

İNSANSIZ HAVA ARACI İLE PAKET DAĞITIMINDA GEZGİN SATICI PROBLEMİNİN GENETİK VE PARÇACIK SÜRÜ OPTİMİZASYON ALGORİTMALARI İLE ÇÖZÜMÜ

Enes Buğra ACAR^{1*}, Cumali KARABEY², Bayram KÖSE³

^{1,2,3}İzmir Bakırçay Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği, İzmir, 35660, Türkiye
Geliş Tarihi/Received Date: 09.02.2023 Kabul Tarihi/Accepted Date: 31.08.2023 DOI: 10.54365/adyumbd.1249391

ÖZET

Bu makalede kargo alanında kullanılmak üzere insansız hava aracının (İHA), polinom olmayan (Non-polynomial) zor problemler arasındaki gezgin satıcı problemi (GSP) Genetik Algoritma (GA) ve Parçacık Sürü Optimizasyonu (PSO) Algoritması ile çözülmüştür. Bu çalışma, kargo lojistiği ve taşımacılığında yeni yöntemler arayan araştırmacılar ve endüstri uzmanları için önemli bir değer taşımaktadır. İHA tabanlı paket dağıtımının etkin bir şekilde optimize edilmesi, lojistik sektörünün verimliliğini artırma potansiyeli taşımaktadır. Öncelikle veri seti olarak TSPLIB'den EIL51 verileri kullanılmıştır. Sonrasında gerçek hayattan kargo teslimatına örnek olarak sıhhi paket taşınması için İzmir ili Menemen ilçesindeki eczanelerin lokasyonları alınmıştır. Bulunan rota sonuçları ile İHA paket teslimatında optimal yol haritası çıkartılmıştır. Sonuçlara bakıldığında, GA'nın algoritma içeriğinden dolayı daha uzun sürdüğü ve GA'nın PSO'ya göre daha optimize edilmiş bir rota sağladığı görülmüştür.

Anahtar Kelimeler: *Gezgin Satıcı Problemi (GSP), Genetik Algoritma (GA), Parçacık Sürü Optimizasyon Algoritma (PSO), Kargo İHA*

SOLUTION OF THE TRAVELING SALESMAN PROBLEM IN PACKAGE DISTRIBUTION WITH DRONE WITH GENETICS AND PARTICLE SWARM OPTIMIZATION ALGORITHMS

ABSTRACT

In this article, the use of emergency medicine and treatment kits in the field of health is solved using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) Algorithm as a traveling salesman problem among non-polynomial hard problems. This work is of significant value to researchers and industry professionals looking for new methods of cargo logistics and transportation. Effectively optimizing package delivery by UAV has the potential to increase the efficiency of the logistics industry. With the route results found, an optimal roadmap for drone package delivery was created. Firstly, EIL51 data from TSPLIB used as the data set. After that, the locations of the pharmacies in the Menemen district of İzmir were taken from real life. We evaluate the results, it was seen that GA took longer due to the algorithm content and GA provided a more optimized route than PSO.

Keywords: *Traveling Salesman Problem (GSP), Genetic Algorithm (GA), Particle Swarm Optimization Algorithm (PSO), Cargo Drone*

1. Giriş

Teknolojinin gelişmesiyle birlikte günümüzde birçok alanda kullanılan insansız hava araçlarının sayısı gün geçtikçe artmaktadır. Savunma sanayide, tarımda, ulaşımda, güvenlikte gibi pek çok alanda hizmet veren insansız hava araçları, sağlık alanında olay yerine otomatik kalp şok cihazı (Automated External Defibrillator: AED) ulaşımının kolaylaştırılmasında, ilaç/doku/kan ürünlerinin ulaşımında,

*e-posta¹ : enesb.acar@gmail.com ORCID ID: <https://orcid.org/0000-0002-4145-7950> (Sorumlu Yazar)

e-posta² : cumali.karabey@bakircay.edu.tr ORCID ID: <https://orcid.org/0000-0001-8413-6148>

e-posta³ : bayram.kose@bakircay.edu.tr ORCID ID: <https://orcid.org/0000-0003-0256-5921>

travma kiti vb. medikal malzeme ulaşımında ve daha birçok sektörde fayda sağlamaktadırlar. Bu sektörlerden biri de medikal kargo sektörüdür. İHA ile paket dağıtımıyla da kargo yoğunluğunun önüne geçilmesi ve sürecin kolaylaştırılması hedeflenmiştir [1-4].

Gelişen teknolojiyle, birçok sektörde teknolojiyle doğru orantılı bir şekilde uyum yakalanmaya çalışılmıştır. Bu sektörlerden biri olan kargo dağıtımlarında, insanların bu hizmete yetişememesi, artan müşteri talepleri, siparişlerin karmaşıklığı, artan ulaşım yolları ile başa çıkılmasında, kargo ve lojistik destek firmalarını teknoloji arayışına itmiştir. Bu firmaların daha kısa sürede, daha kısa menzil ve daha az enerji tüketimi ile dağıtım süreçlerini optimize hale getirmeleri gerekmektedir.

Bu konuda önerilen teknolojilerden biri de insansız hava araçları (İHA/Drone) ile kargo taşımada zaman ve menzil avantajı sağlanabilir. İHA'lar günden güne isimlerini her sektörde duyurduğu gibi bu sektörde de başarılı bir giriş yapmıştır. Kara yollarındaki trafik problemlerinden bağımsız bir şekilde havadan teslimatın yapılmasını sağlamaktadır. Bu özelliği ile hızlı ve esnek bir dağıtım sürecini oluşturmaktadır. Fakat insansız hava araçlarıyla paket dağıtımı bir Gezgin Satıcı Problemi (GSP) olarak çözülmesi çok önemlidir.

Turgut ve Şeker yaptıkları çalışmada İHA'nın taşımacılık sektöründeki yansımalarını araştırmışlardır; yaygınlaştığını ve kullanımının artmaya başladığının tespit etmişlerdir [5]. Nakiboğlu, ürünlerin üreticiden tüketiciye son adım teslimatında yasal düzenlemelerin eksikliğini ve gereğine vurgu yapmıştır [6]. Uslu ve Tekin'in çalışmasında pandemi süreciyle gelişen kargo teknolojisinde dijitalleşme ve temassız teslimatta internet üzerinden araştırma ve literatür tarama sonucuna göre sektörde insansız hava araçlarının kargoda kullanımının artacağı fikri öngörülmüştür [7]. Yetiş ve arkadaşları, araç-İHA iş birliği ile yapılan hibrit kargo dağıtımında optimum mesafe ve maliyette azalma elde edildiğini göstermişlerdir [8].

GSP, aralarındaki uzaklık mesafesi bilinen n adet noktanın her bir noktadan bir kez geçme/uğrama şartı ile en kısa sürede veya en az maliyette rotayı oluşturmasıdır. İHA'larla paket dağıtımında GSP kullanılarak, paketleri sahiplerine teslim etmek için en verimli rotayı nasıl bulacağını belirleme sorununu temsil etmektedir [1-8]. Ayrıca İHA bataryalarından kaynaklı olarak uçuş mesafesi, menzil kısıtı altında optimizasyon problemi olarak gezgin satıcı problemi yardımıyla çözülebilir.

Bu makalede insansız hava araçlarının kısa sürede tüm noktalara gidebilmesi için Gezgin Satıcı Problemi /GSP) olarak çözümünü için birçok algoritma önerilmiştir. Bu makalede iyi bilinen ve yaygın kullanılan Genetik Algoritma ve Parçacık Sürü Optimizasyon Algoritması çözümleriyle, İHA ile paket teslimatında en hızlı ve en optimal dağıtım haritası çıkarılarak teslimat noktaları rotası ile problemi çözüme süreleri hesaplanmıştır.

2. Materyal ve Metod

Gezgin satıcı probleminde, satıcının alacağı yolun tamamının mesafesini azaltmaktır. Burada amaç satıcının uğraması gereken noktaların rotasını belirleyerek ve satıcının bu rotada belli bir sıra ile uğraması ile katedilen mesafenin en aza indirilmesidir. Bu problem için birçok optimizasyon algoritmaları önerilmiştir; Angeniol ve ark. (1988)[9], gezgin satıcı problemini çözmek için kendi kendini organize eden özellik haritalarını kullanan bir yöntem sunmuştur. Problemi çözmek için veriye dayalı bir yaklaşım sunar. Optimizasyon için kendi kendini organize eden özellik haritalarının kullanılması, genellikle hızlı ve etkilidir. Problemin karmaşıklığına bağlı olarak, özellik haritalarının eğitimi ve oluşturulması zaman alabilir. Yeterli miktarda veri olmadığında veya veri kalitesi düşük olduğunda performansı etkilenir. Somhom ve ark. (1997)[10], yine, kendi kendini organize eden bir model kullanarak probleme yaklaşır, veriye dayalı bir çözüm sunar. Bu tür modeller, karmaşık problemlerde etkili olabilir ve çözümün iyileştirilmesine izin verebilir. Modelin oluşturulması ve eğitilmesi için çok fazla zaman ve kaynak gerekebilir. Ayrıca, modelin karmaşıklığı nedeniyle, büyük veri kümeleri veya yüksek boyutlu problemlerle başa çıkmak zor olabilir. Ellabib ve ark. (2007)[11], gezgin satıcı problemini çözmek için değişim stratejileri ile çoklu bir karınca kolonisi sistemi sunmuştur.

Karınca kolonisi algoritmaları, doğadan ilham ile geliştirilen algoritmalar ve karmaşık optimizasyon problemleri için etkili olabilmektedir. Çoklu karınca kolonisi sistemi, çözüm alanında daha iyi keşifler yapabilir ve daha çeşitli çözümler üretebilir. Karınca kolonisi algoritmalarının performansı, parametre ayarlarına ve problem özelliklerine bağlıdır. Ayrıca, büyük veri setleri veya yüksek boyutlu problemlerle başa çıkmak için hesaplama kaynakları gerekebilir. Nguyen ve ark. (2007)[12], gezgin satıcı problemini çözmek için bir genetik algoritma önermişlerdir ve farklı varyasyon operatörleri kullanarak çözümün çeşitliliğinin artırılabilirliğini bulmuşlardır. Genetik algoritmaların performansı, parametre ayarlarına ve genetik operatörlerin seçimine bağlı olup popülasyon tabanlı yaklaşım olarak, büyük veri setleri veya yüksek boyutlu problemlerle çalışırken hesaplama kaynakları gerektirmektedir. Sauer ve Coelho (2008)[13], gezgin satıcı problemini çözmek için yerel bir arama yöntemi ile ayrık bir diferansiyel evrim algoritması sunmuştur. Shi ve ark. (2008)[14], ödül toplayan gezgin satıcı problemini çözmek için zaman pencereli bir karınca kolonisi optimizasyon yöntemi sunmuştur. Xie ve Liu (2008)[15], gezgin satıcı problemini çözmek için çok etmenli bir optimizasyon sistemi sunmuştur. Yi ve ark. (2008)[16], gezgin satıcı problemini çözmek için hızlı elastik bir ağ yöntemi sunmuştur. Chien ve Chen (2009)[17], paralelleştirilmiş genetik karınca kolonisi sistemine dayalı gezgin satıcı problemini çözmek için bir yöntem sunmuştur. Cheng ve Wang (2009)[18], zaman pencereli araç rotalama problemini çözmek için ayrıştırma tekniğine sahip bir genetik algoritma sunmuştur. Liu ve Zeng (2009) [19], gezgin satıcı problemini çözmek için pekiştirmeli öğrenmeli bir genetik algoritma sunmuştur. Naimi ve Taherinejad (2009)[20], GSP çözmek için yerel bir güncelleme sürecinin yeni bir yorumuyla bir karınca kolonisi algoritması sunmuştur. Marinakis ve Marinaki (2010)[21], araç rotalama problemini çözmek için genetik algoritmaları ve parçacık sürüsü optimizasyon algoritmalarını birleştirerek hibrit bir algoritma sunmuştur. Chang ve ark. (2010)[22], gezgin satıcı probleminde aranmamış çözüm uzayını aramak için genetik algoritmalarda dinamik bir çeşitlilik kontrol tekniği sunmuştur [23].

2.1. Gezgin Satıcı Probleminin Matematiksel Modeli

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

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

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

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

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

GSP'nin amaç fonksiyonu Denklem 1'de verilmektedir. Burada $d(i, j)$ ifadesi i ve j noktaları arasındaki mesafeyi göstermektedir. $x(i, j)$ ise i noktasından j noktasına gidilip gidilmediğini ifade etmektedir. Denklem 2 ve Denklem 3'de her bir noktaya yalnız bir kez uğranacağını garanti altına almaya yöneliktir. Denklem 2'ye göre her noktadan sadece bir kez çıkılacak, Denklem 3'e göre her noktaya yalnızca bir kez gidilecektir. Denklem 4'de ise oluşabilecek alt turlardan kurtulmaya yönelik olan alt tur eleme kısıtıdır. Denklem 5'de $x(i, j)$ 'nin 1 olması i noktasından j noktasına gidildiğini, 0 olması ise gidilmediğini göstermektedir [24].

3. Veri

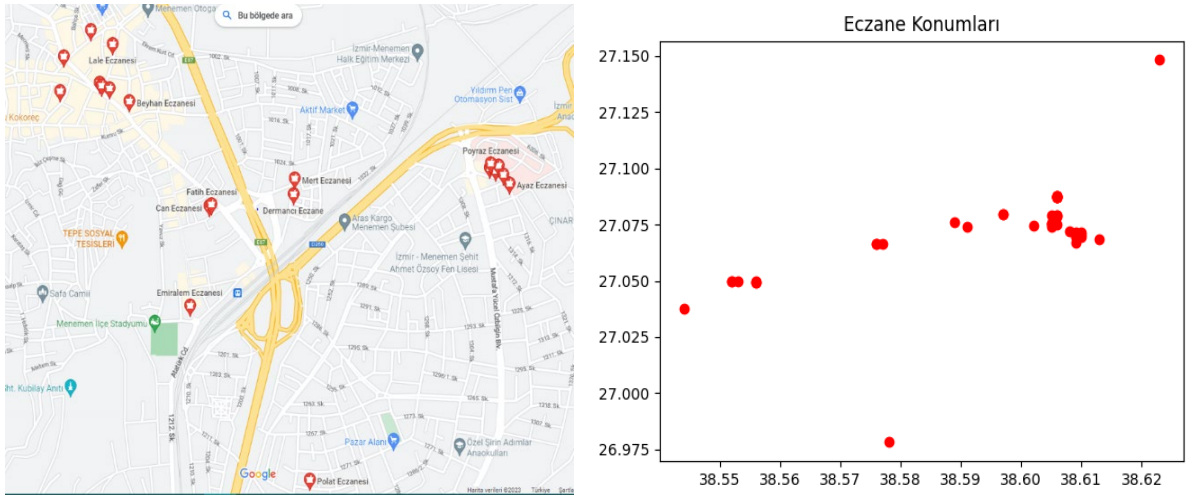
Makalenin bu bölümünde gerçekleştirilen uygulamada kullanılan veriler dair bilgiler verilmiştir. Çalışmada veri seti olarak; TSPLIB'den EIL51 verileri ve İzmir ili Menemen ilçesindeki eczanelerin lokasyonları verileri kullanılmıştır.

3.1. EIL51 Veri Seti

TSPLIB (Travelling Salesman Problem Library), seyahat eden satıcı problemi (Traveling Salesman Problem - TSP) ve benzeri kombinatoriyel optimizasyon problemleri için bir dizi örnek veri seti içeren bir koleksiyondur. EIL51, TSPLIB veri setlerinden biridir ve bu veri seti, 51 şehir (lokasyon) için her şehir için x ve y koordinatları belirtilir. Veri setinde genellikle herhangi bir arazi tipi veya şehir özelliği belirtilmez. Sadece şehirlerin koordinatlarına odaklanılır. EIL51 veri seti, gezgin satıcı problemi (TSP) gibi bir problemi incelemek ve çözmek isteyen araştırmacılar için kullanışlı bir örnek sunmaktadır [31,32].

3.2. İzmir ili Menemen ilçesi Eczaneleri Veri Seti

Bu makalede gerçek veri seti olarak İzmir/Menemen ilçesine ait eczane konumları Şekil 1'de görülmektedir. Bazı eczanelerin birbirine oldukça yakın olduğu ve bu nedenle tüm eczaneleri almak için harita küçültüldüğünde 45 adet eczane yerine yaklaşık 20 adet kadar gözüktüğü gözlenmiştir. Haritanın ölçeği büyütüldüğünde uzakta kalan eczanelerin görünmemesi sebebiyle, tüm eczane lokasyonlarını gösteren kuşbakışı harita resmi kullanılmıştır. İzmir/Menemen ilçesindeki Eczanelerin konumları Google haritalardan alınmış olup x ve y olarak koordinat düzlemine çevrilerek veri seti oluşturulmuştur.



Şekil 1- İzmir Menemen İlçesi Eczaneleri Konumları (Tüm konumlar gösterilmemektedir.)

4. Gezgin Satıcı Problemi İçin Çözüm Algoritmaları

Gezgin satıcı problemi, satıcının alacağı minimum mesafeli yol güzergâhı, aynı zamanda satıcının uğraması gereken noktaların sırasını belirlemektedir. Bu çalışmada insansız hava araçlarının güzergah belirlenmesinde Gezgin Satıcı Problemi /GSP) olarak ele alınmıştır. Diğer taraftan literatürde GSP çözümü için birçok algoritma önerilmiştir [9]. Bu makalede iyi bilinen ve yaygın kullanılan Genetik Algoritma ve Parçacık Sürü Optimizasyon Algoritması ile sağlık sektöründe İHA ile ilaç ve tıbbi malzemelerin acil taşınmasına çözüm olarak ele alınmıştır.

4.1. Genetik Algoritma

İlk olarak 1970’li yıllarda John Holland ve arkadaşlarının yaptığı çalışmalarda ortaya çıkan Genetik Algoritmalar (GA), Yapay Zekâ’nın hızlı gelişen alanlarından [3]. Genetik algoritma, biyolojik evrimi yönlendiren süreçte doğal seçilime dayanan hem kısıtlı hem de kısıtsız optimizasyon problemlerini çözmek için önerilmiş bir yöntemdir. Genetik algoritma, bireysel çözümlerden oluşan bir popülasyonu tekrar tekrar değiştirir. Her adımda mevcut popülasyondan bireyleri ebeveyn olarak seçer ve onları gelecek neslin çocuklarını üretmek için kullanır. Ardışık nesiller boyunca, popülasyon optimal bir çözüme doğru gelişir. Basit bir ifadeyle, bir problemi çözmek için ardışık nesildeki bireyler arasında “en uygun olanın hayatta kalması” simülasyonunu yaparlar. Her nesil, bir popülasyonundan oluşur ve her birey, arama uzayında ve olası çözümde bir noktayı temsil eder. Her birey, dizi olarak temsil edilir ve dizi kromozoma benzer [25].

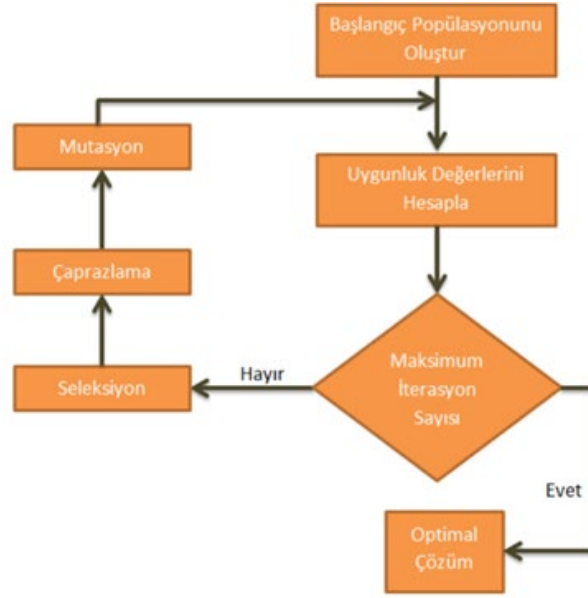
Genetik algoritmalarda kromozomlar, problemde rotayı göstermektedir. Bu rotaların her biri çözümü temsil eder. Kromozomların bir araya gelmesiyle oluşan topluluğa popülasyon denir. Uygunluk fonksiyonunun kullanılması ile hesaplanan Uygunluk değeri çözümün kalitesini belirler. Genetik bilimene benzer şekilde uygulanan Çaprazlama ve mutasyon işlemleri bireylerden daha iyi çözümler elde etmek için yapılır. Oluşan yeni nesiller eski nesil ile karşılaştırılır ve çözümün kalitesine göre eski bireylerin yerine geçerler. Böylelikle yeni bir kromozom popülasyonunun oluşması sağlanır. Problemin çözümünde kullanılan Genetik Algoritmanın Şekil 2’de ki gibi akış diyagramı ve sözde kodu aşağıda verilmiştir [3,4,24,25,26,27]:

Genetik Algoritmanın sözde kodu;

- 1) N adet kromozomlu başlangıç popülasyonunu oluştur.
 - 2) Popülasyonda yer alan kromozomların uygunluk değerlerini hesapla.
 - 3) Yeni topluluğu oluşturmak için aşağıdaki adımları tekrarla:
 - 3.1) Popülasyon içerisinde iki kromozom seç.
 - 3.2) Ebeveynlerden gelen kromozomları çaprazla.
 - 3.3) Oluşan yeni bireye belirli bir olasılıkla mutasyona uğrat.
 - 4) Oluşturulan yeni bireyler ile eski popülasyonun en iyi bireyleri bir oranda karıştırarak yeni bir popülasyonu elde et. (Elitizm).
 - 5) Durdurma koşulu sağlandıysa en iyi çözümü göster. Sağlanmadıysa Adım 2 ye git.
-

4.1.1. Başlangıç Popülasyonunun Oluşturulması

Çözüm bilgileri içeren kromozomların bir araya gelmesiyle oluşan topluluğa popülasyon denir. Genetik algoritmanın başlangıcında popülasyonun oluşturulması gerekmektedir. Popülasyon büyüklüğü, problemin çözüm süresini oldukça etkilemektedir. Popülasyondaki birey sayısının artması çözüm süresini artırırken, birey sayısının azalması istenilen çözüm değerine ulaşamamasına sebep olabilir. Seçilecek probleme göre belirlenen popülasyon büyüklüğü genetik algoritmayı kullanan kişi tarafından iyi belirlenmelidir [3,4,24-28].



Şekil 2- Genetik Algoritma Akış Diyagramı

4.1.2. Uygunluk Fonksiyonu

Başlangıç popülasyonu oluşturulduktan sonra kromozomların uygunluk değerleri hesaplanır. Gezgin satıcı problemindeki uygunluk değeri, en kısa mesafe olan kromozom baz alınarak hesaplanır. Yani her kromozom ve ya birey çözüme aday bir güzergâhtır. Çaprazlama sonrası oluşan yeni kromozomlarda da bu şekilde uygunluk değeri hesaplanmaktadır [3,4,24-28].

4.1.3. Seçilim

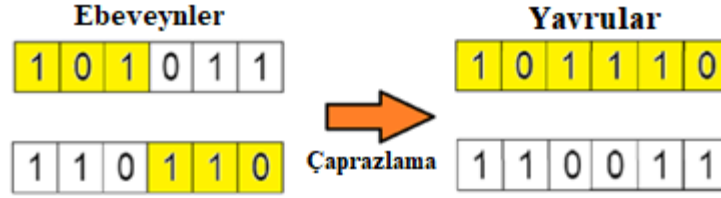
Yeni topluluğu oluşturabilmek için çaprazlama ve mutasyon işlemine alınacak bireyler seçilir. Seçim aşamasında uygunluk değeri fazla olan kromozomların seçilmesi olasılığı yüksektir. En bilinen seçilim yöntemleri; Turnuva Seçilimi, Sıralı Seçilim ve Rulet Seçimidir [24-29].

Rulet seçimi yönteminde popülasyondaki kromozomların uygunluk değerleri toplanır. Burada her bir kromozomun seçilme şansı, kendi uygunluk değerinin tüm bireylerin uygunluk değerlerinin toplamına oranı kadardır. Sıralı seçim yönteminde popülasyondaki kromozomlar, uygunluk değerlerine göre küçükten büyüğe sıralanır. Bu sıraya göre her bir kromozoma sıra numarası verilir. Bu sayede bütün kromozomlara seçilme şansı verilir. Fakat bu durum çözümün geç yakınsamasına sebep olabilir. Turnuva yönteminde ise popülasyondaki kromozomlardan k adet birey rastgele seçilir. Bu bireylerden uygunluk değeri en iyi olan birey alınır. Benzer şekilde bu işlem tekrar uygulanarak ikinci ebeveyn seçilir.

4.1.4. Çaprazlama

Seçim aşamasından sonra seçilen bireyler çaprazlamaya tabi tutulurlar. Çaprazlamadaki amaç, uygunluk değerleri yüksek olan bireylerin çaprazlanarak daha iyi çözümü olan yeni bireyler üretmektir. Literatürde birçok örneği bulunan çaprazlama yönteminin en basiti tek noktalı çaprazlamadır. Bu yöntemde ebeveynler üzerinde ortak bir nokta belirlenir. Oluşacak birinci çocuk için; çaprazlama noktasına kadar gelen genleri ilk ebeveynden, diğer genleriyse

ikinci ebeveynden sırayla alır. Kalan genler ikinci çocukta benzer işlem uygulanır. Aşağıdaki Şekil 3'te örnekte verilen ebeveynlerin ortak noktası 3 kabul edilip tek noktalı çaprazlama işleme tabi tutulmuştur [24-29].



Şekil 3- Genetik Algoritma Çaprazlama Operatörü Örneği

4.1.5. Mutasyon

Oluşturulan yeni bireylerin genlerinin bir ya da birkaçında rastgele değişiklik yapılmasına mutasyon denir. Mutasyonun amacı, oluşan yeni çözümlerin önceki çözümlerle aynı olmasını engelleyerek genetik çeşitliliği sağlamak ve optimal çözüme kısa sürede ulaşmaktır. Böylelikle oluşan yeni bireyler bir sonraki nesle aktarılabilirler. Mutasyon oranı çok düşük (%0.01) tutulmalıdır [3,4,24-30]

4.1.6. Yeni Neslin Oluşturulması

Tüm işlemlerden sonra oluşturulan yeni nesil bireylerin ne yapılacağı belirlenmelidir. Popülasyondaki kötü ebeveynler ile iyi çocukların yer değiştirilebilir, tüm ebeveynler ile tüm çocukların yer değiştirilebilir veya ebeveynler ile çocukların birleştirilerek bütün bir popülasyon oluşturulabilir. Eğer iyi çözümlü çocuklar kötü ebeveynlerin yerine geçerse çözüm süresi kısalmır. Bu şekilde işlem durdurma kriteri sağlanıncaya kadar tekrar edilir [3,4,24-29].

Popülasyondaki mevcut bireyleri yeni popülasyona belirli bir ölçütle seçerken; elitizm, bir stratejisi olarak en iyi bireylerin korunmasını amaçlamaktadır. Sonuçta GA'nın başrimında seçilimin rolü önemlidir. Çaprazlama sonucu oluşan evlat bireyler ile oluşturulan popülasyona eski popülasyondaki en iyi bireylerin belli miktarda karıştırılması yoluyla yeni bir popülasyon oluşturulması (Elitizm) GA da genel bir yaklaşımdır [30].

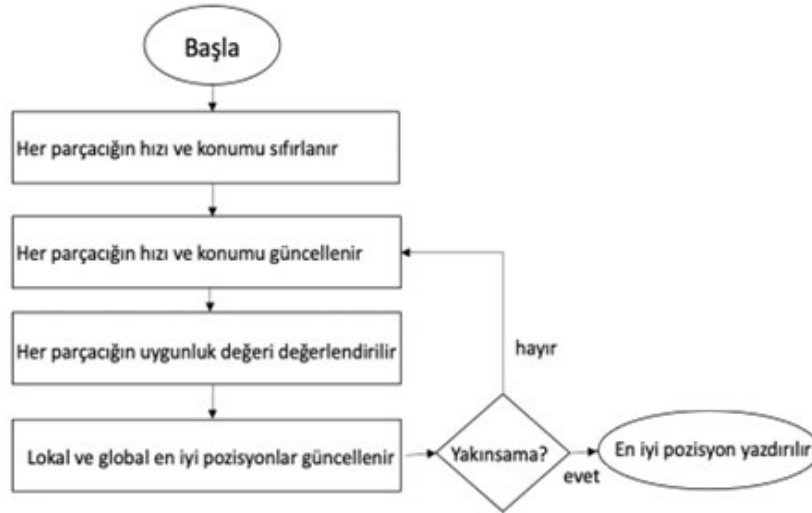
4.2. Parçacık Sürü Optimizasyonu Algoritması

Parçacık sürü optimizasyonu (PSO) orijinal olarak Kennedy ve Eberhart tarafından önerilmiştir. Balık veya kuş gibi hayvan sürülerinin sosyal faaliyetlerinden esinlenmiştir. Sistem önce ilk iterasyon için konumları potansiyel çözümleri temsil eden bir parçacık grubunu rastgele oluşturur. Diğer iterasyonlarda bu global çözüme ait bilgiler tutulur ve sürü içerisindeki parçacıklar yeni bir lokal arayışa giderler. Parçacıkların hareketlerini kendi optimum çözümleri ve en iyi global çözümü etkiler ve geçmişe bağlı kalarak optimum çözüm arayışı yapmaktadır. Parçacıklara ait çözümler kıyaslanır ve en iyi çözüm tutulur, tutulan çözüm eski global çözüm ile kıyaslanır ve en iyi global çözüm elde edilmiş olur belirlenen iterasyon sayısına göre bu parçacıklar arayışa devam eder. D boyutlu arama uzayında m parçacık olduğunu varsayalım. i 'inci parçacığın konumu ve hızı şu şekilde temsil edilir; $X_i = (xi1, xi2, xi3, \dots, xiD)$, $V_i = (vi1, vi2, vi3, \dots, viD)$. Her parçacığın hızı en iyi konum bilgisi tarafından güncellenir. ' $pBest$ ' i 'inci parçacığın o ana kadar aranan en iyi konumudur ' $gBest$ ' ise o ana kadar aranan yakındaki parçacıkların en iyi konum bilgisidir [23]. i 'inci parçacığın konumu ve hızının güncelleme kuralları Denklem (1) ve Denklem (2) deki gibidir [28];

$$V_i = V_i + C1R1(pBest - X_i) + C2R2(gBest - X_i) \quad (6)$$

$$X_i = X_i + V_i \quad (7)$$

Burada $C1$ ve $C2$ pozitif sabitlerdir, ivme katsayıları olarak adlandırılabilir. $R1$ ve $R2$ rastgele sayılardır ve 0-1 aralığında sayılardır. Algoritmanın akış diyagramı Şekil 4’de verilmiştir [29].



Şekil 4- PSO Akış Diyagramı [28]

PSO Algoritmasının sözde kodu;

1. Kontrol parametrelerinin değerlerini belirle
 2. Başlangıç popülasyonu için parçacıkları rastgele seç ve başlangıç koşullarını belirle, değerleri ata
 3. Maksimum iterasyon sayısına ulaşana veya minimum hata koşulu sağlanana kadar
 4. Her parçacığın uygunluk değerini hesapla P_{best} 'ten iyiyse onunla değiştir
 5. Tüm parçacıkların bulunduğu P_{best} değerlerinin en iyisini, tümünün P_{best} 'i olarak ayarla
 6. Her parçacık için parçacık hızını hesapla
 7. Parçacık pozisyonu güncelle
-

5. Uygulama

Problemin ilk veri setinin çözümünde Genetik Algoritma kullanılırken öncelikle yapılması gereken, Genetik Algoritmanın sonuçlarını önemli derecede etkileyen parametrelerden birisi olan popülasyon büyüklüğünün belirlenmesidir. Bu çalışmada popülasyon büyüklüğü, değişken olarak her simülasyonda farklı girilmiştir. Böylelikle Python program ile yazılan kodlarla hangi iterasyon ve popülasyon büyüklüğünde en iyi çözüm verdiği sonuçlara bakılarak anlaşılmıştır. Daha sonra mutasyon oranı için farklı değerler denenerek sonuçlarına bakılmış ve uygun mutasyon oranı 0.008 olarak belirlenmiştir. Problemden permütasyon kodlamalı GA kullanılmıştır. Koda girdi olarak 51 adet nokta verilmiştir. İterasyon sayısı optimumu bulduğunda durdurulma kriterini içerdiğinden değişken olarak verilmiştir.

Problemin ikinci veri setinin Genetik ve Parçacık Sürü Optimizasyon Algoritmaları ile çözümünde ise koordinatları verilen 45 adet eczanenin girdi değerleri bir önceki veri seti ile benzerlik göstermektedir. Bu 45 adet eczanenin koordinatlara göre yerleşiminde bazı eczaneler aynı lokasyonda 3 tane, 4 tane yan yana konumlu olduğundan grafik ve haritada yaklaşık 20 adet kadar gözükmektedir. Bu eczaneler arasında menzil belirlemede, PSO için, popülasyon büyüklüğü her simülasyonda farklı girilmiştir. Böylelikle en iyi çözüm sonuçları; iterasyon sayısı, popülasyon büyüklüğü menzil mesafesi ve süre açısından karşılaştırılarak değerlendirilmesi sağlanmıştır.

6. Optimizasyon için Simülasyon Sonuçları ve Analizi

Makalenin bu bölümünde gerçekleştirilen uygulamaya ait çıktılar bulunmaktadır. Genetik algoritma ve Parçacık Sürü Optimizasyonu algoritması karşılaştırılması yapılırken performans başarısı için “Rota Mesafesi” ve “Başarı Süresi” kriterleri dikkate alınmıştır. Her iki algoritma da iterasyon değerleri ve popülasyon büyüklükleri sabit tutularak karşılaştırmalar yapılmıştır.

6.1. EIL51 Verileri Kullanılarak Elde Edilen Sonuçlar

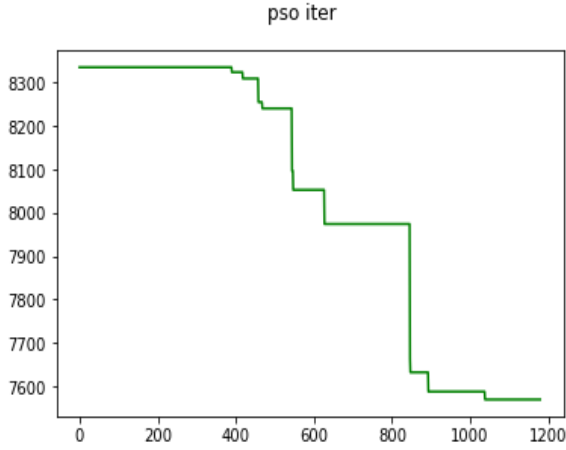
Hazır veriler (EIL-51) kullanılarak yapılan PSO algoritma programı test edilmiştir[31]. Tablo-1 incelendiğinde, PSO ile 1200 iterasyon ve 300 parçacıkla yapılan simülasyon sonucunda 22sn de en kısa mesafeyi 7570 m civarında hesaplanmıştır. Her iterasyon değerinde popülasyon sayısı arttıkça bulunan rota mesafesi kısalmaktadır fakat zaman maliyeti artmaktadır. Tablo-2 incelendiğinde, iterasyon sayısı arttıkça GA ile optimum rotaya daha yaklaşılmaktadır fakat süre maliyeti artışı da beraberinde gelmektedir. Tüm bunlara karşın GA'nın popülasyon büyüklüğünün artışının optimum mesafeye belirgin bir etkisi olmadığı görülmüştür.

Tablo 1 - EIL51 Veri Seti ile PSO Algoritması Sonuçları

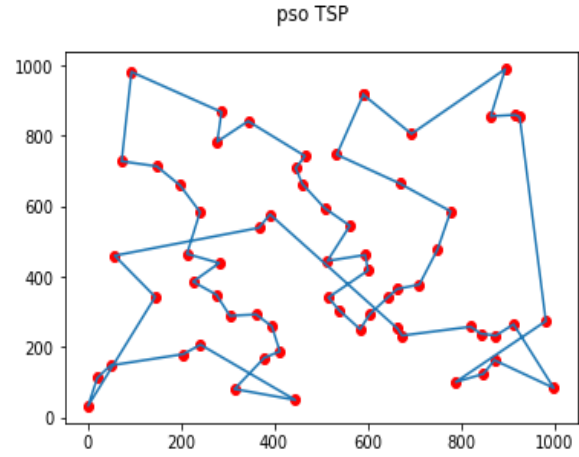
		PARAMETRELER		SONUÇLAR	
		İterasyon Sayısı	Popülasyon Büyüklüğü	Bulunan Rota Mesafesi	Arama Süresi
Başlangıç Rota Mesafesi = 8335,256	100	100	100	8335,256	2.5 sn
		175	175	8335,256	3s
		250	250	8335,256	4.5 sn
		300	300	8335,256	7s
	500	100	100	8335,256	7s
		175	175	8300.06	8s
		250	250	8298,301	9s
		300	300	7956,312	11s
	1000	100	100	8099,967	9.5 sn
		175	175	7778.530	13.5 sn
		250	250	7842,764	16.5 sn
		300	300	7794.566	19s
1200	100	100	8238,405	14s	
	175	175	7852,773	15.5 sn	
	250	250	7587,580	19.5 sn	
	300	300	7569.426	22s	

Tablo 2 - EIL51 Veri Seti ile Genetik Algoritma Sonuçları

		PARAMETRELER		SONUÇLAR	
		İterasyon Sayısı	Popülasyon Büyüklüğü	Bulunan Rota Mesafesi	Arama Süresi
Başlangıç Rota Mesafesi = 8335,256	100	100	100	8264.709	6s
		175	175	7430.007	13s
		250	250	7761.664	25s
		300	300	7559.561	40s
	500	100	100	7649.889	25s
		175	175	7409.437	1dk 8s
		250	250	7302.052	2dk 8s
		300	300	7026.819	3dk
	1000	100	100	6936.572	1dk
		175	175	7420.367	2dk 16s
		250	250	7382.446	4dk 12s
		300	300	7301.284	6dk
1200	100	100	7361.819	1dk 12	
	175	175	7050.740	2dk 45s	
	250	250	6710.492	5dk 10s	
	300	300	6896.164	7dk 12s	

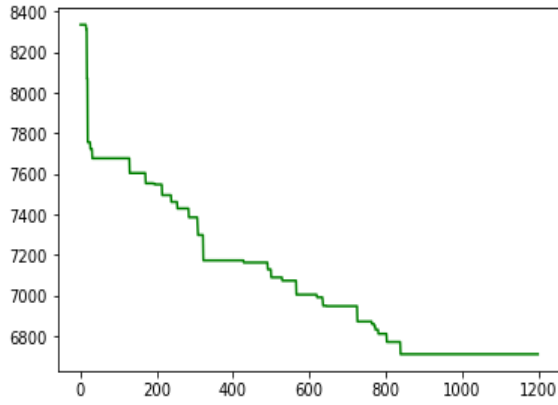


Şekil 5- PSO Algoritma optimum değere yaklaşım grafiği

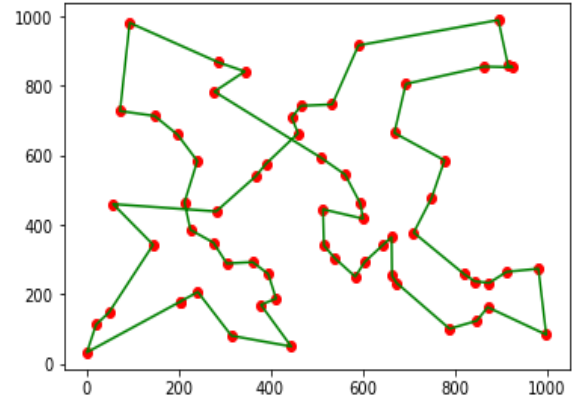


Şekil 6- PSO Algoritma Araması Sonucunda Bulunan Rota

İterasyonla mesafe düşümü, Şekil-5’de 0-1200 aralığı jenerasyonla 7600-8300 aralık ile temsil etmektedir.



Şekil 7- Genetik Algoritma optimum değere yaklaşım grafiği



Şekil 8- Genetik Algoritma Araması Sonucunda Bulunan Rota

Şekil-7 de 0-1200 jenerasyonu, 6800-8400 aralığı mesafeyi temsil etmektedir.

6.2. Gerçek Verileri Kullanarak Uygulama

Gerçek veriler kullanılarak Genetik ve PSO Algoritmaları ile optimum sonuçlara ulaşılmıştır. Bir önceki veri setine göre kullanılan girdi parametrelerinde değişenler sadece şehir yerine Eczane gelmiş ve lokasyon sayısı değişmiştir. Diğer parametreler bir önceki veri seti ile aynı tutulmuştur. Türkiye Cumhuriyeti coğrafi konum olarak, 26-45 doğu meridyenleri ve 36-42 kuzey paralelleri tarafından çevrelenmektedir. Yani 19 meridyen, 6 paralel ile ölçeklendirilmektedir. Türkiye'nin güneyde 36° lik paralel üzerinde bulunan ardışık iki meridyen 81 km ve kuzeyinde ise 42°lik paralel üzerindeki iki meridyen arası ise 74 km'dir. Ekvator paraleli üzerinde iki meridyen arası mesafe 111 km dir[33,34]. Algoritmaların çalışmaları coğrafi konum cinsinden, açılal konum olarak hesaplamalar yapılmıştır. Bu yaklaşımla menziller yaklaşık metrik olarak hesaplanabilir. Tablo 3 ve Tablo 4 incelendiğinde, ilk bakışta en kısa mesafe 1200 iterasyon ve 100 popülasyon büyüklüğü ile 1 dk 39 sn'de GA ile elde edilmiştir. PSO algoritması ile en kısa menzil 1000 iterasyon 100 popülasyonla 14.2 sn'de elde edilmiştir. Yapay zeka temelli olan, sezgisel yapıya bütün algoritmalar genel olarak kesin çözüme yakınsamayı garanti etmezler, ancak gerçek çözüm çok çok küçük hatayla yaklaşabilirler. Bunun için

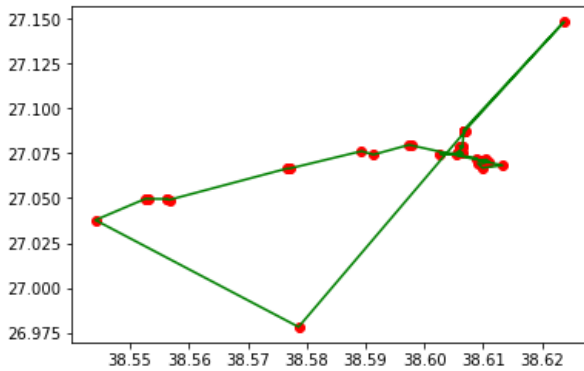
yeterince deneme yapılmalı ve algoritma ile problem modelinin optimum parametreleri elde edilmelidir. Bu süreç zorlu bir süreç olsa da çözüme oldukça yakın sonuçlar, günlük yaşamdaki problemlerimizi çözmeye yeter. Özellikle analitik çözümün olmadığı, matematiksel modeli bir fonksiyon şeklinde ifade edilemeyen problemlerde bu sezgisel yöntemler kullanmak çözüm için çok önemlidir. Bu çalışmada GA ve PSO algoritmaları ile sağlık sektörü için zaman, kargo ağırlığı ve İHA'nın enerji kısıtı altında optimum çözümler üretilebilir.

Tablo 3 - Eczane Konumları ile PSO Algoritma Sonuçları

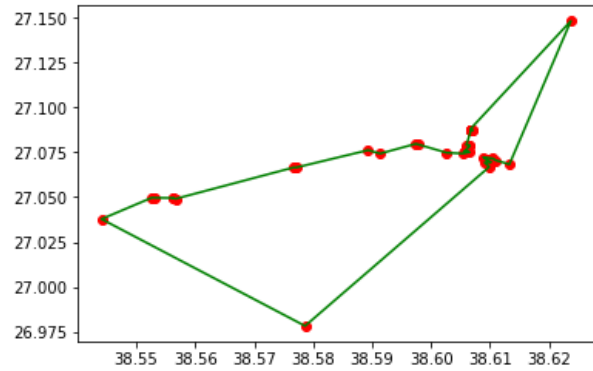
		PARAMETRELER		SONUÇLAR	
		İterasyon Sayısı	Popülasyon Büyüklüğü	Bulunan Rota Mesafesi	Arama Süresi
Başlangıç Rota Mesafesi = 0.4383216461701235	Şehir Sayısı = 45	100	100	0.438321646170	1.6 sn
			175	0.438321646170	1.9 sn
			250	0.438321646170	2.2 sn
			300	0.438321646170	2.5 sn
		500	100	0.437577095647	7.3 sn
			175	0.436800663821	8.6 sn
			250	0.435270717610	10.5 sn
			300	0.434697110422	11.4 sn
		1000	100	0.434314445052	14.2 sn
			175	0.435814669247	15.2 sn
			250	0.435981190229	17.8 sn
			300	0.435570588256	21.5 sn
1200	100	0.436718550590	22.7 sn		
	175	0.437214091859	20.4 sn		
	250	0.4350305874813	22.7 sn		
	300	0.4359089829906	24 sn		

Tablo 4 - Eczane Konumları ile Genetik Algoritma Sonuçları

		PARAMETRELER		SONUÇLAR	
		İterasyon Sayısı	Popülasyon Büyüklüğü	Bulunan Rota Mesafesi	Arama Süresi
Başlangıç Rota Mesafesi = 0.4383216461701235	Şehir Sayısı = 45	100	100	0.4216141576094	8.9 sn
			175	0.4202002744885	25.6 sn
			250	0.4189672619183	38.5 sn
			300	0.4242203103298	50.4 sn
		500	100	0.4226159248675	41.8 sn
			175	0.4174078320156	1 dk 33 sn
			250	0.4173056232760	2 dk 55 sn
			300	0.4166072174686	4 dk 17 sn
		1000	100	0.4157284632578	1 dk 24 sn
			175	0.4180074280467	3 dk 30 sn
			250	0.4181365055331	6 dk 31 sn
			300	0.415352133474	9 dk 40 sn
1200	100	0.4147512988456	1 dk 39 sn		
	175	0.415276421963012	3 dk 54 sn		
	250	0.41486437436170	7 dk 40 sn		
	300	0.41730562327605	10 dk 25 s		



Şekil 9 - PSO Algoritma Araması Sonucunda Bulunan Rota



Şekil 10 - Genetik Algoritma Araması Sonucu Bulunan Rota

7. Sonuç ve Değerlendirme

Bu makalede insansız hava araçlarının kargo taşıma rotasını en kısa sürede tüm noktalara gidebilmesi için Gezgin Satıcı Problemi ele alınmıştır. Problemin çözümü birçok algoritmayla çözümlenmesine karşın bu çalışmada Genetik Algoritma ve Parçacık Sürü Optimizasyon Algoritması

kullanılmıştır. İHA ile ürün teslimatında en hızlı ve en optimal dağıtım haritası çıkarılarak teslimat noktalarına en kısa sürede ve en az iş gücüyle varış rotaları belirlenmiştir.

Rastgele noktalar arasında en kısa mesafeyi bulabilmek için gerçekleştirilen algoritma çözümlerinde, tüm değerlerde Genetik algoritmanın daha iyi sonuçlar verdiği gözlemlenmiştir. İlk veri seti kullanılarak çıkan sonuçların arasında en iyi çözümü veren ise 1200 iterasyonda, 250 popülasyon büyüklüğü olmuştur. İkinci veri seti kullanıldığında ise 1200 iterasyonda 100 popülasyon büyüklüğü en iyi çözümdür.

İki algoritmanın sonuçlarına bakıldığında Genetik algoritmanın parametrelerinden dolayı daha uzun sürmesine rağmen PSO'ya göre daha iyi sonuçlar vermiştir. Buna göre Genetik Algoritmanın PSO'ya göre daha başarılı olduğu söylenilebilir. Gerçek verilerde popülasyon sayısının artırılması sonuca belirgin düzeyde iyileştirmemiştir. İki algoritmada da iterasyon sayısının artması optimum sonucu iyileştirirken zaman maliyetini artırmıştır.

Bu makalede algoritmaların yazımı için Python programlama dili kullanılmıştır. Genetik algoritma çalıştırılırken oldukça uzun süreler harcanırken maliyeti oldukça arttırmıştır. Bu iki optimizasyon algoritmalarının farklı programlama dillerinde karşılaştırılması ve performans ölçümü daha sonraki çalışmalarda yapılabilir.

Gerçek verilerden alınmış bu sonuçlara bakılarak da Genetik Algoritmanın PSO'ya göre daha iyi sonuçlar verdiği söylenilebilir. Böylelikle literatüre, 'Gezgin satıcı problemini çözebilmek için genetik algoritmanın kullanılacağı ve böylelikle İHA kullanımında daha hızlı ve maksimum verim alınacağı ve katkı sağlayacağı öngörülmüştür.' cümlesinin silinmesi ve yerine 'Gerçek verilerle elde edilen rotalar incelendiğinde GA'nın PSO'ya göre daha iyi sonuçlar verdiği görülmektedir. İki algoritmanın sonuçlarına bakıldığında GA'nın parametrelerinden dolayı daha uzun sürmesine rağmen PSO'ya göre daha iyi sonuçlar vermiştir. Gerçek verilerde popülasyon sayısının artırılması sonucu belirgin düzeyde iyileştirmemiştir. İki algoritmada da iterasyon sayısının artması optimum sonucu iyileştirirken zaman maliyetini artırmıştır.

Çıkar Çatışması Beyanı

Makale yazarları herhangi bir kurum, kuruluş, kişi ile kişisel ve finansal çıkar çatışması olmadığını beyan etmektedirler.

Kaynaklar

- [1] Çalışkan F., Yüksel H., Dayik M., Genetik Algoritmaların Tasarım Sürecinde Kullanılması, 2016
- [2] Yildirim T., Simetrik Gezgin Satıcı Problemi İçin Yeni Bir Meta-Sezgisel: Kör Fare Algoritması, Denizli, Şubat - 2014
- [3] Pulat M., Kocakoç D., Gezgin Satıcı Probleminin Genetik Algoritmalarla Çözümünde Başlangıç Popülasyonunun Belirlenmesi, July 2017
- [4] Kuzu S., Önay O., Şen U., Tunçer M., Yildirim B. F., Keskindürk T., Gezgin satıcı problemlerinin metasezgiseller ile çözümü, Cilt/Vol:43, Sayı/No:1, 2014, 1-27
- [5] Turğut, M. ve Şeker, B. (2022). İnsansız Hava Araçlarının (İha) Taşımacılıkta Kullanımına Yönelik Keşfedici Bir Araştırma: Drone Taşımacılığı Ve Uygulamaları. Akıllı Ulaşım Sistemleri Ve Uygulamaları Dergisi, 5 (2), 169-187. Doi: 10.51513/Jitsa.1146992
- [6] Nakiboğlu, G. (2020), Drone Taşımacılığı ve Son-Adım Teslimatta Kullanımı, Çukurova Üniversitesi İİBF Dergisi Cilt:24. Sayı:2, ss.285-298
- [7] Uslu, F., Tekin, Z. (2021), Pandemi Sürecinde Drone Kullanımı: Geleceğin Lojistik Teknolojileri, Uluslararası İktisadi ve İdari Bilimler Kongresi: Krizler, Belirsizlikler ve Arayışlar, 165

- [8] Yetiş, H., Güngör, Z. ve Karaköse, M. (2021). Araç-İHA İş birliği ile Kargo Teslimatları İçin Ortak Rota Optimizasyonu. *Fırat Üniversitesi Fen Bilimleri Dergisi*, 33 (2), 135-144. Retrieved from <https://dergipark.org.tr/en/pub/fufbd/issue/64918/878774>
- [9] Angeniol, B., Vaubois, G. D. L. C., & Le Texier, J. Y. (1988). Self-organizing feature maps and the travelling salesman problem. *Neural Networks*, 1(4), 289-293.
- [10] Somhom, S., Modares, A., & Enkawa, T. (1997). A self-organising model for the travelling salesman problem. *Journal of the Operational Research Society*, 48(9), 919-928.
- [11] Ellabib, I., Calamai, P., & Basir, O. (2007). Exchange strategies for multiple ant colony system. *Information Sciences*, 177(5), 1248-1264.
- [12] Nguyen, H. D., Yoshihara, I., Yamamori, K., & Yasunaga, M. (2007). Implementation of an effective hybrid GA for large-scale traveling salesman problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1), 92-99.
- [13] Sauer, J. G., & dos Santos Coelho, L. (2008, September). Discrete differential evolution with local search to solve the traveling salesman problem: Fundamentals and case studies. In 2008 7th IEEE international conference on cybernetic intelligent systems (pp. 1-6). IEEE.
- [14] Shi, X., Wang, L., Zhou, Y., & Liang, Y. (2008, October). An ant colony optimization method for prize-collecting traveling salesman problem with time windows. In 2008 Fourth International Conference on Natural Computation (Vol. 7, pp. 480-484). IEEE.
- [15] Xie, X. F., & Liu, J. (2008). Multiagent optimization system for solving the traveling salesman problem (TSP). *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2), 489-502.
- [16] Yi, J., Bi, W., Yang, G., & Tang, Z. (2008, November). A fast elastic net method for traveling salesman problem. In 2008 Eighth International Conference on Intelligent Systems Design and Applications (Vol. 1, pp. 462-467). IEEE.
- [17] Chien, C. Y., & Chen, S. M. (2009, July). A new method for handling the traveling salesman problem based on parallelized genetic ant colony systems. In 2009 International Conference on Machine Learning and Cybernetics (Vol. 5, pp. 2828-2833). IEEE.
- [18] Cheng, C. B., & Wang, K. P. (2009). Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert systems with applications*, 36(4), 7758-7763.
- [19] Liu, F., & Zeng, G. (2009). Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, 36(3), 6995-7001.
- [20] Naimi, H. M., & Taherinejad, N. (2009). New robust and efficient ant colony algorithms: Using new interpretation of local updating process. *Expert Systems with Applications*, 36(1), 481-488.
- [21] Marinakis, Y., & Marinaki, M. (2010). A hybrid genetic-Particle Swarm Optimization Algorithm for the vehicle routing problem. *Expert Systems with Applications*, 37(2), 1446-1455.
- [22] Chang, T. K., Huang, C. H., Huang, C. H., Chen, H. C., & Cheng, C. K. (2010). The influence of long-term treadmill exercise on bone mass and articular cartilage in ovariectomized rats. *BMC Musculoskeletal Disorders*, 11(1), 1-7.
- [23] Chen, S. M., & Chien, C. Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. *Expert Systems with Applications*, 38(12), 14439-14450.
- [24] Çolak S., Genetik Algoritmalar Yardımı İle Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama, *Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi*, Cilt 19, Sayı 3, 2010, Sayfa 423-438
- [25] Gerşil M., Alkaya A., Gezgin Satıcı Problemi İçin Sezgisel Metotların Performans Analizi, XI. Üretim Araştırmaları Sempozyumu, 23-24 Haziran 2011
- [26] Dikmen Ha., Dikmen Hü., Elbir A., Ekşi Z., Çelik F., Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması, 18(1), 8-13, 2014
- [27] Cevre U., Özkan B., Uğur A., Gezgin Satıcı Probleminin Genetik Algoritmalarla Eniyilemesi ve Etkileşimli Olarak İnternet Üzerinde Görselleştirilmesi, Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir

- [28] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In MHS'95. Proceedings of the sixth international symposium on micro machine and human science (pp. 39-43). Ieee.
- [29] Kennedy J., Elbarth R. (August 2002), Particle swarm optimization, Purdue School of Engineereing and Technology
- [30] Gencal, M. C. & Oral, M. (2022). Evrimsel algoritmalar için yeni bir meta-iyileştirici: bipolar eşleşme eğilimi . Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 28 (2) , 313-323 . Retrieved from <https://dergipark.org.tr/en/pub/pajes/issue/69632/1110433>
- [31] Onwubolu, G. C., Babu, B. V., & Clerc, M. (2004). Discrete particle swarm optimization, illustrated by the traveling salesman problem. New optimization techniques in engineering, 219-239.
- [32] url1, <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/> (erişim: 15.02.2023)
- [33] url2, <https://www.yeniakit.com.tr/haber/turkiyenin-uzerinden-gecen-paralel-ve-meridyenler-hangileridir-1539841.html> (erişim:15.08.2021)
- [34] url3, <https://www.matematiklezzeti.com/post/meridyenler> (erşim:15.08.2023)