



Bartın Üniversitesi Mühendislik ve Teknoloji Bilimleri Dergisi



Tufal Takviyeli Polimer Matrisli Kompozitin Kuru Kayma Aşınma Davranışları

M. Sabri GÖK, Mustafa ÖZCAN*, Rifat ÜNAL, Mesut ÇAKMAK, Mete ULAŞIR, Emre AYDOĞDU, A. Tayfun KİRAZ, Dilara BOYNUEYRİ, Esra GÜNER, Hasan YILMAZ, Dilara BEYTEKİN

Bartın Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği Bölümü, Bartın / TÜRKİYE

Geliş Tarihi: 02.04.2019 Düzeltme Tarihi: 30.05.2019 Kabul Tarihi: 19.06.2019

Özet

Bu çalışmada, özellikle sıcak kalıplama işlemlerinde kullanılan ve ticari ismi clarofast olan polimerik malzeme kompozit numune haline getirilerek kuru kayma aşınma davranışları incelenmiştir. Takviye edici olarak 61-67, 67-91, 91-125 µm boyutlarında (Fe₂O₃) tufal tozları kullanılmıştır. Kompozit numuneler hazırlanırken içerisine kütlece % 5, 10, 15, 20 ve 25 oranlarında tufal katılmıştır. Aşınma testleri ball on disk aşınma cihazında 6 mm çapında Al₂O₃ bilya ile 7 N luk yük altında 0,15m/sn dönme hızında gerçekleştirilmiştir. 3D optik profilometre ile aşınma izi profili tespit edilmiş ve aşınma kayıpları bu profile göre hesaplanmıştır. Clarofast içerisindeki artan tufal oranına bağlı olarak kompozit numunelerdeki sertlik oranının arttığı görülmüştür. En iyi aşınma direnci % 5 oranında tufal içeren numunelerde görülürken en düşük aşınma direnci % 25 takviyeli numunelerde görülmüştür.

Anahtar Kelimeler: Tufal, Clarofast, Kompozit, Kuru Kayma Aşınması

Dry Sliding Wear Behavior of Iron Scale Reinforced Polymer Matrix Composite

Abstract

In this study, dry sliding wear behavior of composites produced from polymer material which are used in hot molding process and whose name is clarofast have been investigated. Tufal (Fe₂O₃) powders which have 61-67, 67-91, 91-125 µm dimensions were used as reinforcement. Wear tests were performed in ball on disc wear device with Al₂O₃ ball (6 mm diameter), under 7N load, at 0.15 m/s rotational speed. The wear profile was determined with 3D optical profilometer and the wear losses were calculated according to this profile. It has been observed that the hardness ratio of composite samples increased due to the increasing clarofast ratio. The lowest wear resistance was seen in sample containing 5% tufal, while the highest wear resistance was observed in 25% reinforced sample.

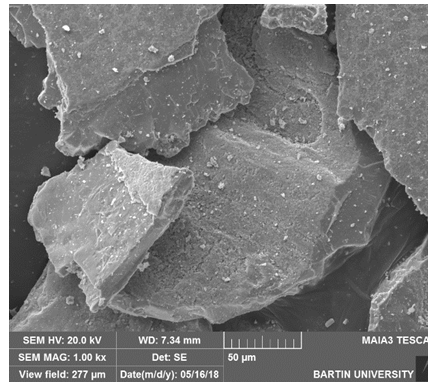
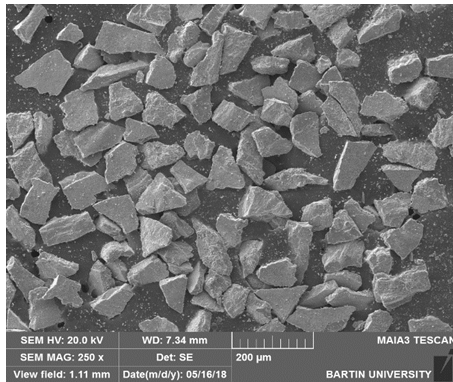
Keywords: Scale, Clarofast, Composite, Dry Sliding Wear

1. Giriş

İki veya daha fazla malzemenin makro-düzeyde bir araya getirilmesiyle oluşturulan ve yeni özelliklere sahip malzemelere Kompozit malzeme denir [1]. Kompozit malzemelere olan ilgi geçen yüzyılın ortalarında bu yana hızlı bir şekilde artmıştır. Matris, takviye fazı ve ara yüzeyden oluşan kompozitlerin geleneksel malzemelerden olan üstünlükleri; hafiflik, yüksek mukavemet, titreşim sönmeme, elektriksel ve korozyon özellikleri olarak sıralanabilir. Kompozit malzemeler yapılarını oluşturan

malzemelere göre kompozit malzemeler ve yapısal bileşenlerinin şekline göre kompozit malzemeler diye gruplandırılabilir[2-3].

Polimer kompozitler iki ve daha fazla bileşenden veya iki veya daha fazla fazdan oluşabilirler. Polimerlere güçlendirici olarak katılan dolgu malzemeleri önemlidir. Çünkü bu dolgu malzemeleri polimer matrise istenilen özellikleri kazandırmaktadır. Günümüzde hibrit kompozit üretimi sahip olduğu üstünlüklerinden dolayı yaygınlaşmaktadır [4-7].



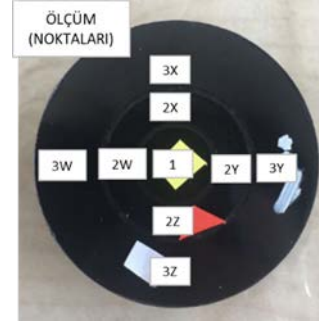
Takviye edici olarak kompozit matrislerin içerisinde SiC, Al₂O₃, B₄C, Zr, Fiberler ticari anlamda kullanılmaktadır. Ancak bütün doğal kaynaklar sınırsız rezerve sahip değildir. Bu nedenle endüstriyel katı atıkların değerlendirilerek ortadan kaldırılması hem ekonomi ve çevre kirliliği için hem de doğal kaynakların korunması

açısından büyük bir önem arz etmektedir [8-16]. Endüstriyel katı atıkların geliştiği güzel atılmaları çevre kirliliğine neden olmaktadır. Bu atıklarda bulunan zararlı maddeler, özellikle toksik etkisi olan

ağır metaller, yağmur ve kar sularının etkisi ile çevredeki sulara ve toprağa taşınmaktadır. Ağır metaller bitki, hayvan ve insan sağlığını önemli ölçüde olumsuz yönde etkilemektedir. Bu nedenle endüstriyel katı atıkların yeniden kullanıma sunulması değerlendirilmesi günümüzde daha da önem kazanmaktadır. Bu amaçla bu çalışmada atık tufaller takviye edici malzeme olarak kullanılmıştır.

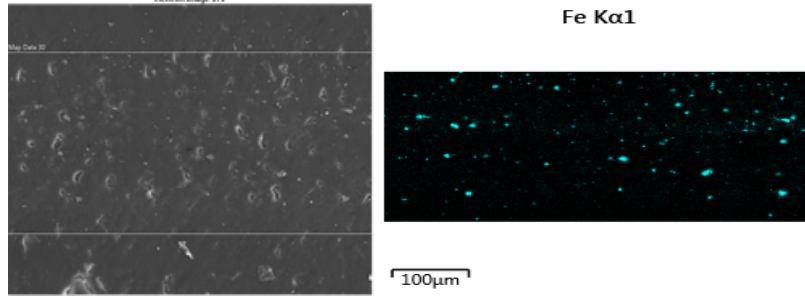
2. Yöntem

Numunelerin metalografik incelemesinde sıcak kalıplama işleminde kullanılan clarofast bu çalışmada kompozit malzeme haline getirilmiştir. Kompozit numune oluşturmak için gerekli tufal tozları haddehaneden temin edilmiştir. Tufal tozları bilyalı değirmende öğütülmüş ve sonrasında farklı boyutlardaki eleklerden geçirilerek boyutlandırılmışlardır. Toz boyutları 33 µm altı, 33-61, 61-67, 67-91-125 µm olarak belirlenmiştir. 33 µm altı ve 91-125 µm boyutlarındaki tozların SEM görüntüleri Şekil 1'de verilmiştir. Şekil 1 a da görüleceği üzere toz boyutunun küçülmesi ile toz içerisindeki artık partikül oranı eleme işleminin zorlaşması nedeni ile artmıştır. Oysa toz boyutunun artması daha temiz ve istenilen boyutlarda homojen toz elde etme kabiliyetini arttırmıştır. Kompozit numuneler üretilirken clarofast içerisine kütlece % 5, 10, 15, 20 ve 25 oranlarında tufal (Fe₂O₃) tozları katılmıştır. Karışımın homojen bir dağılım göstermesi için iki saat süresince bıçaklı bir değirmen yardımı ile karıştırılmıştır. Harmanlama işlemi sonrasında Numuneler 40x10 mm çapında yaklaşık 200 °C sıcaklıkta 30 dk 500 Newtonluk yük altında preslenmişlerdir. Presleme işlemi sonrasında numunelerin makro boyuttaki görüntüsü Şekil 2'de verilmiştir.



Şekil 2. Üretilen kompozit numunelerin fotoğrafı

Numunelerin aşındırma işlemleri pin on disk aşınma cihazında gerçekleştirilmiştir. Testlerde iki farklı yük kullanılmış bunlar 7 ve 15 N olarak belirlenmiştir. Kompozit numunelerin aşındırılmasında 6 mm çapında 2000 HV sertlik değerine sahip alümina bilyalar kullanılmıştır. Aşındırma işlemi süresince numunelere 1500 metrelik bir aşındırma yolu aldırılmıştır. Deneylerin doğruluğunu arttırmak amacıyla her bir test 3 kez tekrarlanmıştır. Numuneler aşındırma cihazına bağlamadan önce ve aşındırma işlemi tamamlandıktan sonra basınçlı hava ile temizlenmişlerdir. Numunelerin hacim kayıplarının belirlenmesinde HUVİTZ marka 3D profilometre kullanılmıştır. Daire şeklinde oluşan aşınma izinin dört farklı bölgesinden alınan alan aşınma izinin çevresel boyu ile çarpılarak numunelerdeki hacim kayıpları belirlenmiştir.



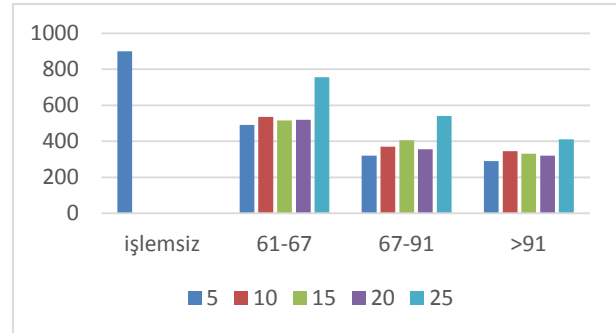
Şekil 3. Üretilen kompozit numunelerin EDS mapping analizi 61-67 µm

3. Bulgular

Kompozit numuneler üretildikten sonra SEM mikroskobu yardımı ile gerçekleştirilen elementel dağılım haritalanması çıkarılmıştır. Şekil 3'te ki resimler üzerinden de görüleceği gibi 61-67 µm, boyutundaki FeO partikülleri homojen olarak yapı içerisinde dağılım göstermişlerdir.

Numunelerin 7 N'luk yük altında aşındırma işlemine tabi tutulduktan sonra eldedilen hacim kayıpları Şekil 4 de verilmiştir. Hem abrasiv aşındırıcı kullanılarak hem de kuru kayma aşınma şartlarında gerçekleştirilen deneylerde yüksek hacim oranında sert takviye ediciye sahip kompozitlerin aşınma dirençlerinin daha yüksek olduğu belirtilmiştir. Axen ve Zum Gahr tarafından Al matrisli partikül takviyeli kompozit numuneler üzerine geliştirilen oran konsept teorisi Wang ve Rack tarafından ilettilerle nisbi penetrasyon derinlik tezi geliştirilmiştir. Aşındırıcının batma derinliğinin takviye edici partikül büyüklüğüne ve bu partiküller arasındaki mesafeye bağlı olduğunu belirtmişlerdir [2-3]. Şekil 4'te verilen grafikten de görüleceği üzere takviyesiz numunenin hacim kaybı 900 mg olarak gerçekleşirken en iyi aşınma direnci gösteren 91-125 µm boyutlarında ki % 5 tufal takviyeli numunenin hacim kaybı 300 mg olarak gerçekleşmiştir. Böylece takviyesiz numune, en iyi aşınma performansı gösteren numuneden 3 kat, en düşük hacim kaybı gösteren numuneden de 0.1 kat daha kötü aşınma performansı göstermiştir. Numunelerin aşınma performansı takviye edici toz oranına bağlı olarak kıyaslandığında kompozit içerisinde artan tufal oranına bağlı olarak düştüğü görülmektedir. En iyi aşınma direnci en düşük tufal takviyeli yani % 5 oranında tufal ile takviye edilen numunelerde görülürken en kötü aşınma direnci en yüksek oranda tufal ile takviye edilen (%25) numunelerde

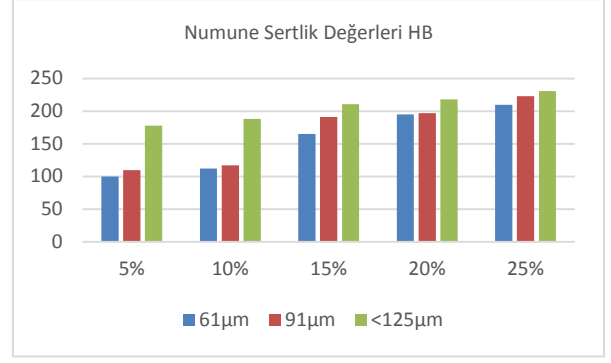
görülmüştür. Hatta diğer takviye oranları ile kıyaslandığında % 25 lik takviye oranının bir kırılma noktası olduğunu söylemek mümkündür. Çünkü bu oranda hacim kayıpları ani bir sıçrama göstermiştir. Bu durumu sebebini kompozit içerisinde artan takviye edici oranına bağlı olarak matris bağın dokusunun zayıflamasına veya aşındırma esnasında aşındırıcı bilya ile kompozit arasında meydana gelen kuvvetlerin varlığına bağlamak mümkündür. Şöyle ki karşı yüzeyin aşındırılabilmesi için aşındırıcı bilyanın bu yüzeye nüfuz etmesi gereklidir. Bunu bilya üzerine uyguladığımız 7 N luk normal yük sağlayacaktır. Burada kompozit numune ne kadar yüksek hacim veya yüzey sertliğine sahip ise aşındırıcının karşı yüzeye batması o kadar zor olacaktır. Numunelere ait sertlik değerleri Şekil 5'deki grafikte verilmiştir.



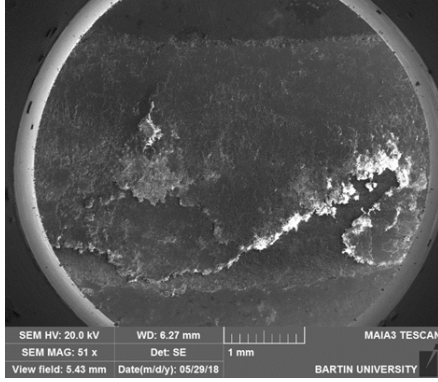
Şekil 4. 7 N luk yük altında numunelerde meydana gelen hacim kayıpları (mm³)

Şekil 5’de görüleceği üzere, hem kompozit numune içerisinde artan tufal oranına bağlı olarak hemde tufal partikül tane büyüklüğünün artmasına bağlı olarak numune hacim sertliklerinin arttığı görülmektedir. Hatta deneylerde elde edilen hacim kayıplarının tam tersine en yüksek sertlik değeri % 25 tufal takviyeli numuneler de görülmüştür.

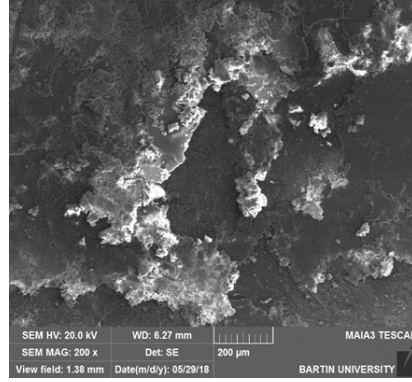
Numunelerin aşındırılması için gerekli diğer kuvvet ise, kesme kuvvetidir. Kompozit yüzeyinden partikül kaldırılabilmesi için bu iki kuvvetin yüksek olması gereklidir. Bu kuvvetlere karşı koyan direnç ise kompozit içerisindeki sert faz olan tufal partikülleridir. Toz boyutuna göre kompozit numunelerin aşınma performansları değerlendirildiğinde toz boyutunun büyümesi ile aşınma direncinin arttığı anlaşılmaktadır. Alahelisten [17-20] ve arkadaşları kompozit malzemelerin tribolojik davranışları malzemelerin mikroyapısal davranışları ve tribolojik sistemin yükleme şartlarına bağlı olduğunu belirlemiştir.



Şekil 5. 7 N luk yük altında numunelerde meydana gelen hacim kayıpları (mm³)

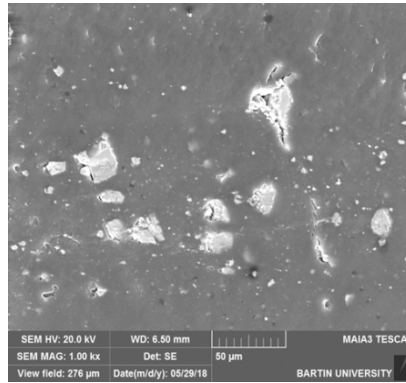


Şekil 6. Takviyesiz kompozit numunenin aşınma sonrası iz SEM fotoğrafı



Bu durumu birim alanda aşındırıcı bilya ile temas eden tufal oranının artması ile ilişkilendirmek mümkündür. Çünkü malzemelerin sertliği arttıkça kendisine batmak isteyen aşındırıcı bilyeye karşı koyma direnci o oranda artmış olacaktır. Artan takviye edici oranını ile birlikte birim alana düşen tufal partiküllerinin sayısında artış olacağından dolayı yumuşak polimer matris aşındırıcının olumsuz etkilerinden korunmuş olacaktır.

Takviyesiz clarofast’ın 7 N luk yük altında aşındırma işlemine tabi tutulduktan sonra aşınma yüzeyinden farklı büyütmeyle alınan SEM fotoğrafları Şekil 6’da sırası ile verilmiştir. Yaklaşık 3 mm genişliğinde meydana gelen aşınma izi aşırı plastik deformasyon sonucunda meydana gelmiştir.



Şekil 7. Tufal tozları (67-91µm) ile takviye edilen kompozit numunenin aşınma sonrası iz SEM fotoğrafı

% 25 Takviyeli clarofast’ın 7 N luk yük altında aşındırma işlemine tabi tutulduktan sonra aşınma yüzeyinden farklı büyütmeyle alınan SEM fotoğrafları Şekil 7’de verilmiştir. Matris içerisinde kırılan tufal taneleri görülmektedir. Kırılan bu sert partiküller daha sonrasında bir abrasiv aşındırıcı etkisi göstermektedir.

% 5 Takviyeli clarofast’ın küçük ve büyük görüntülerdeki SEM fotoğrafları incelendiğinde şekil 8’de pürüzsüz bir yüzey dikkat çekmektedir.

4. Sonuç

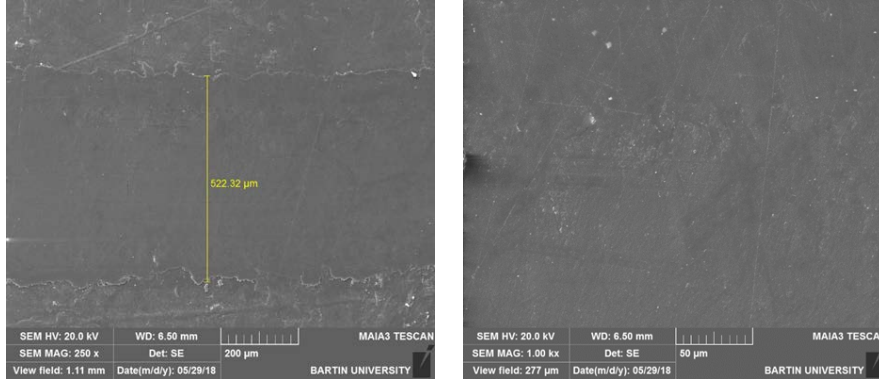
Endüstriyel atık olan tufal tozları farklı boyut ve takviye edici oranlarında takviye edilerek kompozit olarak üretilen clarofast’ın kuru kayma deneyleri sonucunda elde edilen sonuçlar aşağıda özetlenmiştir.

1. Takviye edici oranına bağlı olarak numunelerdeki sertlik değeri 3 kattan daha fazla bir artış göstermiştir.
2. Numunelerin hacim kayıpları; artan yüke bağlı olarak artış, artan takviye edici oranına bağlı olarak düşüş göstermiştir.

Numunelerin aşınma dirençlerinde artan takviye edici oranı ile doğrudan bir ilişki görülmemiştir.

3. Aşınma mekanizması olarak, takviyesiz kompozit numunelerde aşırı plastik deformasyon, mikro çatlama görülürken takviyeli

kompozitlerde daha pürüzsüz bir aşınma tabakası görülmüştür. Ancak takviye edici partiküllerin kırıldığı görülmüştür.



Şekil 8. Tufal tozları (91-125 µm) ile takviye edilen kompozit numunenin aşınma sonrası iz SEM fotoğrafı

Kaynaklar

1. Wypych G. Handbook of fillers. 2nd ed. Toronto: ChemTec Publishing; 1999.
2. Hauptert F, Wetzel B. Reinforcement of thermosetting polymers by the incorporation of micro- and nanoparticles. In: Friedrich K, Fakirov S, Zhang Z, editors. Polym compos from nano- to macro-scale. Springer Science+Business Media, Inc.; 2005. p. 45–62.
3. Hoffman DW, Roy R, Komarneni S. Diphasic Xerogels, a new class of materials: phases in the system Al₂O₃-SiO₂. J Am Ceram Soc 1984;67:468–71.
4. Ashby MF, Brechet YJM. Designing hybrid materials. Acta Mater 2003;51:5801–21.
5. Thwe MM, Liao K. Durability of bamboo-glass fiber reinforced polymer matrix hybrid composites. Compos Sci Technol 2003;63:375–87.
6. Fu SY, Xu G, Mai YW. On the elastic modulus of hybrid particle/short-fiber/ polymer composites. Compos Part B Eng 2002;33:291–9.
7. Karger-Kocsis J. Reinforced polymer blends. In: Paul DR, Bucknall CB, editors. Polym blends, vol. 2. New York: John Wiley & Sons; 2000. p. 395.
8. Zuoyong Dou, Gaohui Wu, Xiaoli Huang, Dongli Sun and Longtao Jiang // Composites: Part A 38 (2007) 186.

9. M. Ramachandra and K. Radhakrishna // Wear 262 (2007) 1450.
10. Onur Sayman // Composite structures 53 (2001) 419.
11. B.G. Park, A.G. Crosky and A.K. Hellier // Composites: Part B 39 (2008) 1270.
12. B.G. Park, A.G. Crosky and A.K. Hellier // Composites: Part B 39 (2008) 1257.
13. Barbara Previtali, Dante Poggi and Cataldo Taccardo // Composites: Part A 39 (2008) 1606.
14. Sanjeev Das, Siddhartha Das and Karabi Das // Composites Science and Technology 67 (2007) 746.
15. Sedat Ozden, Recep Ekici and Fehmi Nair // Composites: Part A 38 (2007) 484.
16. T.S. Srivatsan, Meslet Al Hajri and V.K. Vasudevan // International Journal of Fatigue 27 (2005) 357.
17. O. Sayman, H. Akbulut and C. Meric // Composites and Structures 75 (2000) 55.
18. A. Alahelisten, F. Bergman, M. Olsson and S. Hogmatk, Wear, 165 (1993) 1221
19. N. Axen, and K.H. Zmn Gahr, Wear, 157 (1992) 189.
20. A. Wang and H.L. Rack, Wear, 146 (1991) 337.



Bartın Üniversitesi Mühendislik ve Teknoloji Bilimleri Dergisi



K-Gezgin Satıcı Probleminin Emperyalist Rekabetçi Algoritması ile Kümeleme Tabanlı Optimizasyonu

Oktay KÖSE^{1*}, Pakize ERDOĞMUŞ²

¹Karabük İl Milli Eğitim Müdürlüğü, Karabük / TÜRKİYE

²Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Düzce / TÜRKİYE

Geliş Tarihi: 21.01.2019 Düzeltme Tarihi: 25.03.2019 Kabul Tarihi: 09.05.2019

Özet

Gezgin Satıcı Probleminde (GSP), bir satıcının n adet şehrin hepsine yalnızca bir kez uğrayarak tekrar başladığı şehre dönmesi incelenmektedir. GSP’de amaç en kısa mesafeyi bulup maliyeti düşürmektir. Bu çalışmada Türkiye’nin 81 ili için en kısa turun bulunması amaçlanmıştır. Bu amaçla önce tek gezgin satıcı ile en kısa turun bulunması için, sezgisel algoritmalarından Emperyalist Rekabetçi Algoritması(ERA) kullanılmıştır. Türkiye haritası üzerinde çalışma yapılmış, ERA ile elde edilen sonuçlar, Benzetimli Tavlama (BT), Parçacık Sürü Optimizasyonu (PSO) ve Karınca Kolonisi Algoritması (KKO) ile de karşılaştırılmıştır. Daha sonra 81 il k-means kümeleme ile k=7,8 ve 9 için kümelenecek her küme için alt GSP problemleri çözülmüştür. En sonunda ise bulunan küme merkezlerine en yakın şehirler seçilerek, bu şehirler üzerinden bir GSP turu hesaplanmıştır. Bu çalışmanın sonuçlarının özellikle lojistik açısından kullanılabilir pratik sonuçlar olması için, tüm sonuçlar Türkiye haritası üzerinde işaretlenmiş ve tur uzunlukları sonuçları hem Karayolları Genel Müdürlüğü, hem de Google Map üzerinden hesaplanarak sunulmuştur.

Anahtar Kelimeler: Gezgin Satıcı Problemi, Emperyalist Rekabetçi Algoritması, K-Ortalama, Sezgisel Algoritmalar

Clustering Based Optimization of K-Travelling Salesman Problem with Imperialist Competitive Algorithm

Abstract

In Travelling Salesman Problem(TSP), it is searched for a possible shortest route for a salesman who starts from a city, visits each city only once and returns to the initial city, among n given cities. The aim of TSP is finding the shortest route and minimizing the traveling cost. In this study, it is aimed to find the shortest tour for 81 cities in Turkey. With this aim, firstly, to find the shortest tour for 81 cities, Imperialist Competitive Algorithm(ICO) has been used in this study. ICO is one of the heuristic search algorithms, which has been developed in 2008 for continuous optimization problem solutions. The results have been shown on the map, for a realistic view. The results found by ICO have been compared with Simulated Annealing(SA), Particle Swarm Optimization(PSO) and Ant Colony Optimization(ACO). The results have been shown that PSO and SA can't compete with the ACO and ICO. Although the population number is 100, PSO results don't proceed the other population based algorithms. ACO is the most stable algorithm both in view of solution time and optimal path length. This is because of the fact that ACO is originally developed by Dorigo for TSP problem solution. But the best shortest tour has been found with ICO. So, ICO can compete with ACO for TSP problems. Secondly, the 81 cities have been clustered for k=7,8 ve 9 and each cluster found for these clusters have also been solved as sub TSP problems. So with this way, GSP problem for 81 cities is solved with sub 8,9,10 TSP problems. Since the results can be used practically, especially for logistic, results are shown on map. And the tour lengths have been calculated both with General Directorate of Highways and Google Map data. The second part of this study has been shown that the practical solution of TSP is also giving near optimal solutions.

Keywords: Travelling Salesman Problem, Imperialist Competitive Algorithm, K-Means, Heuristic Algorithms

1. Giriş

Günümüzde lojistik, hükümetler, şirketler ve araştırmacılar için, özellikle taşımacılık hizmetlerini geliştirmede büyük önem taşımaktadır [1]. GSP ve Araç Rotalama Problemi(ARP) en önemli taşımacılık problemlerinden olup, ülkeler için kendi eyalet veya şehirleri için en kısa rotanın bulunması, stratejik bir önem arz etmektedir. GSP bir satıcının n adet şehrin hepsine yalnızca bir kez uğrayarak tekrar başladığı şehre döneceği en kısa turu bulma problemidir. En kısa turu bulma problemi, aslında n adet şehrin bir sıralaması olduğu için, bir kombinatoriyel optimizasyon problemidir [2]. Burada amaç en kısa mesafeyi verecek turu bulup, maliyeti düşürmektir [3].

GSP, 1930’lu yıllarda Karl Menger tarafından matematiksel olarak tanımlanmıştır [4]. Problemden maliyet, noktaların uzaklıkları üzerinden hesaplanmakta ve en kısa tur en düşük maliyeti göstermektedir. Şehir sayısı arttıkça mümkün çözüm sayısı üstel olarak arttığı için bu problemi polinomal sürede çözecek bir algoritma yoktur ve problem NP(Non-Polinomial)-Zor (NP-Hard) problemler içerisinde [5].

Gerçek hayat problemlerine çözüm aranırken, en iyi çözümü, en kısa sürede çözmek gibi birbirleri ile çelişen iki hedef vardır. İşte sezgisel optimizasyon bu iki amacı maksimize eden denge noktasını bulmayı amaçlar. Sezgisel yöntemler, olası tüm çözümler içerisinde çok uzun çözüm süresinde en iyi çözümü bulmak yerine, makul çözüm sürelerinde iyiye yakın çözümleri bulmayı amaçlar

Sezgisel algoritmalar genel olarak, fiziksel süreçler ve yasalardan esinlenen algoritmalar, doğal olaylardan esinlenen algoritmalar ve yiyecek arama davranışından esinlenen algoritmalar olarak sınıflandırıldığı gibi [6] fizik tabanlı, kimya tabanlı, biyolojik tabanlı, müzik tabanlı ve sosyal tabanlı olarak da sınıflandırılabilirler. Fizik tabanlı algoritmalar, ısı işlem ve elektromanyetizma algoritması, kimya tabanlı algoritmalar yapay kimyasal reaksiyon algoritması, biyolojik tabanlı algoritmalar genetik algoritma, KKO, arı kolonisi algoritması, müzik tabanlı algoritmalar armoni arama algoritması ve sosyal tabanlı algoritmalar ise ERA örnek verilebilir[7].

ERA, ilk olarak 2007 yılında IEEE (Uluslararası Elektrik-Elektronik Mühendisliği) Evrimsel Hesaplama (Evolutionary

Computation) konferansında, sürekli optimizasyon problem çözümü için önerilmiş bir algoritma olup, literatürde ERA ile çok sayıda çalışma yapılmıştır. Bu algoritma dünya üzerindeki ülkelerden gruplar oluşturup bu grupları yöneten bir emperyalist ülke oluşturmakta ve daha sonra emperyalist ülkeler arasında yarış oluşturarak tek bir emperyalist ülke kalmasını amaçlamaktadır [8].

ERA, adaptif parametre kontrolü [9], İskelet yapısının optimum dizaynı [10], Mayın patlatıldıktan sonra partikül büyüklüğü dağılımının tahmini [11] ve GSP problem çözümü için kullanılmıştır [12].

Sürekli optimizasyon problem çözümü için geliştirilmiş olan ERA algoritmasının yeni bir versiyonu kesikli optimizasyon problem çözümü için geliştirilmiştir. Geliştirilen algoritma 50-199 arası şehre sahip 14 problem üzerinde test edilmiş ve bazı problemlerde en iyiye yakın çözümleri elde edilmiştir [12]. Bir başka çalışmada geliştirilmiş GSP, ERA ile çözülmüştür [13].

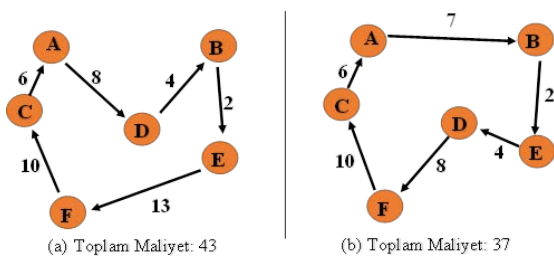
Bu çalışmada lojistik açıdan önemli olan Türkiye'nin 81 ili için GSP problemi için en kısa turun bulunması amaçlanmıştır. Bu amaçla daha önce birçok GSP çözümünde kullanılmış ve başarımlı test edilmiş sosyal tabanlı bir algoritma olan ERA kullanılmıştır. GSP Türkiye haritası üzerinde çalışılmıştır. İllerin koordinatları ve iller arası mesafeler elde edilerek, en kısa tur bulunmaya çalışılmıştır. Problem çözümü için MATLAB© programı kullanılmış ve grafiksel sonuçlar sunulmuştur. Sonra 81 il 7,8 ve 9 kümeye ayrılarak her bir küme için GSP çalıştırılmış ve bulunan turların merkezine en yakın şehirler üzerinden de tüm bu alt turları bağlayan tur hesaplanmıştır.

Bölüm 2'de GSP hakkında genel bilgi verilmiş, Bölüm 3'de ERA algoritmasına yer verildikten sonra, Bölüm 4'de önce Türkiye'deki iller üzerinde 1-GSP en kısa turu ERA algoritması ile bulunmuş, diğer sezgisel algoritmalar ile karşılaştırması yapılmıştır. Daha sonra 81 il önce k=7,8, ve 9 için kümelenecek 7-GSP, 8-GSP ve 9-GSP çalışma sonuçları elde edilmiş ve sonuçlar 1-GSP ile hem zaman hem de tur uzunluğu bakımından karşılaştırılmıştır.

Bölüm 5'te çalışmada elde edilen sonuçlar sunulmuştur.

2. Gezgin Satıcı Problemi

GSP bir şehirden başlayıp bütün şehirlere sadece bir kez uğrayarak tekrar başladığı şehre dönen en kısa mesafeli turu bulma problemidir [14,4]. Şekil 1.'de görüldüğü gibi her şehre en az bir kere uğrayarak yapılan en kısa kapalı tur Hamilton çevrimi olarak adlandırılır [15].



Şekil 1. Hamilton Çevrim Örnekleri

GSP günlük hayatın içinde bir problem olmasından dolayı nakliye ve depolama gibi alanlarda çok kullanılır. GSP V düğümler kümesi, E kenarlar kümesi olmak üzere $G(V,E)$ şeklinde bir graf olarak tanımlanabilir.

Rastgele bir $v \in V$ düğümden başlayıp tüm düğümleri en az bir kez ziyaret ettikten sonra tekrar v düğüme dönme olayı bir "tur" olarak nitelendirilir. Denklem (2.1) de karar değişkenleri x_{ij} için tanımlama yapılmıştır.

$$x_{ij} = \begin{cases} 1, & i' den j' ye gidildiğinde \\ 0, & diğer durumlarda \end{cases} \quad (2.1)$$

c_{ij} = i'nci düğüm ile j'nci düğüm arasındaki mesafe/maliyet olmak üzere bir optimizasyon problemi olarak ele alınır. GSP, amaç fonksiyonu ve kısıtları (2.2),(2.3),(2.4),(2.5) ve (2.6)'te verilen bir optimizasyon problemi olarak modellenir.

Minimum

$$\sum_{i \neq j}^n c_{ij} x_{ij} \quad (2.2)$$

Kısıtları altında;

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (2.3)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \quad (2.4)$$

$$\sum_{i,j \in S}^n x_{ij} \leq |S| - 1 \quad (2.5)$$

$$S \subset V, 2 \leq |S| \leq n - 2 \quad (2.5)$$

$$x_{ij} \in \{0,1\} \quad (2.6)$$

$$i, j = 1, \dots, n \quad i \neq j \quad (2.6)$$

Denklem (2.2) tur mesafesini minimum yapan amaç fonksiyonudur. Denklem (2.3) ve (2.4) her bir düğüme bir kez uğramayı sağlayan kısıtlardır. Denklem (2.5) ve (2.6) oluşabilecek alt turları engellemek için kullanılan kısıttır. [15].

Standart GSP'de amaç, listedeki tüm şehirlere en az bir kere uğrayarak tekrar başladığı şehre dönerek minimum maliyeti bulmaktır. Şehir sayısı çoğaldıkça çözümü bulmak için gereken süre üstel olarak artmaktadır. Mesela; şehir sayısı x ve gidilebileceği komşu şehir sayısı y olursa problemin çözüm sayısı x^y kadar artar [16]. Bu sebeple de şehir sayısı arttıkça problemin çözüm kümesi ve maliyet artmaktadır.

GSP'nin çözümü için genel olarak iki yol izlenir. Bunlardan birincisi çözümün garanti olduğu kesin algoritmalarıdır. Diğeri ise kesin çözümü garanti etmeyen düşük maliyetli kısa sürede yakın çözümler hedefleyen yaklaşık algoritmalarıdır [17].

GSP için kesin yöntemler en iyi çözümü garanti ederler. Sayma, dal-sınır ve dal-kesme yöntemi kesin yöntemler için verilebilecek örneklerdir.

Sayma yöntemi; en sade şekliyle gidilebilecek bütün turlar tek tek hesaplanır ve bu turlar içinde en kısa olan seçilerek çözüm elde edilmiş olur [18]. x şehirli asimetrik bir GSP için çözüm sayısı $(x-1)!$ olarak hesaplanır. Büyük problemler için düşünüldüğünde çözüm imkansız denebilecek boyutlara ulaşmaktadır.

Dal-sınır algoritması; problemi dallara ayırarak çözme yoluna gider. Amaç problemi dallara ayırarak bölümlenmek ve her bölümün ayrı araştırılması yapılarak çözüm yolunu kolaylaştırmaktır [19].

Dal-kesme yöntemi; kesme düzlemi algoritması ile dal-sınır algoritmasının birleşiminden oluşmaktadır. Tam sayı çözümü olan bir problemin kesme düzlemi yaklaşımı ile çözmek verimli olmayacağı için aynı zamanda dallandırma yapmak gerekmektedir. Dal-sınır yöntemi kesme düzlemi algoritması uygulanarak daha performanslı hale getirilebilir. Dallandırma yapmadan kesme yapılacağı gibi her dalı içinde kesme uygulanabilir [20].

GSP çözümünde kullanılan kesin yöntemler şehir sayısı az olan problemlerin çözümü için daha kullanışlıdır. Çok sayıda şehirleri olan bir GSP üzerinde kesin yöntemler ile bir sonuç bulunabilir ama bu yıllar sürebilir. Bundan dolayı sezgisel yöntemleri kullanmaya zorunlu kalınmıştır. Klasik sezgisel yöntemlerin birçoğu 1960-1990 yılları arasında çalışılmış ve daha geniş çözüm kümesine sahip

problemlerde etkili olmuştur [21]. Klasik sezgisel yöntemler; rota yapılandırma yöntemleri, rota geliştirme yöntemleri ve iki aşamalı yöntemler olarak üç grupta yer almaktadır.

Rota yapılandırma yöntemleri; problemi çözerken adım adım problem içinde ilerler ve bir sonuç bulana kadar devam ederler. Çözümün geliştirilmesi ve düzeltilmesi için bir işlem yapmazlar, bulunan sonucun doğruluğu kesin değildir. Bu yöntemlere, en yakın komşu algoritması, açgözlü algoritması, ekleme algoritması örnek olarak gösterilebilir [22].

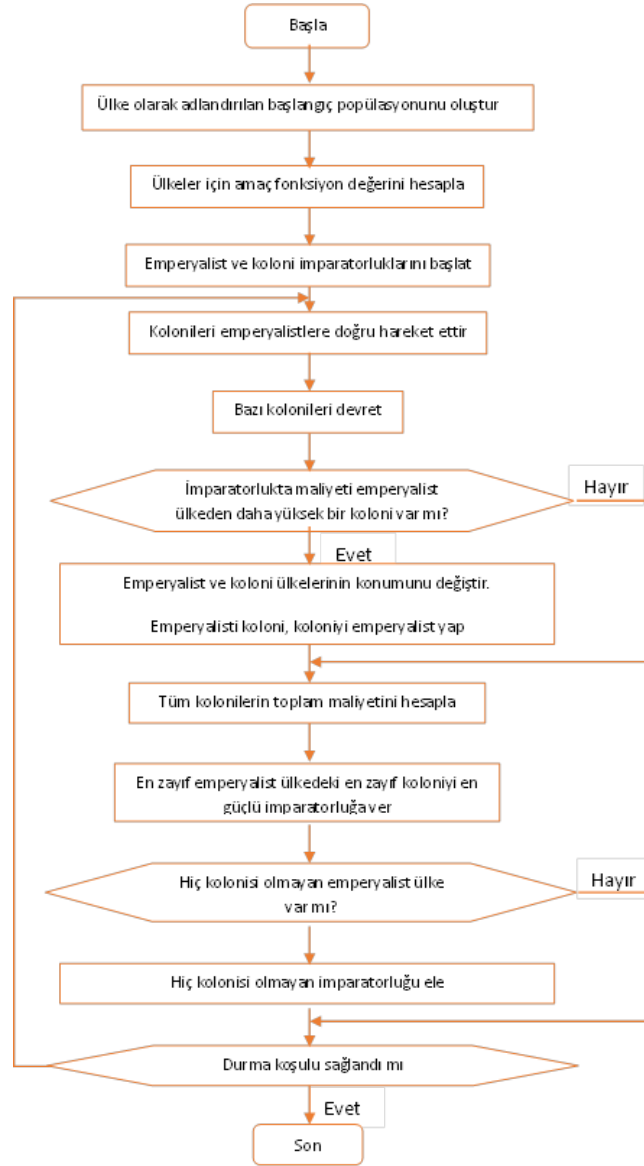
Rota geliştirme yöntemleri; problem için bulunmuş bir çözümün iyileştirilmesini amaçlamaktadırlar. Rota yapılandırma yöntemlerinden farkı ve avantajı daha önceden bulunmuş bir çözüm üzerinde çalışırlar. Bu yöntemler her çözüm kümesi elemanı içinde iyileştirmeler yaparak bulunan sonucun daha iyi hale getirilmesini sağlar. 2-opt, 3-opt, Lin-Kernighan rota geliştirme yöntemleri için örnek olarak verilebilir [22].

İki aşamalı algoritmalar; problemi iki alt problem haline getirir. Bu problemlerin biri rotalama diğeri kümelemedir. Bunların sırası önemli değildir, önce gruplayıp sonra rotalayabilir veya önce rotalayıp sonra gruplayabilir. Süpürme algoritması, taç yaprağı algoritması, Christofide, Mingozzi ve Toth iki aşamalı metodu bunlara örnek verilebilir [23].

Metasezgisel yöntemler, en iyi ve en az maliyetli sonuç için çözüm kümesi içinde araştırma (exploration) ve geliştirme (exploitation) için farklı yaklaşımları bir araya getiren tekrarlayıcı işlemler yapan algoritmalarıdır [24].

Tavlama benzetimi; metallerin çok yüksek sıcaklıklarda eritilmesi ve daha sonra düzenli olarak soğutulmuş düzgün kristal yapı elde edilmesini esas alan bir algoritmadır. Yüksek sıcaklıklar araştırma aşamasına düşük sıcaklıklar geliştirme aşamasına denk düşer. Metalin donma noktası, optimum çözüme denk gelir [25].

Genetik algoritma; en iyi olan hayatta kalır prensibini esas alan popülasyon temelli bir algoritmadır. Problem yapısı için kromozomlar kullanılır. Kromozomlardan başlangıç için rastgele bir popülasyon üretilir. Daha sonra popülasyon içinden seçilen kromozomlara çaprazlama ve mutasyon fonksiyonları uygulanarak yeni kromozomlar oluşturulur. Eğer oluşturulan yeni kromozomlar daha iyi bir sonuç ortaya koymuşsa ileri nesillere aktarılır, daha kötü bir sonuçta sistemden elenirler. Böylece yeni popülasyonlar oluşturulur. Bu süreç sonlandırma kriterine ulaşıncaya kadar tekrarlayan bir şekilde devam eder [26].



Şekil 2. Emperyalist rekabetçi algoritmasının akış diyagramı

PSO; kuş ve balık sürülerinin hareketlerinden esinlenilerek geliştirilmiş bir algoritmadır. PSO'nda her parçacığa ait bir hız bilgisi vardır. Parçacığın hızı diğer parçacıklardan alınan verilerle karşılaştırılır. Her bir aşamada hesaplanan hız ile bir önceki aşamadaki hız karşılaştırılır ve buna göre popülasyon içinde güncelleme yapılarak daha iyi bir nesil oluşturulur. PSO'nun hızlı yakınsaması ve ayarlanması gereken az sayıda parametre içermesi sebebi ile yapay sinir ağı eğitimi, bulanık sistem kontrol parametre ayarlaması ve fonksiyon optimizasyonu gibi bir çok alanda başarı ile uygulanmıştır [27].

KKO; karınca sürülerinin kendi yiyeceklerinin bulmak için yaptıkları hareketlerden esinlenerek çalışmaktadır. Karıncalar yemeklerini bulmak için yuvalarından ayrılırlar ve yemeğe giderken yola feromon sıvısı bırakırlar. Feromenu karıncalar yuvaya dönmek için kullanırlar. Ayrıca feromon daha sonra yemek aramaya çıkacak karıncalar içinde yol gösterici olacaktır. O yoldan ne kadar çok karınca giderse o yol üzerindeki feromon miktarı da yoğun olacağı için daha çok tercih edilen bir rota belirlenmiş olur. Küçük bir ihtimalde olsa her karınca yoğun feromonu takip edecek değildir. Feromon miktarı az olan yoluda seçebilir. Böylece de alternatif yollar ve yiyecek kaynakları bulunmuş olur buda diğer rotalar üzerinde çalışma imkanı sağlar [28].

3. Emperyalist Rekabetçi Algoritma

ERA, her evrimsel algoritma gibi bir başlangıç popülasyonuna sahiptir. Bu popülasyondaki her eleman bir ülke olarak düşünülür ve bu ülkeler arasında bazıları emperyalist olarak belirlenir, diğer ülkeler de bu emperyalist ülkelerin kolonileri olarak belirlenir. Başlangıç popülasyonundaki tüm koloniler başlangıç emperyalist ülkeleri arasında güçlerine doğru orantılı olarak dağıtılır. Koloniler dağıtıldıktan sonra imparatorlukların güçleri, emperyalistlerin gücüne kolonilerin güçleri ortalamasının belirli bir oranı eklenerek hesaplanır.

Daha sonra imparatorluklar arasında emperyalist savaş başlayacaktır. Bu savaşta bir imparatorluk gücünü arttıramazsa bu savaşı kaybeder. Bu savaşta güçsüz imparatorluklar güçsüzleşerek yok olurlarken, güçlü imparatorluklar ise güçlenerek savaşı kazanmış olur. Tek bir imparatorluk kalana kadar bu savaş bu şekilde devam eder. Bütün ülkeler o imparatorluğun kolonisi olur. Yeni oluşan uygun dünya düzeninde bütün ülkeler imparatorluğunun gücüne sahip olacaktır. Algoritmanın akış diyagramı Şekil 2'de gösterilmiştir [8].

Algoritmanın başlangıcında, bağımsız değişkenlerin alt ve üst sınırları aralığında rastgele ilk çözümler üretilir. D boyutlu bir problemde, bir ülke(çözüm) $1 \times D$ dizisi olarak tanımlanır. Bu dizi (3.1) denklemindeki gibi tanımlanır.

$$\text{ülke} = [u_1, u_2, \dots, u_D] \quad (3.1)$$

Bir ülkenin maliyeti (Cost) u_1, u_2, \dots, u_D değerlerindeki değer fonksiyonu ile ölçülür ve (3.2) denklemindeki gibi tanımlanır.

$$\text{Cost} = f(\text{ülke}) = f(u_1, u_2, \dots, u_D) \quad (3.2)$$

Başlangıç için N adet ülke oluşturulur. En güçlü N_{imp} sayıdaki ülke imparatorluk oluşturmak için seçilir. Koloniler için ise geride kalan N_{koloni} gösterilir. Böylece emperyalist ve koloni olmak üzere iki tip ülke oluşturulur.

İlk imparatorlukları oluşturmak için koloniler emperyalist ülkelere güçleri oranında gönderilir ve emperyalistlerin güçleri koloni sayılarıyla orantılı olur. Bu oranı hesaplamak için (3.3) denklemindeki gibi tanımlanır.

$$C_{n_i} = c_n - \lfloor \text{maks} \rfloor_i \{c_i\} \quad (3.3)$$

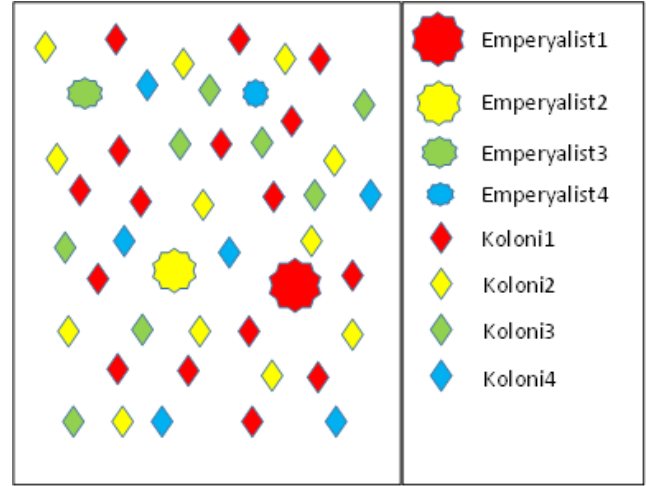
n_i emperyalistin maliyeti C_{n_i} , normalize edilmiş maliyeti ise c_n ile gösterilir. Her emperyalistin normalize edilmiş gücü denklem (3.4)'deki gibidir.

$$u_{n_i} = c_n / (\sum_{i=1}^{N_{\text{imp}}} c_i) \quad (3.4)$$

Emperyalistin normalize edilmiş gücü sahip olduğu koloni sayısına bağlıdır. Başlangıç koloni sayısı da denklem (3.5)'deki gibi gösterilir.

$$N_{\text{koloni}} = \text{round} \{u_{n_i} \cdot N_{\text{koloni}}\} \quad (3.5)$$

İmparatorluğun başlangıç koloni sayısı N_{koloni} ile gösterilirken, N_{koloni} toplam koloni sayısını gösterir. Rastgele seçilen N_{koloni} sayıdaki koloni emperyalist ülkelere dağıtılır. Kolonilerle birlikte emperyalist ülkeler n_i imparatorluğu kurar. Başlangıçta oluşturulan imparatorlukların ve güçleri oranında sahip oldukları koloniler Şekil 3'de gösterilmektedir [29, 30].



Şekil 3. İmparatorlukların başlangıç popülasyonları

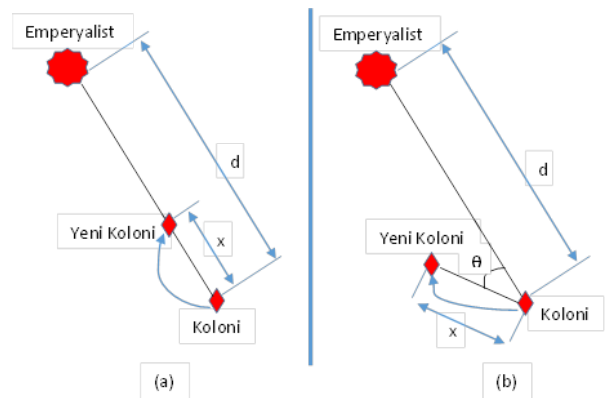
Emperyalist ülkeler güçleri oranında koloni sayılarını arttırlar. Arttırma işlemi kolonilerin emperyalist ülkelere doğru yönelmesiyle başlar. Bir koloninin vektörel biçimdeki x birimlik hareketi Şekil 4 (a)'da gösterilmektedir. x değeri rastgele seçilmiştir. Kolonilerin emperyalist ülkelere doğru hareketi için $\beta > 1$ şartı aranır.

$$x \sim U(0, \beta \cdot x) \quad (3.6)$$

Emperyalist ülkenin diğer kolonilere de ulaşmak için Şekil 4 (b)'deki gibi hareket yönüne rastgele seçilmiş bir θ değerindeki açıdan oluşan sapma eklenir.

$$\theta \sim U(-\gamma, \gamma) \quad (3.7)$$

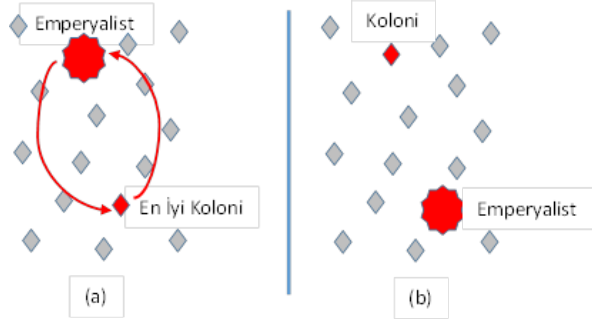
γ değeri ne kadarlık bir sapmanın olacağını belirler ve β ile γ değeri rastgele seçilmiştir [8, 31].



Şekil 4. (a) Koloninin emperyaliste olan hareketi (b) Koloninin yeni pozisyonu

Eğer kolonilerin konumları emperyalistlere göre daha düşük maliyette ise o koloni, emperyalist ülke ile yer değiştirme işlemini gerçekleştirebilir. Şekil 5 (a)'da koloni ve emperyalistin eski yeri,

Şekil 5 (b)'de ise yer değiştirme işlemi gerçekleşikten sonraki koloni ve emperyalistin yeni yeri gösterilmektedir.



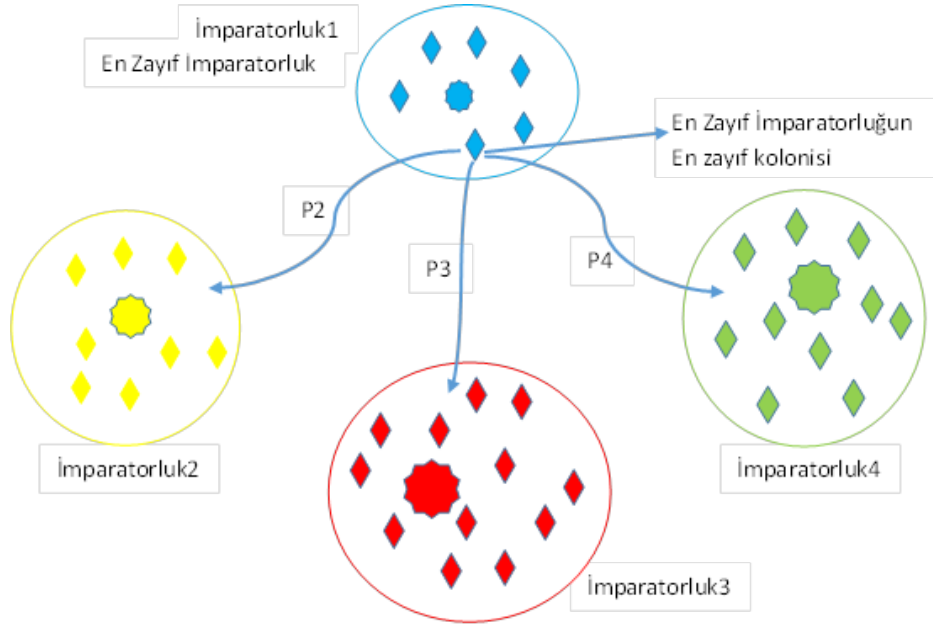
Şekil 5. (a) Koloni ve emperyalistin yer değişimi (b) Yer değiştirmeden sonraki durum

İmparatorlukların toplam gücü o imparatorluğun emperyalist ülkesinin gücüne bağlıdır. Kolonilerde belli bir oranda imparatorluk içindeki güce etki ederler ama asıl belirleyici emperyalist ülkelerin gücüdür. İmparatorluğun toplam gücü (3.8) denklemde gösterilmiştir.

$$T. [C.]_{-n} = \text{Maliyet}([empyalist]_{-n}) + \xi \text{ ortalama} \{ \text{maliyet}([imparatorluk \text{ kolonileri}]_{-n}) \} \quad (3.8)$$

Burada ξ bir ile sıfır arasında bir değer alınır. ξ değeri ne kadar küçük seçilirse imparatorluğun toplam gücündeki emperyalist ülkenin etkisi o kadar fazla olur, büyük olması da kolonilerin etkisini artırır. ξ için bir çok uygulamada 0,1 değeri uygulanır [8].

Bu yarışta bütün imparatorluklar diğer imparatorlukların kolonilerini ele geçirecekleri kendi güçlerini arttırmak isterler. Böylece zayıf imparatorluklar gitgide daha da zayıflamaya, güçlü ülkelere daha güçlü olmaya başlayacaktır. Bu yarışın genel modeli Şekil 6'da gösterilmiştir [30].



Şekil 6. Emperyalist Yarışma

4. ERA ve K-means Kümeleme ile En Kısa 7-GSP, 8-GSP ve 9-GSP Tur Sonuçları

Bu uygulama Matlab2014 programlama dili ile geliştirilmiştir. Uygulamada Türkiye'deki iller ve il merkezleri arasındaki karayolu mesafeleri kullanılarak sonuçlara ulaşılmıştır. Türkiye il merkezlerinin koordinatları ve Karayolları Genel Müdürlüğünden [32] alınan iller arası mesafe tablosu program içine yerleştirilmiş ve uygulamada bu koordinat ve mesafeler üzerinden hesaplamalar yapılmıştır.

Çalışmada tüm sonuçlar en az 30 kere çalıştırılarak en iyi ve ortalama değerler elde edilmiştir. Çalışmada "Intel Core i5 1.70 GHz işlemci, 8 Gb Ram, 64 bit Windows 10 versiyon işletim sistemi" kullanılmıştır.

İlk olarak ERA da popülasyon sayısı 80 alınarak en uygun iterasyon sayısını bulmak için 1000, 2500 ve 5000 değerleri ile 30 kez çalıştırılarak Tablo 1'deki en kötü süre, en iyi süre, ortalama süre, en iyi tur ve en kötü tur değerleri bulunmuştur.

Tablo 1. ERA için iterasyon sayısına göre bulunan sonuçlar

İterasyon sayısı	En kötü süre (sn)	En iyi süre (sn)	Ortalama süre (sn)	En kısa tur (km)	En uzun tur (km)
1000	36,125	34,5172	34,981	12592	13806
2500	87,561	85,3023	85,728	10805	11092
5000	173,567	169,3863	171,294	10517	10952

Tablo 1'e bakıldığında iterasyon sayısı arttıkça süre artsa da tur için daha iyi çözümler bulunmuştur. En iyi tur değerlerinin iterasyon sayısı 5000 alındığında çıktığı görülmüştür.

Ayrıca iterasyon sayısı 5000'den fazla alındığında süre artmaya devam etmiş fakat tur için daha iyi sonuçlar elde edilmemiştir.

Tablo 2'de en iyi değerlerin bulunduğu iterasyon sayısı 5000 ve sonuçların değişmediği görülen iterasyon sayısı 7500 ile elde edilen 30 sonuçtan en iyi 5 sonuç gösterilmiştir.

Daha sonra iterasyon sayısı 5000 sabit tutulup popülasyon sayısı 40,80 ve 120 değerleri ile 30 kere çalıştırılarak Tablo 3'deki en kötü süre, en iyi süre, ortalama süre, en kısa tur ve en uzun tur değerleri bulunmuştur.

Tablo 2 ERA için iterasyon sayısı 5000 ve 7500 değerinin karşılaştırılması

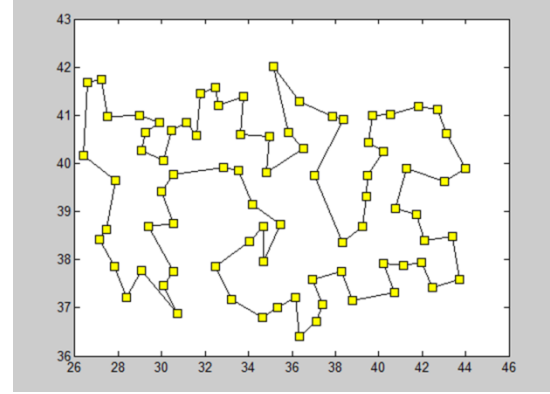
İterasyon sayısı=5000		İterasyon sayısı=7500	
En iyi tur (km)	En iyi tur süresi (sn)	En iyi tur (km)	En iyi tur süresi (sn)
10280	211,4023	10342	330,4614
10304	209,1883	10388	377,8011
10340	216,0314	10464	397,0229
10367	209,8271	10634	396,3461
10424	208,9024	10656	462,8553

Tablo 3 ERA için popülasyon sayısına göre bulunan sonuçlar

İterasyon sayısı	En kötü süre (sn)	En iyi süre (sn)	Ortalama süre (sn)	En kısa tur (km)	En uzun tur (km)
40	156,652	129,3746	148,0889	10394	12136
80	228,9252	169,3863	196,8052	10375	11401
120	237,3044	208,8989	212,4841	10280	10942

Tablo 4'e bakıldığında popülasyon sayısı arttıkça süre artsa da en iyi tur için daha iyi sonuçlar bulunmuştur. Tablo 1 ve Tablo 3'deki verilere göre ERA için en iyi değerlerin iterasyon sayısı 5000, popülasyon sayısı 120 alındığında elde edildiği görülmüştür. Şekil 7'de en iyi tur için alınan program çıktısı verilmiştir.

Ancak bu çizim iki şehir arası Öklid mesafesine dayalı bir çizim olup, gerçek yollar değildir. Şekil 8'de en iyi turun Türkiye haritası üzerindeki gideceği yol çizilerek gösterilmiştir. Bu turlar için başlangıçlar şehir merkezi seçilmiştir. Harita üzerinde tur dışı çizgiler gibi görünen uçlar, şehir merkezine geliş ve bir sonraki hedefe gidiştir. Tablo 4.'de ise ERA için bulunan en iyi turun il sıralaması gösterilmiştir.

**Şekil 7.** ERA için bulunan en iyi tur**Tablo 4.** ERA için en iyi turun il sıralaması

Sıra No	İl Adı	Sıra No	İl Adı	Sıra No	İl Adı	Sıra No	İl Adı
1	MALATYA	22	BATMAN	43	ESKİŞEHİR	64	BİLECİK
2	ELAZIĞ	23	DİYARBAKIR	44	KÜTAHYA	65	SAKARYA
3	TUNCELİ	24	MARDİN	45	AFYONKARAHİSAR	66	DÜZCE
4	ERZİNCAN	25	ŞANLIURFA	46	UŞAK	67	BOLU
5	BAYBURT	26	ADYAMAN	47	ISPARTA	68	ZONGULDAK
6	GÜMÜŞHANE	27	KAHRAMANMARAŞ	48	BURDUR	69	BARTIN
7	TRABZON	28	GAZİANTEP	49	ANTALYA	70	KARABÜK
8	RİZE	29	KİLİS	50	DENİZLİ	71	KASTAMONU
9	ARTVİN	30	HATAY	51	MUĞLA	72	ÇANKIRI
10	ARDAHAN	31	OSMANIYE	52	AYDIN	73	ÇORUM
11	KARS	32	ADANA	53	İZMİR	74	YOZGAT
12	İĞDIR	33	MERSİN	54	MANİSA	75	TOKAT
13	AĞRI	34	KARAMAN	55	BALIKESİR	76	AMASYA
14	ERZURUM	35	KONYA	56	ÇANAKKALE	77	SİNOP
15	BİNGÖL	36	AKSARAY	57	EDİRNE	78	SAMSUN
16	MUŞ	37	NEVŞEHİR	58	KIRKLARELİ	79	ORDU
17	BİTLİS	38	NİĞDE	59	TEKİRDAĞ	80	GİRESUN
18	VAN	39	KAYSERİ	60	İSTANBUL	81	SİVAS
19	HAKKÂRİ	40	KIRŞEHİR	61	KOCAELİ		
20	ŞIRNAK	41	KIRIKKALE	62	YALOVA		
21	SİİRT	42	ANKARA	63	BURSA		



Şekil 8. ERA için en iyi turun Türkiye haritası üzerinde gösterimi

81 il için GSP'nin en kısa turu, gerçek koordinatlar kullanılarak ERA ile bulunmaya çalışılmıştır. Elde edilen sonuçların karşılaştırılması için çalışma yapıldığında, literatürde 81 il için bir çalışma tesbit edilemediğinden, PSO, BT ve KKO ile karşılaştırılmıştır. Tablo 5'de ERA, PSO, KKO ve BT için kullanılan parametre bilgileri verilmiştir. Çalışmada kullanılan parametre değerleri, literatürde bu algoritmalar için tavsiye edilen değerlerdir.

Tablo 5'de verilen değerler ile çalıştırılan GSP, ERA, PSO, KKO ve BT ile her biri ile 30 kez çalıştırılarak, Tablo 6'deki en iyi süre, en kötü süre, ortalama süre, en iyi tur ve en kötü tur değerleri bulunmuştur. Ancak makale yazım sürecinde bir çalışma ile karşılaştırılmış ve bu çalışmanın sonuçları da karşılaştırılmıştır [33].

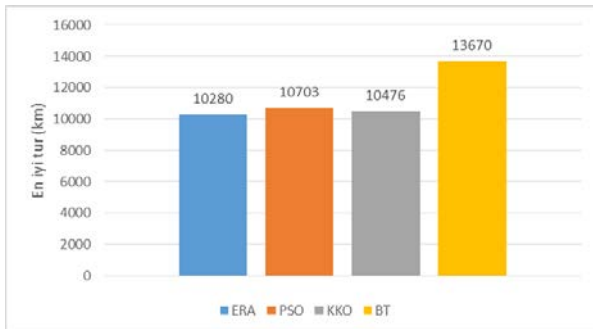
Tablo 5. Uygulamada kullanılan algoritmaların parametre bilgileri

Parametre	ERA	PSO	KKO	BT
Maksimum İterasyon Sayısı	5000	10000	2500	5000
Özel Parametreler	Alpha=1 Zeta=0,1	W:1 W=0,99	Feromon buharlaşma oranı=0,05	Soğuma katsayısı alfa=0,99
Popülasyon sayısı	40	100	40	

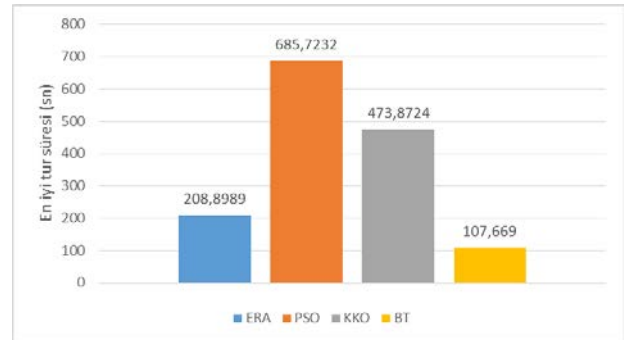
Tablo 6. ERA, PSO, KKO ve BT için elde edilen sonuçlar

Uygulanan algoritma	En kötü süre (sn)	En iyi süre (sn)	Ortalama süre (sn)	En kısa tur (km)	En uzun tur (km)
ERA	237,3044	208,8989	212,4841	10394	12136
PSO	734,3919	685,7232	704,2509	10703	12960
KKO	645,8426	473,8724	486,6273	10476	10719
BT	116,8536	101,9145	107,669	13670	16239
[33] nolu referans Genetik Algoritma			910	9966	

Şekil 9 ve 10'da ERA, PSO, KKO ve BT için en iyi tur ve en kısa süre için karşılaştırılması grafik üzerinde gösterilmiştir.



Şekil 9. ERA, PSO, KKO ve BT'nin en iyi tur için karşılaştırılması



Şekil 10. ERA, PSO, KKO ve BT'nin süre açısından karşılaştırılması

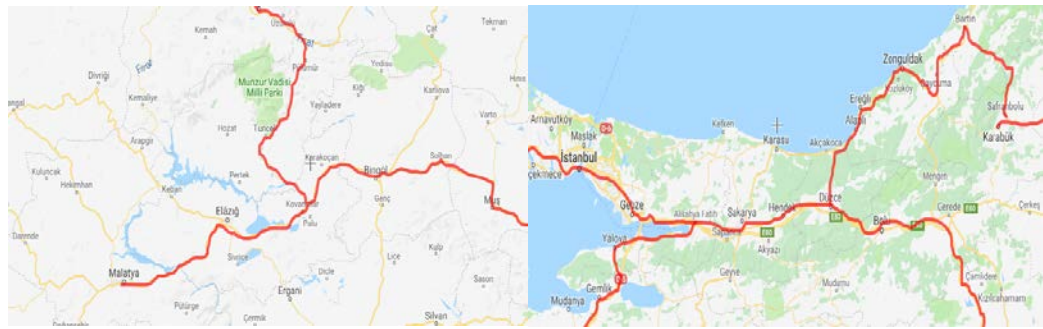
Dört algoritma içinde en iyi tur bulunduğu iterasyon sayıları üzerinden karşılaştırma yapılmıştır. Sonuçlara göre süre açısından BT daha hızlı çalışmış olsa da en iyi tur için, ERA daha iyi ve tutarlı sonuçlar vermiştir.

Şekil 11'de karşılaştırılan çalışma sonucunda bulunan en iyi tur Türkiye haritası üzerinde gösterilmiştir.

Makalede sonuçlar Öklid üzerinden gösterildiği için GSP mantığına aykırı bir durum görünmemektedir. Fakat 33 nolu referansta bulunan en kısa tur, harita üzerinde uygulandığında, Şekil 12'de görüldüğü gibi iki bölgede bir noktadan diğer noktaya geçmesi için daha önce geçilmiş bir noktaya tekrar uğraması gerekmektedir.



Şekil 11. Karşılaştırılan makalenin[33] en iyi turunun Türkiye haritasında gösterimi



Şekil 12. Karşılaştırılan makalenin en iyi tur problemi

Makalede 81 il üzerinden tek bir gezgin satıcı için en kısa tur bulunduktan sonra, aynı probleme daha gerçekçi bir çözüm bulunması arayışına gidilmiştir. 81 ilin tamamının gezilmesi, gerçek hayatta da lojistik açısından, önem arzeden bir GSP problemidir. Ancak tek bir gezgin satıcı yerine k adet küme oluşturularak elde edilen k+1 gezgin satıcı ile problemi çözmek olası çözüm sayısını büyük ölçüde azaltmakta, dolayısıyla çözüm süresini kısaltmaktadır. Teorik olarak düşünüldüğünde, n şehirli bir 1-GSP (n-1)! tur içerirken, k-means ile k-adet alt probleme ayrılmış k-GSP tüm kümeler eşit elemanlı varsayılırsa $k(n/k-1)!$ olası tur içerir. 81 il için düşünüldüğünde 1-GSP 7.1569×10^{118} olası tur içerirken $k=10$ için tüm kümelerin yaklaşık 8 şehir içerdiği varsayımı ile $(81/10-1)! \times 10 = 50400$ olası tur içerir.

Bir dağıtım şirketi için düşünülecek olursa, tek bir dağıtıcının 81 ili gezmesi, hem dağıtımın tek bir araçla yapılamayacağı olması, hem turun çok uzun olması dolayısıyla imkânsızdır.

Bu amaçla çalışmanın bu bölümünde 81 il önce k-means kümeleme ile 7,8 ve 9 bölgeye ayrılmıştır. Bu kümeleme de aynı kümeğe düşen şehirler kendi aralarında bir GSP problemi olarak ele

alınarak çözülmüş ve ERA ile elde edilen en kısa tur sonuçları Tablo 7'de $k=7$ için verilmiştir. Tablo 7'den görüleceği gibi önce her bir bölge için en kısa GSP sonucu verilmiş, daha sonra grup merkezlerine en yakın şehirler üzerinden en kısa tur hesaplanmış ve en son sütunda ise bu yolla elde edilen toplam tur uzunluğu sunulmuştur.

Şekil 13'de $k=7$ için kümeleme sonucu aynı kümeğe düşen iler için turlar gerçek yollar üzerinden gösterilmiştir.

Bu çözümler elde edilirken önce bir lojistik firmasının 7 bölge için sorumlusunun olduğu ve kendi bölgesine ait şehirleri gezerek yaşadığı şehre geri döndüğü varsayılmıştır. Bu sebeple tüm turların hesabında şehir merkezleri kullanılmıştır.

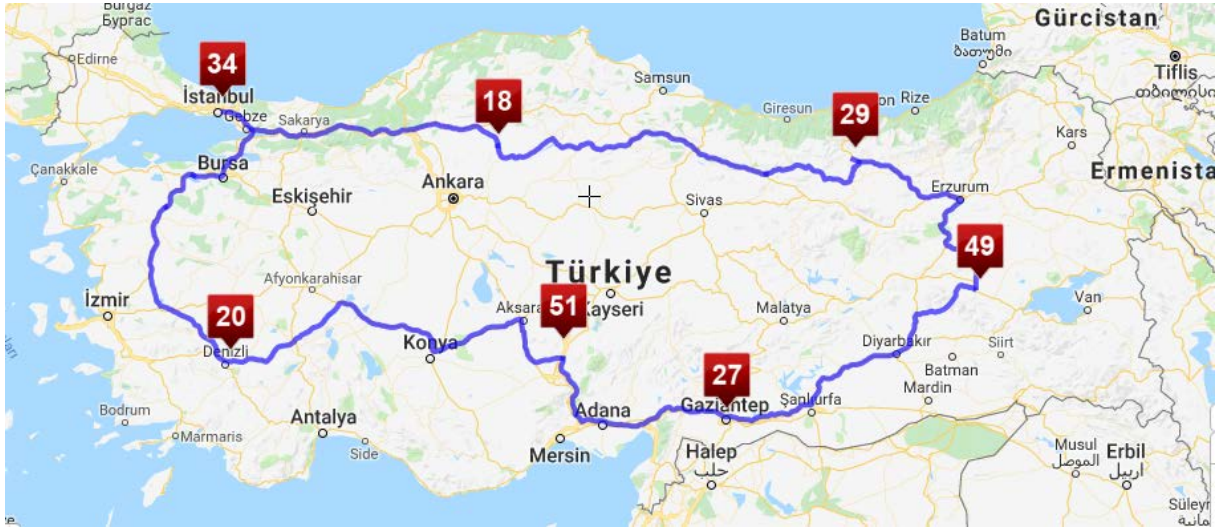
Şekil 14'de ise 7 bölge merkezine en yakın şehirlerin oluşturduğu GSP problemi için en kısa tur bulunmuştur. Kümeleme sonucu elde edilen küme merkezlerinin kullanılmaması, bunun yerine bu merkeze en yakın şehir seçilmiştir. Şekil 15'de ise bulunan her iki sonuç, toplam 8 kapalı tur harita üzerinde gösterilmiştir.

Tablo 7. K-means ile 7 bölge için elde edilen sonuçlar

Sonuçlar	1. Bölge (km)	2. Bölge (km)	3. Bölge (km)	4. Bölge (km)	5. Bölge (km)	6. Bölge (km)	7. Bölge (km)	Toplam (km)	Grup Merkezleri Tur(km)	Tüm Toplam (km)
KGM	1057	1227	1919	2389	1393	1595	1710	11290	-	
Google.map	1142	1221	1733	2347	1338	1549	1620	10950	3604	14554



Şekil 13. k=7 bölge için en iyi sonucun bölge içi tur gösterimi



Şekil 14. k=7 bölge için en iyi sonucun bölge merkezleri ve bölgeler arası tur gösterimi



Şekil 15. k=7 bölge için en iyi sonucun bölgeler arası ve bölge içi tur gösterimi

k=8 için ERA ile elde edilen en kısa tur sonuçları Tablo 8'de verilmiştir. Şekil 16'da k=8 için kümeleme sonucu aynı kümeye düşen iller için turlar gerçek yollar üzerinden gösterilmiştir.

Şekil 17'de ise 8 bölge merkezine en yakın şehirlerin oluşturduğu GSP problemi için en kısa tur bulunmuştur. Şekil 18'de ise her iki sonuç aynı harita üzerinde gösterilmiştir.

Tablo 8. K-means ile 8 bölge için elde edilen sonuçlar

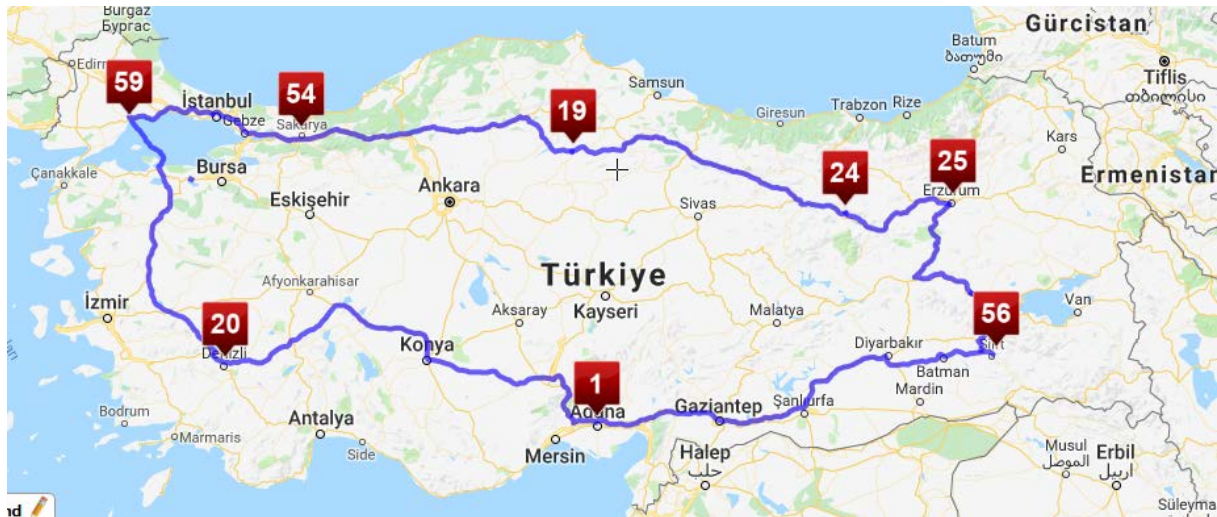
Bölgeler	1. Bölge (km)	2. Bölge (km)	3. Bölge (km)	4. Bölge (km)	5. Bölge (km)	6. Bölge (km)	7. Bölge (km)	8. Bölge (km)	Toplam (km)	Grup Merkezleri Toplamı (km)	Tüm Toplam (km)
Kgm	1876	1174	1767	826	1333	1237	1783	1616	11612		
Google.map	1795	1088	1662	845	1334	1200	1750	1323	10997	3758	14755

Tablo 7 ve 8 incelenecek olursa, toplam tur uzunluğunda 201 km lik bir artış olmuş, bu artışın 154km tur merkezleri arasındaki GSP turunun uzunluğunun artmasından kaynaklandığı, buna karşılık

diğer her bir bölge için oluşturulmuş tur uzunluklarının azaldığı görülmüştür.



Şekil 16. k=8 bölge için en iyi sonucun bölge içi tur gösterimi



Şekil 17. k=8 bölge için en iyi sonucun bölge merkezleri ve bölgeler arası tur gösterimi

k=9 için ERA ile elde edilen en kısa tur sonuçları Tablo 9'da verilmiştir. Şekil 19'da k=9 için kümeleme sonucu aynı kümeye düşen iller için turlar gerçek yollar üzerinden gösterilmiştir.

Şekil 20'de ise 9 bölge merkezine en yakın şehirlerin oluşturduğu GSP problemi için en kısa tur bulunmuştur. Şekil 21'de ise her iki sonuç aynı harita üzerinde gösterilmiştir.



Şekil 18. k=8 bölge için en iyi sonucun bölgeler arası ve bölge içi tur gösterimi



Şekil 19. k=9 bölge için en iyi sonucun bölge içi tur gösterimi



Şekil 20. k=9 bölge için en iyi sonucun bölge merkezleri ve bölgeler arası tur gösterimi



Şekil 21. $k=9$ bölge için en iyi sonucun bölgeler arası ve bölge içi tur gösterimi

5. Sonuç

GSP problemi başta İşletme, Endüstri Mühendisliği ve Bilgisayar Mühendisliği olmak üzere bir çok disiplin tarafından çokça çalışılan bir problemdir. NP-Zor bir problem olduğu için, ayrıntılı optimizasyon algoritmalarının performanslarını değerlendirmek için de, GSP problemleri test problemi olarak kullanılmaktadır. Bu çalışmada ise, Gezgin Satıcı Problemi 81 ili kapsayan bir tur için çözülmeye çalışılmıştır. Tüm sonuçlar birlikte değerlendirildiğinde bu çalışmanın katkıları aşağıda maddeler halinde sıralanmıştır:

- GSP probleminin ERA algoritması da dahil olmak üzere, sezgisel optimizasyon algoritmaları ile çözümünü içeren çok sayıda çalışma mevcuttur. Ancak özellikle Türkiye'nin 81 ili için GSP probleminin sezgisel algoritmalar ile çözümüne ait tek bir çalışma bulunmuştur[33]. Bu çalışma GSP probleminin, 81 il için çözümünü, GA ve KKO ile elde etmiş ve çalışmanın sonuçları bu makalede elde edilen sonuçlar ile karşılaştırılmıştır.
- ERA algoritması 81 il için GSP problemine ilk defa uygulanmıştır. ERA algoritmasının çözümü, en optimum çözüme yakın sonuçlar vermiştir. Bu bakımdan gelecekteki çalışmalarda, GSP problem çözümünde ERA alternatif bir sezgisel optimizasyon algoritması olarak kullanılabilir.
- Çalışmada teorik en kısa turun bulunması işlemi yanında, lojistik bakımından uygulanabilir olası çözümler bulunması amaçlanmıştır. Bu amaçla tarafımızdan yeni bir GSP çözüm yaklaşımı önerilmiştir. Bu yaklaşımda turdaki tüm şehirleri tek bir seferde gezmek yerine k adet birbirinden bağımsız alt tur elde edilmiştir. Bu turların küme merkezine en yakın olan şehirlerin oluşturduğu tur ile de ayrıık olan turlar birleştirilmiştir.
- $k=7,8$ ve 9 için elde edilen tüm sonuçlar harita üzerinde gösterilerek, olası çalışmalarda kullanılması amaçlanmıştır. Bu yolla elde edilen tüm turlar toplamı 14554-15 401 km arasında olmakla birlikte, alt tur uzunlukları k arttıkça azalmıştır. Bu sonuçların her biri alternatif bir çözüm olarak değerlendirilebilir.

Kaynaklar

1. J. Pedro Schmitt, F. Baldo and R. Stubbs Parpinelli, A MAX-MIN Ant System with Short-Term Memory Applied to the Dynamic and Asymmetric Traveling Salesman Problem, 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), Sao Paulo, Brazil, 2018, 1-6 s.
2. Blum, C., & Roli, A., Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM computing surveys (CSUR), 35(3), 268-308, 2003.
3. X. Yang and J. Wang, Application of improved ant colony optimization algorithm on traveling salesman problem, 2016

Chinese Control and Decision Conference (CCDC), Yinchuan, 2156-2160s., 2016.

4. Çolak S., Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, (19) 3, 423 – 438, 2010.
5. Pulat M., Deveci K. İ., Gezin satıcı probleminin çözümünde kullanılan genetik algoritmanın parametrelerinin incelenmesi, Uluslararası İktisadi ve İdari Bilimler Dergisi – IJEAS, (16), 21 – 36, 2017.
6. Erdoğan, P. (2016). Doğadan esinlenen optimizasyon algoritmaları ve optimizasyon algoritmalarının optimizasyonu. Düzce üniversitesi bilim ve teknoloji dergisi, 4(1),293-304 s.
7. Alataş B., Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, FÜ. Fen Bil. Ens., Bilgisayar Mühendisliği ABD, Doktora Tezi, 4 s., 2007.
8. Atashpaz-Gargari, E., Lucas, C., Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition, IEEE Congress on Evolutionary Computation, 4661-4667, 2007.
9. Noorbin, S. F. E. H., & Alfi, A. (2018). Adaptive parameter control of search group algorithm using fuzzy logic applied to networked control systems. Soft Computing, 22(23), 7939-7960
10. Maheri, M. R., & Talezadeh, M. (2018). An Enhanced Imperialist Competitive Algorithm for optimum design of skeletal structures. Swarm and Evolutionary Computation, 40, 24-36
11. Sayevand, K., Arab, H., & Golzar, S. B. (2018). Development of imperialist competitive algorithm in predicting the particle size distribution after mine blasting. Engineering with Computers, 34(2), 329-338.
12. Yousefikhoshbakh, M., & Sedighpour, M. (2013). New imperialist competitive algorithm to solve the travelling salesman problem. International Journal of Computer Mathematics, 90(7), 1495-1505
13. Ardalan, Z., Karimi, S., Poursabzi, O., & Naderi, B. (2015). A novel imperialist competitive algorithm for generalized traveling salesman problems. Applied Soft Computing, 26, 546-555.
14. Davendra, D., Travelling Salesman Problem, Theory and Applications, Intech, Croatia, 2010, 6 s.
15. Pulat M., Karakoç İ. D., Gezin Satıcı Probleminin Genetik Algoritmalarla Çözümünde Başlangıç Popülasyonunun Belirlenmesi, Journal Of Emerging Economies And Policy, 2(1), 95-123, 2017.

16. Kayman A. Y., Gezgin satıcı probleminin çözümünde parçacık sürü optimizasyonu algoritması performansının bulanık c-ortalama yöntemini ile iyileştirilmesi, Kocaeli Üniversitesi, Fen Bil. Ens., Endüstri Mühendisliği ABD, Doktora Tezi, 10 s., 2015.
17. Bahaabadi, M., Mohaymany, A. S., and Babaei, M., "An Efficient Crossover Operator For Travelling Salesman Problem", International Journal Of Optimization In Civil Engineering, 2 (4), 607-619, (2012).
18. Alkan A., Gezgin satıcı problemi için veri madenciliği tabanlı bir model önerisi, Kocaeli Üniversitesi Fen Bil. Ens., Endüstri Mühendisliği ABD, Doktora Tezi, 28 s., 2014.
19. Terzi, Ü., Gezgin Satıcı Problemi için Diferansiyel Gelişim Algoritması Tabanlı Bir Metasezgisel Öneri, Kocaeli Üniversitesi Fen Bil. Ens., Endüstri Mühendisliği ABD, Doktora Tezi, 14 s., 2009.
20. Kesintürk T., Topuk N., Özyeşil O., Araç Rotalama Problemleri İle Çözüm Yöntemlerinin Sınıflandırılması ve Bir Uygulama, İşletme Bilimi Dergisi, 3, 77-106, 2015.
21. Öztürk C., Karınca ve Sürü Optimizasyon Yöntemlerinin İncelenmesi ve Yazılım Uygulamalarının Oluşturulması, Marmara Üniversitesi Fen Bil. Ens., Elektrik ve Elektronik Mühendisliği, Yüksek Lisans Tezi, 26 s., 2006.
22. Kayman A. Y., Gezgin Satıcı Probleminin Çözümünde Parçacık Sürü Optimizasyonu Algoritması Performansının Bulanık C-Ortalama Yöntemi İle İyileştirilmesi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği ABD, Yüksek Lisans Tezi, 15 s., 2015.
23. Matai R., Singh S. P., Mittal L. M., Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches, Editor: Davendra D., Traveling Salesman Problem, Theory and Applications, InTech, Croatia, 1-24, 2010.
24. Göksal F. P., Eşzamanlı Topla-Dağıt Araç Rotalama Problemi İçin Sezgisel Yaklaşımlar: Genetik Algoritma ve Kuş Sürüsü Eniyileme, Gazi Üniversitesi Fen Bil. Ens., Endüstri Mühendisliği, Yüksek Lisans Tezi, 4 s., 2010.
25. Çelik E., Bulut Görev Çizelgesi İçin Benzetilmiş Tavlama Tabanlı Bir Optimizasyon Yaklaşımı, Atatürk Üniversitesi Fen Bil. Ens., Bilgisayar Mühendisliği ABD, Yüksek Lisans Tezi, 36 s., 2018.
26. Yang X. S., Introduction to Mathematical Optimization – From Linear Programming to Metaheuristics, Cambridge International Science Publishing, United Kingdom, 2008.
27. Sallabi, O. M., and El-Haddad, Y., An improved genetic Algorithm to solve the traveling Salesman problem, World Academy of Science, Engineering and Technology, 28, 472-474, 2009.
28. Erdoğan, P. (2013). Particle swarm optimization performance on special linear programming problems. Scientific Research and Essays, 5(12), 1506-1518.
29. Yin P., Ant Colony Search Algorithms for Optimal Polygonal Approximation Of Plane Curves, Pattern Recognition, 36(8), 1783-1797, 2003.
30. Abdechiri, M., Faez, K., Bahrami, H., Adaptive Imperialist Competitive Algorithm, 9th IEEE International Conference on Cognitive Informatics (ICCI), 940-945, 2010.
31. Kızılluk, S., Alataş, B., Sosyal tabanlı güncel sezgisel optimizasyon algoritmaları, C.Ü. İktisadi ve İdari Bilimler Dergisi, 13(2): 39-55, 2012.
32. <http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Root/Uzakliklar.aspx>: Son erişim tarihi: 02-01-2018.
33. Dikmen, H., Dikmen, H., Elbir, A., Ekşi, Z., & Çelik, F. (2014). Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması. Journal of Natural & Applied Sciences, 18(1).