

**Bulanık c-ortalımalı kümeleme ve karınca kolonisi optimizasyonu ile  
çoklu gezgin satıcı problemi**  
**Multiple travelling salesman problem with fuzzy c-means and  
ant colony optimization algorithms**

Gönderim Tarihi / Received: 07.10.2021

Kabul Tarihi / Accepted: 12.12.2022

Doi: [10.31795/baunsobed.1005070](https://doi.org/10.31795/baunsobed.1005070)

Derya DİKBİYİK<sup>\*\*1</sup>

Selçuk ALP<sup>2</sup>

**ÖZ:** Sezgisel algoritmalar, kabul edilebilir sürede optimuma yakın çözümler verebilen ve çok büyük boyutlu optimizasyon problemleri için kullanılabilen algoritmalarlardır. En iyi çözümün bulunacağı garanti edilememekle beraber, bulunan çözümün kabul edilebilir düzeyde olması, çözüme kolay ve hızlı ulaşılabilmesi açısından kullanımı oldukça yaygın olan yöntemlerdir. Sezgisel yöntemlerde problemin çözümüne yönelik yaklaşımlar; karar verme, optimizasyon, bulanık mantık, yapay zeka, makine öğrenmesi, derin öğrenme şeklinde karşımıza çıkar. Bu çalışmada; pek çok alanda uygulaması olan Gezgin Satıcı Problemi (GSP) için optimuma en yakın çözümü hızlı bir şekilde bulabilmek amacıyla Karınca Kolonisi Optimizasyonu (KKO) yöntemi seçilmiştir. Rastgele seçilen verileri gruplandırmak amacıyla da Bulanık C-Ortalımalı Kümeleme (BCO) Algoritması kullanılmıştır. Çalışmada kullanılan veriler BCO algoritması kullanılarak ayrı ayrı 3, 4 ve 5 kümeye ayrılmış; elde edilen veri setleri Çoklu KKO ile değerlendirilmiş ve sonuçlar karşılaştırılmıştır. 3'lü kümeleme ile elde edilen rotanın uzunluğunun 4'lü kümeden %4,61, 5'li kümeden ise %2,37 daha kısa olduğu belirlenmiştir.

**Anahtar Kelimeler:** Karınca kolonisi optimizasyonu, Bulanık c-ortalımalı kümeleme, Çoklu gezgin satıcı problemi

**ABSTRACT:** Heuristic algorithms are algorithms that can provide near-optimum solutions for very large-scale optimization problems in an acceptable time. They are commonly used methods in terms of being at an acceptable level and reaching a solution easily and quickly, although it cannot be guaranteed that the best solution will be found. Problem solving approaches being used in heuristic methods; decision making, optimization, fuzzy logic, artificial intelligence, machine learning, deep learning. In this paper; chosen method in order to find the best solution of route optimization for the Travelling Salesman Problem (TSP), which has applications in many areas, is that Ant Colony Optimization (ACO). For classification of the data, that is randomly selected, Fuzzy C-Mean Clustering (FMC) Algorithm has being used. The data set has being separated into 3, 4 and 5 clusters and all cluster data sets obtained were evaluated with Multiple ACO and then the results were compared. It was determined that the length of the route obtained by clustering 3 was 4.61% shorter than the cluster of 4 and 2.37% shorter than the cluster of 5.

**Keywords:** Ant colony optimization, Fuzzy c-means clustering, Multiple travelling salesman problem

<sup>\*\*</sup> Sorumlu Yazar / Corresponding Author

<sup>1</sup> Doktora Öğr., Yıldız Teknik Üniversitesi Makine Fakültesi/Endüstri Mühendisliği Bölümü/Yöneylem Araştırması Anabilim Dalı, [derya.dikbiyik@std.yildiz.edu.tr](mailto:derya.dikbiyik@std.yildiz.edu.tr), <https://orcid.org/0000-0002-2964-4200>

<sup>2</sup> Doç. Dr., Yıldız Teknik Üniversitesi Makine Fakültesi/Endüstri Mühendisliği Bölümü/Yöneylem Araştırması Anabilim Dalı, [alp@yildiz.edu.tr](mailto:alp@yildiz.edu.tr), <https://orcid.org/0000-0002-6545-4287>

## EXTENDED ABSTRACT

### Literature review

Travelling Salesman Problem is a kind of problem that describing easily, but solving this problem is not easier describing it. Travelling starts with a particular starting point, the seller's route include n number of locations and then finishes with arriving starting point again. In this kind of problem, as the number of locations increase the solution area also grows, thus reaching the optimum solution becomes longer and more difficult. In such cases where analytical solution methods are insufficient in solving the problem, instead of scanning the entire solution space, heuristic and metaheuristic methods that make logical deductions with partial scans in the solution space can be used.

Heuristic and metaheuristic methods do not assure the solution of the problem, however make a great contribution to reducing the solution cost. These algorithms can give near-optimum solutions in an acceptable time and can be used for very large-scale optimization problems. They are very common methods in terms of being at an acceptable level and reaching a solution easily and quickly, although it cannot be guaranteed that the best solution will be found. Some approaches to solving the problem in heuristic and meta-heuristic methods; decision making, optimization, fuzzy logic, artificial intelligence, machine learning, deep learning.

Traveling Salesman Problem; after it was first defined mathematically in 1954, it was used for optimization modeling of many problems such as vehicle routing and shipment planning, machine programming and assignment, order picking. With increasing of the problem complexity, studies and applications related to the heuristic and metaheuristic approach have been started. Among these applications, the Ant Colony Optimization Algorithm has taken place among the most frequently drawn on methods.

### Methodology

In this study; The Ant Colony Optimization method was chosen for the purpose of quickly finding the proximately best solution to the optimum for the Traveling Salesman Problem, which has applications in many fields. Fuzzy C-Mean Clustering Algorithm have been used to group randomly selected data.

Fuzzy C-Mean Clustering Algorithm is one of the primarily preferred fuzzy clustering methods. According to the fuzzy logic principle, all elements of each sets belong to with membership values varying in the range  $[0,1]$ . However, the sum of membership values for all classes should be "1". Based on the distance from the center of the clusters for each element, the cluster belongs to the element is determined according to the membership degrees. The greater the proximity to the center, the greater the membership value to that cluster.

Ant Colony Optimization, on the other hand, is a metaheuristic method developed on the basis of the ants' sense of access to food sources and their sense of direction. Ants are, as social animals, that live in a colony in nature, have a certain division of labor between them and solve problems together by helping.

Ant Colony Optimization has been designed by examining the feeding behaviour of real ants. In ant colonies, the ability to find the shortest path between nest and food sources is highly developed. This complex behavior arises when the members of the colony leave pheromones on their way to the food source by any route they choose, that is, they communicate with each other indirectly. For the ants that follow the others, there is the behavior of choosing the path containing the strongest pheromone. In alternative routes, the increase or decrease in the amount of pheromone determines the preference of the route. Ants leave more pheromones on the shortest path than they will travel more often, so the search process naturally shifts towards the shortest paths.

For Ant Colony Algorithm Optimization, parameters, transition rules, local pheromone update and general pheromone update rules are determined. With a parameter that determines the importance of the pheromone amount of the relevant path and allows the previous iteration results to be transferred to the progressive iterations, the probability of selecting the paths in which the pheromone is concentrated

increases, and the coincidence decreases. Another parameter that determines the influence of the path length on the selection of the next point decreases the possibility of searching for alternative solutions. And also the other parameter is defined to determine the evaporation rate.

For the Multiple Traveling Salesman Problem with Ant Colony Optimization the solution differs. One seller goes to some locations, then the second seller goes to some of the un-visited locations, and the process is completed with the participation of all sellers and visits to all locations. The tours are completed with each ant leaving the starting point and touring the places not visited. When the ants start the tour from the starting point or the artificial location, the tour of the first seller begins, and the tour of another seller starts when the ants pass from the starting point or the artificial point.

Clustering algorithms; separates the data set into different branches or clusters in compliance with a particular criterion. Thus, although the similarity rate between the elements of the sets is quite high, there are considerable differences with the elements of other sets. Fuzzy C-Mean Clustering Algorithm assigns the data set into C number of fuzzy sets and calculates the clustering center for each group. An iterative formula using the Lagrange Multiplier method is applied to reduce the objective function.

### **Findings and discussion**

For the application, the starting point coordinates and the coordinates of the 30 locations to be visited were determined and then the values of these 30 locations were loaded into the Matlab application to cluster with the Fuzzy C-Mean Clustering Algorithm. With this Algorithm, three different clusters were made, namely, 3-Clustering, 4-Clustering and 5-Clustering. The outputs obtained were uploaded back to the Matlab application, using them as inputs for Ant Colony Optimization. The values obtained for all three clusters were evaluated with tables and graphics and the results obtained were compared.

### **Results and recommendations**

While evaluating, it should be kept in mind that the distance between points is the Euclidean Distance. Scheduling should be made for the routes to be created, considering the roads open to access and the traffic situation. In addition, while planning the route, delivery time and cost should be evaluated together. If the priority is fast delivery or if the delay in delivery is subject to penalties or sanctions, it must be considered as a priority in the evaluation. The method is a metaheuristic application, so that the solution cannot be guaranteed to be optimum, but it has been evaluated as the best solution calculated in a short time and at an acceptable level.

## Giriş

Ulaşım ürünleri, insanların, canlıların bir noktadan diğer bir noktaya nakledilmesini ifade etmektedir ve tarih boyunca şehirleşme, uygarlaşma ve ilerleme gibi kavramların ölçütlerinden biri olmuştur. Ulaştırma sektörü, günümüzde dünya ölçeğinde ekonomik olarak önemli bir yere sahiptir. İlişki derecesine bağlı olarak bu sektör, diğer sektörleri de etkilemektedir. Ulaşım sektörü, başta ekonomi olmak üzere turizm, sağlık, eğitim vb. birçok alanı bölgesel, ulusal ve uluslararası düzeyde etkileyebilen önemli bir sektördür (Aytekin, 2022: 18).

Sezgisel algoritmalar, karar vericilerin bekleme kabul edebileceği (katlanabileceği) sürede, problemin en iyi çözümüne yakın çözümlere ulaşabilen ve bunun yanı sıra çok büyük boyutlu (çok sayıda değişken ve/veya kısıt içeren) optimizasyon problemleri için de kullanılabilen algoritmalar. Sezgisel algoritmalar, en iyi çözümün bulunacağı garanti etmemekle birlikte, en iyi çözüme ne kadar yakın ya da uzak bulunduğu da gösteremez. Buna karşın Sezgisel algoritmalar, ulaşılan çözümün kabul edilebilir bir düzeyde olmasını, ayrıca bu çözüme kolay ve hızlı bir şekilde ulaşılabilmesi sağlamaktadırlar. Sezgisel algoritmalar kullanılan yöntemlerde problemin çözümüne yönelik yaklaşımlar; karar verme, optimizasyon, bulanık mantık, yapay zeka, makine öğrenmesi ve derin öğrenme şeklinde gruplara ayrılabilirler. Bu çalışmada; pek çok alanda uygulaması bulunan Gezgin Satıcı Problemi (GSP) için en iyi çözüme, en yakın çözümü hızlı bir şekilde bulabilmek için Sezgisel algoritmalarından biri olan Karınca Kolonisi Optimizasyonu (KKO) yöntemi kullanılmıştır. Rasgele belirlenen lokasyonların kümelenmesi için BCO algoritmasından yararlanılmıştır. Literatürde, bu çalışmada da önerilen çözümün etkinliğinin test edilmesi için lokasyonların rasgele belirlendiği çalışmalar bulunmaktadır. Bu çalışmalara Arthur ve Friendewey (2017), Sureja ve Chawda (2012) Ahmed ve Pandit (2001) ve Frieze (1987) çalışmaları örnek olarak verilebilir. Çalışmada kullanılan veriler BCO algoritması kullanılarak ayrı ayrı 3, 4 ve 5 kümeyle ayrılmış; elde edilen veri setleri Çoklu KKO ile değerlendirilmiş ve sonuçlar karşılaştırılmıştır. Bu çalışma ile sezgisel yöntemlerde başlangıç durumunun, çözüme ulaşmada (daha iyi çözüme ulaşma ya da çözüme daha hızlı ulaşma) önemli olduğu ortaya konulmaya çalışılmıştır. Problemden yer alan lokasyonların 3, 4 ve 5 kümeyle rasgele atanması yerine önceden gruplandırılarak çözümlere ulaşılmıştır.

Çalışma, dört bölümden oluşmaktadır. Birinci bölüm giriş bölümüdür. İkinci bölümde, gezgin satıcı problemi tanıtılmış ve çalışmada kullanılan Karınca Kolonisi Optimizasyonu ve Bulanık C-Ortalama Kümeleme Algoritması yöntemleri anlatılmış ve literatürde yapılan çalışmalara yer verilmiştir. Üçüncü bölümde ise uygulamada kullanılan GSP, BCO ve KKO ile ilgili bilgiler sunulmuştur. Dördüncü bölümde, gerçekleştirilen uygulamalar hakkında bilgi verilmiştir. Bulgular ve tartışmalara beşinci bölümde yer verilmiştir. Çalışmanın altıncı ve son bölümünde ise sonuçlar sunulmuş ve gelecek çalışmalar için önerilerde bulunulmuştur.

## Literatür taraması

Çalışmada, çoklu GSP'nin çözümü için BCO algoritması ve KKO birlikte kullanılmıştır. Bu bölümde GSP, BCO ve KKO ile ilgili kısa bilgilendirme yapıldıktan sonra literatürde yapılmış çalışmalara yer verilecektir.

GSP, kolay tanımlanabilen bir problem çeşididir, ancak problemin çözümü, tanımlanması kadar kolay değildir. Belirli bir başlangıç noktasından harekete başlayan gezgin satıcı, n sayıda lokasyonu dolaşır ve yine başlangıç noktasına dönerek turu tamamlar. Tanımlanan problemde n, lokasyon sayısı arttıkça çözüm uzayı hızla büyümekte ve optimum çözüme ulaşmak uzun ve zorlu bir süreç haline gelmektedir. Analitik çözüm yöntemlerinin problemin çözümünde yetersiz kaldığı bu tür durumlarla karşılaşıldığında çözüm uzayının tamamını taramak yerine, çözüm uzayında kısmi taramalar ile mantıksal çıkarımlar yapan meta-sezgisel yöntemler kullanılabilir. Problemin çözümünü garanti etmemekle beraber, meta-sezgisel yöntemler çözüm maliyetinin azaltılmasına büyük katkı sağlamaktadır (Kuzu vd., 2014: 2).

GSP; matematiksel ifadesi ilk kez Dantzig vd. (1954) tarafından tanımlanmış ve sonrasında pek çok problemin optimizasyon modellemesi için ilham vermiştir (Şahin, 2019: 912). Bu çalışmalar genellikle; araç rotalama ve sevkiyat planlama (Christofides ve Eilon, 1969), makine programlama ve atama (Gilmore ve Gomory, 1964; Nicholson, 1967), sipariş toplama (Ratliff ve Rosenthal, 1983), olarak karşımıza çıkmaktadır.

Problemlerin karmaşıklaşmasıyla GSP için sezgisel yaklaşım (Karg, 1964) ile ilgili çalışmalar yapılmış ve KKO Algoritmasının GSP için uygulanması (Lin ve Kernighan, 1973) en sık başvurulan yöntemler arasında yerini almıştır. Günümüze kadar bu çalışmaların pek çok örneği uygulanmış ve literatüre kazandırılmıştır.

Dinamik GSP için çoklu KKO kullanımı (Mavrovouniotis vd., 2014), GSP için Karınca Kolonisi Hiper Sezgiseli (Aziz, 2015), GSP alan probleminin KKO ile çözümü (Lammel vd., 2016), parametre ayarlarının KKO üzerinde etkinliği (Shrivastava ve Kumar, 2018), açık deniz rüzgar çiftliği için optimize edilmiş toplayıcı topolojisi tasarımında KKO ile çoklu GSP optimizasyonu (Srikakulapu ve Vinatha, 2018), KKO ile zamandan bağımsız GSP optimizasyonu (Tomanova ve Holly, 2020); KKO ile optimize edilmiş GSP uygulamasına yönelik güncel çalışmalardan bazılarıdır.

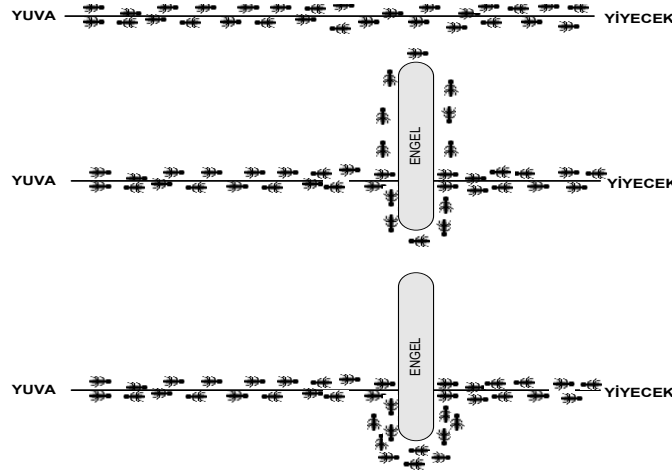
Bulanık C-Ortalımalı (BCO) kümeleme algoritması; en yaygın kullanılan bulanık kümeleme yöntemlerinden biridir. Bulanık mantık prensibine göre tüm elemanların kümeler üyeliği değerleri  $[0,1]$  aralığında değişmektedir. Bununla beraber, her bir elemanın tüm sınıflara ait üyeliği dereceleri toplamının "1" olması gerekmektedir. Her eleman için kümelerin merkezine olan uzaklık baz alınarak üyeliği derecelerine göre elemanın hangi kümeye ait olduğu belirlenir. Merkeze olan yakınlık arttıkça o kümeye olan üyeliği değeri daha büyük olur.

BCO kümeleme algoritması; Dunn tarafından 1973'te ortaya atılmış, algoritma Bezdek tarafından 1981 yılında geliştirilmiştir (Işık ve Çamurcu, 2007: 33). BCO algoritmasında Her bir kümeye ait elemanların küme merkezine olan uzaklıklarını minimuma indirilmesini amaçlanmaktadır.

BCO ile ilgili literatürde birçok çalışma bulunmaktadır. BCO kümeleme algoritması ile yapılan güncel çalışmalara, BCO ile geleneksel kümeleme yöntemlerinin karşılaştırılması (Guo vd., 2020), Büyük veri kümeleme (Shen vd., 2020), BCO kullanarak eksik değer ifadesi (Mausor vd.,2020), BCO ile hasta istatistikleri analizine dayalı kalp hastalığının belirlenmesi (Menaiilov vd., 2020), BCO izleme kümelemesine dayalı çimento klinker üretiminin ısıl verimlilik tahmin modeli (Liu vd, 2020), Tümör segmentasyonu için beyin MRI görüntülerinin BCO ile bölünmesi (Thilagam vd., 2020) çalışmaları örnek olarak verilebilir.

Karınca Kolonisi Optimizasyonu (KKO); karıncaların besin kaynaklarına ulaşma mantığı ve yön seçme duyularından yola çıkılarak geliştirilmiş meta-sezgisel bir yöntemdir. Karıncalar sosyal hayvanlar olarak, doğada koloni halinde yaşarlar. Kendi aralarında özel bir iş bölümü bulunmaktadır ve problemlerle karşılaştıklarında yardımlaşmakta ve birlikte çözüm üretmektedir.

KKO, gerçek karıncaların beslenme davranışları incelenerek kurgulanmıştır. Karınca kolonilerinde, yuva ile yiyecek kaynakları arasındaki en kısa olan yolu bulabilme yeteneği oldukça gelişmiştir. Koloni üyelerinin seçtikleri herhangi bir yolu kullanarak yemek kaynağına giderken bıraktığı feromon ile yani birbirleriyle dolaylı bir şekilde haberleşmesiyle bu karmaşık davranış ortaya çıkmaktadır. Takipteki diğer karıncalar için ise feromon miktarı en yüksek olan yolu tercih etme davranışı söz konusudur. Alternatif yollarda feromon miktarındaki artış ya da azalış yolun tercih edilme durumunu belirler. En kısa yolu kullanan karıncalar daha sık gidip gelerek daha yoğun feromon bırakırlar, dolayısıyla arama süreci doğal olarak en kısa olan yollara doğru kayar (Cura, 2008: 113-114).

**Şekil 1:** Yiyecek arayan karıncaların feromon oranına göre izlediği yol

KKO ile ilgili literatürde farklı alanlarda yapılan güncel çalışmalara, finans (Terzi ve Şen, 2021), tedarik zinciri yönetimi (Hong vd., 2018), optimizasyon problemleri (Liao vd., 2014, Abachizadeh ve Tahani, 2009), görüntü algılama (Tian vd., 2008), atama problemleri (D'Acerno vd., 2006, Wiesemann ve Stützel 2006), rotalama (Alp ve Özalp, 2020, Pala ve Aksaraylı, 2018, Pinto ve Barán, 2005) çalışmaları örnek olarak verilebilir.

Bu çalışma ile gezgin satıcı probleminin KKO yöntemi ile çözümünün etkinleştirilmesi için (uygun çözüme daha hızlı ulaşılması) önce noktaların BCO Algoritması kümelmesi sağlanmıştır. Bu sayede çözüm daha hızlı ulaşılması sağlanmıştır.

### Yöntem

Bu bölümde, önce gezgin satıcı problemi tanımlanmış daha sonra çalışmada kullanılan algoritmalar olan Bulanık C-Ortalama Kümeleme (BCO) Algoritması ve Karınca Kolonisi Optimizasyonu (KKO) yöntemleri anlatılmıştır.

#### **Bulanık c-ortalama kümeleme**

Kümeleme algoritması; veri setini spesifik bir kritere göre farklı dallara veya kümelere ayırır. Böylece kümelerin elemanları arasındaki benzerlik oranı oldukça fazla olduğu halde diğer kümelerin elemanları ile oldukça büyük farklar vardır. Hedeflenen kümeleme şekline göre, Bulanık C-Ortalama Kümeleme, Olasılıklı C-Ortalama Kümeleme, Bulanık Olasılıklı C-Ortalama Kümeleme gibi çeşitli kümeleme algoritmaları vardır (Guo vd., 2020: 240). İlk olarak 1974'te Dunn tarafından önerilen BCO kümeleme algoritması 1981'de Bezdek tarafından geliştirilmiş ve pek çok uygulanma alanı bulmuştur (Türe ve Başer, 2015: 18).

Başlangıç, BCO algoritmasının performansı için hayati öneme sahiptir. Algoritma, objektif fonksiyonun optimizasyonu için gerekli koşulları sağlar, ancak BCO algoritmasının global minimum bulmasını garanti etmez (Valente de Oliveira ve Pedrycz, 2007: 399).

BCO algoritması  $X = \{x_1, x_2, \dots, x_N\}$  veri setini C bulanık küme içine atar ve her grup için kümeleme merkezini hesaplar. Algoritmanın fonksiyonu (Işık ve Çakmurcu, 2007: 33-34):

$$\min_{U,V} \left( J_{FMC}(U,V) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_j - v_i\|^2 \right), \quad 0 \leq u_{ij} \leq 1 \quad (1)$$

$$\sum_{i=1}^c u_{ij} = 1 \quad (2)$$

$u_{ij}$ ,  $i$ 'nci kategoriye ait  $j$ 'nci üyenin üyelik derecesi;  
 $m$ , kümeleme sonuçlarının bulanıklaştırma derecesini gösteren bulanık ağırlıklı dizin;  
 $v_i$ ,  $i$ 'nci kümeleme merkezi.

Amaç fonksiyonunu minimize etmek için Lagrange Çarpımı yönteminin kullanıldığı yinelemeli formül:

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}}}, \quad i = 1, \dots, C; j = 1, \dots, N \quad (3)$$

$$v_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m}, \quad i = 1, \dots, C \quad (4)$$

### Karınca kolonisi optimizasyonu

#### Geçiş kuralının belirlenmesi

KKO problemlerinde,  $k$  karıncasının  $i$  noktasından  $j$  noktasına gitme olasılığı  $p_k(i, j)$  olarak ifade edilir ve Eşitlik 1'de verildiği gibi hesaplanmaktadır (Junie ve Dingwei, 2006: 3).

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in j_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta} & \text{eğer } j \in j_k(i) \\ 0 & \text{Diğer durumlarda} \end{cases} \quad (5)$$

$\tau(i, j)$ ,  $i$  ve  $j$  noktaları arasındaki feromon miktarı  
 $\eta(i, j)$ ,  $i$  ve  $j$  noktaları arasındaki mesafenin tersi  $1/\delta(i, j)$   
 $\alpha \in [0, 1]$  ve  $\beta \in [0, 1]$ , ayarlanabilir parametreler  
 $i$ , karıncanın bulunduğu nokta  
 $j$ , karıncanın gideceği noktayı ifade etmektedir.

#### Yerel feromon güncellemesi

Karıncaların  $i$  noktası ile  $j$  noktası arasında gidip gelirken feromon miktarına yaptığı toplam katkı  $\Delta\tau_{ij}$  ifadesi ile tanımlanmaktadır. Problemdeki karıncaların bir önceki iterasyonda geçmiş olduğu noktalar arasına bıraktıkları feromon düzeyleri buharlaşma oranı  $\rho$  da dikkate alarak hesaplayan ifade Eşitlik 2'de verilmiştir.

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t+1) \quad (6)$$

$\tau_{ij}(t)$ ,  $t$  iterasyonuna kadar biriken feromon düzeyi  
 $\Delta\tau_{ij}^k(t+1)$ ,  $t$  iterasyonundaki feromon düzeyi  
 $\rho$  ( $0 \leq \rho \leq 1$ ), feromon buharlaşma parametresi

$$\Delta\tau_{ij}^k(t+1) = \begin{cases} \frac{1}{L^k(t+1)} & k \text{ karıncası } (i, j) \text{ yolunu kullanmışsa} \\ 0 & \text{diğer durumda} \end{cases} \quad (7)$$

$L^k(t + 1)$ ,  $k$  karıncasının toplam tur uzunluğu

### Genel feromon güncellemesi

En uygun çözümü veren karıncanın geçmiş olduğu noktalar arasına bıraktıkları feromon miktarı genel feromon güncellemesi olarak ifade edilmekte ve Eşitlik 4'deki gibi hesaplanmaktadır.

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^k(t + 1) \quad (8)$$

$$\Delta\tau_{ij}^k(t + 1) = \begin{cases} \frac{1}{L_{best}(t + 1)} & (i, j) \text{ en iyi tura ait ise} \\ 0 & \text{Diğer durumda} \end{cases} \quad (9)$$

$L_{best}(t + 1)$ , geçerli iterasyonda bulunan en iyi turun uzunluğu

### Parametreler

$\alpha$  değeri; ilgili yoldaki feromon seviyesinin önemini belirleyen ve daha önceki iterasyon sonuçlarının sonraki iterasyonlara aktarılmasını sağlayan bir parametredir.  $\alpha$  değeri yüksek ise feromon düzeyinin yüksek olduğu yollar daha büyük olasılıkla seçilir ve tesadüfilik azalır.

$\beta$  değeri; yol uzunluğunun bir sonraki noktanın seçimine olan etkisini belirleyen bir parametredir.  $\beta$  değerindeki artış bir sonraki yolun seçimindeki tesadüfiliği arttırırken,  $\beta$  değerindeki düşüş alternatif çözümlerin araştırılması ihtimalini de azaltmaktadır.

$\rho$ , buharlaşma oranını ifade eder.

### Çoklu gezgin satıcı problemi için karınca kolonisi optimizasyonu algoritması:

Çoklu GSP'nin KKO ile çözümü farklılık göstermektedir. Bir satıcı bazı lokasyonlara gider, ardından ikinci satıcı gidilmemiş lokasyonlardan bazılarını gider ve süreç tüm satıcıların katılımı ve tüm lokasyonların ziyaret edilmesi ile tamamlanır.

$$\begin{cases} 2 \leq tn_i \leq l_i \\ \sum_i^m tn_i = n - 1 \end{cases} \quad (i = 1, 2, \dots, m) \quad (10)$$

$tn_i$ ,  $i$  satıcısının gideceği lokasyon sayısı

$l_i$ ,  $i$  satıcısının gideceği maksimum lokasyon sayısı

$m$ , satıcı sayısı

$n$ , toplam lokasyon sayısı

Turlar her karıncanın başlangıç noktasından çıkarak gidilmemiş lokasyonları gezmesi ile tamamlanır. Karıncaların başlangıç noktası veya yapay lokasyondan tura başlamasıyla  $i$  satıcısının  $i$ 'nci zamanda turu başlar. Bir sonraki satıcının turu ise karıncaların başlangıç noktası veya yapay noktadan geçişi ile olur (Junjie ve Dingwei, 2006: 2).

Bu çalışmada Çoklu GSP'nin çözümü için doğrudan KKO uygulaması kullanılmamış; bunun yerine BCO Algoritması ile kümeleme yapıldıktan sonra her veri seti için KKO uygulanmıştır.

### Uygulama

Bu çalışmada çoklu bir GSP problemi için uygun rotaların (3, 4 ve 5) belirlenmesi amaçlanmıştır. Önce BCO ile lokasyonlar kümelere ayrılmış daha sonra her bir lokasyonda en uygun rota KKO ile belirlenmiştir.



Çalışmada rasgele oluşturulan toplam orta büyüklükteki bir problem olması için 31 lokasyon kullanılmıştır. Lokasyonların tümüne uğrayacak olan 3, 4 ve 5 satıcı bulunduğu durumlar ayrı ayrı ortaya konulmuş, rotalar belirlenmiş, elde edilen çözümler incelenmiş ve karşılaştırılmıştır.

Uygulamada kullanılacak olan,  $(x_0, y_0)$  başlangıç noktası koordinatları olmak üzere rastgele seçilen 31 lokasyona ait  $(x_i, y_i)$  koordinatları Tablo 1’de verilmiştir:

**Tablo 1:** Rastgele seçilmiş lokasyonların  $(x, y)$  koordinatları

Nokta	$x_i$	$y_i$
0	37,6485	24,9652
1	99,2393	54,1482
2	20,3073	52,1406
3	37,5416	94,1816
4	70,8670	63,7501
5	16,3497	71,8557
6	32,3962	42,4441
7	80,9012	24,8357
8	21,8956	93,5987
9	32,8388	80,2128
10	46,1998	76,1633
11	35,2330	6,1752
12	7,3539	90,6653
13	50,5193	65,5628
14	27,4199	94,7949
15	56,7635	54,5241
16	2,7511	30,5318
17	61,3717	8,8515
18	42,3611	42,1534
19	64,5295	36,7533
20	30,6118	85,2528
21	29,0784	98,3923
22	43,0515	5,3853
23	55,0274	35,7135
24	12,2461	99,9067
25	13,0425	41,5017
26	22,8788	38,0647
27	10,6923	10,4579
28	9,3037	58,9612
29	18,8572	14,6320
30	68,3495	37,6575

BCO Algoritması kullanılarak başlangıç noktası hariç olmak üzere, 30 lokasyon için Matlab ile 3 farklı kümeleme yapılmıştır:

### **3’lü bulanık kümeleme ve kko algoritması**

30 lokasyon için her biri 10 elemanlı olmak üzere 3 küme BCO Algoritması tarafından oluşturulmuş; başlangıç noktası her bir kümeye eklenmek suretiyle lokasyonlar KKO Algoritmasına veri olarak yüklenmiştir.

Başlangıç Noktası = [37.6485, 24.9652] olarak belirlendiği durumda 3 satıcı için uğraması gereken ve BCO algoritması ile belirlenen lokasyonlar Tablo 2’de sunulmuştur.

**Tablo 2:** 3’lü kümeleme sonuçları

1. Küme	2. Küme	3. Küme
2	1	3
6	4	5
11	7	8
16	13	9
22	15	10
25	17	12
26	18	14
27	19	20
28	23	21
29	30	24

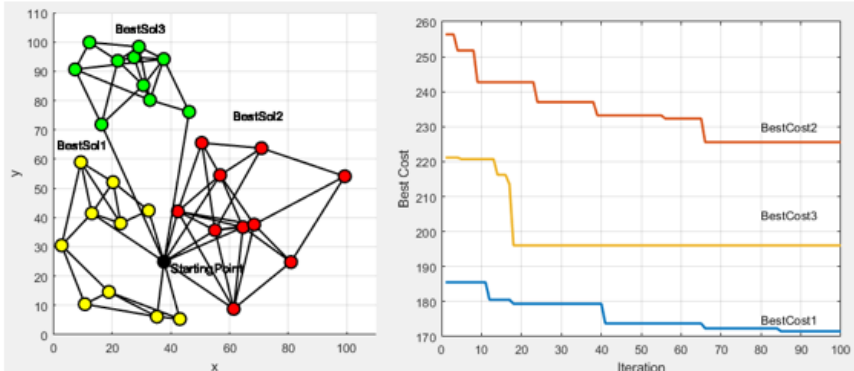
Tablo 3, BCO algoritması ile belirlenen lokasyonlar için her bir satıcının en uygun rotası KKO Algoritması kullanılarak belirlenen sonuçları göstermektedir.

**Tablo 3:** 3’lü kümeleme için kko algoritma sonuçları

Küme	Tur	Mesafe
1. Küme	0 → 6 → 26 → 25 → 2 → 28 → 16 → 27 → 29 → 11 → 22 → 0	171,5087
2. Küme	0 → 18 → 15 → 13 → 4 → 1 → 7 → 17 → 19 → 30 → 23 → 0	225.5878
3. Küme	0 → 5 → 12 → 24 → 8 → 14 → 21 → 3 → 20 → 9 → 10 → 0	196.0835
<b>TOPLAM:</b>		<b>593.1800</b>

3 satıcı için belirlenen en uygun rotalar ve çözüme ulaşmak için gerçekleştirilen tekrar sayıları Şekil 2’de gösterilmiştir.

**Şekil 2:** 3’lü kko algoritması için en iyi çözüm ve en iyi sonuç grafikleri



#### 4'lü bulanık kümeleme ve kko algoritması

30 lokasyon için 4 küme BCO Algoritması tarafından oluşturulmuştur. BCO sonucunda her bir satıcı için uğraması gereken lokasyon sayısı farklı olarak bulunmuştur. Her bir satıcının uğraması gereken lokasyon sayısı kümelerdeki eleman sayısı olarak aşağıdaki gibidir:

- 1. Küme: 5 eleman
- 2. Küme: 8 eleman
- 3. Küme: 7 eleman
- 4. Küme: 10 eleman

Başlangıç noktası her bir kümeye eklenmek suretiyle lokasyonlar KKO Algoritmasına veri olarak yüklenmiştir.

Başlangıç Noktası = [37.6485, 24.9652] olarak belirlendiği durumda 4 satıcı için uğraması gereken ve BCO algoritması ile belirlenen lokasyonlar Tablo 4'de sunulmuştur.

**Tablo 4:** 4'lü kümeleme sonuçları

1. Küme	2. Küme	3. Küme	4. Küme
11	1	2	3
17	4	6	5
22	7	16	8
27	13	18	9
29	15	25	10
	19	26	12
	23	28	14
	30		20
			21
			24

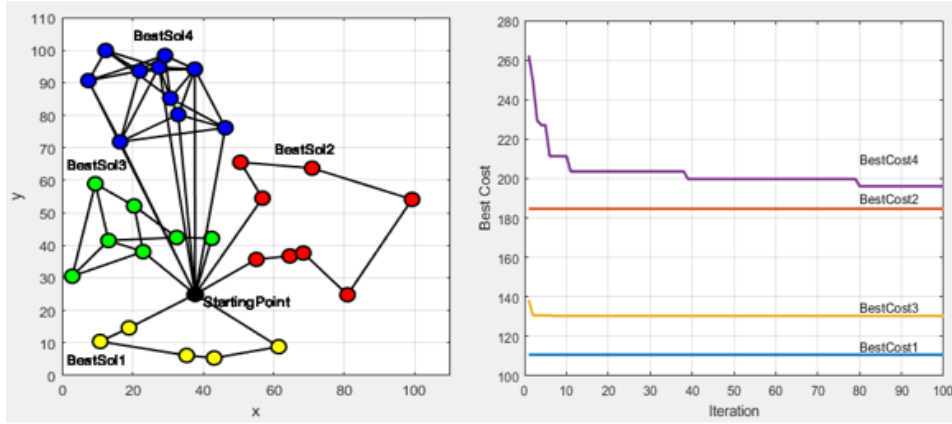
BCO algoritması ile belirlenen lokasyonlar için her bir satıcının en uygun rotası KKO Algoritması kullanılarak belirlenmiş ve Tablo 5'de verilmiştir.

**Tablo 5:** 4'lü kümeleme için kko algoritma sonuçları

Küme	Tur	Mesafe
1. Küme	0 → 29 → 27 → 11 → 22 → 17 → 0	110.7083
2. Küme	0 → 15 → 13 → 4 → 1 → 7 → 30 → 19 → 23 → 0	184.7021
3. Küme	0 → 26 → 16 → 25 → 28 → 2 → 6 → 18 → 0	130.3649
4. Küme	0 → 5 → 12 → 24 → 8 → 14 → 21 → 3 → 20 → 9 → 10 → 0	196.0835
<b>TOPLAM:</b>		<b>621.8588</b>

4 satıcı için belirlenen en uygun rotalar ve çözüme ulaşmak için gerçekleştirilen tekrar sayıları Şekil 3’de gösterilmiştir.

**Şekil 3:** 4’lü kko algoritması için en iyi çözüm ve en iyi sonuç grafikleri



### 5’li bulanık kümeleme ve kko algoritması

30 lokasyon için 5 küme BCO Algoritması tarafından oluşturulmuştur. BCO sonucunda her bir satıcı için uğraması gereken lokasyon sayısı farklı olarak bulunmuştur. Her bir satıcının uğraması gereken lokasyon sayısı kümelerdeki eleman sayısı olarak aşağıdaki gibidir.

- 1. Küme: 5 eleman
- 2. Küme: 6 eleman
- 3. Küme: 9 eleman
- 4. Küme: 4 eleman
- 5. Küme: 6 eleman

Başlangıç noktası her bir kümeye eklenmek suretiyle lokasyonlar KKO Algoritmasına veri olarak yüklenmiştir.

Başlangıç Noktası = [37.6485, 24.9652] olarak belirlendiği durumda 5 satıcı için uğraması gereken ve BCO algoritması ile belirlenen lokasyonlar Tablo 6’da sunulmuştur.

**Tablo 6:** 5’li kümeleme sonuçları

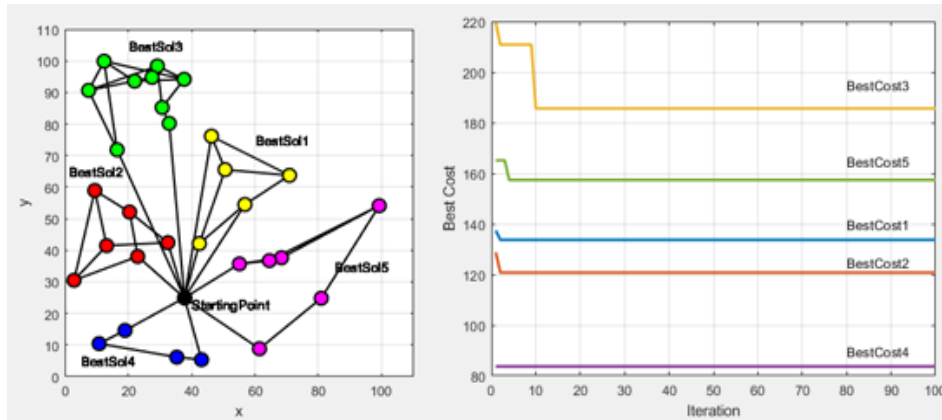
1. Küme	2. Küme	3. Küme	4. Küme	5. Küme
4	2	3	11	1
10	6	5	22	7
13	16	8	27	17
15	25	9	29	19
18	26	12		23
	28	14		30
		20		
		21		
		24		

BCO algoritması ile belirlenen lokasyonlar için her bir satıcının en uygun rotası KKO Algoritması kullanılarak belirlenmiş ve Tablo 7’de verilmiştir.

**Tablo 7:** 5'li kümeleme için kko algoritma sonuçları

Küme	Tur	Mesafe
1. Küme	0 → 15 → 4 → 10 → 13 → 18 → 0	110.7083
2. Küme	0 → 6 → 2 → 28 → 25 → 16 → 26 → 0	120.8242
3. Küme	0 → 9 → 20 → 3 → 21 → 14 → 8 → 24 → 12 → 5 → 0	185.6715
4. Küme	0 → 29 → 27 → 11 → 22 → 0	83.6966
5. Küme	0 → 7 → 30 → 1 → 23 → 17 → 19 → 0	157.4257
<b>TOPLAM:</b>		<b>607.6170</b>

5 satıcı için belirlenen en uygun rotalar ve çözüme ulaşmak için gerçekleştirilen tekrar sayıları Şekil 4'te gösterilmiştir.

**Şekil 4:** 5'li kko algoritması için en iyi çözüm ve en iyi sonuç grafikleri

### Bulgular ve tartışma

Bu çalışmada Çoklu GSP için KKO kullanılmış ve veri setini gruplara ayırmak üzere BCO Algoritması kullanılmıştır. Veri seti 3, 4 ve 5'li kümelere ayrılarak her kümeleme için KKO Algoritması Matlab ile çalıştırılmış ve 100 iterasyon sonucu elde edilen her bir sonuç için çözüme elde edilmiş bu çözümlere ait tablo ve grafikler Uygulama bölümünde ayrıntılı bir biçimde sunulmuştur.

Her 3 kümeleme için elde edilen sonuçlar Tablo 8'de özetlenmiştir. Buna göre; bir başlangıç noktası ve 30 lokasyondan ibaret olan ziyaret noktaları için 3'lü kümeleme ile elde edilen sonuçlar, yapılan çalışma için en kısa mesafe olarak hesaplanmıştır. 3'lü kümelemeyi 5'li kümeleme takip etmektedir. 4'lü kümeleme en uzun mesafe olarak hesaplanmıştır.

**Tablo 8:** Sonuçların karşılaştırılması

Kümeleme	Toplam Mesafe
3'lü Kümeleme	593,1800
4'lü Kümeleme	621,8588
5'li Kümeleme	607,6170

3'lü kümeleme sonrasında yapılan çözüm sonucunda elde edilen rotanın uzunluğunun 4'lü kümelemeden sonra elde edilen çözümden %4,61, 5'li kümelemeden elde edilen rotanın uzunluğundan ise %2,37 daha kısa olduğu belirlenmiştir.

Toplam mesafesi 593,1800 olan 3'lü kümeleme sonrası elde edilen çözümde, her bir rotanın ortalama mesafesi 197,0835, en kısa rota 171,5087 ve en uzun rota ise 225,5878 birim olarak bulunmuştur. Toplam mesafesi 621,8588 olan 4'lü kümeleme sonrası elde edilen çözümde, her bir rotanın ortalama mesafesi 155,447, en kısa rota 110,7083 ve en uzun rota ise 196,0835 birim olarak bulunmuştur. Toplam mesafesi 607,6170 olan 5'li kümeleme sonrası elde edilen çözümde ise her bir rotanın ortalama mesafesi 11,6653, en kısa rota 83,6966 ve en uzun rota ise 185,6715 birim olarak bulunmuştur.

### Sonuç ve öneriler

En kısa mesafe 3'lü kümeleme sonucunda elde çözümün olduğu görülmektedir. 4'lü kümeleme incelendiğinde toplam mesafe olarak en yüksek sonuç karşımıza çıkmaktadır. 3'lü kümeleme ile karşılaştırıldığında hem araç maliyetini hem de toplam mesafeye bağlı olarak yakıt maliyetini arttırmaktadır. Bu nedenle tercih edilmesi uygun değildir. 5'li kümeleme değerlendirildiğinde toplam mesafe açısından 3'lü kümeleme ile 4'lü kümeleme arasında kaldığı görülmektedir. Yine araç ve yakıt maliyetlerini arttırmakla beraber diğer kriterler de göz önüne alınarak değerlendirildiğinde 5'li kümeleme tercih edilebilir.

3'lü kümeleme uygulanacak sonuç olarak değerlendirmeye alındığı takdirde lokasyonların ziyareti için görev; her biri tek seferde tamamlanmak üzere, 3 kişiye veya 1 kişi için 3 farklı güne atanabilir. Diğer kümelemeler de benzer şekilde değerlendirilebilir. Karar verici satıcı/araç sayısı maliyeti ile kat edilen toplam mesafe maliyetini birlikte dikkate alarak kendisi için en uygun seçeneğe karar verebilir

Çözümler değerlendirme yapılırken noktalar arasındaki mesafenin Öklidyen Uzaklık olduğu unutulmamalıdır. Çizilecek rotalar için ulaşımına açık yollar ve trafik durumu dikkate alınarak çizelgeleme yapılmalıdır.

Öte yandan rotalama yapılırken teslim süresi ve maliyet beraber değerlendirilmelidir. Öncelikli olan hızlı teslimat ise veya teslimatın gecikmesi ceza veya yaptırıma tabi ise mutlaka değerlendirmede öncelikli olarak ele alınmalıdır. Eğer değerlendirmede öncelikli unsur maliyet ise gerekli araç sayısı, toplam mesafe ve yakıt maliyeti gibi kriterler öncelikli olacaktır.

Yöntem (KKO algoritması); meta-sezgisel bir uygulama olduğundan elde edilen çözümün optimum olması garanti edilemez, ancak kısa sürede ve kabul edilebilir ve uygulanabilir düzeyde, hesaplanan en iyi çözüm olarak değerlendirilmiştir. Gelecek çalışmalarda aynı veriler farklı meta-sezgisel uygulamalar (genetik algoritmalar, parçacık sürü optimizasyonu, yapay arı kolonisi vb.) ile değerlendirilip sonuçlar karşılaştırılabilir, uygulama birkaç defa tekrar edilerek sonuçlar kendi içinde karşılaştırılıp değerlendirilebilir veya iterasyon sayısı artırılarak döngünün kendi içinde çok daha fazla sayıda tekrarlanması suretiyle sonuçlar karşılaştırılarak tekrar değerlendirme yapılabilir. Çalışmada lokasyon sayısı 31 olarak belirlenmiştir. Farklı lokasyon sayıları kullanılarak çözümler yapılabilir ve elde edilen çözümler karşılaştırılabilir.

Çalışma; karar vericiye karar verme sürecinde alternatiflerin belirlenmesi amacıyla rasgele üretilen veriler kullanılarak yapılmıştır. Gerçek veriler ile çalışılarak ilgili firma için karar destek mekanizması oluşturulabilir.

### Kaynakça

- Abachizadeh, M., ve Tahani, M. (2009). An ant colony optimization approach to multi-objective optimal design of symmetric hybrid laminates for maximum fundamental frequency and minimum cost. *Structural and Multidisciplinary Optimization*, 37(4), 367-376.
- Ahmed Z. H. ve Pandit, S. N. (2001). The travelling salesman problem with precedence constraints. *Opsearch*, 38(3), 299-318.
- Alp S. ve Özalp M. (2020). Uygun dağıtım rotası belirlenmesi probleminde hibrit sezgisel bir yöntem önerisi: bir kargo firması örneği. *Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi*, 3(1), 59-70.
- Arthur J.L. ve Frenmdewey J.O. (2017). Generating travelling-salesman problems with known optimal tours. *Journal of the Operational Research Society*, 39(2), 153-159.
- Aytekin İ. (2022). Türkiye’de karayolu ve demiryolu ulaştırma hizmetleri ile kalkınma arasındaki nedensellik ilişkisinin analizi. *Anadolu İktisat ve İşletme Dergisi*, 6(1), 17-35.
- Aziz Z.A. (2015). Ant colony hyper-heuristics for travelling salesman problem. *IEEE International Symposium on Robotics and Intelligent Sensors (IEEE IRIS2015)*, Procedia Computer Science 76: 534-538.
- Bezdek J.C. (1981). Pattern recognition with fuzzy objective function algorithms. Plenum Press.
- Christofides N. ve Eilon S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, 20, 309-318.
- Cura T. (2008). Modern sezgisel teknikler ve uygulamaları. Papatya Yayınları.
- Dantzig G., Fulkerson R. ve Johnson S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4), 393-410.
- Dunn, J.C. (1974). A fuzzy relative ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3), 32-57.
- D’Acierno, L., Montella, B., ve Lucia, F. D. (2006). A stochastic traffic assignment algorithm based on ant colony optimisation. In *International Workshop on Ant Colony Optimization and Swarm Intelligence*, (25-36).
- Gilmore P.C. ve Gomory R.E. (1964). Sequencing a one state-variable machine: a solvable case of the traveling salesman problem. *Journal of Operations Research*, 12(5), 655-679.
- Guo Y., Zi Y. ve Jiang Y. (2020). Contrastive study of distributed multitask fuzzy c-means clustering and traditional clustering algorithms. In *2020 5th International Conference on Communication, Image and Signal Processing (239-245)*. IEEE.
- Hong J., Diabat A., Panicker V. V. ve Rajagopalan S. (2018). A two-stage supply chain problem with fixed costs: An ant colony optimization approach. *International Journal of Production Economics*, 204, 214-226.
- Frieze A. M. (1987). On the exact solution of random travelling salesman problems with medium size integer coefficients. *SIAM Journal on Computing*, 16(6), 1052-1072.
- Işık M. ve Çamurcu A.Y. (2007). K-means, k-medoids ve bulanık c-means algoritmalarının uygulamalı olarak performanslarının tespiti. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 6(11), 31-45.
- Junjie P. ve Dingwei W. (2006). An ant colony optimization algorithm for multiple travelling salesman problem. *Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC’06)*.
- Karg R.L. ve Thompson G.L. (1964). A heuristic approach to solving travelling salesman problems. *Journal of Management Science*, 10(2), 225-248.
- Kuzu S., Önay O., Şen U., Tunçer M., Yıldırım B.F. ve Keskindürk T. (2014). Gezgin satıcı problemlerinin metasezgiseller ile çözümü. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 43(1), 1-27.

- Lammel B., Gryzłak K., Dornberger R. ve Hanne T. (2016). An ant colony system solving the travelling salesman region problem. *Computational and Business Intelligence (ISCBI), 4th International Symposium on* :125-131. 2016.
- Liao T., Stützle T., de Oca M. A. M. ve Dorigo M. (2014). A unified ant colony optimization algorithm for continuous optimization. *European Journal of Operational Research*, 234(3), 597-609.
- Lin S. ve Kernighan B.W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Journal of Operations Research*, 21(2), 498-516.
- Liu Z., Wang X., Yu H. ve Xing R. (2020). Thermal efficiency prediction model of cement clinker production based on fuzzy c-means monitoring clustering, *39th Chinese Control Conference (CCC) Chinese Control Conference (CCC)*: 5391-5394.
- Mausor F.H., Jaafar J. ve Taib S.M. (2020). Missing values imputation using fuzzy c means based on correlation of variable. *International Conference on Computational Intelligence (ICCI) Computational Intelligence (ICCI)*:261-265.
- Mavrovouniotis, M., Shengxiang Y. ve Xin Y. (2014). Multi-colony ant algorithms for the dynamic travelling salesman problem. *Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*:9-16.
- Meniaïlov I., Chumachenko D. ve Bazılevych K. (2020). Determination of heart disease based on analysis of patient statistics using the fuzzy c-means clustering algorithm. *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) Data Stream Mining and Processing (DSMP)*:333-336.
- Nicholson T. A. J. (1967). A sequential method for discrete optimization problems and its application to the assignment, travelling salesman, and three machine scheduling problems. *IMA Journal of Applied Mathematics*, 3(4), 362–375.
- Ratliff H.D. ve Rosenthal A.S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Journal of Operations Research*, 31(3), 507-521.
- Pala O. ve Aksaraylı M. (2018). Çok amaçlı kapasite kısıtlı araç rotalama problemi çözümünde bir karınca kolonisi optimizasyon algoritması yaklaşımı. *Alphanumeric Journal* , 6 (1) , 37-48 .
- Pinto D. ve Barán B. (2005). Solving multiobjective multicast routing problem with a new ant colony optimization approach. In *Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking*:(11-19).
- Shen Y., Pedrycz W., Chen Y., Wang X. ve Gacek A. (2020). Hyperplane division in fuzzy c-means: clustering big data. *IEEE Transactions on Fuzzy Systems IEEE Trans. Fuzzy Syst. Fuzzy Systems, IEEE Transactions on*. 28(11):3032-3046.
- Shrivastava K. ve Kumar S. (2018). The effectiveness of parameter tuning on ant colony optimization for solving the travelling salesman problem. *8th International Conference on Communication Systems and Network Technologies (CSNT)*.
- Srikakulapu R. ve Vinatha U. (2018). Optimized design of collector topology for offshore wind farm based on ant colony optimization with multiple travelling salesman problem. *Journal of Modern Power Systems and Clean Energy*, 6(6): 1181-1192.
- Sureja N.M. ve Chawda B.V. (2012). Random Travvelling Salesman Problem usşng Genetic Algorithms. *International Journal of Computing*, 2(2): 359-363.
- Şahin Y. (2019). Sezgisel ve metasezgisel yöntemlerin gezgin satıcı problemi çözüm performanslarının kıyaslanması. *BAİBÜ Sosyal Bilimler Enstitüsü Dergisi*, 19(4), 911-932.
- Terzi S. ve Şen İ. K. (2021). Finansal Başarısızlık Tahmininde Karınca Kolonisi Optimizasyon Yaklaşımı: Borsa İstanbul'da Bir Araştırma. *İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 20(40), 525-541.
- Thilagam M., Arunesh K. ve Rajeshkann, A. (2020). Analysis of brain MRI images for tumor segmentation using fuzzy c means algorithm. *International Conference on Electronics and Sustainable Communication Systems (ICESC) Electronics and Sustainable Communication Systems (ICESC), 2020 International Conference on*.: (183-186).
- Tian J., Yu W. ve Xie S. (2008). An ant colony optimization algorithm for image edge detection. *Congress On Evolutionary Computation (Ieee World Congress On Computational Intelligence)*:751-756.



- Tomanova P. ve Holy V. (2020). Ant colony optimization for time-dependent travelling salesman problem. *In Proceedings of the 2020 4th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence: (47-51)*. New York.
- Türe H. ve Başer F. (2015). Bulanık c-ortalama kümeleme algoritması ile ülke risk değerlendirmesi. *İstanbul Üniversitesi İktisat Fakültesi Ekonometri ve İstatistik Dergisi*, 23, 16-33.
- Valente de Oliveira J. ve Pedrycz W. (2007). *Advances in fuzzy clustering and its applications*. New York: John Wiley & Sons, Ltd.
- Wiesemann W. ve Stützle T. (2006). Iterated ants: An experimental study for the quadratic assignment problem. *In International Workshop on Ant Colony Optimization and Swarm Intelligence: (179-190)*. Springer, Berlin, Heidelberg.

#### **Etik kurul onayı**

Yapılan çalışmada anket kullanılmaması ve analizi yapılan verilerin rastsal örnekleme olarak yazarlar tarafından üretilmiş olması sebebi ile bu araştırma etik kurul izni gerektirmeyen çalışmalar arasında yer almaktadır.

#### **Araştırmacıların katkı oranı beyanı**

Yazarlar çalışmaya eşit oranda katkı sağlamıştır.

#### **Çıkar çatışması beyanı**

Bu çalışmada herhangi bir potansiyel çıkar çatışması bulunmamaktadır.