



Gezgin Satıcı Problemi İçin Yeni Bir Çözüm Yaklaşımı: TPORT

A Novel Solution Approach for Travelling Salesman Problem: TPORt

Kenan Karagül^{1*}

¹ Pamukkale Üniversitesi, Honaz Meslek Yüksekokulu, Yönetim ve Organizasyon Bölümü, Denizli, TÜRKİYE
Sorumlu Yazar / Corresponding Author *: kkaragul@pau.edu.tr

Geliş Tarihi / Received: 21.02.2019

DOI:10.21205/deufmd.2019216312

Kabul Tarihi / Accepted: 05.05.2019

Araştırma Makalesi/Research Article

Atıf Sekizi/How to cite: KARAGÜL, K. (2019). Gezgin Satıcı Problemi İçin Yeni Bir Çözüm Yaklaşımı: TPORt. DEUFMD, 21(63), 819-832.

Öz

Bu çalışmada, yöneylem araştırması alanının en çok çalışılan problemlerden biri olan gezgin satıcı ve ulaşırma problemleri üzerinde durulmakta ve aralarındaki ilişkiden faydalanan yeni bir çözüm algoritması önerilmektedir. Ulaşırma problemleri için bir çok başlangıç çözüm algoritması önerilmiştir. Benzer bir mantık ve sezgi ile simetrik gezgin satıcı problemine başlangıç çözümü üretmek için TPORt adı verilen bir yaklaşım önerilmiştir. Bu yöntem gezgin satıcı problemini daha etkin çözmek için yıllardır süren araştırmalara bir katkı sağlamak için önerilmiştir. Önerilen yöntemde gezgin satıcı uzaklık matrisi, bir ulaşırma tablosu gibi ele alınarak, matris üzerinde yapılan özel bir normalizasyon işlemi ile gezgin satıcı problemi için başlangıç çözümü elde edilmektedir. Daha sonra, elde edilen başlangıç çözümünün performansı 2-Opt algoritması ile geliştirilmektedir. Geliştirilen sezgisel, En Yakın Komşu algoritması ile yakınlık gösterdiği için gezgin satıcı problemlerinin çözüm performansları En Yakın Komşu algoritması ve doğrudan başlangıç çözümüne uygulanan 2-Opt algoritması sezgisellerinin çözümleri ile karşılaştırılmıştır. Önerilen yaklaşım sıkılıkla kullanılan gezgin satıcı test problemleri ve bilimsel yazında yer alan bir grup problem ile analiz edilmiştir. Ortalama çözüm değeri optimalden %26 sapma gösterirken, En Yakın Komşu algoritması için optimalden sapma %16 olarak gerçekleşmiştir. Ancak 2-Opt ile hem TPORt hem de En Yakın Komşu algoritmalarının çözümleri geliştirildiğinde, sırasıyla %4 ve %3 optimalden ortalama sapma elde edilmiştir. Bu bağlamda önerilen çözüm yaklaşımının çözüm performansı açısından rekabetçi olduğu ileri sürülebilir. Ayrıca çözüm süreleri açısından yapılan karşılaştırmalarda önerilen yöntemle En Yakın Komşu algoritması arasında önemli düzeyde fark vardır. Sonuç olarak, önerilen yöntemin çözüm hızı açısından üstün, çözüm kalitesi bakımından kıyaslanan yöntemlere göre rekabetçi olduğu gösterilmiştir. Özellikle, problem boyutu büyüdükçe kıyaslanan yöntemlerin çözüm süresi neredeyse sabit bir seviyede seyrederken En Yakın Komşu algoritmasının çözüm süreleri üstel bir eğilim göstermiştir.

Anahtar Kelimeler:Gezgin Satıcı Problemi, Ulaşırma Problemi, En Yakın Komşu Algoritması, 2-Opt Algoritması

Abstract

In this study, one of the most frequently studied problems in the field of operations research, the traveling salesman problem and the transportation problem, are considered, and a new solution algorithm that takes advantage of the relationship between them is proposed. Many initial solution algorithms have been proposed for transportation problems. In this study, with a similar approach

and intuition, an approach called TPORT is proposed for the initial solution of the symmetric traveling salesman problem. This method has been proposed to contribute to years of research to solve the traveling salesman problem more effectively. In the proposed method, the traveling salesman distance matrix is treated as a transportation table, and a special normalization process on the matrix provides the initial solution of the traveling salesman problem. Then, the performance of the initial solutions is improved with 2-Opt algorithm. As the developed heuristic has similarities with the nearest neighbor algorithm, the solution performances of the traveling salesman problems were compared with the solutions of the nearest neighbor algorithm and the solutions of the directly applied 2-Opt algorithm. The proposed approach has been analyzed by the well-known traveling salesman test instances and a group of test instances from the literature. The average solution performance of the proposed method has 26% deviation from the optimal, whereas the performance of the nearest neighbor algorithm has a 16% deviation from the optimal. However, when the solutions of TPORT and nearest neighbor algorithm were improved with 2-Opt, the average deviation was obtained as 4% and 3%, respectively. In this context, it can be argued that the proposed solution approach is competitive in terms of solution performance. Also, there is a huge difference between the proposed method and the nearest neighbor algorithm in terms of solution times. As a result, it has been shown that the proposed method is superior in terms of solution speed and competitive in terms of solution quality. In particular, as the problem size is increased, the solution time of the comparable methods is almost constant, while the solution time of the nearest neighbor algorithm shows an exponential increasing trend.

Keywords: Traveling Salesman Problem, Transportation Problem, Nearest Neighbor Algorithm, 2-Opt Algorithm

1. Giriş

Ulaştırma problemi (UP), m adet kaynaktan n adet talep noktasına ürünlerin en az maliyetle dağıtımının planlanması problemidir. Atama problemi (AP) ise tek kişi sadece tek işe atanmak üzere, n adet işe n adet kişinin en az maliyetle atanması problemini ifade eder [1-4] UP ve AP birbiriley benzer şekilde ele alınan, yakın ilişkili problemlerdir. Genel olarak bakıldığından her iki problem de kapasiteli ağ akış probleminin özel türüdür [5]. Diğer taraftan, yine bilimsel yazında çok çalışılan ve temel yöntemi araştırması problemlerinden biri olan gezgin satıcı problemi (GSP) yukarıda bahsedilen problemler ile büyük oranda benzerlik gösterir. Bu bağlamda Ulaştırma, Atama ve GSP arasında önemli teorik ilişkiler olduğu ileri sürülebilir [3].

Günümüzde gerek araştırmacılar gerekse endüstriyel uygulayıcılar GSP için hızlı ve etkin çözümler üretebilecek yöntemler geliştirmek için yoğun çaba harcamaktadır. Her zaman daha hızlı ve optimal ve/veya optimalden minimum sapmayla çözüme ulaşmayı sağlayacak yöntemlerin önerilmesi ile ilgili literatür çalışmaları artarak devam etmektedir. GSP'ni bu kadar özel yapan olgu ise teorik ve endüstriyel bir çok problem için temel model niteliği taşımasıdır. GSP'nin NP-Zor sınıfında yer alan kombinasyonel bir problem olması da onu araştırmacılar için çekici yapmaktadır.

Bu çalışmada, GSP için başlangıç çözümü elde etmek için UP'lerine başlangıç çözümü üreten yaklaşımları temel alan bir sezgisel önerilmektedir. UP, AP ve GSP arasındaki benzerlikler üzerinden hareketle yeni çözüm yaklaşımları sunulmaya çalışılmıştır. Bu bağlamda, UP için başlangıç çözüm üretebilecek TPORT algoritması kullanılarak GSP başlangıç çözümleri elde edilmektedir. Bu çıktılarla GSP çözüm sezgiselleri kullanılarak GSP için hızlı ve etkin çözümler bulunması hedeflenmektedir.

Çalışmanın bilimsel yazın başında UP ve GSP ile ilgili bilimsel yazın kısaca sunulmuştur. Araştırmancının yöntemi başında kısaca En Yakın Komşu, 2-Opt ve TPORT algoritmaları açıklanmıştır. Analiz çalışması ve bulgular başında önerilen yöntem ve En Yakın Komşu algoritması ve bu yöntemlerin ürettiği çıktıların 2-Opt ile geliştirilmesi ile elde edilen çözümler karşılaştırılmıştır.

GSP tanımı ve anlatılması kolay, ancak çözümü oldukça zor bir problemdir. Küçük boyutlu problemler için kesin yöntemlerle çözümlere erişmek mümkündür. Ancak problem boyutları büyümeye başladığında kesin yöntemlerle çözüme erişmek imkansız hale gelmektedir. GSP'nin bilimsel yazında NP-Zor sınıfında yer alan bir problem olduğu gösterilmiştir. Yöntem araştırması alanında klasik bir problem olan GSP yüzyılı aşkın bir

süredir araştırmacıların en çok ilgisini çeken problemlerden birisi olmayı sürdürmüştür ve sürdürmektedir. Bu nedenle GSP yazınının büyük bölümü sezgisel ve metasezgisel yaklaşımalarla GSP çözümlerine adanmıştır. Sezgisel ve metasezgisellerin yanında farklı yapısal yaklaşımalarla da GSP için çözüm arayışları vardır. Sezgisellerin kullanılmasıyla optimal ve/veya optimale yakın çözümlere makul sürelerde ulaşılabilirken [6]. Bilimsel yazının çok büyük bölümü GSP çözümü için sezgisel ve/veya metasezgisellere ayrıldığı için tüm yazının ortaya konması olası değildir. Bu noktada, genetik algoritma [7-8], aksikan genetik algoritma [9], evrimsel hesaplamaya dayalı harmoni arama algoritması [10], parçacık sürü optimizasyonu [11], karınca kolonisi optimizasyonu [12], tabu arama [13], benzetimli tavlama [14] GSP'nin çözümünde sezgisel ve metasezgisel yöntemlerin kullanılmasına örnek olarak verilebilir.

Halim ve Ismail [15] en yakın komşu (EYK), genetik algoritma (GA), benzetimli tavlama, karınca kolonisi optimizasyonu (ACO) ve ağaç fizyolojisi optimizasyon algoritmalarının (TPO) GSP çözüm performanslarını karşılaştırmıştır. Antosiewicz vd. [16] 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. Chitty [17] büyük boyutlu GSP test problemlerinin çözümü için Karınca Kolonisi Optimizasyonu (ACO) yöntemini çözüm yaklaşımı olarak kullanmıştır. Karagül vd. [18] tarafından araç rotalaması problemleri için başlangıç çözüm yaklaşımı önerilmiştir. Szabo [19] tarafından GSP problemlerinin benzetimli tavlama çözümü için üç farklı başlangıç çözümü ile performans analizleri yapılmış ve benzetimli tavlama yaklaşımının hangi başlangıç çözümü ile daha iyi sonuçlar ürettiği analiz edilmiştir.

Bu çalışmada önerilen TPORT yöntemi, En Yakın Komşu (EYK) algoritması ile karşılaştırılmıştır. GSP için geliştirilmiş bir çok başlangıç çözümü üreten algoritma bilimsel yazında yer almaktadır. EYK ile TPORT uygulama ve çözüm yaklaşımı açısından benzerlik göstermektedir. Her ikisi de oldukça basit ve hızlıdır. Adil bir karşılaştırma sağlanmak istediği için özellikle çözüm süreleri açısından EYK seçiminin uygun olacağı düşünülmüştür. Her iki yöntemin ürettiği çözümler GSP başlangıç çözümleri olarak ele alınmış ve 2-Opt yaklaşımı ile geliştirilen çözümler üzerinden de karşılaştırmalar

yapılmıştır. Analizler iki grup test problemi üzerinde yapılmıştır.

2. Materyal ve Metot

Bu çalışmada önerilen yöntem TPORT algoritması ile elde edilen GSP başlangıç çözümleri 2-Opt algoritması ile geliştirilmiştir. Aynı problemler En Yakın Komşu algoritması kullanılarak da çözülmüş ve yine elde edilen başlangıç çözümler 2-Opt algoritması ile geliştirilmiştir. Önerilen yöntemin detaylarına geçmeden önce En Yakın Komşu ve 2-Opt algoritmaları kısaca açıklanacaktır. Daha sonra önerilen TPORT algoritması ortaya konulacaktır.

2.1. En Yakın Komşu Algoritması

En yakın komşu algoritması, GSP'yi çözmek için kullanılan ilk algoritmalarдан biridir. Gezgin satıcı, rastgele bir şehirden başlar ve tüm şehirler ziyaret edilinceye kadar bulunduğu şehirden sonra ona en yakın şehri ziyaret eder. Hızlı bir şekilde uygun bir Hamilton turu bulur ancak bulunan çözüm optimal değildir. EYK algoritmasının adımları aşağıda özetlenmiştir [20-23];

A1. Boş rota dizisi oluşturulur, başlangıç düğümü belirlenir. Belirlenen düğüm rota dizisine atanır.

A2. İlk düğümden diğer düğümlere olan uzaklıklar belirlenir.

A3. Belirlenen uzaklıklar arasında en kısa olanı seçilir ve ikinci düğüm olarak rota dizisine atanır.

A4. Tüm düğümler tamamlanana kadar Adım 2 ve Adım 3 tekrar edilir.

Atanan düğümler sırasıyla rotayı oluşturur.

2.2. 2-Opt Algoritması

Bu algoritma, farklı başlangıç çözüm algoritmaları ile elde edilen rotaların iyileştirilmesi için Croes tarafından 1958 yılında geliştirilmiştir[24]. Literatürde GSP üzerine yapılan pek çok çalışmada 2-Opt algoritması hem çözüm yöntemi hem de tur geliştirici bir sezgisel olarak kullanılmıştır. 2-Opt algoritmasının adımları aşağıda verilmiştir [20, 22, 25-27];

A1. Rastsal olarak rotada düğüm çiftleri belirlenir.

A2. Rota bozulmayacak şekilde, düğüm çiftlerinin bağlantıları değiştirilir.

A3. Yeni oluşan rota önceki rotaya göre daha kısa mesafeli ise düğüm çiftlerine karşı gelen kenarlar yerlerinde kalır, maliyet tasarrufu sağlanamamışsa düğüm çiftleri eski yerlerine alınır.

Takip eden bölümde önerilen çözüm yaklaşımının detayları sunulmuştur.

2.3. Önerilen Çözüm Yöntemi: TPORT

Önerilen çözüm yaklaşımı UP için başlangıç çözümleri üreten bir sezgisel baktır açısıyla üretilmiştir. Bu nedenle aynı zamanda UP için başlangıç çözüm üretme potansiyeline sahip olduğu ileri sürülebilir. EK-1'de TPORT yaklaşımının Matlab kodu açık bir şekilde verilmiştir. Ancak bu kodun bazı bileşenleri, veri dosyasının okunması, uzaklık matrisinin hesaplanması ve 2-Opt fonksiyonları okuyucu tarafından geliştirilebilir ve/veya web üzerinde yapılacak kısa bir araştırma ile En Yakın Komşu algoritması da dahil olmak üzere elde edilebilir.

TPORT algoritması GSP için başlangıç çözümleri üretmek üzere tasarlanmıştır. Bu yaklaşımın çıkış noktası GSP uzaklık matrisinin bir ulaşırma tablosu gibi ele alınarak, her bir satırının toplam değerleri bulunarak bu

toplamlara göre satırların normalizasyonu ile elde edilen bir matris üzerinden GSP turunun oluşturulması düşüncesidir. Bu yaklaşım bilimsel yazındaki GSP başlangıç çözüm yaklaşımlarından elde edilen çözüm matrisi ve matris üzerinden GSP turunun elde edilmesi süreci ile farklılık gösterir.

Önerilen yöntemin daha iyi anlaşılabilmesi için küçük bir problem üzerinden adımlar açıklanmaya çalışılacaktır. Yöntemin çözüm aşamalarını göstermek için Burkardt [28] tarafından hazırlanan 5 ve 15 düğümlü iki GSP kullanılmıştır. Beş düğümlü problem üzerinde çözüm aşamaları örneklelmıştır. Beş düğümlü örnek GSP probleminin uzaklık matrisi ve çözüm aşamaları Şekil 1'de gösterilmektedir. Onbeş düğümlü problemin ise sadece bilgisayar ile çözümü verilmiştir. Onbeş düğümlü GSP için xy koordinatları ve uzaklık matrisi Tablo 1'de sunulmaktadır. Bilgisayar ile elde edilen çözüm ise Tablo 2'te verilmiştir. 5 düğümlü GSP için çözüm adımları Şekil 1'den takip edilebilecek şekilde verilmiştir.

Uzaklık Matrisi						Satır Top						Satır Top						Adım-1: Çözüm Matrisi						Adım-2: Köşegen elemanlarını max yap							
1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5			
1	0	3	4	2	7	1	0	1.188	0.25	0.125	0.438	1	0	1.188	0.25	0.125	0.4375	1	0	1.1875	0.25	0.125	0.4375	1	0	1.1875	0.25	0.125	0.4375	1	
2	3	0	4	6	3	2	0	1.188	0	0.25	0.375	0.188	2	0	1.188	0	0.25	0.375	0.188	2	0	1.1875	0	0.25	0.375	0.1875	2				
3	4	4	0	5	8	3	4	0.19	0	0.5	0.8	0.381	3	0.19	0.19	0	0.238	0.381	3	0.1905	0.1905	0	0.2381	0.381	3	0.1905	0.1905	0	0.2381	0.381	3
4	2	6	5	0	6	4	2	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.1053	0.3158	0.2632	0	0.3158	4	0.1053	0.3158	0.2632	0	0.3158	4
5	7	3	8	6	0	5	7	0.292	0.125	0.333	0	0	5	0.292	0.125	0.333	0.25	0	5	0.2917	0.125	0.3333	0.25	0	5	0.2917	0.125	0.3333	0.25	0	5

Adım-3: 1. satır enk değeri bul.						Adım-4: Rotaya ekle, sütunu max yap						Adım-5: 4. satır konumlan, enk değeri bul						Adım-6: Rotaya ekle, sütunu max yap												
1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		
1	0.188	0.25	0.125	0.438		1	0.188	0.25	0.125	0.438		1	0.188	0.25	0.125	0.438		1	0.1875	0.25	0.125	0.4375		1	0.1875	0.25	0.125	0.4375		1
2	0.188	0.25	0.275	0.188		2	0.188	0.25	0.375	0.188		2	0.188	0.25	0.375	0.188		2	0.1875	0.25	0.375	0.1875		2	0.1875	0.25	0.375	0.1875		2
3	0.19	0.19	0	0.238	0.381	3	0.19	0.19	0	0.238	0.381	3	0.19	0.19	0	0.238	0.381	3	0.1905	0.1905	0	0.2381	0.381	3	0.1905	0.1905	0	0.2381	0.381	3
4	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.1053	0.3158	0.2632	0	0.3158	4	0.1053	0.3158	0.2632	0	0.3158	4
5	0.292	0.125	0.333	0.25	0	5	0.292	0.125	0.333	0	0	5	0.292	0.125	0.333	0.25	0	5	0.2917	0.125	0.3333	0.25	0	5	0.2917	0.125	0.3333	0.25	0	5

Rotat=[1]						Rotat=[4-1]						Rotat=[4-1-2]						Rotat=[4-1-2-1]						Rotat=[4-1-2-5]						Rotat=[4-1-2-5-3]						
1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		
1	0.188	0.25	0.125	0.438		1	0.188	0.25	0.125	0.438		1	0.188	0.25	0.125	0.438		1	0.1875	0.25	0.125	0.4375		1	0.1875	0.25	0.125	0.4375		1	0.1875	0.25	0.125	0.4375		1
2	0.188	0.25	0.275	0.188		2	0.188	0.25	0.375	0.188		2	0.188	0.25	0.375	0.188		2	0.1875	0.25	0.375	0.1875		2	0.1875	0.25	0.375	0.1875		2	0.1875	0.25	0.375	0.1875		2
3	0.19	0.19	0	0.238	0.381	3	0.19	0.19	0	0.238	0.381	3	0.19	0.19	0	0.238	0.381	3	0.1905	0.1905	0	0.2381	0.381	3	0.1905	0.1905	0	0.2381	0.381	3						
4	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.1053	0.3158	0.2632	0	0.3158	4	0.1053	0.3158	0.2632	0	0.3158	4						
5	0.1	0.125	0.333	0	0	5	0.1	0.125	0.333	0	0	5	0.1	0.125	0.333	0.25	0	5	0.1	0.125	0.3333	0.25	0	5	0.1	0.125	0.3333	0.25	0	5						

Adım-7: 1. satır konumlan, enk değeri bul						Adım-8: Rotaya ekle, sütunu max yap						Adım-9: 2. satır konumlan, enk değeri bul						Adım-10: Rotaya ekle, sütunu max yap												
1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		1	2	3	4	5		
1	0.188	0.25	0.125	0.438		1	0.188	0.25	0.125	0.438		1	0.188	0.25	0.125	0.438		1	0.1875	0.25	0.125	0.4375		1	0.1875	0.25	0.125	0.4375		1
2	0.188	0.25	0.275	0.188		2	0.188	0.25	0.375	0.188		2	0.188	0.25	0.375	0.188		2	0.1875	0.25	0.375	0.1875		2	0.1875	0.25	0.375	0.1875		2
3	0.19	0.19	0	0.238	0.381	3	0.19	0.19	0	0.238	0.381	3	0.19	0.19	0	0.238	0.381	3	0.1905	0.1905	0	0.2381	0.381	3	0.1905	0.1905	0	0.2381	0.381	3
4	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.105	0.316	0.263	0	0.316	4	0.1053	0.3158	0.2632	0	0.3158	4	0.1053	0.3158	0.2632	0	0.3158	4
5	0.1	0.125	0.333	0	0	5	0.1	0.125	0.333	0	0	5	0.1	0.125	0.333	0.25	0	5	0.1	0.125	0.3333	0.25	0	5	0.1	0.125	0.3333	0.25	0	5

Rotat=[1]						Rotat=[4-1]						Rotat=[4-1-2]						Rotat=[4-1-2-1]						Rotat=[4-1-2-5]	
-----------	--	--	--	--	--	-------------	--	--	--	--	--	---------------	--	--	--	--	--	-----------------	--	--	--	--	--	-----------------	--

Adım 3: Çözüme başlamak için Çözüm matrisinin 1. satırına bakılır ve satırdaki en küçük değerli eleman bulunur. Şekil 1, Adım 3 incelendiğinde 1. Satırdaki enk eleman 4. Sütuna karşılık gelen 0,125 değeridir.

Adım 4: Adım 3'te bulunan en küçük değerli eleman 4. sütunda yer aldığı için 4. sütun tamamı 1 değeri ile doldurulur. Rota dizisine [4] eklenir.

Adım 5: 4. satıra konumlanılır ve 4. satırdaki en küçük değerli eleman aranır. Örnek için en küçük değer 0,105 ile 1. sütuna denk düşer.

Adım 6: Rotaya 1 eklenir ve 1. Sütun 1 ile doldurulur. Böylece Rota dizisi [4-1] olur.

Adım 7: 1. Satıra konumlanılır ve en küçük değerli eleman bulunur, örnek için bu değer 0,188 olur. Bu değer 2 sütununa karşılık gelir.

Adım 8: Rota dizisine 2 eklenir ve böylece rota dizisi [4-1-2] olur. Sütun 2 1 değerleri ile doldurulur.

Adım 9: 2. Satıra konumlanılır ve en küçük değerli eleman bulunur. Örnek için bu değer 0,188'dir ve sütun 5'e karşılık gelir.

Adım 10: Rota dizisine 5 eklenir ve böylece [4-1-2-5] elde edilir.

Adım 11: 5. satıra konumlanılır ve en küçük değerli eleman bulunur. Örnek için bu değer 0,333 olur. Bu sütun 3'e karşılık gelir.

Adım 12: Rota dizisine 3 eklenir böylece [4-1-2-5-3] dizisi elde edilir. Rotaya eklenecek başka düğüm kalmadığı için algoritma sonlanır. Böylece 5 düğümlü GSP için elde edilen çözüm rotası, Şekil 1 ve TPORT algoritma adımları üzerinden [4-1-2-5-3] ve maliyet 21 olarak bulunmuştur. Problemin optimal çözüm rotası [1-3-2-5-4] ve maliyet 19 olarak verilmiştir.

Tablo 1 incelendiğinde onbeş düğümlü GSP için analitik xy koordinatları ve hesaplanan uzaklık matrisi verilmiştir.

Tablo 1. Onbeş düğümlü örnek problem koordinatlar ve uzaklık matrisi

Koordinatlar		Uzaklık Matrisi														
x	y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-0,00000004	0,00000004	0	29	82	46	68	52	72	42	51	55	29	74	23	72	45
-28,87328622	-0,00000087	29	0	55	46	42	43	43	23	23	31	41	51	10	52	20
-79,29157917	21,40333076	82	55	0	68	46	55	23	43	41	29	79	20	65	31	51
-14,65773817	43,38954970	46	46	68	0	82	14	72	31	62	42	21	51	51	43	65
-64,74726053	-21,89817134	68	42	46	82	0	74	23	52	20	46	82	58	46	66	23
-29,05846931	43,21672877	52	43	55	14	74	0	61	23	55	31	32	37	51	29	59
-72,07853197	-0,18158346	72	43	23	72	23	61	0	42	23	31	78	37	51	46	32
-36,03664897	21,61354829	42	23	43	31	52	23	42	0	32	15	37	32	32	31	37
-50,48083829	-7,37447224	51	23	41	62	20	55	23	32	0	29	62	46	29	51	10
-50,58590268	21,58819661	55	31	29	42	46	31	31	15	29	0	51	20	41	23	37
-0,13582038	28,72928968	29	41	79	21	82	32	78	37	62	51	0	65	42	59	61
-65,08656384	36,06246931	74	51	20	51	58	37	37	32	46	20	65	0	61	10	55
-21,49832607	-7,31941595	23	10	65	51	46	51	51	32	29	41	42	61	0	62	23
-57,56872447	43,25055624	72	52	31	43	66	29	46	31	51	23	59	10	62	0	60
-43,07002585	-14,55483969	45	20	51	65	23	59	32	37	10	37	61	55	23	60	0

Tablo 2. Onbeş düğümlü örnek problem çözüm matrisi ve GSP rota ve maliyeti

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0,04	0,11	0,06	0,09	0,07	0,1	0,06	0,07	0,07	0,04	0,1	0,03	0,1	0,06
2	0,06	1	0,11	0,09	0,08	0,08	0,08	0,05	0,05	0,06	0,08	0,1	0,02	0,1	0,04
3	0,12	0,08	1	0,1	0,07	0,08	0,03	0,06	0,06	0,04	0,11	0,03	0,09	0,05	0,07
4	0,07	0,07	0,1	1	0,12	0,02	0,1	0,04	0,09	0,06	0,03	0,07	0,07	0,06	0,09
5	0,09	0,06	0,06	0,11	1	0,1	0,03	0,07	0,03	0,06	0,11	0,08	0,06	0,09	0,03
6	0,08	0,07	0,09	0,02	0,12	1	0,1	0,04	0,09	0,05	0,05	0,06	0,08	0,05	0,1
7	0,11	0,07	0,04	0,11	0,04	0,1	1	0,07	0,04	0,05	0,12	0,06	0,08	0,07	0,05
8	0,09	0,05	0,09	0,07	0,11	0,05	0,09	1	0,07	0,03	0,08	0,07	0,07	0,07	0,08
9	0,1	0,04	0,08	0,12	0,04	0,1	0,04	0,06	1	0,05	0,12	0,09	0,05	0,1	0,02
10	0,11	0,06	0,06	0,09	0,1	0,06	0,06	0,03	0,06	1	0,11	0,04	0,09	0,05	0,08
11	0,04	0,06	0,11	0,03	0,11	0,04	0,11	0,05	0,08	0,07	1	0,09	0,06	0,08	0,08
12	0,12	0,08	0,03	0,08	0,09	0,06	0,06	0,05	0,07	0,03	0,11	1	0,1	0,02	0,09
13	0,04	0,02	0,11	0,09	0,08	0,09	0,09	0,05	0,05	0,07	0,07	0,1	1	0,11	0,04
14	0,11	0,08	0,05	0,07	0,1	0,05	0,07	0,05	0,08	0,04	0,09	0,02	0,1	1	0,09
15	0,08	0,03	0,09	0,11	0,04	0,1	0,06	0,06	0,02	0,06	0,11	0,1	0,04	0,1	1

Çözüm Rotası / Çözüm Maliyeti: 284														
13	2	15	9	5	7	3	12	14	10	8	6	4	11	1
Optimal Rota / Optimal Maliyet: 284														
1	13	2	15	9	5	7	3	12	14	10	8	6	4	11

Tablo 2 incelendiğinde, TPORt yaklaşımı için kullanılan çözüm matrisi verilmiştir. Tablo 2'ün devamında TPORt kodu ile elde edilen çözüm rotası [13-2-15-9-5-7-3-12-14-10-8-6-4-11-1] ve buna ait hesaplanan maliyet 284'tür. Aynı şekilde problemin bulunduğu kaynakta [28] optimal çözüm rotası [1-13-2-15-9-5-7-3-12-14-10-8-6-4-11] ve optimal maliyet 284 olarak verilmiştir. Beş düğümlü ve onbeş düğümlü iki problem incelendiğinde; TPORt yaklaşımında ilk problem için optimalden sapma %10,52 gerçekleşirken, onbeş düğümlü GSP için optimal çözüm bulunmuştur. Önerilen TPORt yaklaşımı için Matlab kodu verilmiş, aynı zamanda hesap tablosu ile beş düğümlü bir GSP ile yaklaşımın tüm detayları adım adım anlatılmıştır. Ayrıca onbeş düğümlü bir GSP ile bilgisayar ile TPORt yaklaşımının çözümü verilmiştir. Bu noktadan sonra önerilen çözüm yaklaşımının test problemleri üzerindeki performansının araştırmasına geçilebilir.

3. Bulgular

TPORt yaklaşımının algoritması ve örnek problem uygulamaları bir önceki bölümde verilmiştir. Önerilen yöntemin analizi için iki ayrı test problemi grubu kullanılmıştır. İlk grup problemler Antosiewicz vd. [16] makalesinden kişisel iletişim ile alınmıştır. Bu problem

grubundaki 6 problem orijinal olarak makalenin yazarları tarafından üretilmiş son iki problem ise TSPLIB [29] kaynağından almıştır. İkinci grup problemler doğrudan TSPLIB'den seçilmiştir. Birinci grup problemler küçük boyutlu problemler olarak değerlendirilebilir. İkinci grupta yer alan problemler ise küçük, orta ve büyük problemler olarak sınıflandırılmıştır.

Yapılan tüm analizler Matlab 2016b paket programında Windows 10 işletim sistemi yüklü Intel Core i7-4800MQ, 2.70 GHz, 16 MB dizüstü bilgisayarda tek çekirdek ile gerçekleştirılmıştır.

TPORt yaklaşımının analizi için bu iki grup problem iki aşamada analiz edilmiştir. İlk aşamada önerilen yöntemin ürettiği çözümler başlangıç GSP çözümleri olarak değerlendirilmiştir. Bu çözümler 2-Opt algoritması ile geliştirilmiştir. Aynı test problemleri En Yakın Komşu algoritması (EYK) ile çözülmüş ve elde edilen çözümler 2-Opt algoritması ile geliştirilmiştir. Önerilen yöntem ve EYK algoritması uzaklık maliyeti ve benzetim süresi performansları açısından karşılaştırılarak analiz edilmiştir. Tablo 3'te A, B, C, D, F, G, H ve Opt olarak kodlanan ifadelerin açıklamaları verilmiştir.

Tablo 3. Analiz tablolarında kullanılan kısaltmalar

Kod	Açıklama
A	TPORT Çözümü
B	TPORT+2-Opt Çözümü
C	EYK Çözümü
D	EYK+2-Opt Çözümü
E	TPORT Süre (sn)
F	TPORT sonrası 2-Opt Süre (sn)
G	EYK Süre (sn)
H	EYK sonrası 2-Opt Süre (sn)
Opt	Optimal Çözüm

Tablo 4 incelendiğinde birinci grup yani küçük boyutlu problemler için optimal, A, B, C ve D çözümleri yer almaktadır.

Tablo 4. Küçük boyutlu problemlere ilişkin çözüm performansları

No	P. Adı	Opt	A	B	C	D
1	20 (a)	397	518	402	407	404
2	20 (b)	367	435	367	380	367
3	50 (a)	560	802	598	683	584
4	50 (b)	571	706	590	614	587
5	80 (a)	709	910	735	801	722
6	80 (b)	687	861	703	770	697
7	att48	333	408	345	373	350
8	eil76	538	660	565	608	554

Tablo 5 incelendiğinde A çözüm yaklaşımının saniye olarak değerleri E sütununda gösterilmiştir. Aynı şekilde diğer sütunlarda da diğer yaklaşımın çözüm süreleri saniye olarak verilmiştir.

Tablo 5. Küçük problemler için çözüm süreleri

No	P. Adı	E	F	G	H	Z1=E+F	Z2=G+H
1	20 (a)	37,03x10 ⁻⁴	6,53 x10 ⁻⁴	20,76 x10 ⁻⁴	36,77 x10 ⁻⁴	43,56x10 ⁻⁴	57,53x10 ⁻⁴
2	20 (b)	35,02 x10 ⁻⁴	5,96 x10 ⁻⁴	15,06 x10 ⁻⁴	16,84 x10 ⁻⁴	40,98 x10 ⁻⁴	31,90 x10 ⁻⁴
3	50 (a)	37,03 x10 ⁻⁴	9,75 x10 ⁻⁴	71,12 x10 ⁻⁴	88,99 x10 ⁻⁴	46,78 x10 ⁻⁴	160,11 x10 ⁻⁴
4	50 (b)	37,82 x10 ⁻⁴	8,55 x10 ⁻⁴	79,01 x10 ⁻⁴	25,62 x10 ⁻⁴	46,37 x10 ⁻⁴	104,63 x10 ⁻⁴
5	80 (a)	43,99 x10 ⁻⁴	11,35 x10 ⁻⁴	196,21 x10 ⁻⁴	12,83 x10 ⁻⁴	55,34 x10 ⁻⁴	209,04 x10 ⁻⁴
6	80 (b)	50,48 x10 ⁻⁴	12,10 x10 ⁻⁴	196,59 x10 ⁻⁴	56,76 x10 ⁻⁴	62,58 x10 ⁻⁴	253,35 x10 ⁻⁴
7	att48	38,12 x10 ⁻⁴	9,99 x10 ⁻⁴	97,54 x10 ⁻⁴	11,62 x10 ⁻⁴	48,11 x10 ⁻⁴	109,16 x10 ⁻⁴
8	eil76	41,23 x10 ⁻⁴	11,09 x10 ⁻⁴	176,29 x10 ⁻⁴	11,94 x10 ⁻⁴	52,32 x10 ⁻⁴	188,23 x10 ⁻⁴

Tablo 6'de küçük test problemleri için görelî karşılaştırmalar tablosu verilmiştir. Bu tabloda Opt sütununda optimal çözüm ile diğer yaklaşımın görelî karşılaştırmaları için 1 her problem için referans değer olarak kullanılmıştır. Tablo 6, denklem 1 ile hesaplanmıştır.

$$\left(\frac{\text{Çözüm}}{\text{Optimal Çözüm}} \right) = \frac{\text{Çözüm Değeri}}{\text{Optimal Değer}} \times 100 \quad (1)$$

Çözüm : A, B, C, D

Bu durumda optimal çözüm 1 ile ifade edilirse tablodaki oranlardan 1'den ne kadar uzak ise çözüm performansı o kadar kötüye gidiyor demektir. Bu bakış açısı ile Tablo 6 ortalama satırına bakıldığından optimalden TPорт yaklaşımı %27 ortalama ile EYK algoritması %11 sapma ile çözüm bulmuştur. Ancak her iki yöntemde 2-Opt uygulanarak elde edilen

çözüm değerlerine bakıldığından her ikisi de %3 sapma ile sonuca ulaşmıştır. Bu değerler de TPорт yaklaşımının iyi bir başlangıç çözüm olmaya yönelik ilk göstergeleri olduğu ileri sürülebilir.

Tablo 6. Küçük problemler için çözüm performanslarının göreli karşılaştırması

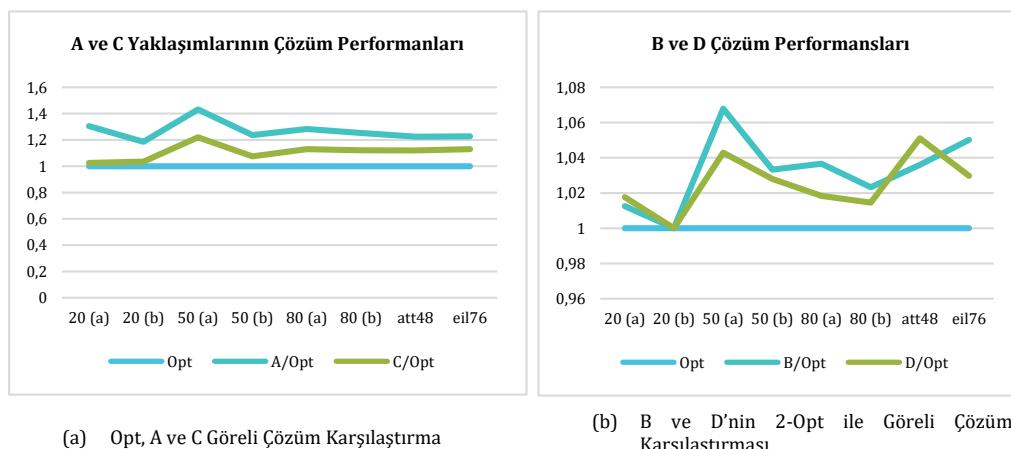
No	P. Adı	Opt	A/Opt	B/Opt	C/Opt	D/Opt
1	20 (a)	1	1,30	1,01	1,03	1,02
2	20 (b)	1	1,19	1,00	1,04	1,00
3	50 (a)	1	1,43	1,07	1,22	1,04
4	50 (b)	1	1,24	1,03	1,08	1,03
5	80 (a)	1	1,28	1,04	1,13	1,02
6	80 (b)	1	1,25	1,02	1,12	1,01
7	att48	1	1,23	1,04	1,12	1,05
8	eil76	1	1,23	1,05	1,13	1,03
	Ortalama	1	1,27	1,03	1,11	1,03

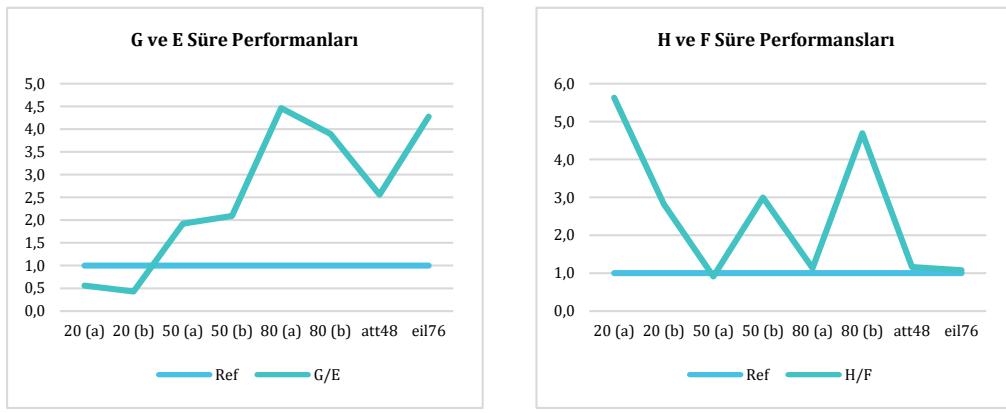
Aynı bakış açısı ile Tablo 7 incelendiğinde referans değer olarak 1 alınmıştır. 1'den küçük değerler EYK ve EYK'ya uygulanan 2-Opt sürelerinin daha etkin olduğunu gösterir. Ancak 1'den büyük olan değerler TPORT çözümlerinin EYK'dan kaç kat hızlı olduğunu gösterir. Bu bağlamda G/E sütunu için ortalama değerlere bakıldığına TPORT çözüm süreleri 2,52 kat EYK'dan hızlı sonuca gitmektedir. Aynı şekilde H/F sütunu incelendiğinde başlangıç çözümleri TPORT ve EYK için 2-Opt uygulama süreleri karşılaştırıldığında 2,5 kat ortalama olarak TPORT çözümleri daha hızlı geliştirilebilmektedir.

Tablo 7. Küçük problemler için sürelerin göreli karşılaştırması

No	P. Adı	Ref	G/E	H/F
1	20 (a)	1,0	0,6	5,6
2	20 (b)	1,0	0,4	2,8
3	50 (a)	1,0	1,9	0,9
4	50 (b)	1,0	2,1	3,0
5	80 (a)	1,0	4,5	1,1
6	80 (b)	1,0	3,9	4,7
7	att48	1,0	2,6	1,2
8	eil76	1,0	4,3	1,1
	Ortalama	1,0	2,52	2,55

Tablo 4, 5, 6 ve 7 için verilen tartışmaların grafikleri Şekil 2'de a, b, c ve d şeklinde gösterilmiştir.





(c) G ve E Çözüm Yaklaşımlarının Görelî Süre Karşılaştırması

(d) H ve F yaklaşımlarının 2-Opt ile Görelî Süre Karşılaştırması

Şekil 2. Küçük problemler için performans grafikleri

Tablo 8'da TSPLIB'den seçilmiş olan küçük, orta ve büyük problemler için çözüm değerleri gösterilmiştir.

Tablo 8. TSPLIB test problemlerinin çözümleri

No	P. Adı	Opt	A	B	C	D
1	eil51	426	563	440	482	434
2	berlin52	7542	8920	7842	8181	7711
3	st70	675	852	690	796	727
4	pr76	108159	153150	114283	130921	113404
5	rat99	1211	1566	1317	1437	1259
6	kroA100	21282	27015	22063	24698	21389
7	eil101	629	808	640	746	639
8	ch130	6110	7658	6435	7129	6414
9	ch150	6528	8150	6620	7113	6606
10	rat195	2323	2752	2392	2612	2393
11	d198	15780	18240	16036	17620	16033
12	a280	2579	3256	2740	3008	2675
13	rd400	15281	19176	15878	18431	15840
14	pcb442	50778	61745	53274	58950	52916
15	rl1323	270199	332001	284015	312845	280799

Tablo 9'da ikinci grup test problemleri için çözüm süreleri verilmiştir ve tüm süreler saniye olarak ifade edilmiştir. Tablo 9'da Z1, TPORT ve 2-Opt çözüm sürelerinin birlikte ölçümünü, Z2

ise EYK ve 2-Opt sürelerinin birlikte ölçümünü göstermektedir.

Tablo 9. TSPLIB test problemlerinin çözüm süreleri

No	P. Adı	E ($\times 10^{-4}$)	F ($\times 10^{-4}$)	G ($\times 10^{-4}$)	H ($\times 10^{-4}$)	(Z1=E+F) ($\times 10^{-4}$)	(Z2=G+H) ($\times 10^{-4}$)
1	eil51	40,23	8,49	76,63	8,00	48,72	84,63
2	berlin52	38,30	8,14	79,96	7,52	46,44	87,48
3	st70	43,52	11,99	151,45	13,35	55,51	164,8
4	pr76	42,10	12,39	177,12	13,13	54,49	190,25
5	rat99	43,56	15,25	314,98	19,98	58,81	334,96
6	kroA100	40,39	16,20	286,07	14,84	56,59	300,91
7	eil101	47,75	16,55	317,93	20,35	64,30	338,28
8	ch130	43,51	25,50	459,87	22,46	69,01	482,33
9	ch150	48,05	28,65	624,67	26,62	76,7	651,29
10	rat195	49,65	43,08	1095,20	40,86	92,73 ⁴	1136,06
11	d198	49,86	67,94	1068,74	51,85	117,8	1120,59
12	a280	60,22	129,53	2876,58	179,14	189,75	3055,72
13	rd400	108,47	663,43	10257,92	493,11	771,90	10751,03
14	pcb442	103,28	459,09	12457,59	480,06	562,37	12937,65
15	rl1323	1139,07	22065,85	512582,86	19366,55	23204,92	531949,41

Tablo 10'de göreli çözüm performansları verilmiştir. Burada her bir problem için diğerine göre üstün olan çözüm yaklaşımının değeri koyu renkli olarak işaretlenmiştir. Tablo 10'deki son satırda yer alan ortalamalar, incelendiğinde TPORT yaklaşımı %26, EYK sezgiseli ise %16 oranında optimalden uzak çözümlere erişmiştir. Ancak her iki yönteme 2-Opt uygulanarak elde

edilen çözümlerde sırasıyla %4 ve %3 sapmalar elde edilmiştir. Burada EYK+2-Opt yaklaşımının TPORT+2-Opt yaklaşımına göre ortalamada %1 daha başarılı olduğu görülmektedir. Ancak bu fark çözüm süreleri gözönüne alındığında TPORT+2-Opt yaklaşımın rekabetçi bir çözüm yaklaşımı olduğu iddiasını çürütmektedir.

Tablo 10. TSPLIB test problemlerinin göreli çözüm oranları

No	P. Adı	Opt	A	C	B	D
1	eil51	1,00	1,32	1,13	1,03	1,02
2	berlin52	1,00	1,18	1,08	1,04	1,02
3	st70	1,00	1,26	1,18	1,02	1,08
4	pr76	1,00	1,42	1,21	1,06	1,05
5	rat99	1,00	1,29	1,19	1,09	1,04
6	kroA100	1,00	1,27	1,16	1,04	1,01
7	eil101	1,00	1,28	1,19	1,02	1,02
8	ch130	1,00	1,25	1,17	1,05	1,05
9	ch150	1,00	1,25	1,09	1,01	1,01
10	rat195	1,00	1,18	1,12	1,03	1,03
11	d198	1,00	1,16	1,12	1,02	1,02
12	a280	1,00	1,26	1,17	1,06	1,04
13	rd400	1,00	1,25	1,21	1,04	1,04
14	pcb442	1,00	1,22	1,16	1,05	1,04
15	rl1323	1,00	1,23	1,16	1,05	1,04
Ortalama		1,00	1,26	1,16	1,04	1,03

Tablo 11 incelendiğinde iki yöntem için göreli karşılaştırmalar verilmiştir. Burada G/E sütununun ortalaması incelendiğinde TPорт yaklaşımı EYK yaklaşımından ortalama olarak 54 kat daha hızlı çözüme gittiği gözlenmektedir. Bu problemler için TPорт yaklaşımına uygulanan 2-Opt ise EYK yaklaşımına uygulanan

2-Opt çözüm süresinden ortalama olarak fark oluşturmamıştır. Son sütun olan Z2/Z1 sütunu önerilen ve referans alınan yöntemlerin 2-Opt rota geliştirme süreleri dahil olmak üzere toplam zaman açısından karşılaştırıldığında ortalama olarak önerilen yaklaşımın 9,30 kat daha hızlı olduğu görülmektedir.

Tablo 11. TSPLIB test problemlerinin göreli çözüm süreleri oranları

No	P. Adı	Opt	G/E	H/F	Z2/Z1
1	eil51	1,00	1,90	0,94	1,74
2	berlin52	1,00	2,09	0,92	1,88
3	st70	1,00	3,48	1,11	2,97
4	pr76	1,00	4,21	1,06	3,49
5	rat99	1,00	7,23	1,31	5,70
6	kroA100	1,00	7,08	0,92	5,32
7	eil101	1,00	6,66	1,23	5,26
8	ch130	1,00	10,57	0,88	6,99
9	ch150	1,00	13,00	0,93	8,49
10	rat195	1,00	22,06	0,95	12,25
11	d198	1,00	21,44	0,76	9,51
12	a280	1,00	47,77	1,38	16,10
13	rd400	1,00	94,57	0,74	13,93
14	pcb442	1,00	120,63	1,05	23,01
15	rl1323	1,00	450,00	0,88	22,92
Ortalama		1,00	54,18	1,00	9,30

Analiz sonuçları bir bütün olarak değerlendirildiğinde hem küçük hem de büyük problemler için EYK+2-Opt yaklaşımının rekabetçi bir yaklaşım olduğu değerlendirilebilir. Bununla birlikte çözüm süreleri açısından yapılan karşılaştırmalarda TPорт yaklaşımının EYK algoritmasından üstün olduğu; EYK algoritması çözüm süresi üstel bir eğilim izlerken TPорт yaklaşımının çözüm süresinin göreli olarak çok az bir değişim gösterdiği gözlenmektedir.

4. Tartışma ve Sonuç

Bu çalışmada GSP problemi için TPорт adı verilen yapısal bir başlangıç çözümü oluşturma yöntemi önerilmiştir. TPорт başlangıç çözümleri, EYK algoritması başlangıç çözümleri ile iki grup problem üzerinden uzaklık maliyeti ve benzetim süresi performansları açısından karşılaştırılmıştır. Ayrıca TPорт ve EYK algoritması ile elde edilen GSP çözümleri, 2-Opt ile geliştirilmiş ve tekrar uzaklık maliyeti ve benzetim süresi performansları açısından karşılaştırılmıştır. Yapılan tüm analizler TPорт yaklaşımının uzaklık maliyeti performansı açısından rekabetçi çözümler ürettiğini göstermektedir. Bununla birlikte, benzetim süresi performansları üzerinden yapılan analizler TPорт yaklaşımının EYK algoritmasına göre üstünlüğünü ortaya koymaktadır.

GSP için iyi bir çözüm algoritmasının ne olduğunu yönelik açık bir soru, hızlı çözüm üreten, çok sayıda tekrar çalıştırılmaya izin veren basit bir iyileştirme yöntemi uygulanmanın mı yoksa üretilen çözümü önemli ölçüde iyileştiren karmaşık ve uzun süren bir yöntemi uygulanmanın mı daha iyi olduğu hakkındadır. Bu çalışmada önerilen TPорт benzeri basit bir algoritma büyük ölçüde ilk çözüme

bağlıdır, ancak EYK algoritması gibi daha karmaşık yöntemler çok daha fazla çalışma süresi alacağı için yalnızca birkaç kez uygulanabilir, böylece ele alınan çözüm uzayının örnekleşmesi azalmaktadır.

Bu çalışmada hızlı çözüm üreten, çok sayıda tekrar çalıştırırmaya izin veren basit bir iyileştirme yöntemi önerilerek bu alanda rekabetçi bir algoritmanın literatüre kazandırılması hedeflenmiştir. Önerilen yapısal yaklaşımın, basit ve hızlı çözüm yeteneği sayesinde yeni yapısal çözüm yaklaşımları, sezgiseller ve meta-sezgiseller ile birlikte kullanılması veya GSP çözümü dışında farklı optimizasyon alanlarına uygulanması ileriye dönük çalışma konuları olarak ele alınabilir. Farklı uygulama alanları GSP'nin temel problem olarak alındığı, çizelgeleme problemleri, araç rotalama problemleri, insansız hava araçlarının rotalanması, askeri alanlardaki bazı problemlerin çözümleri, robotların rotalama algoritmalarının geliştirilmesi gibi alanlar ele alınabilir.

Kaynakça

- [1] Kuhn, H.W. 1955. The Hungarian Method for the Assignment Problem, Naval Research Logistics Quarterly, Cilt. 2, Sayı. 1-2, s. 83-97.
- [2] Munkres, J. 1957. Algorithms for the Assignment and Transportation Problems, Journal of the Society for Industrial and Applied Mathematics, Cilt. 5, Sayı. 1, s. 32-38.
- [3] Burkard, R.E. 1979. Travelling Salesman and Assignment Problems: A Survey, Annals of Discrete Mathematics, Cilt. 4, s. 193-215.
- [4] Winston, W.L. 2003. Operations Research: Applications and Algorithms. 4th edition. Cengage Learning.
- [5] Dantzig, G.B., Thapa, M.N. 1997. Linear Programming 1: Introduction. Springer-Verlag New York, USA.
- [6] Ratliff, H.D., Rosenthal, A.S. 1983. Order Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. Operations Research, Cilt. 31, Sayı. 3, s. 507-521.
- [7] Zhao, F., Li, S., Sun, J., Mei, D. 2009. Genetic Algorithm for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem. Computers & Industrial Engineering, Cilt. 56, Sayı. 4, s. 1642-1648.
- [8] Joines, A., Kay, M.G., Karabacak, M.F., Karagül, K., Tokat, S. 2017. Performance analysis of Genetic Algorithm Optimization Toolbox via Traveling Salesperson Problem. ss. 213-221. Sayers, W. ed. Contemporary Issues in Social Sciences and Humanities, UK, AGP Research, London.
- [9] Sahin, Y., Karagül, K. 2019. Solving Travelling Salesman Problem Using Hybrid Fluid Genetic Algorithm (HFGA). Pamukkale University Journal of Engineering Sciences, Ahead of Print: PAJES-81084 | DOI: 10.5505/pajes.2018.81084.
- [10] Karagul, K., Aydemir, E., Tokat, S. 2016. Using 2-Opt Based Evolution Strategy for Travelling Salesman Problem. An International Journal of Optimization and Control: Theories & Applications (IJOCTA), Cilt. 6, Sayı. 2, s. 103-113.
- [11] Dorigo, M., Gambardella, L.M. 1997. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, Cilt. 1, Sayı. 1, s. 53-66.
- [12] Mavrovouniotis, M., Yang, S. 2013. Ant Colony Optimization with Immigrants Schemes for the Dynamic Travelling Salesman Problem with Traffic Factors. Applied Soft Computing, Cilt. 13, Sayı. 10, s. 4023-4037.
- [13] Gendreau, M., Laporte, G., Semet, F. 1998. A Tabu Search Heuristic for the Undirected Selective Travelling Salesman Problem. European Journal of Operational Research, Cilt. 106, Sayı. 2-3, s. 539-545.
- [14] Malek, M., Guruswamy, M., Pandya, M., Owens, H. 1989. Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem. Annals of Operations Research, Cilt. 21, Sayı. 1, s. 59-84.
- [15] Halim, A.H., Ismail, I. 2017. Combinatorial Optimization: Comparison of Heuristic Algorithms in Travelling Salesman Problem. Archives of Computational Methods in Engineering, s. 1-14. DOI: 10.1002/net.3230200605.
- [16] Antosiewicz, M., Koloch, G., Kamiński, B. 2013. 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, Cilt. 7, Sayı. 1, s. 46-55.
- [17] Chitty, D. M. 2017. Applying ACO To Large Scale TSP Instances. UK Workshop on Computational Intelligence, s. 104-118. Springer, Cham. arXiv:1709.03187. DOI: 10.1007/978-3-319-66939-7_9.
- [18] Karagul, K., Kay, M. G., Tokat, S. 2018. A New Method for Generating Initial Solutions of Capacitated Vehicle Routing Problems. Gazi University Journal of Science, Cilt. 31, Sayı. 2, s. 489-513.
- [19] Szabo, J. 2016. Comparison of Methods for Generating Initial Solution for Simulated Annealing. Central European Researchers Journal, Cilt. 2, Sayı. 1, s. 37-41.
- [20] Şahin, Y., Kulak, O. 2013. Depo Operasyonlarının Planlanması İçin Genetik Algoritma Esası Modeller. Uluslararası Alanya İşletme Fakültesi Dergisi, Cilt. 5, Sayı. 3, s. 141-153.
- [21] Kızılataş, G., Nuriyeva, F. 2013. On the Nearest Neighbor Algorithms for the Traveling Salesman Problem. In: Nagamalai D., Kumar A., Annamalai A. (eds) Advances in Computational Science, Engineering and Information Technology. Advances in Intelligent Systems and Computing, vol 225. Springer, Heidelberg.
- [22] Kay, M. 2016. MATLOG: Logistics Engineering Using Matlab. Mühendislik Bilimleri ve Tasarım Dergisi, Cilt. 4, Sayı. 1, s. 15-20. Retrieved from <http://dergipark.gov.tr/jesd/issue/20875/224091>.
- [23] Matworks, File Exchange, <https://www.mathworks.com/matlabcentral/fileexchange/25542-nearest-neighbor-algorithm-for-the-travelling-salesman-problem>, (Erişim Tarihi: 18.02.2019).
- [24] Croes, G.A. 1958. A Method for Solving Traveling-Salesman Problems. Operations Research, Cilt. 6, Sayı. 6, s. 791-812.

- [25] Eryavuz, M., Gencer, C. 2001. Araç Rotalama Problemine Ait Bir Uygulama. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt. 6, Sayı. 1, s. 139-155. . Retrieved from <http://dergipark.gov.tr/sduiibfd/issue/20850/223589>
- [26] Kuang, E. 2012. A 2-opt-based Heuristic for the Hierarchical Traveling Salesman Problem. <http://honors.cs.umd.edu/reports/kuang.pdf>, (Erişim Tarihi: 18.02.2019).
- [27] Sathyan, A., Boone, N., Cohen, K. 2015. Comparison of Approximate Approaches to Solving the Travelling Salesman Problem and its Application to UAV Swarming. International Journal of Unmanned UAV Swarming Systems Engineering (IJUSEng), Cilt. 3, Sayı. 1, s. 1-16.
- [28] Burkardt, J. 2019. Data for the Traveling Salesperson Problem. <https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html>, (Erişim Tarihi: 18.02.2019).
- [29] Universität Heidelberg. "Index of /software/TSPLIB95 /tsp". <http://comopt.ifb.uni-heidelberg.de/software/TSPLIB95/tsp/> (Erişim Tarihi: 18.11.2018).

EK- 1. Önerilen yöntem TPORT için Matlab kodu

tport.m

```
clear; clc;
%GSP Problemi Verileri: eil51.tsp
xy=dosyaOku(eil51.tsp)
%TSP veri dosyasını oku, GSP xy koordinatlarını al
D=UzaklıkMatrisiHesapla(xy);
%Uzaklık matrisini hesaplayan fonksiyon
[m,n]=size(D); xD=D;
xD(xD==Inf)=0;
rowSum=zeros(1,m);
for i=1:m
    rowSum(i)=sum(xD(i,:));
% Yeni matrisin satır toplamı hesabı
end
M=zeros(m,m);
for i=1:m
    M(i,:)=xD(i,:)./rowSum(i);
%Çözüm matrisinin oluşturulması
end
M(M==0)=1; Z=M;
%Hazırlanan matristen GSP çözümünün oluşturulması
rotaMx=zeros(1,m);
t=1; ss=1;
while t<=m
    [~,bx]=min(Z(ss,:));
    rotaMx(t)=bx;
%Önerilen yöntemin GSP çözümü: rotaMx
Z(:,bx)=1; ss=bx; t=t+1;
end
MalivetTSP=Malivet(rotaMx,D);
%Önerilen yöntemin GSP maliyeti
[2OptRota,2OptMalivet]=2Opt(rotaMx,D);
% Önerilen yöntemin ürettiği rotanın 2-Opt ile iyileştirilmesi
%Geliştirilen Rota: 2OptRota,
%GSP'nin Yeni Maliyet: 2OptMalivet
```
