algoritmalar (GA) üzerinde durulmuştur.

GA genetik ve evrim ilkelerine dayanan olasılıksal, sağlam (robust) ve sezgisel bir arama algoritmasıdır (Javidi vd., 2015: 27). GA Holland tarafından ortaya atıldıktan sonra GA'nın çözüm performansını önemli ölçüde etkileyen operatörlerinden biri olan

çaprazlama operatörleri araştırmacılar tarafından ortaya atılmıştır.

Bu çalışmada çaprazlama operatörlerini ele alan çalışmalar üzerinde durulmuştur. GSP'nin GA çözümü için geliştirilmiş olan çaprazlama operatörlerinin 57 tanesi TSPLIB örnek olaylarla birlikte tablolaştırılmıştır. Amacımız çalışmalara baktığımızda operatörlerin TSPLIB'deki hangi örnek olaylarla daha sık kullanıldığını ve operatörün hangi yaklaşımla o veri seti için daha iyi sonuç verdiğini elde ederek araştırmacılara yol göstermektir.

<sup>1</sup> Araş. Gör., Dokuz Eylül Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Ekonometri Bölümü, meryem.pulat@deu.edu.tr  
**ORCID:** 0000-0003-0642-5619

<sup>2</sup> Prof. Dr., Dokuz Eylül Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Ekonometri Bölümü, ipek.deveci@deu.edu.tr  
**ORCID:** 0000-0001-9155-8269

## 2 GEZGİN SATICI PROBLEMİ

GSP ilk olarak 1930 'lu yılların başlarında Karl Menger tarafından tanımlandığından bu yana, araştırmacı ve akademisyenler tarafından üzerinde en çok çalışılan kombinasyonel optimizasyon problemlerinden birisi olmuştur (Çolak,2010: 424). Tanımlanması kolay olmasına rağmen çözümü zordur ve NP-zor (Nonpolynomially-hard) problemler arasında yer almaktadır (Çolak, 2010: 424). Polinomial (P) denklemler deterministik bir polinom zaman algoritması tarafından çözülebilen tüm karar problemlerinin kümesini gösterir (Jong ve Spears, 1989: 124). Çözümlemesi ve incelenmesi kolay olan denklemlerdir ve kısa sürede problemi çözen yöntemler mevcuttur. Eğer bir denklem ya da sistem polinomial değilse; NP çözülmesi zor sistemlerdir. NP problemleri çözmek için kısa sürede kesin çözüm bulan yöntemler mevcut değildir (Elmas, 2016: 383).

GSP'nin birçok araştırmacının dikkatini çekmesi ve aktif bir araştırma alanı olmaya

$$\text{Minimize: } z = \sum_{i=1}^n \sum_{j=1}^n x(i,j)d(i,j) \quad (1)$$

Kısıtlar:

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

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

$$\sum_{i,j \in S} x(i,j) \leq |S| - 1, \forall S \in (1,2, \dots, n) \quad (4)$$

$$x(i,j) = \begin{cases} 1, & i \text{ noktasından } j \text{ noktasına gidiliyor ise} \\ 0, & i \text{ noktasından } j \text{ noktasına gidilmiyor ise} \end{cases} \quad (5)$$

(1) numaralı eşitlikte GSP'nin amaç fonksiyonu verilmiştir. Amaç fonksiyonunda  $d(i,j)$   $i$  ve  $j$  noktaları arasındaki mesafeyi,  $x(i,j)$  karar değişkeni ise  $i$  noktasından  $j$  noktasına gidilip gidilmediğini göstermektedir. (2) numaralı eşitlik her noktadan sadece bir kez çıkılacağını, (3) numaralı eşitlik ise her noktaya yalnızca bir kez gidileceğini ifade etmektedir. (4) numaralı eşitlikte oluşabilecek olan alt turlardan kurtulmaya yönelik olan alt tur eleme kısıtıdır. (5) numaralı eşitlikte ise  $x(i,j)$ 'nin 1 olması,  $i$  noktasından  $j$  noktasına gidildiğini; 0 olması ise gidilmediğini göstermektedir (Çolak, 2010: 425).

devam etmesinin başlıca üç nedeni vardır (Gopal vd., 2015: 151):

1) Çok sayıda gerçek dünya problemi GSP ile modellenenmektedir.

2) GSP, NP-zor problem sınıfında yer almaktadır.

3) NP-zor problemlerde problem büyük boyutlu olduğunda problemi çözmek için etkili bir yol bulunamamıştır.

GSP'inde şehirlerin sonlu kümesi ve şehir  $i$  den şehir  $j$  ye seyahat maliyeti verilmektedir. Burada amaç, satıcının her şehri tam olarak bir kez ziyaret etme ve sonra başlangıç şehrine dönmesi koşuluyla minimum maliyete sahip  $n$  şehrin permütasyonunu bulmaktır (Ray vd.,2004:1). GSP temel olarak simetrik ve asimetric olarak iki gruba ayrılmaktadır.  $d(i,j) = d(j,i)$  ise GSP simetrik, aksi takdirde asimetrictir (Ahmed, 2010: 96).

GSP matematiksel olarak şöyle ifade edilebilir:

GSP'nin tanımlanması kolay olmasına rağmen optimal çözümünü elde etmek çok zordur ve literatürde NP-zor (Non polynomial-hard) problemler arasında yer alır (Potvin, 1996:339). Bu problemin temel zorluğu şehir sayısının artmasına bağlı olarak meydana gelen tur sayısının büyüklüğüdür. Şehir sayısı  $n$  olan simetrik gezgin satıcı problemi (SGSP) için  $(n-1)!/2$  olası çözüm, şehir sayısı  $n$  olan asimetric gezgin satıcı problemi (AGSP) için  $(n-1)!$  olası çözüm vardır (Ahmed, 2010: 96). Bu problemleri çözmek için önerilen algoritmaları "kesin (exact)" algoritmalar ve "sezgisel (heuristic)" algoritmalar olarak iki

ana grupta toplamak mümkündür (Potvin, 1996: 339).

### 2.1. Kesin Algoritmalar

Bu algoritmalar genellikle GSP'nin tam sayılı doğrusal programlama formülünden türetilen yaklaşımlardır. Problemlerin karmaşıklığından dolayı kesin yöntemlerin etkinliği küçük boyutlu problemlerle sınırlıdır. Ancak kesin yöntemler optimal sonuçları vermektedir (Deep ve Mebrahtu, 2011: 2).

### 2.2. Sezgisel Algoritmalar

Kesin yöntemler kullanarak çözülmesi zor olan sorunları çözmek için kullanılır. Sezgisel yöntemler kesin yöntemler ile karşılaştırıldığında kısa sürede çözüm bulmasına rağmen, optimal bir çözümü garanti edemez (Deep ve Mebrahtu, 2011: 2). Pratikte büyük boyutlu GSP'ni çözmek için sezgisel algoritmalar kesin algoritmalara göre daha çok tercih edilmektedir.

Genel olarak GSP sezgiselleri, tur oluşturan sezgiseller (tour construction procedures), tur geliştiren sezgiseller (tour improvement procedures) ve tur oluşturma ve tur geliştirmeye dayalı olan melez yöntemler (composit procedures) olarak sınıflandırılmıştır (Potvin, 1996: 339).

**Tur oluşturan sezgiseller:** Tur oluşturan algoritmalarda ortak özellik, bir çözüm bulunduğuna durur ve onu geliştirmek için uğraşmaz (Nilsson, 2003: 1). Örnek olarak En yakın komşu (Nearest Neighbour) ve Greedy verilebilir.

**Turu geliştiren sezgiseller:** Turu geliştirmeyi amaçlar. 2-opt ,3-opt ve Lin-Kernighan gibi yerel optimizasyon algoritmalarının yanında; genetik algoritmalar, tavlama benzetimi ,karınca koloni algoritması gibi sezgisel yöntemler örnek olarak verilebilir (Gerşil ve Alkaya, 2005: 406).

**Melez yöntemler:** Hem tur oluşturma hem de tur iyileştirme sezgisellerinin bir arada kullanıldığı algoritmalar. GSP' nin çözümü için en güçlü sezgiseller arasındadır (Potvin, 1996: 342).

GA özellikle geleneksel optimizasyon yöntemlerinin daha az etkin olduğu zor optimizasyon problemlerini çözmek için uygundur (Yang vd., 2010: 1). GA karar değişkenleri yerine problemi kodlama ile ilgilenir. Hiçbir ilgi alanı bilgisi gerektirmez ve aramayı yönlendirmek için stokastik geçiş kurallarını kullanır. En geleneksel yöntemler tek yönlü arama yaparken GA aday çözümlerin bir dizisini kullanarak çok yönlü arama yapar. GA bu nedenle araştırmacılar tarafından en sık kullanılan ve aynı zaman da GSP'nin çözümü için en çok tercih edilen sezgisel tekniklerden biridir (Pongcharoen vd., 2007: 215).

## 3 GENETİK ALGORİTMALAR

GA evrime dayalı algoritmaların bir türüdür. Bu prensibe dayanan tekniklerin hepsini temsilen evrimsel hesaplama terimi yaygın olarak kullanılmaya başlanmıştır. Bu sınıfa örnek olarak genetik algoritmalar, evrimsel programlama ve diferansiyel geliş algoritmaları sayılabilir. Herhangi bir evrimsel algoritma aşağıdaki beş elemana ihtiyaç duyar:

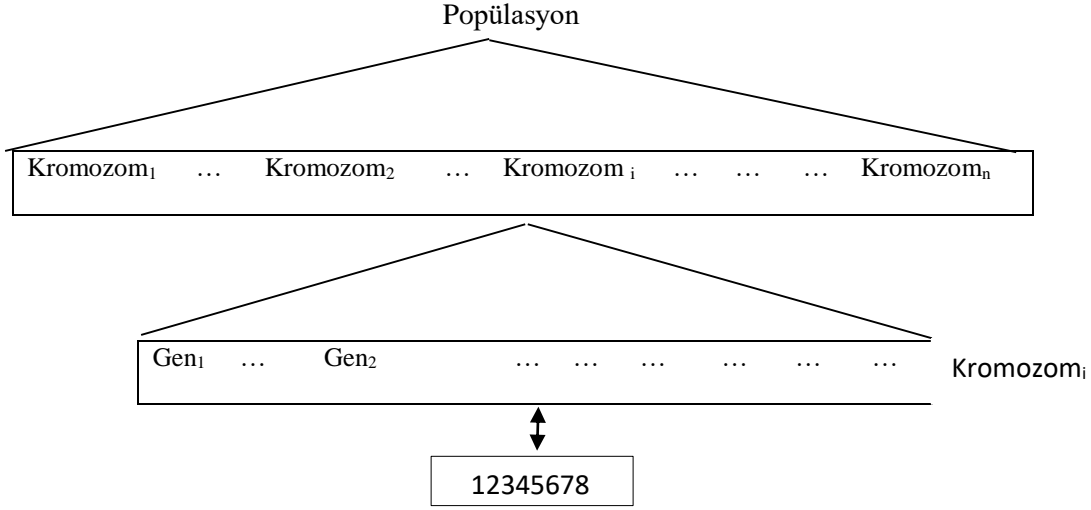
- Problem için çözümlerin genetik temsili (representation)
- Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem
- Çözümleri uygunluk açısından değerlendirmeye tabi tutacak değerlendirme fonksiyonu
- Genetik kompozisyonu değiştirecek operatörler
- Kontrol parametre değerleri (popülasyon büyüklüğü, operatörlerin uygulanma olasılığı).

Goldberg'in tanımına göre, GA rastlantısal arama tekniklerini kullanarak çözüm bulmaya çalışan parametre kodlama esasına dayanan sezgisel bir arama tekniğidir (Elmas, 2016). 1970'li yılların başlarında John Holland tarafından ortaya atılan GA, Darwin'in evrim teorisinden ilham alır. İyi nesiller yaşamlarını korurken kötü nesillerin yok olması ilkesine dayanır. GA farklı uygulama alanlarındaki problemleri optimize edebilmek için kullanılan

yöntemlerden biridir. En yaygın örneği de GSP'dir (Javidi vd., 2015: 27).

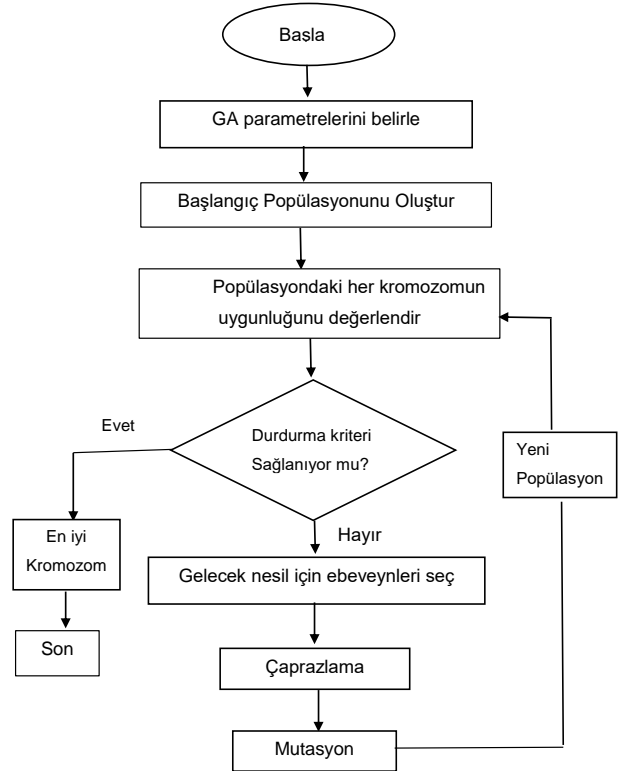
Şekil-2'de görüldüğü gibi GA popülasyona kodlanmış çeşitli problem çözümleriyle başlar;

uygunluk fonksiyonu her bireyin uygunluğunu değerlendirmek için uygulanır sonra yeni nesil seçim, çaprazlama ve mutasyon süreci yoluyla oluşturulur (Kumar vd., 2012: 98).



Şekil 1: Popülasyonun Yapısı

Bu algoritmada problemin arama uzayı bireylerin toplamı olarak temsil edilir. Bu bireyler genellikle kromozom olarak adlandırılan karakter dizileri (ya da matrisler) ile temsil edilir. GA'nın kullanım amacı en iyi genetik materyal ile arama uzayından bireyi bulmaktır. Bireyin kalitesi değerlendirme fonksiyonu ile ölçülür. İncelenecek arama alanı parçasına popülasyon denir. Popülasyon yapısı Şekil-1 de görülmektedir. Popülasyon kromozomlardan, kromozomlar ise genlerin birleşiminden oluşmaktadır. İlk olarak başlangıç popülasyonu seçilir ve bu popülasyonun kalitesi belirlenir. Sonra her iterasyonda ebeveynler popülasyondan seçilir. Bu ebeveynler popülasyona eklenecek çocukları üretir. Bundan sonra, başlangıç popülasyon büyüklüğünü azaltmak için seçim kriterlerine göre popülasyondan bazı bireyler çıkarılır. Algoritmanın her iterasyonu nesil (generation) olarak adlandırılır (Larranga vd., 1999: 131).



Şekil 2: Genetik Algoritma Akış Diyagramı

### 3.1. Gösterim (Representation)

Bireylerin nasıl kodlanacağı ilgilenilen probleme göre değişir. GA kullanarak GSP'ni çözebilmek için çok sayıda farklı gösterim kullanılmıştır. Bunlar; ikili (binary), yol (path), komşuluk (adjacency), sıralı (ordinal) ve matris (matrix) gösterimidir (Laaranga vd., 1999: 134).

GSP bir sıralama problemi olarak ele alındığından yol (path) gösterimi kullanılır. Bu gösterimde ziyaret edilmesi gereken n şehir, n elemanlı listeye göre sıralanmaktadır. Şehir listenin j. elemanı ise; ilgili şehir ziyaret edilmesi gereken j. şehirdir. 8 tane şehrimizin olduğunu varsayarsak şu şekilde gösterilir:

(32417586)

Bazı bilinen ve yaygın olarak kullanılan çaprazlama ve mutasyon operatörü olmasına rağmen her tür kodlama ve kromozom yapısı için uygun çaprazlama ve mutasyon operatörü bulunamaz. Klasik GA operatörleri kullanımı uygun değildir çünkü her gen ziyaret edilecek şehirlerin birini gösterir ve GSP'nde her şehir tam olarak bir kez ziyaret edilmelidir. Bir başka ifadeyle gen tekrarına izin verilmemelidir. Gen tekrarı olursa geçerli bir tur olmaz. Bu yüzden bu gösterime göre çok sayıda çaprazlama ve mutasyon operatörü geliştirilmiştir (Larranga vd., 1999: 137).

### 3.2. Başlangıç Popülasyonunun Oluşturulması

GA'yı diğer sezgisel arama yöntemlerinden ayıran en önemli özellik, (Elmas, 2016)'da belirtildiği gibi çözümü noktadan noktaya değil, noktaların oluşturduğu yığın içinde aramasıdır. Arama alanı popülasyondan rasgele bir şekilde oluşturulan tüm mümkün çözümleri içerir. Bu amaçla, GA gen havuzu olarak adlandırılan dizilerden oluşan bir başlangıç popülasyonunun rasgele üretilmesiyle başlar. Başlangıç popülasyonu oluşturulduktan sonra daha iyi çözümlere karşılı gelen daha iyi popülasyonlar oluşturmak için sırasıyla seçim, çaprazlama, mutasyon operatörleri uygulanır.

Genel olarak, başlangıç yığını rasgele oluşturulur. Ama GA'nın performansını

iyileştirmek için probleme özgü çeşitli sezgiseller geliştirilebilir. Başlangıç popülasyonunun büyüklüğü de her sorun için problemin karmaşıklığına bağlıdır. Popülasyon büyüklüğü genel performansı ve GA verimliliğini etkiler (Javidi vd., 2015: 31). Bu yüzden dikkatli seçilmeli ve farklı popülasyon büyüklükleri test edilmelidir. Popülasyon büyüklüğü GA'nın performansını iki yönden etkilemektedir (Karaboğa, 2014):

- I. Popülasyon büyüklüğünün aşırı küçük olması araştırma alanının yetersiz örneklenmesine sebep olacağından kontrollü ırsamayı sağlamak zorlaşmaktadır ve arama belirli bir alt optimal noktaya doğru sürüklenmektedir.
- II. Popülasyon büyüklüğü aşırı büyük olması nesil (generation) gelişimi için oldukça uzun bir süreye ihtiyaç duymaktadır.

### 3.3. Uygunluk Fonksiyonu:

Başlangıç popülasyonu oluşturulduktan sonra yapılacak ilk işlem bu popülasyonda yer alan kromozomların uygunluk değerinin hesaplanmasıdır. Bir çözümün uygunluk değeri ne kadar yüksekse yaşam ve çoğalma şansı o kadar fazladır ve bir sonraki nesil de temsil edilme oranı da o kadar yüksektir.

GSP bir minimizasyon problemidir. Bu nedenle amaç fonksiyonuna ait uygunluk değeri  $1/f(x)$  şeklinde ifade edilmelidir. Burada  $f(x)$  fonksiyonu kromozom ile temsil edilen turun maliyetinden hesaplanır (Ahmed, 2010: 98).

### 3.4. Seçim (Selection)

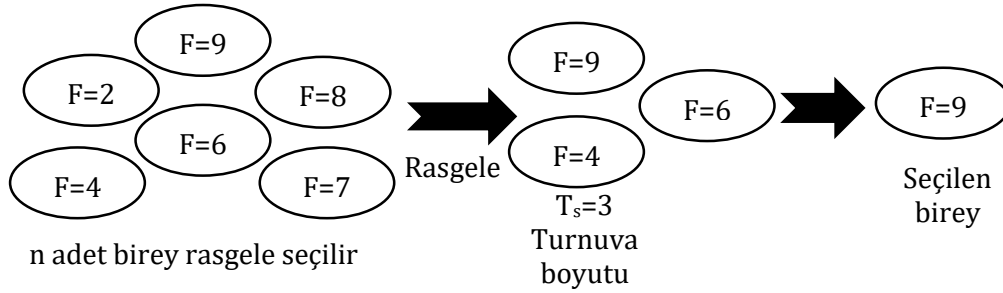
Seçim mekanizması hangi bireyin eşleştirme (mating) için seçileceğini ve her bireyden ne kadar çocuk üretileceğini belirler. Seçim stratejisi "daha iyi birey ebeveyn olmada daha yüksek şansa sahiptir" temel prensibinden oluşur. Bu nedenle, seçim kötü (zayıf) çözümleri atarak popülasyon içinden daha iyi çözümlere ulaşmada önemli bir işlev görür.

İyi arama tekniği global optimumu bulmak için arama (kötü çözümler yeni nesle gitmek için şansa sahip olmalıdır) ve kullanma (iyi nesiller yeni nesile kötü nesilden daha çok gitmeli)

arasında iyi bir denge bulmak önemlidir. GA kullanılan farklı seçim stratejileri algoritmanın performansını önemli ölçüde etkiler.

Bu çalışmada 3 seçim yöntemi açıklanmıştır: Turnuva seçimi (Tournament Selection), Orantılı Rulet Tekerleği (Proportional Roulette Wheel Selection), Sıraya Dayalı Rulet Tekerleği (Rank-based Roulette Wheel Selection).

### 3.4.1. Turnuva Seçimi (Tournament Selection)



Şekil 3: Turnuva Yöntemi ile Seçim (Razali ve Geraghty, 2011: 3)

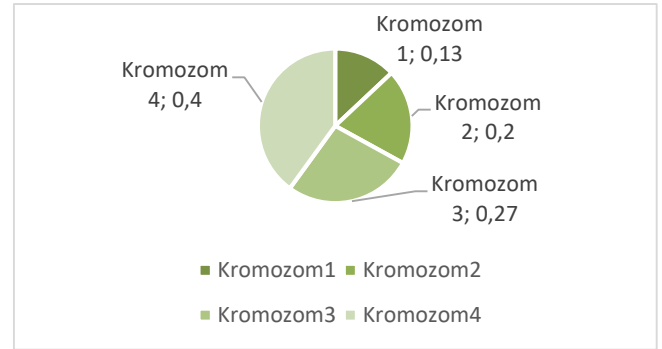
### 3.4.2. Orantılı Rulet Tekerleği Seçimi (Proportional Roulette Wheel Selection)

Bu seçimde çember n adet parçaya bölünür. Her aralık bir diziyi temsil eder. Her dizinin uygunluk değeri toplam uygunluk değerine bölünür. Popülasyondaki n adet bireyin amaç fonksiyonlarına ait uygunluk değerleri sırasıyla ( $f_1, f_2, \dots, f_n$ ) ise i. bireyin seçilme olasılığı ( $P_i$ ):  $P_i = f_i / \sum_{i=1}^n f_i$  formülü ile bulunur.

Böylece yığın içindeki her dizinin çözüm kümesi içinde [0-1] arasındaki yeri bulunur. Bireyler yüzdelik olarak çemberde yer temsil eder (Razali ve Geraghty, 2011: 3).

	Uygunluk Değeri ( $f_j$ )	Olasılık ( $P_j$ )
Kromozom1	0,2	$0,2/1,5 = 0,13$
Kromozom2	0,3	$0,3/1,5 = 0,2$
Kromozom3	0,4	$0,4/1,5 = 0,27$
Kromozom4	0,6	$0,6/1,5 = 0,4$
	$\sum_{j=1}^n f_j = 1,5$	

Turnuva seçimi en popüler seçim yöntemidir. Turnuva seçiminde n birey popülasyondan rasgele seçilir ve seçilen bireyler birbiriyle yarışır. En yüksek uygunluğa sahip birey kazanır ve yeni nesil popülasyonuna dahil edilmektedir. Her turnuvada yarışacak birey sayısı turnuva boyutu olarak adlandırılır ve genellikle üçtür.



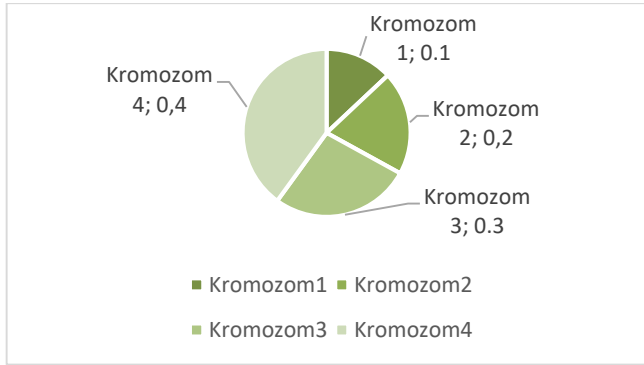
Şekil 4: Orantılı Rulet Tekerleği

En uygun bireyler rulet tekerleğinde en büyük parçaya sahipken, en az uygun olanlar rulet tekerleğinde daha küçük parçaya sahiptir. En büyük uygunluğa sahip olanların seçilmiş olma ihtimali daha çoktur. Tekerlek döndürülür ve durduğunda bu dilimlerden birine denk gelir muhtemelen en geniş olanlarında biridir. Ama tüm kısımların genişliğiyle orantılı bir olasılık ile şansa sahiptir. Avantajı, popülasyondaki bireylerin hiçbirini atmaz, hepsine seçilme şansı verir. Böylece popülasyonun çeşitliliği korunur.

### 3.4.3. Sıraya Dayalı Rulet Tekerleği Seçimi (Rank-based Roulette Wheel Selection)

İlk olarak kendi uygunluk değerine göre popülasyondaki bireyler sırlanır ve sonra uygunluk değerinden ziyade sırasına göre seçim olasılıkları hesaplanır (Razali ve Geraghty, 2011: 4)

	Uygunluk Değeri ( $f_j$ )	Sıra	Olasılık ( $P_i$ )
Kromozom1	0,2	1	1/10 = 0,1
Kromozom2	0,3	2	2/10 = 0,2
Kromozom3	0,4	3	3/10 = 0,3
Kromozom4	0,6	4	4/10 = 0,4
		$\Sigma = 10$	



Şekil 5: Sıraya Dayalı Rulet Tekerleği

### 3.5. Çaprazlama

Çaprazlama (Crossover) iki ebeveynin eşleşerek çocuk üretme sürecidir. Seçim işleminden sonra çaprazlama ile popülasyon daha iyi bireyler ile zenginleştirilir ve farklı çözümler arasında bilgi alışverişi sağlar (Michalewicz, 1992: 17).

GA'nın en önemli operatörü çaprazlamadır. Çünkü GA'nın performansı bu operatörlere bağlıdır (Esmkhan ve Zamanifar, 2012: 1). GA Holland tarafından tanıtıldıktan sonra birçok GA çaprazlama operatörü araştırmacılar tarafından ortaya atılmıştır. Çaprazlama tekniğinde temel parametre çaprazlama olasılığı ( $P_c$ ) dır. Bu olasılık ne sıklıkla

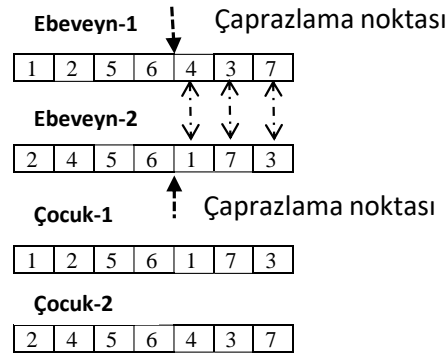
çaprazlamanın yapılacağını tanımlayan bir parametredir (Javidi vd., 2015:31). Bu değer çözümün kalitesi üzerinde önemli etkiye sahiptir. Bu yüzden bu değer seçilirken dikkatli olunmalıdır.

GA için kullanılan klasik çaprazlama operatörleri şehir tekrarına neden olduklarından GSP için uygun değildir. Örneğin, GA'da en çok kullanılan tek noktalı çaprazlamanın uygulandığını varsayalım,

Ebeveyn 1: (1 2 5 6 4 3 7)

Ebeveyn 2: (2 4 5 6 1 7 3)

Çaprazlama noktası 1 ile L-1 (burada L: kromozom uzunluğunu gösterir.) arasında rasgele seçilir. Çaprazlama noktasından sonraki bireyler karşılıklı yer değiştirilerek iki yeni birey elde edilir (Elmas, 2016). Çaprazlama noktası olarak üçüncü noktayı seçtiğimizi varsayalım. Üçüncü noktadan sonraki bireyleri ebeveyn1 ve ebeveyn 2 de karşılıklı olarak değiştirilmektedir.



Bu değişim sonucunda şehir tekrarı olduğu için geçersiz tur olur. Bu problemler için özel operatörler geliştirilmiştir ve halen geliştirmeye devam etmektedir. Çalışma kapsamında literatürde geliştirilmiş çaprazlama operatörlerinden 57 tanesi incelenmiştir. Ancak, burada yazının kapsamını dar tutmak amacıyla literatürde GSP için en çok kullanılan 8 çaprazlama operatörü aşağıda ayrıntılı olarak açıklanmıştır

#### 3.5.1. Kısmi Eşleştirmeli Çaprazlama

Kısmi eşleştirmeli çaprazlama (Partially-Mapped Crossover-PMX) Goldberg ve Lingle (1985) tarafından geliştirilmiştir:

Ebeveyn 1: (1 2 3 4 5 6 7 8)

Ebeveyn 2: (3 7 5 1 6 8 2 4)

İlk olarak, ebeveyn turlarından dizi boyunca iki kesim noktası rasgele seçilir. Burada, ilk kesim noktasının 3. ve 4. bit arasında ve ikincisinin 6. ve 7. bit arasında seçildiğini varsayalım bu durumda,

Ebeveyn 1: (1 2 3 I 4 5 6 I 7 8)

Ebeveyn 2: (3 7 5 I 1 6 8 I 2 4)

Kesim noktaları arasındaki alt diziler (E1: 456, E2: 168) eşleştirme bölümleri (mapping sections) denir. Örneğimizde eşleştirmeler:  $4 \leftrightarrow 1$ ,  $5 \leftrightarrow 6$  ve  $6 \leftrightarrow 8$  dir.

İlk ebeveynin eşleştirme 2.çocuğa kopyalanır ve ikinci ebeveynin eşleştirme bölümü ilk çocuğa kopyalanır.

Çocuk 1: (- - - I 1 6 8 I - -)

Çocuk 2: (- - - I 4 5 6 I - -)

Daha sonra çocuk i (i=1,2) i. inci ebeveynin elemanları kopyalanarak oluşturulur. Bir şehir çocukta zaten varsa eşleştirmeye göre yerleştirilir. Örneğin; birinci çocuğun ilk elemanı ebeveyn 1'e baktığımızda 1 olmalı ama zaten 1 var çocukta var. Bu yüzden eşleştirmelere bakarız.  $1 \leftrightarrow 4$  eşleştirmesinden dolayı 4 olur. İkinci elemanı 2, 2 yok zaten çocuğa eklenir bu şekilde çocuk oluşturulur. Bu operatörde her iki ebeveynde bazı elemanların mutlak pozisyonları korunur (Larranga vd., 1999: 138-139).

Çocuk 1: (4 2 3 I 1 6 8 I 7 5)

Çocuk 2: (3 7 8 I 4 5 6 I 2 1)

### 3.5.2. Dairesel Çaprazlama

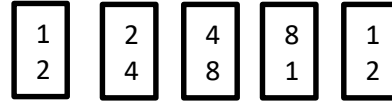
Oliver ve arkadaşları (1987) tarafından önerilmiştir.

Ebeveyn 1: (1 2 3 4 5 6 7 8)

Ebeveyn 2: (2 4 6 8 7 5 3 1)

**Çocuk-1** alt kümelerine odaklanır. Şehirler için alt küme oluştururken ebeveyn 1 de birinci şehirden başlarsak bunun ebeveyn 2 de aynı pozisyonda şehir 2 mevcut; ebeveyn 1 de şehir

2'nin ebeveyn 2 de aynı pozisyonda şehir 4 mevcut; ebeveyn 1 de şehir 4'ün ebeveyn 2 de aynı pozisyonunda şehir 8 mevcut bu şekilde alt küme oluşturulmaktadır. Alt küme oluşturmaya tekrar başa döndüğünde son verilir. Bu örnek için 1-2 alt kümesi geldiğinden durulmalıdır başa dönmüştür. Her iki ailenin pozisyonunda aynı alt kümede yer alıyorsa, bu şehirler ilk aileden çocuğa aynı pozisyonda kopyalanır, kalan pozisyonlar ikinci ebeveyndeki şehirler ile doldurulur (Potvin, 1996: 352).



Şekil 6. Dairesel Çaprazlama için Alt Kümeler

E1: (1 2 3 4 5 6 7 8) Çocuk 1: (1 2 6 4 7 5 3 8)

E2: (2 4 6 8 7 5 3 1) Çocuk 2: (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 korunur.

### 3.5.3. Sıralı Çaprazlama

Sıralı çaprazlama (Order Crossover-OX1) Davis (1985) tarafından önerilmiştir. Şehirlerin konumu değil sırası önemlidir. Ebeveynler alt turu seçer ve diğer ailedeki şehirlerin sırasını koruyarak oluşturur. İlk olarak kesim noktaları rasgele seçilir. Burada, ilk kesim noktasının 2. ve 3. bit arasında ve ikincisinin 5. ve 6. bit arasında seçildiğini varsayalım bu durumda,

Ebeveyn 1: (1 2 I 3 4 5 I 6 7 8)

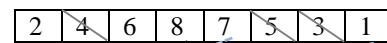
Ebeveyn 2: (2 4 I 6 8 7 I 5 3 1)

Kesim noktaları arasındaki kısım çocuğa kopyalanır.

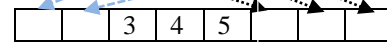
Çocuk 1: (- - I 3 4 5 I - - -)

Çocuk 2: (- - I 6 8 7 I - - -)

Ebeveyn-2



Çocuk 1



İkinci kesim noktası



Sonra ikinci kesim noktasından başlayarak zaten mevcut olan şehirleri atlayarak diğer ebeveynde görüldükleri sırada çocuğa kopyalanır dizin sonuna ulaşıldığında ilk pozisyondan devam eder (Larranga vd., 1999: 353-354).

Çocuk 1: (7 1 I 3 4 5 I 2 6 8)

Çocuk 2: (4 5 I 6 8 7 I 1 2 3)

### 3.5.4. Sıraya Dayalı Çaprazlama

Sıraya dayalı çaprazlama (Order Based Crossover-OBX2) Syswerda (1991) tarafından önerilmiştir. Ebeveynde birkaç rasgele pozisyon seçilir ve bu ebeveynde seçilen pozisyondaki şehirlerin sırası diğer ebeveynde aynı sırada verilir. İkinci ebeveyn de 2., 3., 6. pozisyonlardaki şehirler (4 6 5) birinci ebeveynde 4., 5. ve 6. Pozisyonlardadır, bu pozisyonlardaki şehirler çocuğa ilk ebeveyndeki sıra ile kopyalanır.

Ebeveyn 1: (1 2 3 4 5 6 7 8)

Ebeveyn 2: (2 4 6 8 7 5 3 1)

Çocuk 1: (1 2 3 4 6 5 7 8)

Birinci ebeveynden 2., 3. ve 6. pozisyonları seçildiğini varsayalım;

Ebeveyn 1: (1 2 3 4 5 6 7 8)

Ebeveyn 2: (2 4 6 8 7 5 3 1)

Bu pozisyondaki şehirler ikinci ebeveynde 1., 3. ve 7. pozisyonlardadır. Bu durumda çocuk;

Çocuk: (2 4 3 8 7 5 6 1)

### 3.5.5. Pozisyona Dayalı Çaprazlama

Syswerda (1991) tarafından önerilmiştir. Pozisyonların rasgele seçimi ile başlar; 2., 3. ve 6. pozisyonların seçildiğini varsayalım;

Ebeveyn 1: (1 2 3 4 5 6 7 8)

Ebeveyn 2: (2 4 6 8 7 5 3 1)

(2↔4, 3↔6, 6↔5)

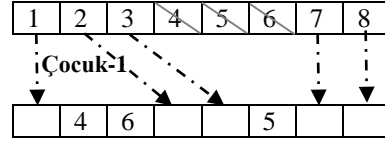
Diğer ebeveynde ikinci ebeveynde pozisyonlara denk gelen şehirler çocuğa kopyalanır.

Çocuk 1: (- 4 6 - - 5 - -)

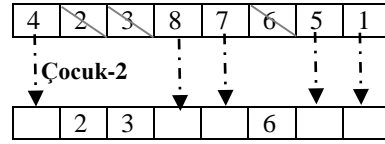
Çocuk 2: (- 2 3 - - 6 - -)

Kalanlar ebeveyndeki sıraya göre kopyalanır.

Ebeveyn-1



Ebeveyn-2



Çocuk 1: (1 4 6 2 3 5 7 8)

Çocuk 2: (4 2 3 8 7 6 5 1)

### 3.5.6. Kenar Birleştirme Çaprazlama

Kenar birleştirme çaprazlama (Genetic Edge Recombination Crossover-ERX) Whitley ve arkadaşları (1989) tarafından önerilmiştir. Turun kenarları kalıtsal bilginin taşıyıcıları olarak görülmektedir. Bu operatörde kenarların değerleri önemlidir. Çocuğa maksimum bilgi miktarını geçirmek için ebeveynlerin kenarları koruması amaçlanmıştır. Bu operatörde çaprazlama işlemini gerçekleştirmek için kenar haritası (edge map) kullanılmaktadır. Turların aşağıdaki gibi olduğunu varsayarsak;

Ebeveyn 1: (1 2 3 4 5 6)

Ebeveyn 2: (2 4 3 1 5 6)

Bu turlar için kenar haritası şu şekildedir:

**Tablo 1:** Turlar için Kenar Haritası

Şehir	Bağlantılı Şehirler
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

Operatör şu şekilde çalışır (Larranga vd., 1999: 357-358):

## M. PULAT - İ. DEVECİ KOCAKOÇ

Adım1: Başlangıç şehrini iki ebeveyn turunun birinden rasgele ya da 4.adımda belirtilen kritere göre seç. Bunu mevcut şehir (current city) olarak belirle.

Adım2: Kenar haritasından mevcut şehri kaldır.

Adım3: Mevcut şehir kenar listesine girmişse adım 4'e git, aksi takdirde adım 5'e git.

Adım4: Mevcut şehrin kenar listesindeki şehirlerin kendi kenar listesinde daha az bağlantıya sahip şehir mevcut şehir olur. Adım 2'ye git.

Adım5: Ziyaret edilmemiş şehirler kalmadıysa, dur. Aksi takdirde, ziyaret edilmemiş bir şehri rasgele seç ve adım 2'ye git.

Yukarıdaki turlar için bu operatörü örneklerirsek:

1. Çocuk turu ebeveynlerin başlangıç şehrinden biri ile başlatılır. Başlangıç şehirleri 1 ve 2 her ikisi de 4 kenara sahiptir. Rasgele bir şekilde şehir 2 seçildi.

2. Şehir 2 için kenar listesi sonraki şehir için adayları gösterir. 1, 3, 4, 6 aday şehirlerdir. 3, 4 ve 6 şehirleri iki kenara sahiptir, şehir 1 ise üç kenara sahip olduğundan kabul edilemez. Şehir 3 ün rasgele seçildiğini varsayalım.

3. Şehir 3 için aday şehirler 1 ve 4 tür. 4 daha az kenara sahip olduğundan seçilir.

4. Şehir 4 sadece şehir 5 kenarına sahiptir, şehir 5 seçilir.

5. Şehir 5 için aday şehirler 1 ve 6 olur. Her ikisi de bir kenara sahiptir, bu durumda rasgele bir şekilde şehir 1 seçilir.

6. Şehir 1 için kenar listesinde sadece şehir 6 kalmıştır. Tur sonucu:

(2 3 4 5 1 6)

### 3.5.7. Maksimum Korumacı Çaprazlama

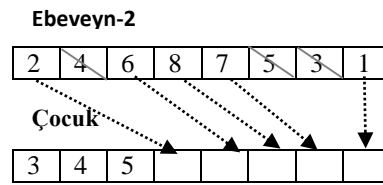
Mühlenbein ve arkadaşları (1988) tarafından önerilmiştir. PMX operatörü ile benzer şekilde çalışır. İlk olarak ilk aileden rasgele alt dizi seçilir. Alt dizi uzunluğu 10'a eşit ya da daha büyük uzunluğa sahip olanlarda 2 ile bölümüne eşit ya da daha küçük olarak belirlenir. Alt dizi

uzunluğunda bu kısıtlamanın sebebi ebeveynlerin herhangi birinden çok fazla bilgi kaybetmeden yeterli bilgi alınmasını sağlamak içindir.

Ebeveyn 1: (1 2 3 4 5 6 7 8)

Ebeveyn 2: (2 4 6 8 7 5 3 1)

Birinci ebeveynden (3 4 5) alt dizisi seçildi. İkinci ebeveynden bunları çıkartılır. (3 4 5) alt dizisi başa yazılır, ikinci ebeveynden kalanlar da aynı sırada kalan pozisyonların devamına yazılır.



Çocuk: (3 4 5 2 6 8 7 1)

Bu çaprazlama operatörünün avantajı, minimum sayıda köşenin bozulmasına izin vermektedir.

### 3.5.8. Ağgözlü Çaprazlama

Grenfestette (1985) tarafından ortaya atılmıştır. Ebeveynlerin birinden ilk şehri seçer, her iki ebeveynde de bu seçilen şehrin yanındaki şehirlerden daha yakın olanı tura seçer. Bir şehir zaten turda varsa, daha önce seçilmişse diğer şehir seçilir. Her iki şehirde turda varsa seçilmemiş bir şehir rasgele seçilir.

Ebeveyn 1: (1 2 3 7 5 6 4)

Ebeveyn 2: (4 1 3 2 7 6 5)

Şehir 4 ikinci ebeveynden çocuğun ilk şehri olarak seçilir. Her iki ebeveynde şehir 4 sonraki kenarlar: (4, 5) ve (4, 1) ve bu iki kenar arasındaki mesafe karşılaştırılır. Şehir 4 ve 1 arasındaki mesafe daha kısa olduğunu varsayarsak 1 seçilir.

Çocuk: (4 1 - - - -)

Şehir 1'den sonraki kenarlar (1, 2) ve (1, 3) şehir 1 ve 2 arasındaki mesafe daha kısa olduğundan 2 seçilir.

Çocuk: (4 1 2 - - -)

Şehir 2'den sonraki kenarlar (2, 3) ve (2, 7) şehir 2 ve 3 arasındaki mesafe daha kısa olduğundan 3 seçilir.

Çocuk: (4 1 2 3 - - -)

Şehir 3'den sonraki kenarlar (3, 7) ve (3, 2) şehir 3 ve 2 arasındaki mesafe daha kısa ama o turda zaten var bu durumda şehir 7 seçilecektir. Bu şekilde çocuk oluşturulur.

Aynı yöntemle diğer çocukta oluşturulabilir.

Çocuk: (4 - - - - -)

### 3.6. Mutasyon (Mutation)

Diğer önemli bir genetik algoritma operatörü mutasyondur. 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.

Mutasyon işlemi çaprazlama sonucu oluşan çocukların tümüne değil, belli bir yüzdesine uygulanır. Buna mutasyon oranı denir. Mutasyonun uygulanma olasılığı (Pm) genellikle çok düşük alınır (Ahmed, 2010: 101).

## 4 ÇAPRAZLAMA OPERATÖRLERİNİN ÖRNEK OLAYLAR BAZLI İNCELENMESİ

GSP'nin tarihi boyunca araştırmacılar önerilen çözüm yöntemlerindeki ilerlemeyi ölçmek için standart test örneklerine dayanmıştır. TSPLIB'de GSP için genellikle verilerin alındığı yerdir. Gernard REINELT Kütüphanesi TSPLIB olarak adlandırılır. 1990'da oluşturulmuştur. Rasgele şehirlerin oluşturulduğu problemler de ortaya atılan yöntemin geçerliliğini test etmek için kullanılmıştır. Ama TSPLIB'de problemin iyi bilinen çözümleri de vardır bu sayede kıyaslama yapmak mümkündür. TSPLIB'de 14 şehirden 85.900 şehir boyutuna kadar 100'ün üzerinde simetrik ve asimetrik örnek vardır.

Amacımız çalışmalara baktığımızda operatörlerin TSPLIB'deki hangi örnek olaylarla daha sık kullanıldığını ve operatörün hangi yaklaşımla o veri seti için daha iyi sonuç verdiğini elde ederek araştırmacılara yol göstermektir. Küçük (17-100 şehir), orta (100-300 şehir), büyük (300+ şehir) ve rassal oluşturulmuş gezgin satıcı problemlerinde GA çözümü için geliştirilmiş olan çaprazlama operatörlerinin 57 tanesi TSPLIB örnek olaylarla birlikte tablolaştırılmıştır. Tablo 2,3,4,5'te satırlarda çaprazlama operatörü sütunlarda ise TSPLIB'den alınmış örnekler ve rasgele oluşturulmuş şehirler yer almaktadır.

**Tablo 2:** Küçük Boyutlu (17-100 Şehir) Gezgin Satıcı Problemleri İçin Kısmi Tablo

OPERATOR \ TSPLIB	GR17	GR24	FR126	BAYG29	BAYS29	EIL30	OLIVER30	DANTZIG42	ATT48	EIL51	BERLİN52	ST70	EIL76	PR76
OX										3,34,36,49,51	28,32,42	34,49	34,36,39,51	34
GREEEDY CROSSOVER										4,33			33	
IMPROVED GX										4				
SCX				5						5	5	47	5	5
ERX				5				20,21		7,34	5,25,32,42	34,47	20,5,21,39,45	5,34
PMX		44					10,31	10,20,21		10,30,31,34,36,44,33,49	25	34,47,49	10,20,21,31,34,36,39,44,33	34
MODIFIED OX							10	10		51			51	
CS1										10			10	
CE1														
CS2														
CE2														
PC1														

**M. PULAT - İ. DEVECİ KOCAKOÇ**

**Tablo 3: Orta Boyutlu (100-3000 Şehir) Gezgin Satıcı Problemleri İçin Kısmi Tablo**

TSPLIB OPERATOR	EİL101	LİN105	PR107	PR124	CH130	SWISS42	PR144	CH150	KROA150	KROB150	PR152	U159	BRG180	RAT195	D198
OX	36, 51	34		34	32,39, 42				14						
GREEEDY CROSSOVER	4														
IMPROVED GX	4														
SCX	5	5											5		5
ERX	5,20,21	5, 34		34	32,39, 42		25	25	20,21	25	25		5	25	5, 25, 35, 48
PMX	10,20, 21,36, 44, 51	44		34	39	44	25	25	14,20, 21	25, 44	25			25, 44	25,48
MODİFİED OX	10														40
CS1				9			9					9			
CE1				9			9					9			
CS2				9			9					9			
CE2				9			9					9			
PC1				9			9					9			

**Tablo 4: Büyük Boyutlu (300+ Şehir) Gezgin Satıcı Problemleri İçin Kısmi Tablo**

TSPLIB OPERATOR	LİN318	RD400	FL417	PR439	PCB442	ATT532	U574	RAT575	RD657	RAT783	P1002	U1060	UM1084
OX		14			14,35, 50			14					
GREEEDY CROSSOVER	4,33												
IMPROVED GX	4												
SCX													
ERX	21, 25, 48		48		21, 25, 35, 48	21	48						
PMX	21, 25, 33, 48	14	48		14,21, 25, 35, 48, 50	21, 46	48	14		46	46		
MODİFİED OX					40					40			
CS1	9	9							9				9
CE1	9	9							9				9
CS2	9	9							9				9
CE2	9	9							9				9
PC1	9	9							9				9

**Tablo 5:** Rasgele Oluşturulmuş Şehirlere Ait Kısmi Tablo

TSPLIB OPERATOR	5	10	12	15	20	25	30	50	75	100	150	10000
OX	49	49			37, 49							
PMX	49	49	45		49	6	17					
MODIFIED OX		1		1	1	6	1					
OX1							17					
OX2(DAVIS)						6	17					
EDGE-2							17					
CX INVER OVER(10)							17					22
GSX				20						26		
GSX-1												
MCX		1		1	1		1					
SX		1		1	1		1					
PBX CROSSOVER & CLEAN UP MODIFIED ORDER CROS.							17					
					43		13	13	13	13	13	
EPMX			45									

Şehirler arasındaki mesafenin rasgele oluşturulduğu çalışmalar Tablo 5'te verilmiştir. Tablonun her bir hücrelerinde de o operatörü ve örnek olayı ele alan makaleler bulunmaktadır. Bu makalelere ilişkin kaynakça da Tablo 6 da verilmiştir. Tablonun tamamına <https://tinyurl.com/y8z5vezt> adresinden ulaşılabilir. Bu web adresindeki Excel dosyasının birinci sayfası şehirlerin ve operatörlerin tamamını içermektedir. İkinci sayfasında da numaralandırılmış biçimde kaynakça yer almaktadır. Sadece çaprazlama operatörünün ele alınmasının nedeni literatürdeki çalışmalarda çok sayıda çaprazlama operatörü geliştirilmişken çok az sayıda mutasyon operatörü geliştirilmiş olmasıdır. Aynı zamanda mutasyonun uygulanma olasılığı çok düşükken çaprazlama olasılığı çok yüksek alınmasıdır. Mutasyon olasılığı genellikle çok düşük olduğundan mutasyonun etkisi kromozomlarda az görülmektedir.

Tabloda siyah renkli olanlar örnek olaylar üzerinde farklı operatörler birbiriyle kıyaslanmıştır. Bu kıyaslama yapılırken farklı oranlarda çaprazlama ve mutasyon operatörü, farklı seçim yöntemleri, farklı başlangıç popülasyonu büyüklüğünün kombinasyonu kullanılarak yapılmıştır. Bu farklılıklara dikkat edilmelidir. Gri olanlarda ise operatörler tek başına birbirleriyle değilde GA içinde GA ile farklı yöntemler birbiriyle kıyaslanmıştır.

Tabloya baktığımızda küçük boyutlu örneklerde kesin yöntemlerle kısa sürede optimal çözümü elde etmek mümkündür ama problem boyutu arttıkça çözüm elde etmek zorlaşmaktadır. Bu yüzden küçük boyutlu örnekler üzerinde GA ile çok fazla çalışma yapılmamıştır. Onları GA gibi sezgisellerle çözmeye gerek yoktur. Problem boyutu küçük olduğunda kesin yöntemlerle optimal çözüm bulunabilmektedir ancak sezgisel yöntemler optimal çözümü garanti edememektedir.

Tablo 6: Numaralandırılmış Kaynakça

Numara	Kaynakça	Numara	Kaynakça
1	<u>Majumdar, J. ve Bhunia A.K. (2011).</u>	33	<u>Ismkhan, H., ve Zamanifar, K. (2015).</u>
2	<u>Larranaga, P., vd. (1999).</u>	34	<u>Gog, A., ve Chira, C. (2011).</u>
3	<u>Javidi, M. M., vd. (2015).</u>	35	<u>Nguyen, H. D., vd. (2000).</u>
4	<u>Ismkhan, H., ve Zamanifar, K. (2012).</u>	36	<u>Kumar, R., vd. (2013).</u>
5	<u>Ahmed, Z. H. (2010).</u>	37	<u>Abdel-Moetty, S. M., vd. (2012).</u>
6	<u>Naveen, K., ve Karambir, R. K. (2012).</u>	38	<u>Amous, S. K., vd. (2008).</u>
7	<u>Deep, K., ve Mebrahtu, H. (2011).</u>	39	<u>Affenzeller, M., ve Wagner, S. (2003).</u>
8	<u>Ray, S. S., vd. (2004).</u>	40	<u>Ray, S. S., vd. (2005).</u>
9	<u>Allaoua, H., ve Brahim, B. (2015).</u>	41	<u>Osaba, E., vd. (2013).</u>
10	<u>Girdhar Gopal, R. K., vd. (2015).</u>	42	<u>Affenzeller, M., ve Wagner, S. (2003).</u>
11	<u>Ghoseiri, K., ve Sarhadi, H. (2007).</u>	43	<u>Elhaddad, Y. R., ve Gannous, A. S. (2012).</u>
12	<u>Mathias, K., and Whitley, D. (1992).</u>	44	<u>Deep, K., ve Mebrahtu, H. (2012).</u>
13	<u>Mitchell, G. G., vd. (2000).</u>	45	<u>Tao, Z. (2008).</u>
14	<u>Martinovic, G., ve Bajer, D. (2011).</u>	46	<u>Merz, P. (2002).</u>
15	<u>Freisleben, B., ve Merz, P. (1996).</u>	47	<u>Khan, I. H. (2015).</u>
16	<u>Katayama, K., vd. (1999).</u>	48	<u>Chen, S., ve Smith, S. F. (1996).</u>
17	<u>Starkweather, T., vd. (1991).</u>	49	<u>Rani, K., ve Kumar, V. (2014).</u>
18	<u>Maekawa, K., vd. (1997).</u>	50	<u>Schneider, J. J., ve Kirkpatrick, S. (2006).</u>
19	<u>Tang, A. Y. C., ve Leung, K. S. (1994).</u>	51	<u>Kumar, R., vd. (2013).</u>
20	<u>Li, L., ve Zhang, Y. (2007).</u>	52	<u>Çolak, S. (2010).</u>
21	<u>Tsai, H. K., vd. (2004).</u>	53	<u>Potvin, J. Y. (1996).</u>
22	<u>Tao, G., ve Michalewicz, Z. (1998).</u>	54	<u>Agarwal, T., ve Singh, K. (2013).</u>
23	<u>Merz, P., ve Freisleben, B. (1997).</u>	55	<u>Nilsson, C. (2003).</u>
24	<u>Freisleben, B., ve Merz, P. (1996).</u>	56	<u>Gersil, M., ve Alkaya, A. (2011).</u>
25	<u>Contreras-Bolton, C., ve Parada, V. (2015).</u>	57	<u>Yang, Y., vd. (2010).</u>
26	<u>Sengoku, H., ve Yoshihara, I. (1998).</u>	58	<u>Pongcharoen, P., vd. (2007).</u>
27	<u>Nguyen, H. D., vd. (2002).</u>	59	<u>Karaboğa, D. (2014).</u>
28	<u>Abdoun, O., vd. (2012).</u>	60	<u>Mohebpour, G. H., ve Delavar, A. G. (2014).</u>
29	<u>Heymendran, J., vd. (2015).</u>	61	<u>Elmas, Ç., (2007).</u>
30	<u>Iyotishree, R. K. (2012).</u>	62	<u>Razali, N. M., ve Geraghty, J. (2011).</u>
31	<u>Chawla, G., ve Bala, M. Y. (2014).</u>	63	<u>Louis, S. J., ve Li, G. (2000).</u>
32	<u>Affenzeller, M., ve Wagner, S. (2004).</u>		

GSP'nin GA ile çözümünü geliştirmek için çok sayıda yaklaşım önerilmiştir. Literatürde incelediğimiz çalışmalara göre:

- Genellikle başlangıç çözümü rasgele oluşturulur. Ancak Nearest-Neighbour sezgiseli gibi farklı başlangıç çözümleri

ile oluşturulduğunda daha iyi sonuçlar elde edilmiştir. Ele alınan probleme özgü başlangıç çözüm önerilebilir.

- Farklı popülasyon büyüklüğünün GA'nın sonuçları üzerinde önemli etkiye

sahiptir. Bu yüzden farklı popülasyon büyüklüğü test edilmelidir.

- Farklı seçim yöntemleri ile birlikte de çaprazlama operatörleri test edilmiştir. Farklı seçim yöntemleri denenebilir ve önerilebilir.
- Çok sayıda çaprazlama operatörü geliştirilmiş ve geliştirilmeye devam etmektedir. Bunlardan bazıları var olan operatörleri temel alarak bazı adımlarında değişiklikler yapılmış geliştirilmiş ya da var olanlarda farklı olarak oluşturulmuştur. Çaprazlama operatörünün GA'nın performansını önemli ölçüde etkilediği görülmüştür. Aynı zamanda çaprazlama oranı da büyük etkiye sahiptir farklı oranları test edilmelidir.
- Az sayıda olsa da bu probleme özgü mutasyon operatörü geliştirilmiştir. Farklı mutasyon operatörleri değişen mutasyon oranları ile birlikte sonuçlar test edilmelidir.
- Bazı çalışmalarda çaprazlama oranı sabit mutasyon oranı her iterasyonda belirli değerler arasında değiştirilmiştir. Bazılarında mutasyon olmaksızın test edilmiştir.

- Lokal arama sezgisellerini kullanarak lokal olarak optimum olan popülasyondan global optimum bulmak için GA uygulanmıştır. GSP için hızı ve kullanım kolaylığından dolayı tercih edilen yerel arama algoritması 2-opt, GA'nın performansını iyileştirmiştir.

## 5 SONUÇ

Çalışmamızda GSP' nin çözümünde kullanılan etkili yöntemlerden biri olan GA'nın temel operatörlerinden biri olan çaprazlama operatörleri incelenmiştir. Çaprazlama operatörleri TSPLIB örnek olayları ile birlikte farklı çalışmalarda ele alınmıştır. Tabloda aynı örnek olayda operatörün farklı yöntemlerle ele alındığı çalışmalar yer alır. Bu çalışmalar incelenerek veri setinde belli bir operatör için uygun yöntem belirlenebilir.

Çaprazlama operatörü; başlangıç popülasyonunun nasıl oluşturulduğuna, popülasyon büyüklüğünün boyutuna, seçim yöntemine, çaprazlama oranı, mutasyon yöntemi, mutasyon oranı farklı sezgisellerle birlikte kullanımına bağlı olarak farklı sonuçlar vermektedir. Bunları doğru bir şekilde belirlediğimizde daha az sapma ve daha kısa hesaplama zamanı ile daha iyi bir çözüm elde edilebilir.

## KAYNAKÇA

ABDEL-MOETTY, S. M., HEAKIL, A. O. (2012), "Enhanced Traveling Salesman Problem Solving Using Genetic Algorithm Technique with Modified Sequential Constructive Crossover Operator", *International Journal of Computer Science and Network Security (IJCSNS)*, 12(6), 134.

ABDOUN, O., ABOUCHABAKA, J., TAJANI, C. (2012), "Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem", *International Journal of Emerging Sciences*, 2(1), 61-77.

AFFENZELLER, M., WAGNER, S. (2003), "A Self-Adaptive Model for Selective Pressure Handling

Within the Theory of Genetic Algorithms", *In International Conference on Computer Aided Systems Theory*, 384-393.

AFFENZELLER, M., WAGNER, S. (2003, June), "SASEGASA: An Evolutionary Algorithm for Retarding Premature Convergence by Self-Adaptive Selection Pressure Steering", *In International Work-Conference on Artificial Neural Networks*, 438-445. Springer, Berlin, Heidelberg.

AFFENZELLER, M., WAGNER, S. (2004), "Reconsidering the Selection Concept of Genetic Algorithms from A Population Genetics

Inspired Point of View”, *Cybernetics and Systems*, 701-706.

AGARWAL, T., SINGH, K. (2013), “Using New Variation Crossover Operator of Genetic Algorithm for Solving the Traveling Salesman Problem”, *MIT International Journal of Computer Science and Information Technology*, 3(1), 35-37.

AHMED, Z. H. (2010), “Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator”, *International Journal of Biometrics & Bioinformatics (IJBB)*, 3(6), 96-105.

ALLAOUA, H., BRAHIM, B. (2015), “A Mono Crossover Genetic Algorithm for TSP”, *Global Journal on Technology, Issue 7 (2015): 4th World Conference on Innovation and Computer Sciences (INSODE-2014)*.

AMOUS, S. K., LOUKIL, T., ELAOU, S., DHAENENS, C. (2008), “A New Genetic Algorithm Applied to the Traveling Salesman Problem”, *International Journal of Pure and Applied Mathematics*, 48(2), 1-16.

CHAWLA, G., BALA, M. Y. (2014), “Solving Optimization Problem by Hybrid Genetic Algorithm Using Hill Climbing in Replacement Operator”, *Journal of Recent Research Aspects*, 2(4), 73-78.

CHEN, S., SMITH, S. F. (1996), “Commonality And Genetic Algorithms”, *Carnegie Mellon University, The Robotics Institute*.

CONTRERAS-BOLTON, C., PARADA, V. (2015), “Automatic Combination of Operators in a Genetic Algorithm to Solve the Traveling Salesman Problem”, *PloS one*, 10(9).

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., MEBRAHTU, H. (2011), “Combined Mutation Operators of Genetic Algorithm for the Travelling Salesman problem”, *International Journal of Combinatorial*

*Optimization Problems and Informatics*, 2(3), 1-23.

DEEP, K., MEBRAHTU, H. (2012), “Variant of Partially Mapped Crossover for the Travelling Salesman Problems”, *International Journal of Combinatorial Optimization Problems and Informatics*, 3(1), 38-60.

ELHADDAD, Y. R., GANNOUS, A. S. (2012), “Two Individual Genetic Algorithm”, *World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 6(3), 209-212.

ELMAS, Ç. (2016), *Yapay Zekâ Uygulamaları*, Seçkin Yayıncılık, Ankara.

ESMKHAN, H. I., ZAMANIFAR, K. (2012), “Developing Improved Greedy Crossover to Solve Symmetric Traveling Salesman Problem”, *International Journal of Computer Science Issue*, 4(3), 1-6.

FREISLEBEN, B., MERZ, P. (1996, September), “New Genetic Local Search Operators for the Traveling Salesman Problem”, *In International Conference on Parallel Problem Solving from Nature*, 890-899.

FREISLEBEN, B., MERZ, P. (1996), “A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems”, *In Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE, 616-621.

GERŞİL, M., ALKAYA, A. (2011), “Gezgin Satıcı Problemi için Sezgisel Metotların Performans Analizi”, *XI. Üretim Araştırmaları Sempozyumu, 23-24 Haziran 2011*, 405-412.

GHOSEIRI, K., SARHADI, H. (2007), “A 2opt-DPX Genetic Local Search for Solving Symmetric Traveling Salesman Problem”, *In 2007 IEEE International Conference on Industrial Engineering and Engineering Management*, 903-906.

GIRDHAR GOPAL, R. K., JAWA, I., KUMAR, N. (2015), “Enhanced Order Crossover for Permutation Problems”, *International Journal*



*of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization)*, 4(2).

GOG, A., CHIRA, C. (2011, May), "Comparative Analysis of Recombination Operators in Genetic Algorithms for The Travelling Salesman Problem", *In International Conference on Hybrid Artificial Intelligence Systems*, 10-17. Springer, Berlin, Heidelberg.

GOPAL, G., KUMAR, R., JAWA, I., KUMAR, N. (2015), "Enhanced Order Crossover for Permutation Problems", *International Journal of Innovative Research in Science, Engineering and Technology*, 4(2), 151-157.

HEYMENDRAN, J., PRIYATHARSAN, U., HEMIJA, P. (2015), "A Representation with Novel Crossover Technique of the Genetic Algorithm for the Travelling Salesmen Problem", *Research Journal of Mathematical and Statistical Sciences*, 3(2), 1-3.

ISMKHAN, H., ZAMANIFAR, K. (2015), "Study of Some Recent Crossovers Effects on Speed and Accuracy of Genetic Algorithm, Using Symmetric Travelling Salesman Problem", *International Journal of Computer Applications*, 80 (6).

ISMKHAN, H., ZAMANIFAR, K. (2012), "Developing Improved Greedy Crossover to Solve Symmetric Traveling Salesman Problem", *Int J. Comput. Sci.*, 9 (4), 121-126.

JAVIDI, M. M., FARD, R. H., JAMPOUR, M. (2015), "Research in Random Parameters of Genetic Algorithm and Its Application on TSP and Optimization Problems", *Walailak Journal of Science and Technology*, 12(1), 27-34.

JONG, K. A. De, SPEARS, W. M. (1989), "Using Genetic Algorithms to Solve NP-Complete Problems", *In:ICGA*, 124-132.

JYOTISHREE, R. K. (2012), "Novel Knowledge Based Tabu Crossover in Genetic Algorithms", *International Journal*, 2(8).

KARABOĞA, D. (2014), *Yapay Zekâ Optimizasyon algoritmaları*, Nobel Yayın Dağıtım, İstanbul.

KATAYAMA, K., HIRABAYASHI, H., NARIHISA, H. (1999), "Performance Analysis for Crossover Operators of Genetic Algorithm", *Systems and Computers in Japan*, 30(2), 20-30.

KHAN, I. H. (2015), "Assessing Different Crossover Operators for Travelling Salesman Problem", *IJ Intelligent Systems and Applications*, 11, 19-25.

KUMAR, R., GOPAL, G., KUMAR, R. (2013), "Alpha Cut based Novel Selection for Genetic Algorithm", *International Journal of Computer Applications*, 13-17.

KUMAR, N., KARAMBIR, KUMAR, R. (2012), "A Comparative Analysis of PMX, CX and OX Crossover operators for solving Travelling Salesman Problem", *International Journal of Latest Research in Science and Technology*, 1(2).

KUMAR, R., GOPAL, G., KUMAR, R. (2013), "Novel Crossover Operator for Genetic Algorithm for Permutation Problems", *International Journal of Soft Computing and Engineering (IJSCE)*, 3(2), 252-258.

LARRANAGA, P., KUJERS, C. M. H., MURGA, R. H., INZA, I., DIZDAREVIC, S. (1999), "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators", *Artificial Intelligence Review*, 13(2), 129-170.

LI, L., ZHANG, Y. (2007, August), "An Improved Genetic Algorithm for the Traveling Salesman Problem", *In International Conference on Intelligent Computing*, 208-216.

LOUIS, S. J., ve LI, G. (2000), "Case Injected Genetic Algorithms for Traveling Salesman Problems", *Information sciences*, 122(2-4), 201-225.

MAEKAWA, K., MORI, N., TAMAKI, H., KITA, H., NISHIKAWA, Y. (1997), "A Genetic Solution for The Traveling Salesman Problem by means of a Thermodynamical Selection Rule", *Transactions of the Society of Instrument and Control Engineers*, 33(9), 939-946.

MAJUMDAR, J., BHUNIA A.K. (2011), "Genetic Algorithm for Asymmetric Traveling Salesman

Problem with Imprecise Travel Times”, *Journal of Computational and Applied Mathematics*, 235 (9), 3063–3078.

MARTINOVIC, G., BAJER, D. (2011), “Impact of Double Operators on The Performance of a Genetic Algorithm for Solving the Traveling Salesman Problem”, *In International Conference on Swarm, Evolutionary and Memetic Computing, Springer, Berlin, Heidelberg*, 290-298.

MATHIAS, K., WHITLEY, D. (1992), "Genetic Operators, The Fitness Landscape and the Traveling Salesman Problem", *In PPSN*, 219-228.

MERZ, P. (2002, July), “A Comparison of Memetic Recombination Operators for The Traveling Salesman Problem”, *In Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 472-479.

MERZ, P., FREISLEBEN, B. (1997, April), “Genetic Local Search for the TSP: New Results”, *In Proceedings of 1997 Ieee International Conference on Evolutionary Computation (Icec'97)*, 159-164.

MICHALEWICZ, Z. (1992), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Vergal, Berlin.

MITCHELL, G. G., O'DONOGHUE, D., TRENAMAN, A. (2000), “A New Operator for Efficient Evolutionary Solutions to The Travelling Salesman Problem”, *In Proc. Applied Informatics*, 771-774.

MOHEBPOUR, G. H., DELAVAR, A. G. (2014), “CCGDC: A New Crossover Operator for Genetic Data Clustering”, *The Journal of Mathematics and Computer Science*, 11, 191-208.

NAVEEN, K., KARAMBIR, R. K. (2012), “A Comparative Analysis of PMX, CX and OX Crossover Operators for Solving Travelling Salesman Problem”, *International Journal of Latest Research in Science and Technology*, 1(2), 98-101.

NGUYEN, H. D., YOSHIHARA, I., YASUNAGA, M. (2000), “Modified Edge Recombination

Operators of Genetic Algorithms for The Traveling Salesman Problem”, *In 2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000*, 2815-2820.

NGUYEN, H. D., YOSHIHARA, I., YAMAMORI, K., YASUNAGA, M. (2002), “Greedy Genetic Algorithms for Symmetric and Asymmetric TSPs”, *IPSJ Transactions on Mathematical Modeling and Its Applications*, 43(10), 165-175.

NILSSON, C. (2003), “Heuristics for the Traveling Salesman Problem”, *Linköping University, Sweden*, 1-6.

OSABA, E., CARBALLEDO, R., DÍAZ, F., PERALLOS, A. (2013, May), “Analysis of the Suitability of Using Blind Crossover Operators in Genetic Algorithms for Solving Routing Problems”, *In 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 17-22. IEEE.

PONGCHAROEN, P., CHAINATE, W., THAPATSUWAN, P. (2007), “Exploration of Genetic Parameters and Operators through Travelling Salesman Problem”, *ScienceAsia*, 33(2), 215–222.

POTVIN, J.-Y. (1996), “Genetic Algorithms for the traveling salesman problem”, *Annals of Operations Research*, 63(3), 337–370.

RANI, K., KUMAR, V. (2014), “Solving Travelling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator”, *International Journal of Research in Engineering & Technology*, 2(2), 27-34.

RAY, S. S., BANDYOPADHYAY, S., PAL, S. K. (2005), “New Genetic Operators for Solving TSP: Application to Microarray Gene Ordering”, *In International Conference on Pattern Recognition and Machine Intelligence*, 617-622.

RAY, S. S., BANDYOPADHYAY, S., PAL, S. K. (2004), “New Operators of Genetic Algorithms for Traveling Salesman Problem”, *In Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, (2), 497–500.

RAZALI, N. M., GERAGHTY, J. (2011), "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP", *In Proceedings of the World Congress on Engineering*, Hong Kong, 2(1), 1-6.

SCHNEIDER, J. J., KIRKPATRICK, S. (2006), "Application of genetic algorithms to TSP", *Stochastic Optimization*, 415-422.

SENGOKU, H., YOSHIHARA, I. (1998, January), "A Fast TSP Solver Using GA on JAVA", *In Third International Symposium on Artificial Life and Robotics (AROB III'98)*, 283-288.

STARKWEATHER, T., MCDANIEL, S., MATHIAS, K. E., WHITLEY, L. D., ve WHITLEY, C. (1991, July), "A Comparison of Genetic Sequencing Operators", *In ICGA* 69-76.

TANG, A. Y. C., LEUNG, K. S. (1994, October), "A Modified Edge Recombination Operator for the Travelling Salesman Problem", *In International Conference on Parallel Problem Solving from Nature*, 180-188.

TAO, Z. (2008, October), "TSP Problem Solution Based on Improved Genetic Algorithm", *In 2008 Fourth International Conference on Natural Computation*, Vol.1, 686-690.

TAO, G., MICHALEWICZ, Z. (1998, September), "Inver-Over Operator for the TSP", *In International Conference on Parallel Problem Solving from Nature*, 803-812.

TSAI, H. K., YANG, J. M., TSAI, Y. F., KAO, C. Y. (2004) "Some Issues of Designing Genetic Algorithms for Traveling Salesman Problems", *Soft Computing*, 8(10), 689-697.

YANG, Y., DAI, H., LI, H. (2010), "Adaptive Genetic Algorithm with Application for Solving Traveling Salesman Problems", *International Conference on Internet Technology and Applications*, 1-4.

<https://tinyurl.com/y8z5vezt>