

Gezgin satıcı probleminin melez akışkan genetik algoritma (MAGA) kullanarak çözümü

Solving travelling salesman problem using hybrid fluid genetic algorithm (HFGA)

Yusuf ŞAHİN^{1*}, Kenan KARAGÜL²

¹İşletme Bölümü, İktisadi ve İdari Bilimler Fakültesi, Mehmet Akif Ersoy Üniversitesi, Burdur, Türkiye.

ysahin@mehmetakif.edu.tr

²Lojistik Bölümü, Honaz Meslek Yüksekokulu, Pamukkale Üniversitesi, Denizli, Türkiye.

kkaragul@pau.edu.tr

Geliş Tarihi/Received: 12.01.2018, Kabul Tarihi/Accepted: 08.05.2018

* Yazışılan yazar/Corresponding author

doi: 10.5505/pajes.2018.81084

Araştırma Makalesi/Research Article

Öz

Gezgin Satıcı Problemi (GSP), bir satıcının bütün şehirleri sadece bir defa ziyaret ederek başlangıç noktasına dönmeyi sağlayan en kısa rotanın belirlendiği problemidir. GSP, araç rotalamadan baskılı devre kartı montajına kadar birçok problemin temelini oluşturur. Bu problem, optimizasyon alanında çalışan kişilerden büyük ilgi görmüştür, ancak özellikle büyük ölçekli veri kümeleri için çözülmesi zordur. Bu çalışmada, GSP'nin çözümü için Akışkan Genetik Algoritma, En Yakın Komşu ve 2-Opt sezgiselleri üzerine kurulu melez bir yöntem sunulmaktadır. Önerilen yöntemin performansı literatürde bulunan En Yakın Komşu, Genetik Algoritma, Tabu Arama, Karınca Kolonisi Optimizasyonu ve Ağaç Fizyolojisi Optimizasyon algoritmaları kullanılarak elde edilen çözüm değerleri ile kıyaslanmıştır. Önerilen yöntemin sonuçları çözüm süresi ve kalitesi bakımından üstünlük göstermektedir.

Anahtar kelimeler: Gezgin satıcı, Meta-sezgiseller, Akışkan genetik algoritma

Abstract

Travelling Salesman Problem (TSP) is a problem in which the shortest possible route enabling a salesman to return to the starting point after visiting all the cities exactly once is determined. Travelling Salesman Problem is the basis for many problems ranging from vehicle routing to printed circuit boards assembly. This problem has been attracting great attention from researchers in the field of optimization; nevertheless it is difficult to solve TSP, especially for large-scale data sets. This paper presents a hybrid solution method based on Fluid Genetic Algorithm, Nearest Neighbor and 2-Opt methods for the solution of TSP. The performance of the proposed method is evaluated with the solution values of the Nearest Neighbor, Genetic Algorithm, Tabu Search, Ant Colony Optimization and the Tree Physiology Optimization algorithms in the literature. The solution results show the superiority of the proposed method in terms of solution time and quality.

Keywords: Traveling salesman, Metaheuristics, Fluid genetic algorithm

1 Giriş

Gezgin Satıcı Problemi (GSP), bir satış elemanının bir dizi şehir ziyaret edip yaşadığı şehre dönerken izlemesi gereken en kısa rotanın belirlendiği bütünleşik bir optimizasyon problemidir. Optimizasyon alanında yoğun ilgi gören probleme teorik açıdan bakıldığında, n tane köşe noktası üzerinde, her bir nokta sadece bir kez ziyaret edilmek koşuluyla, ilk çıkış noktasında sona eren seyahat için en kısa mesafeyi veren rotanın belirlendiği problemidir [1]. Tanımlaması kolay ancak çözümü oldukça zor olan problemde şehir sayısının artışı, kesin çözüm yöntemleri ile kabul edilebilir bir süre zarfında çözümün bulunabilmesini zorlaştırır. Bu özelliği nedeniyle NP-Zor sınıfında yer alan GSP'nin çözümü için sezgisel ve metasezgisel yöntemler sıklıkla kullanılmaktadır.

İlk defa 1930'lu yıllarda matematiksel olarak ifade edilen GSP'nin simetrik, asimetric ve çok satıcılı olmak üzere üç türünden bahsedilebilir [2],[3]. Simetrik GSP'de iki şehir arasındaki gidiş ve dönüş mesafeleri aynı iken ($c_{ij}=c_{ji}$), asimetric GSP'de ise bu mesafeler birbirinden farklıdır ($c_{ij}\neq c_{ji}$). Çok satıcılı GSP (mGSP) ise birden fazla satıcının yer aldığı GSP'dir. Baskılı devre kartlarının montajı [4], ekip planlaması [5], okul otobüsü rotalama [6], görev planlaması [7], araç rotalama [8] ve depolarda sipariş toplama [9] gibi problem türleri için oluşturulan modellerin esin kaynağı GSP'dir. Bu

problemlerin tamamı GSP'ye yeni kısıtlar eklenmesi ile oluşturulmuş problem türleridir.

GSP'nin çözümü için literatürde birçok algoritma ve çözüm yöntemi önerilmiştir. Lin-Kernighan-Helsgaun algoritmasını kullanan Concorde yazılımı ile 85.900 şehir, Rice ve Princeton Üniversitesi bilgisayarlarının 22 yıllık çalışmasıyla ve aynı bilgisayarlar ve yazılım ile Almanya'nın 15.112 şehri için optimal sonuçlar elde edilmiştir [35]. NP-Zor sınıfında yer alması nedeniyle GSP'nin tam çözümünün elde edilmesi karmaşık ve zaman alıcıdır. Bu durum GSP'nin çözümü için sezgisel ve metasezgisel yöntemlerin gerekliliğini ortaya koymaktadır [9]. Metasezgisel yöntemler birçok optimizasyon problemine kabul edilebilir bir süre içerisinde etkili çözümler sunduğu için yaygın olarak kullanılan yöntemlerdir. GSP'ye küçük boyutlu problemler için kesin çözüm yöntemleri uygulanabilirken, büyük çaplı problemlerin çözümü için daha çok genetik algoritma [11], parçacık sürü optimizasyonu [12], karınca kolonisi [13], tabu arama [14], benzetimli tavlama [15], kör fare algoritması [36] ve harmoni arama algoritması [10] gibi metasezgisel yöntemler kullanılmaktadır.

Snyder ve Daskin [16], genetik algoritma ile 2-opt tur geliştirici sezgiselini birleştirerek GSP'ye çözüm aramıştır. Chang ve diğ. [17], sistem performansını daha da artırmak için evrim süreci boyunca dinamik bir eşik kontrol mekanizması önermiştir. Önerilen yöntem, genetik algoritmanın küresel yakınsama davranışını iyileştirmek için problem özelliklerinden bağımsız

olarak uygulanabilen bir yöntemdir. Razali ve Geraghty [18] farklı ebeveyn seçim stratejileri için genetik algoritmanın GSP'de çözüm performansını değerlendirmiştir. Elde edilen sonuçlar turnuva ve orantılı rulet tekeri yöntemlerinin küçük problemler için sıra esaslı rulet tekeri yöntemine göre daha iyi sonuçlar verdiğini, ancak problem boyutu arttıkça erken yakınsamaya duyarlı hale geldiğini belirlemişlerdir. Majumdar ve Bhunia [19], asimetrik GSP'nin çözümü için aralık değerli uygunluk fonksiyonuna sahip bir genetik algoritma önermiştir. Deep ve Mebrahtu [20] GSP'nin çözümünde performansı arttırmak için kombine iki mutasyon yöntemi önermişlerdir. Önerdikleri evrik değişim (inverted exchange) ve evrik yer değiştirme (inverted displacement) yöntemlerinin performansları mevcut dört mutasyon operatörü ile karşılaştırılmıştır.

Antosiewicz ve diğ. [21] GSP'nin çözümü için altı adet metasezgisel yöntemi karşılaştırmıştır. Yapılan karşılaştırma sonucunda benzetimli tavlama en iyi çözümleri bulurken, tabu arama düşük varyanslı hızlı sonuçlar üretmiştir. Ahmed [22] sıralı kümelenmiş gezgin satıcı probleminin çözümü için sıralı çaprazlama, 2-opt ve yerel arama sezgisellerinin kullanıldığı melez bir genetik algoritma önermiştir. Kuzu ve diğ. [23] sekiz farklı metasezgisel yöntemle GSP'yi çözümler sonuçları karşılaştırmıştır. Karşılaştırılan yöntemler çözüm kalitesinin sürekliliği bakımından değerlendirildiğinde, genetik algoritma ve karınca kolonisi optimizasyon algoritmalarının daha iyi sonuçlar ortaya koyduğu görülmektedir. Kang ve diğ. [24] GSP'nin çözümü için görüntü işleme birimlerinin (GPU) paralel hesaplama gücünden istifade eden çaprazlama yöntemlerinin genetik algoritmadaki paralel biçimde uygulandığı bir yöntem önermiştir. Önerilen çaprazlama yönteminin GSP için önceki çaprazlama yöntemleri gibi yapıcı bir yaklaşım olduğu belirtilmektedir. Halim ve Ismail [25] en yakın komşu (EYK), genetik algoritma (GA), tavlama benzetimi (TB), karınca kolonisi (KKO) ve ağaç fizyolojisi optimizasyon algoritmalarının (AFOA) GSP çözüm performanslarını karşılaştırmıştır. Rana ve Srivastava [26] ağırlıklı sezgiseli ile elde ettikleri çözümleri genetik algoritmanın başlangıç popülasyonuna eklemek suretiyle GSP'ye çözüm aramıştır.

Yapılan çalışmalar incelendiğinde genellikle genetik operatörlerinin değişimi ile çözüm performansının geliştirilmesi üzerine odaklanıldığı görülmektedir. Bu çalışmada, rota kurucu ve geliştirici sezgiseller olan En Yakın Komşu (EYK) ve 2-Opt sezgiselleri Jafari-Marandi ve Smith [27] tarafından geliştirilen Akışkan Genetik Algoritma (AGA) isimli yönteme entegre edilerek GSP'nin çözümü için melez bir çözüm yöntemi (MAGA) önerilmiştir. Çalışma, yeni bir yöntem olan Akışkan Genetik Algoritmanın klasik çözüm yöntemleri ile bütünleşik olarak kullanılması bakımından özgün bir çalışmadır. Çalışmanın ikinci bölümünde GSP'nin matematiksel modeli, üçüncü bölümde GA ve AGA yöntemleri, dördüncü bölümde MAGA'nın GSP'ye uyarlanması, beşinci bölümde deneysel çalışma ve takip eden bölümde sonuç ve öneriler yer almaktadır.

2 Gezgin satıcı problemi

V , n tane köşe noktasının kümesi, A bağlantıyı oluşturan bir dizi yay veya kenar ve C (c_{ij}) A ile ilişkili bir mesafe (veya maliyet) matrisi olmak üzere $G=(V,A)$, bir çizgeyi ifade eder. GSP, böyle bir çizge üzerinde köşe noktalarından sadece bir kez geçerek oluşturulan minimum mesafeli Hamilton turunun bulunması problemi [25]. Simetrik GSP, yönlendirilmemiş bir çizge üzerinde tanımlanırken, asimetrik GSP yönlendirilmiş bir çizge

üzerinde tanımlanır. Yönlendirilmemiş bir çizge üzerindeki bağlantılar için oluşturulan mesafe matrisinde (C) iki bağlantı arasındaki gidüş ve dönüş maliyetleri eşit iken ($c_{ij}=c_{ji}$), diğerinde bu maliyetler birbirinden farklıdır. Burada C , mesafe veya seyahat maliyetinin yer aldığı matris olmak üzere GSP'nin matematiksel modeli aşağıda sunulmuştur [3].

Karar Değişkenleri:

$$x_{ij} = \begin{cases} 1, & \text{eğeri noktasından } j \text{ noktasına gidilirse} \\ 0, & \text{aksihalde} \end{cases}$$
$$\min Z = \sum_{i < j} c_{ij} \cdot x_{ij} \quad (1)$$

Kısıtlar:

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2 \quad (k \in V) \quad (2)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad (S \subset V, 3 \leq |S| \leq N - 3) \quad (3)$$

$$x_{ij} = 0 \text{ veya } 1 \quad (i, j) \in E \quad (4)$$

Yukarıda sunulan 0-1 tamsayı programlama modelinde i şehirden sonra j şehrine gidilmesi halinde x_{ij} 1 değerini alırken, aksi halde 0 değerini alır. (1) No.lu eşitlik toplam seyahat mesafesinin minimizasyonunu ifade eden amaç fonksiyonudur. (2) No.lu eşitlik bir şehre yalnız bir giriş ve bir çıkış olmasını ifade eden derece kısıtıdır. (3) No.lu kısıt ile alt tur oluşumu engellenmeye çalışılmaktadır.

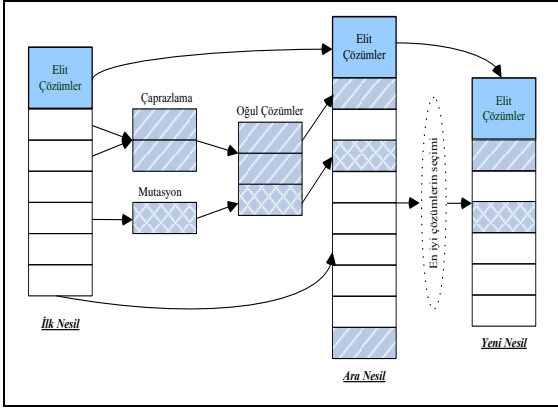
3 Akışkan genetik algoritma

Standart bir genetik algoritma başlangıç rassal çözümler kümesi ile çözüme başlar. Her bir çözümün gösterimi için çözüm kümesinde (toplulukta) yer alan bireyler kromozomlardır. Kromozomların gösteriminde problemin yapısına bağlı olarak 0-1 veya permutasyon kodlaması kullanılır. Kromozomlar gen olarak adlandırılan kısımlardan oluşur. Uygulanan iterasyonlar neticesinde kromozomlar değişime uğratılır ve yeni nesiller oluşturulur. Genetik algoritmada kromozomların değişimi ve yeni nesillerin oluşturulması için çaprazlama ve mutasyon operatörleri kullanılır. Çaprazlamada iki kromozom eşleştirilerek oğul bireyler meydana gelirken, mutasyon gende oluşturulan değişim ile yeni birey oluşumu sağlar. Oluşan her yeni nesilde kromozomların çözüm kalitesi uygunluk fonksiyonu ile değerlendirilir. Topuluk hacminin sabit tutulabilmesi için oluşan yeni nesilde çözüm kalitelerine göre seçim işlemi yapılır ve geri kalan kromozomlar elenir. Belirlenen sonlandırma ölçütleri (süre veya uygunluk değerinin yakınsaklığı) sağlanıncaya kadar iterasyonlar devam ettirilerek en iyi çözümü veren kromozomların bulunması amaçlanır [28],[29]. Genetik Algoritmanın genel işleyişi Şekil 1'de gösterilmektedir.

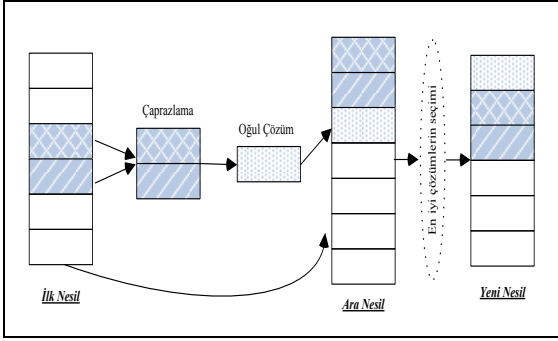
3.1 Akışkan genetik algoritmanın temelleri

Literatürde genetik algoritma ile ilgili birçok çalışma olmasına karşın kromozom, birey, çaprazlama ve mutasyon gibi kavramlardan oluşan bu yöntemin özünü ilgili bir çalışma yapılmamıştır. AGA bazı temel farklılıklara sahip bir genetik algoritmadır. Bu temel farklılıklar Tablo 1'de gösterilmektedir [27]. AGA, kromozom ve bireylerin (çözümlerin) gösterimi, çaprazlama ve mutasyon açısından standart genetik algoritmadan ayrılmaktadır. Bunların yanı sıra *bireysel*

öğrenme oranı (individual learning rate), küresel öğrenme oranı (global learning rate), çeşitlilik oranı (diversity rate), bireyin doğuşu (born an individual) ve nesil gen tasarımı (generation blueprint) gibi kavramların eklenmesiyle genetik algorithmadan farklılaştırılmıştır. AGA'nın genel işleyişi Şekil 2'de gösterilmektedir [27].



Şekil 1:GA'nın genel işleyişi [30].



Şekil 2: AGA'nın genel işleyişi.

Tablo 1: GA ve AGA arasındaki en önemli farklar.

	GA	AGA
Kromozom	✓	Değişmiş
Birey (çözüm)	✓	Değişmiş
Uygunluk değeri	✓	✓
Çözümleme fonksiyonu	✓	✓
Çaprazlama	✓	Değişmiş
Mutasyon	✓	×
Seçim fonksiyonu	✓	✓
Durdurma kriteri	✓	✓
Bireysel öğrenme oranı	×	✓
Global öğrenme oranı	×	✓
Çeşitlilik oranı	×	✓
Bireyin doğumu	×	✓
Nesil gen tasarımı	×	✓

3.1.1 Kromozom gösterimi

GA pratik bir optimizasyon yöntemidir. Bir popülasyonun içerisinde yer alan çözümlerin ifade edilebilmesi için problem türüne göre farklı kromozom gösterimleri kullanılabilir. Belirli bir problem çözülmeye çalışılırken düşünülmesi gereken konuların en başında kromozomun, yani çözümün gösterimi gelmektedir. GA'nın ilk uygulamalarında ikili dizi (binary

strings) yapısı kullanılmış olmasına karşın, günümüz uygulamalarında problemin türüne göre farklı gösterimler kullanılmaktadır. Karakter dizileri, permütasyon kodlama, değer kodlama ve ağaç kodlama çözüm gösterimleri ikili kodlamanın yanı sıra yaygın olarak kullanılan diğer kromozom gösterim yöntemleridir.

GA ile karşılaştırıldığında AGA'nın en temel farklılıklarından birisi kromozom gösterimidir. AGA'da gen içerisindeki değer bireyin 0 yerine 1 değeri alma olasılığını ifade eder. Gen içerisindeki değerler birey doğum fonksiyonu (*born-an-individual function*) ile hesaplanır. Şekil 3'te gösterilen kromozomda 3. genin 1 değerini alma olasılığı 0.85'tir.

0.36	0.58	0.85	0.97	0.69	0.61	0.18	0.90
0	0	1	1	1	1	0	1

Şekil 3: AGA'da örnek kromozom gösterimi.

3.1.2 Birey doğum fonksiyonu

AGA'da kromozom ve birey arasındaki bağlantı birey doğum fonksiyonu ile sağlanmaktadır. Birey doğum fonksiyonu, giren kromozoma bağlı olarak GA'ya benzer bir şekilde cevap üretir. Birey doğum fonksiyonu her bir gen için ilgili genin değerine göre 0 veya 1 değerlerinin üretimi için kullanılır. Örneğin Şekil 3'te gösterilen kromozomda 3 No.lu genin 0 yerine 1 değerini alma olasılığı %85'tir. Teorik olarak bakıldığında bu örnekte sunulan kromozom ile 2^8 farklı birey farklı olasılık değerlerinde ortaya çıkabilir [27].

Birey doğum fonksiyonu kullanılırken global öğrenme oranı ve nesil gen tasarımı olmak üzere iki parametre kullanılır. Nesil gen tasarımı her bir iterasyonda bütün kromozomlardaki değerlerin ortalaması olarak alınır. Bu değer ilk iterasyon için 0.5'tir. Fonksiyonun kromozom ve nesil gen tasarımı olarak iki girdisi vardır. Genin 0 veya 1 değerini alma olasılığı Eşitlik (5) ile hesaplanır. Eşitlikten görüleceği üzere iki durum etkili olasılık değerini (EPV) 1'in altında ve 0'ın üstünde tutmaya çalışmaktadır. Bu durumlar AGA'nın lokal optimuma takılmasını engellemek için tanımlanmaktadır. Eğer EPV çeşitlilik oranının altına incek olursa gen 0 değerini alacaktır. Bu hesaplamalar yapılırken kullanılan birey doğum fonksiyonu Eşitlik (5)'te sunulmuştur [27].

Notasyon:

- η_g : Global öğrenme oranı,
- PVC_i : i . gen için kromozomdan alınan olasılık değeri,
- PVB_i : i . gen için nesil planından alınan olasılık değeri,
- η_{DR} : Çeşitlilik oranı,
- EPV_i : Etkili olasılık değeri.

$$\begin{cases} \eta_g \times PVB_i + (1 - \eta_g) \times PVC_i < \eta_{DR} & EPV_i = \eta_{DR} \\ \eta_g \times PVB_i + (1 - \eta_g) \times PVC_i > 1 - \eta_{DR} & EPV_i = 1 - \eta_{DR} \\ aksihalde & EPV_i = \eta_g \times PVB_i + (1 - \eta_g) \times PVC_i \end{cases} \quad (5)$$

3.1.3 Çaprazlama operatörü

Şekil 1'de özetlenen genetik algoritma sürecinde farklı kromozomların eşleştirilmesi ile oğul bireylerin (yeni çözümlerin) oluştuğu yapı çaprazlama sürecidir. Çaprazlama yapılmasının sebebi arama uzayının farklı noktalarında bulunan kromozomların belirli bölümlerinin karşılıklı olarak yer değiştirmesini sağlayarak aramayı daha önce ziyaret edilmeyen farklı bölgelere taşımaktır. Standart

genetik algoritmada iki kromozomun çaprazlaması neticesinde iki oğul birey ortaya çıkar. Mevcut nesil içerisinde çaprazlamaya tabi tutulacak kromozom sayısı çaprazlama oranı (p_c) ile belirlenir. Bu oran genellikle [0.5,1.0] aralığında belirlenir. Tek nokta, çok nokta, düzgün, parçalı eşleştirilmiş çaprazlama ve sıra esaslı çaprazlama gibi yöntemler literatürde yaygın olarak kullanılmaktadır [30],[31].

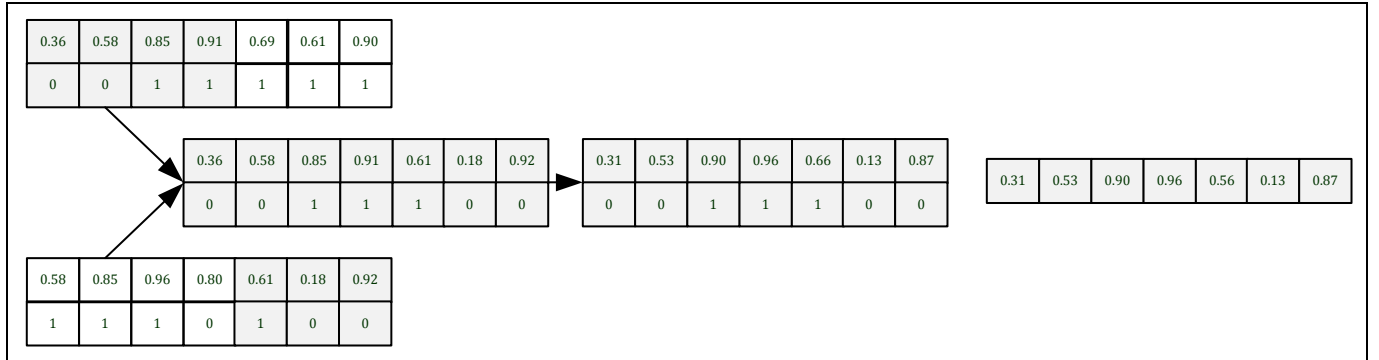
AGA'da ise standart genetik algoritmadan farklı olarak çaprazlama operatörü uygulanan iki kromozomdan bir tane oğul birey oluşur. Bu süreçte rastgele olarak karıştırılan iki kromozomdan üretilen oğul birey öğrenme oranına göre değişikliğe uğrar. Oğul bireyde gen 1 değerini aldıysa olasılık değerine öğrenme oranı eklenirken, 0 değerini aldıysa olasılık değerinden aynı oran çıkarılır. Kullanılan bu yeni yapı sayesinde iyi bir popülasyon çeşitliliği sağlandığı için algoritmanın yerel optimuma yakalanma ihtimali azalmaktadır. Bu durum aynı zamanda algoritmada mutasyon operatörü kullanma ihtiyacını da ortadan kaldırmaktadır. AGA'da birey öğrenme oranınının 0.05 olduğu çaprazlama uygulaması Şekil 4'te gösterilmiştir [27].

4 MAGA'nın gezgin satıcı probleminde uygulaması

4.1 GSP İçin Çözüm Gösterimi

Buraya kadar olan kısımda genel hatları ile AGA'nın özellikleri anlatılmıştır. Bu kısımda yöntemin GSP'ye nasıl uygulandığı açıklanmaya çalışılmıştır. MAGA'nın GSP'ye uygulamasında her bir gene öncelikle bir indis değeri verilir. Bu sayede Şekil 5'te gösterilen kromozom yapısı ortaya çıkar. Şehir sayısı kadar gen olacağı için şehir sayısı kadar indis tanımlanacaktır.

Ardından Şekil 5'te gösterilen kromozomda örnek olması için rastgele verilen olasılık değerlerine göre genlerin sıralanmasıyla Şekil 6'da gösterilen GSP gösterimi elde edilir.



Şekil 4: AGA'da çaprazlama operatörü.

Kromozom	0.60	0.85	0.13	0.70	0.27
İndis	1	2	3	4	5

Şekil 5: GSP için kromozom gösterimi.

Kromozom	0.13	0.27	0.60	0.70	0.85
GSP Çözümü	3	5	1	4	2

Şekil 6: Kromozomun GSP çözümüne dönüşümü.

Bu noktada rota belirleme işlemi için entegre EYK+2-Opt sezgiseli devreye sokulur. Bu yöntemlerin adımları kısım 4.2 ve 4.3'te sunulmuştur.

4.2 EYK sezgiseli

En Yakın Komşu, rota belirlemek için konumsal verilerin kullanıldığı sezgisel bir yöntemdir. Yaygın bir kullanıma sahip bu yöntem, belirli bir konuma en yakın konumun rotaya eklenmesi suretiyle aracın genel rotasının belirlendiği bir yöntemdir. EYK yöntemi hızlı ve etkili sonuçlar üretmesi sebebiyle çözüm rotasını belirlemek için çalışma kapsamında kullanılmıştır. Genel işleyişi beş adımdan oluşan bu yöntemin GSP için adımları aşağıda sunulmuştur [31].

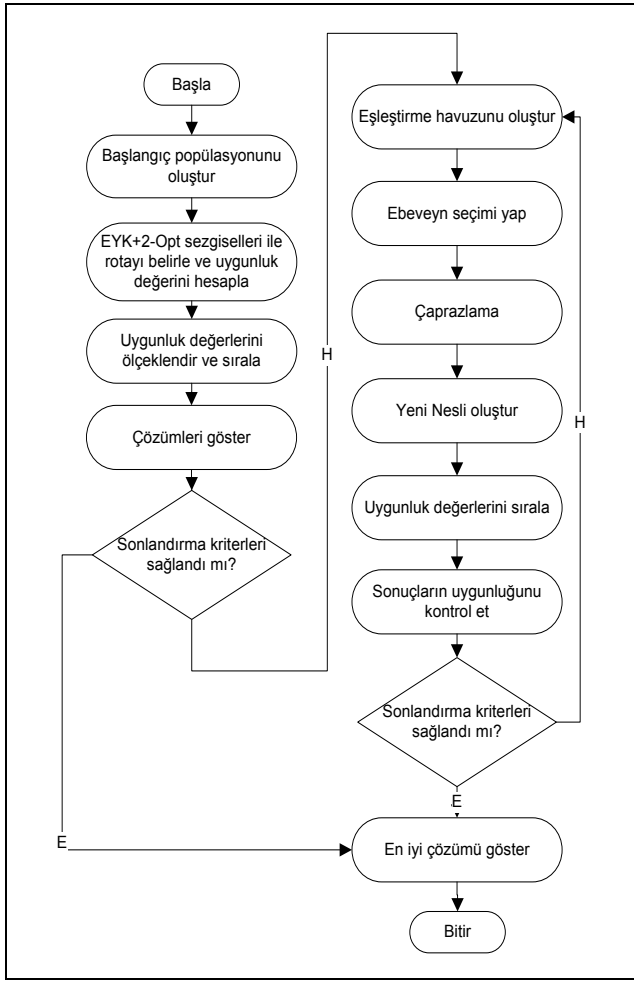
- Adım 1:* Başlangıç konumundan en kısa mesafeli şehri belirle,
- Adım 2:* İlk şehrin konumundan diğer şehirlerin konumlarına olan mesafeyi belirle,
- Adım 3:* Mevcut mesafeler arasında en kısa olanı seç ve ikinci şehri belirle,
- Adım 4:* Listedeki bütün şehirler tamamlanana kadar Adım 2 ve 3 ü tekrar et,
- Adım 5:* Şehirlerin seçim sırasına göre şehir konumlarını birleştir ve rotayı göster.

4.3 2-Opt Sezgiseli

EYK ile elde edilen rotaların geliştirilmesi için rota geliştirici 2-Opt sezgiseli [32] çalışma kapsamında diğer yöntemlerle bütünleşik bir şekilde kullanılmıştır. 2-opt algoritmasının GSP için işleyişi şu şekildedir [33]:

- Adım 1:* Rassal olarak turdaki şehir çiftlerini belirle,
- Adım 2:* Tur bozulmayacak şekilde, şehir çiftlerinin yerini değiştir,
- Adım 3:* Yeni oluşan tur önceki tura göre bir gelişme sağlamış ise şehir çiftlerini yeni yerlerinde bırak, aksi halde eski yerine iade et.

GSP'nin çözümü için önerilen melez çözüm yönteminin genel akışı ise Şekil 7'de gösterilmektedir.



Şekil 7:MAGA'nın akış şeması.

5 Deneysel çalışmalar

Deneysel çalışmalar kapsamında Halim ve İsmail [25] tarafından kullanılan 15 adet iki boyutlu (EUC_2D) veri kümesi tercih edilmiştir. Bu veri kümeleri öklidyen uzaklık formülü ile hesaplama yapılabilen verilerden oluşmaktadır. Önerilen melez yöntem için MATLAB programlama dilinde bir yazılım hazırlanmış, test işlemleri Intel Dual Core 2.40 GHz, 8 GB RAM özelliklerine sahip Linux işletim sistemi üstünde çalışan bir bilgisayarda tek işlemci ile gerçekleştirilmiştir. Çalışma kapsamında önerilen yöntem için küçük çaplı bir veri kümesi ile yapılan denemelerin ardından Tablo 2'de yer alan parametre kümesi kullanılmasına karar verilmiştir.

Tablo 2: Parametre kümesi.

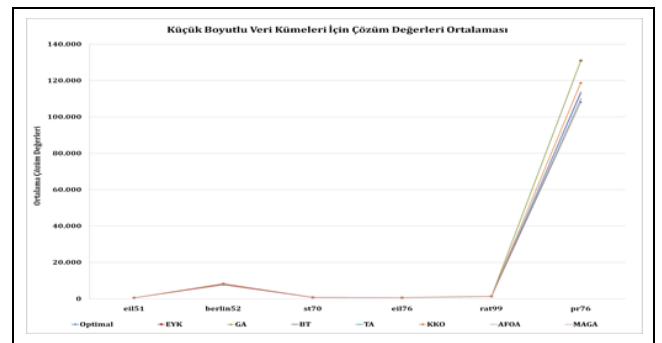
Parametre	Değer
Maksimum İterasyon Sayısı	100
Popülasyon Büyüklüğü	50
Çaprazlama Oranı	0.50
Birey Öğrenme Oranı (ILR)	0.15
Global Öğrenme Oranı (GLR)	0.15
Çeşitlilik Oranı (DR)	0.70

Elde edilen sonuçlar Halim ve İsmail [25]'de yer alan sonuçlarla kıyaslanmıştır. Deneyler için kullanılan veri kümeleri [34] Tablo 3'te gösterilmektedir. Seçilen veri kümeleri kıyaslanmanın yapıldığı çalışmada olduğu gibi düğüm sayısına göre küçük ($n < 100$), orta boyutlu ($190 > n \geq 100$) ve büyük boyutlu ($n > 190$) şeklinde sınıflandırılmıştır.

Tablo 3: Kullanılan veri kümeleri.

Veri Kümesi	Boyutu	Düğüm Sayısı	Optimal Çözüm
eil51	Küçük	51	426
berlin52	Küçük	52	542
st70	Küçük	70	675
eil76	Küçük	76	538
pr76	Küçük	76	108.159
rat99	Küçük	99	1.211
kroA100	Orta	100	21.282
eil101	Orta	101	629
ch130	Orta	130	6.110
ch150	Orta	150	6.528
rat195	Büyük	195	2.323
d198	Büyük	198	15.780
a280	Büyük	280	2.579
rd400	Büyük	400	15.281
pcb442	Büyük	442	50.778

Tablo 3'te gösterilen veri kümeleri kullanılarak 5 tekrarlı deneyler gerçekleştirilmiş, elde edilen çözüm sonuçları Tablo 4'te, çözüm süreleri ise Tablo 5'te özetlenmiştir. Tablo 4'ten de görüleceği üzere, diğer yöntemlerle kıyaslandığında bütün veri kümeleri için AGA en iyi çözüm değeri ortalamalarını sunmaktadır. 15 veri kümesinin 5 tanesinde optimal çözüm değerine ulaşılmıştır. Elde edilen çözümlerin ortalamalarına bakıldığında %0-2.13 arasında değişen oranlarda en iyi çözüme yakınsama sağlanmıştır. Çözüm sürelerini karşılaştırabilmek için kıyaslama yapılan çalışmanın yazarlarından yöntemlere ait çözüm süresi detayları istenmiş ancak olumsuz cevap alınmıştır. Bu nedenle kıyaslama diğer çalışmada verilen grafikler üzerinden yapılmıştır. Küçük, orta ve büyük boyutlu problemler için elde edilen çözüm değerlerinin ortalamalarına ilişkin karşılaştırmalar sırasıyla Şekil 8, Şekil 9 ve Şekil 10'da gösterilmiştir.



Şekil 8: Ortalama çözüm değeri (küçük boyutlu problemler).

Tablo 4: GSP problemleri için her bir yöntem ile elde edilen sonuçlar.

Veri Kümesi	Yöntem	Optimal Çözüm	Ortalama Çözüm	En İyi Çözümde Sapma (%)	Standart Sapma	En İyi Çözüm	En Kötü Çözüm
eil51	EYK	426	505.08	18.56	1.17	503.17	505.99
	GA	426	454.10	6.60	1.35	452.90	455.90
	BT	426	439.13	3.08	2.29	437.42	443.04
	TA	426	439.10	3.08	4.00	434.01	443.58
	KKO	426	467.46	9.73	0.91	466.54	468.43
	AFOA	426	437.26	2.64	1.65	435.28	438.91
	MAGA	426	427.40	0.33	0.97	426.00	429.00
berlin52	EYK	7542	8182.78	8.50	1.66	8180.66	8185.26
	GA	7542	7946.40	5.36	280.66	7546.00	8269.00
	BT	7542	7960.67	5.55	44.69	7903.77	8020.72
	TA	7542	7740.10	2.63	148.90	7544.37	7937.87
	KKO	7542	7922.32	5.04	44.55	7872.59	7985.41
	AFOA	7542	7705.80	2.17	101.59	7544.00	7810.00
	MAGA	7542	7542	0.00	0.00	7542	7542
st70	EYK	675	761.51	12.82	0.91	760.67	762.99
	GA	675	700.72	3.81	10.07	685.75	711.07
	BT	675	696.33	3.16	1.17	695.10	698.00
	TA	675	690.27	2.26	9.23	680.99	703.74
	KKO	675	756.55	12.08	11.20	739.87	768.75
	AFOA	675	697.12	3.28	2.19	694.91	699.86
	MAGA	675	675.1	0.01	0.32	675	676
eil76	EYK	538	612.26	13.80	0.73	611.38	613.16
	GA	538	570.03	5.95	5.77	560.83	575.70
	BT	538	567.15	5.42	1.84	564.68	569.22
	TA	538	561.71	4.41	5.07	554.54	568.65
	KKO	538	590.69	9.79	2.87	586.43	594.06
	AFOA	538	556.77	3.49	11.03	545.65	574.31
	MAGA	538	544.9	1.28	0.32	544	545
pr76	EYK	108159	130921.92	21.05	1.06	130920.67	130923.18
	GA	108159	122981.65	13.70	6098.68	117673.20	130630.95
	BT	108159	113000.23	4.48	1788.28	110620.77	115152.60
	TA	108159	109930.19	1.64	688.21	109046.25	110943.51
	KKO	108159	118733.31	9.78	2051.79	116259.14	121226.86
	AFOA	108159	113843.92	5.26	2777.70	111341.00	117865.61
	MAGA	108159	108175.1	0.01	27.35	108159	108234
rat99	EYK	1211	1368.75	13.03	1.54	1366.44	1370.53
	GA	1211	1285.61	6.16	1.15	1284.62	1287.56
	BT	1211	1277.36	5.48	0.85	1276.54	1278.66
	TA	1211	1243.47	2.68	8.35	1233.45	1252.59
	KKO	1211	1324.30	9.36	2.47	1320.54	1326.40
	AFOA	1211	1265.85	4.53	0.91	1264.74	1267.18
	MAGA	1211	1222.3	0.93	1.25	1220	1223
kroA100	EYK	21282	24697.83	16.05	1.65	24695.35	24699.54
	GA	21282	22726.20	6.79	504.18	22278.00	23368.00
	BT	21282	22277.50	4.68	708.29	21837.88	22782.68
	TA	21282	22521.64	5.82	215.30	22293.45	22794.73
	KKO	21282	22941.68	7.80	29.83	22908.97	22990.15
	AFOA	21282	22463.60	5.55	445.45	21795.00	22852.00
	MAGA	21282	21300	0.08	21.26	21282	21353
eil101	EYK	629	735.98	17.01	0.40	735.43	736.37
	GA	629	685.89	9.04	3.81	680.67	689.56
	BT	629	672.13	6.86	5.93	664.29	679.72
	TA	629	667.61	6.14	5.16	661.66	674.41
	KKO	629	752.91	19.70	3.66	748.03	757.40
	AFOA	629	675.00	7.31	9.15	658.66	679.54
	MAGA	629	630	0.16	0.00	630	630
ch130	EYK	6110	7198.30	17.81	1.95	7195.17	7200.18
	GA	6110	6610.80	8.20	159.12	6426.00	6777.00
	BT	6110	6558.70	7.34	136.79	6335.90	6699.94
	TA	6110	6717.06	9.94	451.71	6214.81	7334.39
	KKO	6110	6913.99	13.16	11.76	6900.30	6929.02
	AFOA	6110	6515.28	6.63	80.60	6409.03	6594.13
	MAGA	6110	6164.80	0.90	25.65	6129.00	6204.00

Tablo 4: Devamı.

Veri Kümesi	Yöntem	Optimal Çözüm	Ortalama Çözüm	En İyi Çözümünden Sapma (%)	Standart Sapma	En İyi Çözüm	En Kötü Çözüm
ch150	EYK	6528	7077.89	8.42	1.09	7076.50	7079.11
	GA	6528	7004.76	7.30	3.46	7000.54	7009.40
	BT	6528	7061.83	8.18	97.04	6951.57	7176.90
	TA	6528	6862.34	5.12	180.33	6616.01	7051.91
	KKO	6528	7350.48	12.60	15.47	7331.64	7370.45
	AFOA	6528	6942.43	6.35	39.11	6900.20	6994.48
	MAGA	6528	6569.40	0.63	1.90	6564.00	6570.00
rat195	EYK	2323	2628.38	13.15	1.57	2625.65	2629.56
	GA	2323	2414.52	3.94	5.90	2407.45	2420.54
	BT	2323	2537.99	9.25	24.82	2497.54	2560.45
	TA	2323	2373.94	2.19	11.66	2359.36	2388.40
	KKO	2323	2465.11	6.12	40.07	2401.43	2499.44
	AFOA	2323	2573.47	10.78	51.66	2516.24	2656.84
	MAGA	2323	2347.00	1.03	0.00	2347.00	2347.00
d198	EYK	15780	18062.37	14.46	0.84	18061.17	18063.17
	GA	15780	16582.86	5.09	172.00	16405.87	16829.83
	BT	15780	16380.49	3.81	247.95	16035.16	16728.84
	TA	15780	16083.48	1.92	32.19	16043.14	16124.63
	KKO	15780	18031.92	14.27	155.01	17806.47	18206.47
	AFOA	15780	16645.64	5.49	126.95	16440.25	16767.17
	MAGA	15780	15937	0.99	0.00	15937	15937
a280	EYK	2579	3094.21	19.98	0.43	3093.78	3094.89
	GA	2579	2789.83	8.82	47.47	2787.75	2894.43
	BT	2579	2830.18	9.74	87.54	2766.43	2976.77
	TA	2579	2800.79	8.60	13.18	2786.31	2816.81
	KKO	2579	2867.85	11.20	88.22	2733.74	2965.85
	AFOA	2579	2790.54	7.89	13.63	2763.00	2795.04
	MAGA	2579	2634	2.13	0.00	2634	2634
rd400	EYK	15281	18219.35	19.23	1.96	18216.53	18221.99
	GA	15281	16567.29	8.42	145.90	16346.38	16752.11
	BT	15281	16816.65	10.05	47.86	16763.14	16880.70
	TA	15281	20723.56	35.62	15.13	20703.33	20739.47
	KKO	15281	19259.06	26.03	75.28	19165.00	19365.00
	AFOA	15281	18190.84	19.04	116.68	18049.89	18372.33
	MAGA	15281	15606	2.13	0.00	15606	15606
pcb442	EYK	50778	58952.63	16.10	2.18	58950.14	58955.99
	GA	50778	55718.90	9.73	750.90	54424.78	56337.00
	BT	50778	57421.04	13.08	730.06	56207.01	57987.05
	TA	50778	83123.01	63.70	42.22	83059.00	83172.00
	KKO	50778	63436.70	24.93	504.87	62543.00	63741.52
	AFOA	50778	60750.43	19.64	4264.65	56742.00	65929.80
	MAGA	50778	51757	1.93	0,00	51757	51757

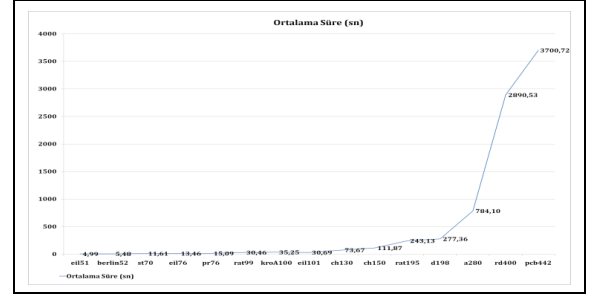
Tablo 5: Veri kümeleri için ortalama çözüm süresi.

Veri Kümesi	Boyut	Ortalama Çözüm Süresi (sn.)
eil51	Küçük	4.99
berlin52	Küçük	5.48
st70	Küçük	11.61
eil76	Küçük	13.46
pr76	Küçük	15.09
rat99	Küçük	30.46
kroA100	Orta	35.25
eil101	Orta	30.69
ch130	Orta	73.67
ch150	Orta	111.87
rat195	Büyük	243.13
d198	Büyük	277.36
a280	Büyük	784.10
rd400	Büyük	2890.53
pcb442	Büyük	3700.72

Tablo 4, Tablo 5 ve devamında yer alan grafikler incelendiğinde önerilen yöntemin bütün veri kümeleri için optimal çözüme

daha yakın ortalama çözüm değerleri sağladığı görülmektedir. Kıyaslanan çalışmada yer alan çözüm süresi grafiklerinden TA

ve KKO'nun küçük çaplı problemlere bile 5.000 sn. dolayında çözüm üretmektedir. Küçük çaplı problemler için MAGA'nın en uzun ortalama çözüm süresi 30.46 sn.'dir. Orta boyutta problemler için TA ve KKO 10.000 sn. civarında çözüm üretirken, bu süre MAGA için en fazla ch150 veri kümesinde 111.87 sn. olarak gerçekleşmiştir. 195 şehrin yer aldığı rat195 veri kümesinden 442 şehirlil pcb442 veri kümesine kadar olan büyük boyutlu problemler için çözüm süreleri 243.13 ile 3700.72 sn. arasında değişmektedir. Karşılaştırmada kullanılan diğer yöntemlerden EYK dışında kalanlar deneysel çalışmalarda kullanılan veri kümelerine 100-1400 sn. arasında değişen sürelerde çözüm üretebilmiştir. Bu yöntemlerin çözüm sürelerinden bazıları MAGA'ya göre iyi olmasına karşın, çözüm değerleri önerilen yöntem ile kıyaslandığında daha geride kalmaktadır. Buradan hareketle, MAGA'nın hızlı ve etkin çözümler ürettiği sonucuna ulaşılabilir. Şekil 11'de yer alan yöntemlerin optimal değerlerden sapma yüzdeleri grafiği incelendiğinde bu fark daha net görülmektedir. Şekil 12'de ise Tablo 5'te sunulan sürelerle ait grafik yer almaktadır.



Şekil 12: Veri kümeleri için MAGA'nın ortalama çözüm süreleri.

6 Sonuçlar

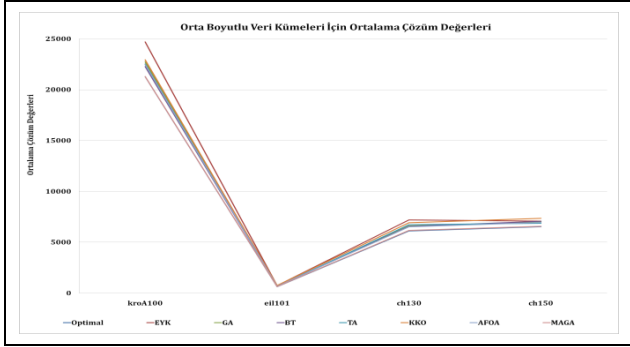
Çalışma kapsamında literatürde geniş yer bulan Gezin Satıcı Problemi ele alınmıştır. Problemin çözümü için EYK ve 2-Opt sezgisellerinin entegre edildiği MAGA adı verilen bir yöntem sunulmuştur. Deneysel çalışma için Heidelberg Üniversitesi'nin web sayfasından [34] alınan veri kümeleri kullanılmıştır. Standart GA'dan Tablo 1'de verilen özellikler bakımından farklılık gösteren bu yöntemin performansını analiz edebilmek için 15 veri kümesi ile deneyler yapılmıştır. Yapılan bu deneyler neticesinde çözüm süresi ve kalitesi bakımından dikkat çekici sonuçlara ulaşılmıştır. Kullanılan veri kümelerinden 5 tanesinde optimal çözüme ulaşma başarısı gösteren MAGA yöntemi, bütün veri kümeleri için çözüm kalitesi ve süresi bakımından diğer yöntemlerden daha iyi sonuçlar ortaya koymuştur. Entegre EYK+2-Opt sezgisellerinin rota belirleme safhası için yöntemle eklenmesi hızlı ve iyi sonuçlar alınmasında en büyük etkidir. Gelecekte yapılacak olan çalışmalarda, kullanılan matematiksel model ve yazılım üzerinde yapılacak ufak değişiklikler ile GSP'nin uzantısı olan Kapasiteli Araç Rotalama, Envanter Rotalama ve Sipariş Toplama Problemi gibi problem türlerine de yöntemin rahatlıkla uygulanabileceği ve iyi sonuçlar alınabileceği değerlendirilmektedir. Bu çalışmada yapılmamakla birlikte, yeni bir yöntem olan AGA'da kullanılacak parametre değerleri için belirli bir deney tasarımı planına bağlı olarak en iyi değerlerin belirlenmesi için varyans analizi çalışmaları yapılabilir.

7 Teşekkür/Acknowledgment

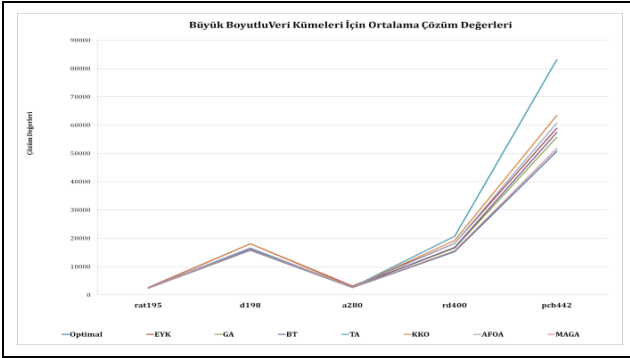
Yöntemin GSP çözümüne uygulanması sürecindeki değerli destekleri için AGA'nın geliştiricileri R. Jafari-Marandi ve Brian K. Smith'e teşekkür ederiz. Yardımcı ve yapıcı önerileri için anonim hakemlere ve yayın kuruluna minnettarız. We would like to thank R. Jafari-Marandi and Brian K. Smith, the developers of the FGA, for their valuable support in the process of applying the method to the GSP solution. We are grateful to the anonymous reviewers and the editorial board for their helpful and constructive advice.

8 Kaynaklar

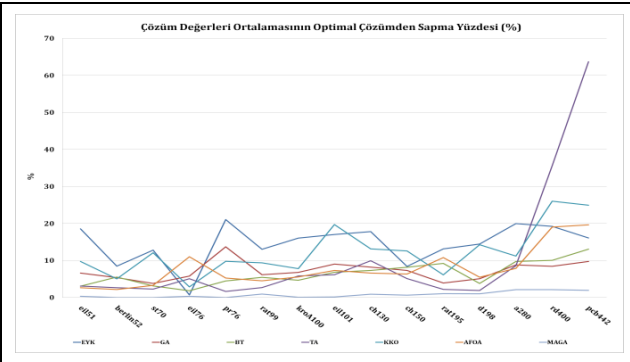
- [1] Potvin JY. "Genetic algorithms for the traveling salesman problem". *Annals of Operations Research*, 63(3), 337-370, 1996.
- [2] Goyal S. "A survey on travelling salesman problem". *43rd Midwest Instruction and Computing Symposium*, Eau Claire, Wisconsin, USA, 16-17 April 2010.
- [3] Matai R, Singh S, Mittal ML. *Traveling Salesman Problem: an Overview of Applications, Formulations and Solution Approaches*. Editor: Davendra D. Traveling salesman problem, theory and applications, London, United Kingdom, 1-24, InTech, 2010.



Şekil 9: Ortalama çözüm değeri (orta boyutlu problemler).



Şekil 10: Ortalama çözüm değeri (büyük boyutlu problemler).



Şekil 11: Yöntemlerin optimal çözümden sapma yüzdeleri (%).

- [4] Burke EK, Cowling PI, Keuthen R. "New models and heuristics for component placement in printed circuit board assembly". *International Conference on Information Intelligence and Systems*, Bethesda, Maryland, USA, 31October-3 November, 1999.
- [5] Hoffman KL, Padberg M. "Solving airline crew scheduling problems by branch-and-cut". *Management Science*,39(6), 657-682, 1993.
- [6] Park J, Byung-In K. "The school bus routing problem: A review". *European Journal of Operational Research*, 202(2), 311-319, 2010.
- [7] Yu Z, Jinhai L, Guochang G, Rubo Z, Haiyan, Y. "An implementation of evolutionary computation for path planning of cooperative mobile robots". *4th World Congress on Intelligent Control and Automation*, Shanghai, China, 10-14 June 2002.
- [8] Mazzeo S, Irene L. "An ant colony algorithm for the capacitated vehicle routing". *Electronic Notes in Discrete Mathematics*, 18, 181-186, 2004.
- [9] Ratliff HD, Rosenthal AS. "Orderpicking in a rectangular warehouse: a solvable case of the traveling salesman problem". *Operations Research*, 31 (3), 507-521, 1983.
- [10] Karagül K, Aydemir, E, Tokat S. "Using 2-Opt based evolution strategy for travelling salesman problem". *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 6(2), 103-113, 2016.
- [11] Zhao F, Li S, Sun J, Mei D. "Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem." *Computers & Industrial Engineering*, 56(4), 1642-1648, 2009.
- [12] Dorigo M, Gambardella LM, "Ant colony system: a cooperative learning approach to the traveling salesman problem." *IEEE Transactions on evolutionary computation*,1(1), 53-66, 1997.
- [13] Mavrovouniotis M, Yang S. "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors". *Applied Soft Computing*, 13(10), 4023-4037, 2013.
- [14] Gendreau M, Laporte G, Semet F. "A tabu search heuristic for the undirected selective travelling salesman problem". *European Journal of Operational Research*,106(2-3), 539-545, 1998.
- [15] Malek M, Guruswamy M, Pandya M, Owens H. "Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem". *Annals of Operations Research*, 21(1), 59-84, 1989.
- [16] Snyder LV, Daskin MS. "A random-key genetic algorithm for the generalized traveling salesman problem". *European Journal of Operational Research*, 174(1), 38-53, 2006.
- [17] Chang PC, Huang WH, Ting CJ. "Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems.". *Expert systems with applications*,37(3), 1863-1878, 2010.
- [18] Razali NM, Geraghty J. "Genetic algorithm performance with different selection strategies in solving TSP". In *Proceedings of the world congress on engineering*, 2, 1134-1139, 2011.
- [19] Majumdar J, Bhunia AK. "Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times". *Journal of Computational and Applied Mathematics*, 235(9), 3063-3078, 2011.
- [20] Deep K, Adane, HM. "New variations of order crossover for travelling salesman problem". *International Journal of Combinatorial Optimization Problems and Informatics*, 2(1),2-13, 2011.
- [21] Antosiewicz M, Koloch G, Kamiński B. "Choice of best possible metaheuristic algorithm for the travelling salesman problem with limited computational time: quality, uncertainty and speed". *Journal of Theoretical and Applied Computer Science*, 7(1), 46-55, 2013.
- [22] Ahmed ZH. "An experimental study of a hybrid genetic algorithm for the maximum traveling salesman problem". *Mathematical Sciences*,7(1), 10, 2013.
- [23] Kuzu S, Önay O, Şen U, Tunçer M, Yıldırım BF, Keskinürk, T. "Gezgin satıcı problemlerinin metasezgiseller ile çözümü". *Istanbul University Journal of the School of Business*,43(1), 1-27, 2014.
- [24] Kang S, Kim SS, Won J, Kang YM. "GPU-based parallel genetic approach to large-scale travelling salesman problem". *The Journal of Supercomputing*, 72(11), 4399-4414, 2016.
- [25] Halim AH, Ismail I. "Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem". *Archives of Computational Methods in Engineering*, 2017. <https://doi.org/10.1007/s11831-017-9247-y>
- [26] Rana S, Srivastava SR. "Solving travelling salesman problem using improved genetic algorithm". *Indian Journal of Science and Technology*,10(30), 1-6, 2017.
- [27] Jafari-Marandi R, Smith, BK. "Fluid genetic algorithm (FGA)". *Journal of Computational Design and Engineering*, 4(2), 158-167, 2017.
- [28] Gen M, Cheng R. *Genetic Algorithms and Engineering Design*. New York, USA, Wiley, 1997.
- [29] Şahin Y, Eroğlu, A. "Kapasite kısıtlı araç rotalama problemi için metasezgisel yöntemler: bilimsel yazın taraması". *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*,19(4), 337-355, 2014.
- [30] Şahin Y. Depo Operasyonları ve Sipariş Dağıtım Faaliyetlerinin Sezgisel Yöntemler Kullanarak Eş Zamanlı Optimizasyonu. Doktora Tezi, Süleyman Demirel Üniversitesi, Isparta, Türkiye, 2014.
- [31] Eiben AE, Smith JE. *Introduction to Evolutionary Computing*. Berlin, Germany, Springer-Verlag Heidelberg, 2003.
- [32] Croes GA. "A method for solving large scale symmetric traveling salesman problems to optimality". *Operations Research*, 6, 791-812, 1958.
- [33] Eryavuz M, Gencer C. "Araç rotalama problemine ait bir uygulama". *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 6(1), 139-155, 2001.
- [34] Universität Heidelberg. "Index of /software/TSPLIB95/tsp". <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/> (18.11.2017).
- [35] Joines A, Kay, MG, Karagül K, Tokat S. *Performance analysis of Genetic Algorithm Optimization Toolbox via Traveling Salesperson Problem*. Editor: Sayers W. Contemporary Issues in Social Sciences and Humanities, 213-221, London, UK, AGP Research, 2017.
- [36] Yıldırım T, Kalaycı CB, Mutlu Ö. "Gezgin satıcı problemi için yeni bir meta-sezgisel: kör fare algoritması". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 22(1), 64-70, 2016.