



## Capacitated vehicle routing problem with simulated annealing algorithm with initial solution improved with fuzzy c-means algorithm

Ahmet Fatih Eker<sup>1</sup> , Ahmet Yunus Çil<sup>2</sup> , İbrahim Çil<sup>1\*</sup>

<sup>1</sup>Department of Industrial Engineering, Faculty of Engineering, Sakarya University, 54050, Sakarya, Turkey.

<sup>2</sup>Department of Industrial Engineering, Faculty of Engineering, Kocaeli University, 41380, Kocaeli, Turkey.

### Highlights:

- Capacitated vehicle routing problem with pick-up or delivery is considered
- Initial solution method developed with fuzzy c-means is developed for the problem
- Performance of the model is tested on test instances

### Keywords:

- Vehicle routing problem
- Simulated annealing
- Fuzzy c-means
- Optimization

### Article Info:

Research Article  
Received: 24.08.2020  
Accepted: 20.08.2021

### DOI:

10.17341/gazimmfd.784653

### Correspondence:

Author: İbrahim Çil  
e-mail: icil@sakarya.edu.tr  
phone: +90 532 450 7393

### Graphical/Tabular Abstract

In this study, the initial solution of Simulated Annealing method is improved with Fuzzy c-means. The vehicle routing problem with capacitated aims to determine the optimal set of routes for a fleet of vehicles serving customers with various demand. This method, which is used to find overlapping clusters through data, has a flexible structure and has a better classification ability than other clustering algorithms.

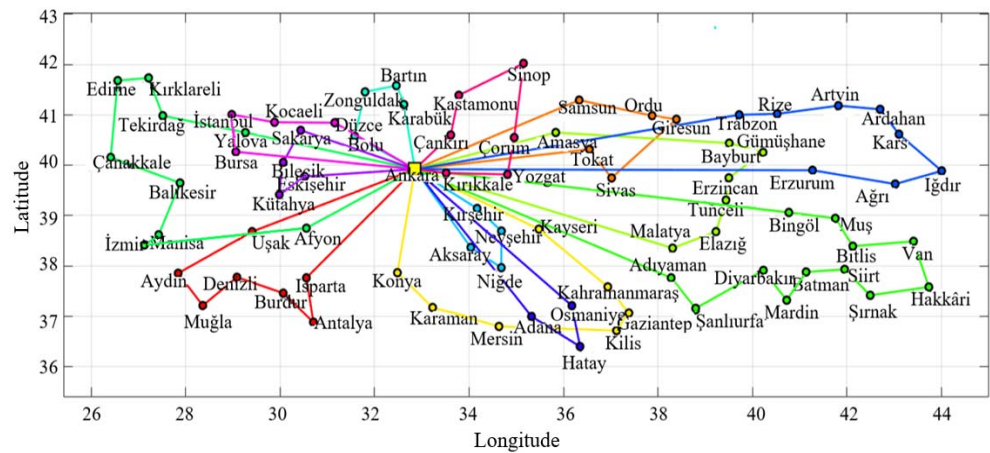


Figure A. Final solution of 13 cluster vehicle routing problem

**Purpose:** The aim of this study is to obtain efficient solution procedures for the vehicle routing problems for which minimization of route distances is of importance.

### Theory and Methods:

Metaheuristic methods are sensitive to the initial solution. So the nodes were clustered with the Fuzzy c means to improve the initial solution. The result from here is given as an input to the Simulation Annealing algorithm. The routing problem of the capacity vehicle has been solved.

### Results:




Both the accuracy and the efficiency of the developed method are proven by the computational results.

### Conclusion:

In this paper, a vehicle routing problem with capacitated is addressed. Solution procedures which are able to find good feasible solutions within a small amount of time are obtained. This method can be used when small and complex problems need to be solved. Proposed mathematical model and heuristic algorithm give better results.



## Bulanık c-ortalama algoritması ile başlangıç çözümü iyileştirilmiş tavlama benzetimi algoritması ile kapasiteli araç rotalama problemi

Ahmet Fatih Eker<sup>1</sup> , Ahmet Yunus Çil<sup>2</sup> , İbrahim Çil<sup>1\*</sup> 

<sup>1</sup>Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 54050, Serdivan, Sakarya, Türkiye

<sup>2</sup>Kocaeli Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 41380, İzmit, Kocaeli, Türkiye

### Ö N E Ç I K A N L A R

- Toplamalı veya dağıtım kapasiteli araç rotalama problemi üzerinde çalışıldı
- Problem için bulanık c-ortalama ile iyileştirilmiş başlangıç çözümü yöntemi geliştirilmiştir
- Modelin performansı test edildi ve standart model ile karşılaştırıldı

#### Makale Bilgileri

Araştırma Makalesi  
Geliş: 24.08.2020  
Kabul: 20.05.2021

#### DOI:

10.17341/gazimmfd.784653

#### Anahtar Kelimeler:

Tavlama benzetimi,  
kapasiteli araç rotalama  
problemi,  
optimizasyon,  
bulanık c-ortalama

#### ÖZ

Bu çalışmada, popüler bir problem olan Araç Rotalama Problemi üzerinde çalışılmıştır. Problemi çözmek için meta-sezgisel bir yöntem olan Tavlama Benzetimi kullanılmıştır. Genel olarak tavlama benzetimi algoritması, metallerin tavlama işlemini taklit eden değişken sıcaklık parametresine göre tekrarlayan bir işlemdir. Bizim çalışmamız için bu yöntemin en büyük sorunu, algoritmayı başlatmak için kullanılan başlangıç çözümünü rassal olarak oluşturmasıdır. Bu sebepten dolayı optimum çözüme ulaşmak için kullanılan arama uzayı büyük olduğundan, çözüm süresi (veya iterasyon sayısı) artacaktır. Daha iyi bir başlangıç çözümüyle optimum çözüme ulaşmak daha kısa zaman alacaktır. Ulaşmak istediğimiz optimum çözüm minimum mesafe olduğundan, başlangıç çözümünü iyileştirmek için Bulanık c ortalama kullanılarak rotalar kümelendi. Bulanık mantık gereği, her verinin 0-1 arasında birden fazla kümeye dahil olabilmesi durumu, algoritmanın her çözümünde başlangıç çözümünü değiştireceğinden dolayı optimum çözüme yaklaşma durumu olacaktır. Aynı veriler ve aynı parametreler kullanılarak rassal başlangıç çözümü kullanan tavlama benzetimi ve bulanık c ortalama ile başlangıç çözümü iyileştirilmiş tavlama benzetimi ile problem çözülmüştür. Bulanık c ortalama yöntemi başlangıç arama uzayını %57 oranında azaltmıştır. Dolayısıyla bulanık c ortalama aynı çözüm süresinde ve aynı iterasyon sayısında optimum çözüme daha yakın sonuçlar vermiştir.

## Capacitated vehicle routing problem with simulated annealing algorithm with initial solution improved with fuzzy c-means algorithm

### H I G H L I G H T S

- Pickup or delivery capacitated vehicle routing problem was studied
- Initial solution method improved with fuzzy c means is developed for the problem
- The performance of the model has been tested and compared to the standard model

#### Article Info

Research Article  
Received: 24.08.2020  
Accepted: 20.08.2021

#### DOI:

10.17341/gazimmfd.784653

#### Keywords:

Simulated annealing,  
capacitated vehicle routing  
problem,  
optimization,  
fuzzy c-means

#### ABSTRACT

In this study, a popular problem, the Vehicle Routing Problem, has been studied. Simulated annealing, a meta-heuristic method, was used to solve the problem. In general, the simulated annealing algorithm is an iterative process according to the variable temperature parameter, which mimics the annealing process of metals. The biggest problem with this method for our study is that it randomly generates the initial solution used to start the algorithm. For this reason, since the search space used to reach the optimum solution is large, the solution time (or number of iterations) will increase. With a better initial solution, it will take less time to reach the optimum solution. Since the optimum solution we want to reach is the minimum distance, the routes are clustered using fuzzy c mean to improve the initial solution. Due to fuzzy logic, the case that each data can be included in more than one cluster between 0-1 will approach the optimum solution since it will change the initial solution in each solution of the algorithm. By using the same data and the same parameters, the initial solution is improved with fuzzy c mean using a random initial solution, and the problem is solved with Simulated Annealing. Fuzzy c mean method reduced the initial search space by 57%. Therefore, Fuzzy c mean gave results closer to the optimum solution in the same solution time and the same iteration number.

\*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : ahmet.eker2@ogr.sakarya.edu.tr, ahmetyunustc@gmail.com

\*icil@sakarya.edu.tr / Tel: +90 532 450 7393

## 1. GİRİŞ (INTRODUCTION)

Endüstride lojistik, firmalar tarafından dikkate alınması gereken en önemli unsurlardan biridir. Birçok firma, müşteri ihtiyaçlarını en iyi şekilde karşılayabilmek için ürün ve hizmetlerinin tasarımına ve üretimine odaklansa da bu ürünler müşteriye ulaşmazsa, firma başarısız olur ve ayakta kalamaz. Bir firma, lojistiği etkin ve verimli bir şekilde yönetebiliyorsa firma maliyetlerini, enerjiyi ve zamanı, kârını etkileyecek şekilde optimize edebilir. Başka bir deyişle, firmalar bir dağıtım probleminde, ürünlerin nasıl dağıtılacağı ve lojistiğin nasıl düzgün bir şekilde organize edileceği ile ilgilenmelidirler, böylece yüksek bir kâr elde edilebilir.

Tedarik zinciri yönetiminin önemli bir parçası, tedarik zinciri içinde ürünlerin taşınması için lojistik operasyonların koordinasyonudur. Bir firmanın tedarik zincirindeki müşterilere hizmet vermek için teslimat rotalarını tasarlama görevi, literatürde Araç Rotalama Problemi (ARP) olarak bilinir. Uygulamada, müşteri talepleri müşteri lokasyonlarına varmadan önce kesin olarak bilinmeyebilir. Bu gibi durumlarda, araç kapasitesi gerçekte karşılaşılan talebi karşılayamayabilir. Bu nedenle müşterilere hizmet vermeye devam etmeden önce kapasiteyi yenilemek için merkezi bir depoya geri dönüş gerektirebilir. Bu potansiyel rota hatalarını etkili bir şekilde yöneten rotalar tasarlamak önemli bir husus ve zor bir iştir.

Gerçek hayatta talepler, araç kapasiteleri, teslim tarihleri, araçların ortalama hızları, tüketicilerin farklı yerlerde bulunması vb. gibi dağıtım sürecinde yerine getirilmesi gereken birçok engel vardır. Bu sorunlarla başa çıkmak için, firmaların tüm engelleri ve kısıtlamaları karşılayabilmeleri için araç rotalarını optimize etmeleri gerekmektedir. Bu araç rotası optimizasyonu, ARP olarak bilinir. ARP, iyi bilinen bir kombinatoriyal optimizasyon problemidir ve ilk olarak Dantzig ve Ramser tarafından tanıtılmıştır [1]. Acil durum hazırlığı ve afet yardımı, katı atık ve süt endüstrisi, sokak temizliği, engelli insanların taşınması, okul otobüsü rotalama vb. gibi birçok uygulama alanına sahiptir [2]. ARP; malları, tüketicilere en uygun rota ile teslim etmeyi ve depoya girip çıkmak için kullanılan araç sayısını en aza indirmeyi amaçlamaktadır.

1964'te Clarke ve Wright [3] ARP'yi genişleterek Kapasiteli Araç Rotalama Problemi (KARP)'ni elde ettiler ve bu problem için ilk buluşsal yöntemi önerdiler. KARP'de bir depo, bir dizi müşteri (n), bir dizi kapasiteli (k) araç (m) ve her müşterinin bir talebi (d<sub>i</sub>) vardır. KARP'deki görev, tüm müşterilere tam olarak bir kez hizmet verecek ve hiçbir aracın toplam kapasitesini (k) aşan bir müşteri grubunu ziyaret etmeyecek şekilde araç rotalarını oluşturmaktır [4]. Araştırma, araçların müşteriye ulaşana kadar taleplerin bilinmediği Stokastik Talepli Kapasiteli Araç Rotalama Problemi (KARPST) üzerine de odaklanmıştır.

ARP'yi çözmek için kesin yöntemler, hesaplama açısından zordur ve müşteri sayısı büyük olduğunda makul hesaplama zamanında en uygun rotayı bulması garanti edilmez. Bu

nedenle, ARP'leri çözmek için yaygın olarak kullanılan teknikler, sezgisel veya meta-sezgisel uygulamalara dayanan algoritmik yöntemlerin kullanımına odaklanır. ARP'ye uygulanan sezgisel prosedürlerin örnekleri arasında Parçacık Sürü Optimizasyonu [5], Yapay Arı Kolonisi Optimizasyonu, Karınca Kolonisi Optimizasyonu [6], Tabu Araması [7], Tavlama Benzetimi [8], Kısıtlı Programlama Algoritmaları ve Genetik Algoritmalar vardır.

## 2. KAYNAK ARAŞTIRMASI (LITERATURE SURVEY)

1959'daki ilk çalışmadan bu yana birçok yayın yapılmış ve ARP'nin kapsamı genişletilmiştir. Son on yılda, büyük problemleri çözmek için teknik çözüm konusunda önemli ilerlemeler kaydedilmiştir. İlgili gören bir diğer konu, teknolojik yeniliklerin ARP'ye dâhil edilmesidir. Bunlar arasında küresel konumlandırma sistemleri, radyo frekansı tanımlama ve yüksek kapasiteli bilgisayarların bilgi işlemesi alanları bulunmaktadır [9].

Dantzig ve Ramer tarafından formüle edilen ARP, gezgin satıcı probleminin bir uzantısıdır. ARP, bir depoda bulunan ve bir dizi müşteriye hizmet vermek için rotalanması gereken bir dizi araç olduğunu varsayar. ARP'nin amacı, belirli bir dizi kısıtlama altında araç maliyetini ve toplam rota maliyetini en aza indirmektir [10]. ARP'nin ortaya çıkışından bu yana birçok varyasyonu gerçek dünyadaki durumlara uygulanmıştır ve literatürde kapsamlı araştırmalar mevcuttur. Bu çalışmalar için Min vd. [11] aşağıdaki tabloyu oluşturmuştur.

**Tablo 1.** ARP'nin sınıflandırılması (Classification of VRP)

Sınıflandırma standardı	Problem türleri	
1 Malların yönü	Tek yönlü	Çift yönlü
2 Talep türü	Deterministik	Dinamik
3 Depo sayısı	Tekli	Çoklu
4 Araç sayısı ve Tipi	Tekli	Çoklu
5 Araç kapasitesi	Belirli	Belirsiz
6 Depo kapasitesi	Belirli	Belirsiz
7 Kademe sayısı	Tekli	Çoklu
8 Zaman pencereleri	Sert ve Yumuşak	Yok
9 Amaç fonksiyonu	Tekli	Çoklu

Balinski ve Quandt, toplam maliyeti, faaliyetler açısından tanımlamaktadırlar. Faaliyetlerdeki siparişlerin ağırlığını kullanarak toplam maliyeti hesaplamışlardır [12]. Ek kısıtlar ile birlikte problem en kolay kapasite kısıtlamalarından daha karmaşık kısıtlama kombinasyonlarına kadar çeşitli yönlerde genişletilebilir. Ghiani, Laporte, Musmanno, yaygın operasyonel kısıtların; araç sayısı, talep kapasitesi kısıtlaması, rota süresi, zaman aralıkları, müşteriye özel araç kısıtlaması ve müşteri önceliği ilişkileri olduğunu belirtmektedirler [13].

Klasik ARP modellerini çözmek için çeşitli kesin ve sezgisel algoritmalar geliştirilmiştir. Kesin algoritmaları Doğrudan Ağaç Arama Yöntemleri, Dinamik Programlama ve Tam Sayılı Doğrusal Programlama olarak üç ana kategoride sınıflandırabiliriz. Alt Sınır Ataması ve Dal Sınır

Algoritması, ARP ve Çoklu Gezgin Satıcı Problemi (MTSP) arasındaki ilişkiyi kullanır. Diğer bir problem ise her müşteriyi sadece bir kez ziyaret ederken depoda başlayan ve biten bir dizi minimum maliyet rotasının oluşturulmasıdır [14]. Simetrik ARP için sabit sayıda MTSP'nin k-derece merkez ağacının gevşemesine dayanan başka bir çözüm algoritması geliştirilmiştir. 10-25 arasında değişen müşteri sayısı başarılı bir şekilde çözülebilmektedir [15]. Eilon, Watson-Gandy ve Christofides, ARP'yi sabit sayıda araçla çözmek için dinamik bir programlama modeli önermektedir. Optimum değeri hesaplamak için minimum maliyet yineleme işlevini bulmuşlardır [16]. Christofides, daha sonraki bir çalışmada 50 müşteri için çözümü iyileştirilmiş bir sonuç elde etmiştir [17]. Balinski ve Quandt, ARP'ler için bir Bölümleyici Kümeleme ve Sütun Oluşturma Algoritması geliştirmişlerdir. Bu algoritmanın zorluğu hesaplama yollarının maliyetli olması ve çok sayıda ikili değişkene sahip olma riskidir. Uygulanabilir çözümler çok az olduğu için sorun sıkı bir şekilde kısıtlanmış problemlerde işe yaramaktadır [12].

ARP için sezgisel algoritmalar, TSP için geliştirilen algoritmalarından türetilmiştir [18]. Clarke ve Wright'ın algoritması, ARP'leri sınırsız sayıda araçla çözer [19]. Algoritma, müşterileri maksimum tasarrufu sağlamak için birleştirir ve iyileştirme mümkün olmayana kadar devam eder. Probleme, araç sabit maliyetleri ve filo boyutları eklemek gibi yöntemin farklı varyasyonları önerilmiştir [20]. Wren ve Holliday, bir veya birkaç depoya sahip ARP'ler için bir tarama algoritması ve kutupsal koordinatlarla temsil edilen Öklid düzleminde bulunan köşeler önermektedir. Algoritma, en küçük açığa sahip bir müşteriyi seçer ve mevcut kapasitesi varsa müşteriyi bir araca yerleştirir. Oluşturulan her rota için, rotalarda kat edilen toplam mesafeyi minimum düzeye indirmek için bir TSP çözülür [21]. Gendreau, Hertz ve Laporte, bir dizi çözüm ve iyileştirme adımı oluşturmak için bir Tabu Arama yöntemi geliştirdiler [12]. Algoritma tarafından oluşturulan rotalar uygulanabilir değilse, fizibilite sapmasını göstermek için amaç fonksiyonunda bir ceza eklenmektedir. Zhang ve ark. [23]; Karınca Kolonisi Optimizasyonu yönteminin geliştirilmesi ve geçerliliğinin test edilmesinde ARP'de maliyet ve zaman faktörlerini ele alarak bu yöntemde katkıda bulunmuştur.

Zaman Pencereci Araç Rotalama Problemi (ZPARP)'nde, her müşteri, en erken ( $e_i$ ) ve en geç ( $l_i$ ) kabul zamanlarını temsil ettiği bir seyahat maliyeti, seyahat süresi, talep ve zaman penceresi ile ilişkilendirilir. Zaman pencereleri, sorunun niteliğine bağlı olarak sert veya yumuşak kısıtlamalar olarak kabul edilebilir. ZPARP için kesin yöntemleri, Lagrange gevşemeye dayalı yöntemler, sütun oluşturma ve dinamik programlama olarak üç kategoride sınıflandırabiliriz [24].

Okul otobüsü rotalama ve çizelgeleme problemi ayrıca zaman pencerelerini de dikkate alır [25]. ARP için yaratıcı yöntemler ortaya koyan Ünsal vd. [26] ise kümeleme analizi ve yapay zekâ tekniklerini kullanmıştır. Bu iki yöntem okul

servisinin rotalanmasıyla ilgili iyi sonuçlar vermiştir. Swersey ve Ballard, sorunu rotalama ve zamanlama olarak incelemişlerdir [27]. Rotaların önceden oluşturulduğunu varsayarlar, bu nedenle sadece zamanlama problemi çözülür. Her rotaya tek bir otobüs atanır, öğrencileri alır ve belirli bir zaman aralığında okula varır. Problem, her otobüsün yalnızca bir okula tahsis edildiğini varsayar. Amaç kullanılan okul otobüsü sayısını en aza indirmektir. Swersey ve Ballard problemi çözerek iki tamsayı programlama modelini formüle ettiler. Formülasyonları ABD'deki bir bölgenin gerçek verilerine uyguladılar. Yöntem tatmin edici sonuçlar vermiş ve ihtiyaç duyulan otobüs sayısını %25 oranında azaltmıştır.

Sezgisel algoritmalar ARP'yi çözmek için yaygın olarak kullanılmaktadır. Baker ve Schaffer, Clarke ve Wright'ın ARP sezgisel tasarruf yönteminin bir uzantısı olan Sezgisel Rota Oluşturma yöntemini tasarlamışlardır [28]. Tasarrufları, mesafe ve zamanın bir kombinasyonu olarak tanımlamışlardır. Solomon, benzer bir buluşsal yöntem tanımlamıştır ancak zaman; tasarruf işlevine dâhil edilmemiştir [29]. Landeghem, müşteriler arasındaki zamanı ölçmek için bir sezgisel tasarruf yöntemi geliştirmiştir [30]. Bu buluşsal yöntemler zaman karmaşıklığına sahiptir. r-Opt Sezgiseli, r müşterilerini diğer r müşterileri ile değiştirir. 2-Opt veya 3-Opt, zaman aralığı ihlali riskine rağmen sorunu çözebilmektedir. Potvin ve Rousseau, bu tür problemleri çözmek için 2-Opt ve r-Opt algoritmalarını geliştirmiştir [31]. Meta-sezgisel yöntemler de ARP çözümünde kullanılır. Bu yöntemler arasında Tavlama Benzetimi, Tabu Araması, Karınca Kolonisi Optimizasyonu ve Genetik Algoritma yer alır.

González vd. [32] genetik algoritma tabanlı meta-sezgisel yeni bir çözüm yöntemi için algoritma geliştirmişlerdir ve bu yöntemi Solomon veri seti üzerinde test etmişlerdir. Puspita vd. [33] bir atık toplama problemi üzerinde karma tam sayılı doğrusal matematiksel model değişkenleri sayısını azaltarak yeni bir yöntem geliştirmişlerdir. Xia vd. [34] kesikli bölünmüş teslimatlı kapasite kısıtlı araç rotalama problemi üzerinde çalışmışlardır. Bu yöntemi çözmek için de tabu arama yöntemi kullanılmıştır. Novoa-Flores vd. [35] çoklu depo zaman pencereli araç rotalama problemi üzerinde çalışmışlardır. Bu problemi çözmek için klasik sezgisel iki aşamalı yöntem kullanmışlardır.

Son zamanlarda, bazı daha karmaşık ARP uzantıları tüm dünyadaki akademisyenlerin büyük ilgisini çekmektedir. Bu problemlerin bir örneği de [36] ve diğer birçok çalışmada ele alınan Open ARP'dir. Belirsiz ARP, birçok akademisyenin ilgisini çeken bir başka araştırma alanıdır. Liu ve Lai [37] tarafından hazırlanan çalışmalar, belirsiz talepleri dikkate alan çalışmalara örnektir.

Bayrak ve Özyörük [38], Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi ve Bölünmüş Talepli Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi için iki farklı model sunmuşlardır. Ayrıca, orta ve büyük problemlerin çözümü için öneride bulunmuşlardır. Atasagun ve Karaoğlan [39],

zaman bağımlı araç rotalama problemi ve topla-dağıt araç rotalama problemini birlikte ele alarak, Zaman Bağımlı Eşzamanlı Topla-Dağıt Araç Rotalama Problemi ilk kez tanımlamış ve çözümü için bir matematiksel model önermişlerdir. Küçükaydın [40], zaman pencereli araç rotalama problemini çözmek için yinelemeli yerel arama meta-sezgiseli ve değişken komşuluk arama meta-sezgiselinden yararlanarak iki sütun türetme temelli mat-sezgisel geliştirmiştir. Aydoğdu ve Özyörük [41], Dinamik Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi ele almış ve problemin çözümü için Rassal İteratif Yerel Arama Değişken Komşu İniş algoritması adında yeni bir algoritma geliştirmişlerdir. Geliştirilen sezgisel algoritmanın matematiksel model ile elde edilen sonuçlara yakın sonuçlar verdiği görülmüştür. Yalın lojistik yönetimi ve milkrun yöntemleri taşıma ve dağıtım maliyetlerini azaltmada kullanılan diğer yaklaşımlardır [42-44].

Müşterileri ARP çözümede gruplamak literatürde yeni değildir, Yoshiike ve Takefuji [45], birinci aşamanın, müşterileri gruplandırma için maksimum nöron modelini kullanarak birleştirildiği ve ikinci aşamanın, elastik ağ modelini kullanarak her müşteri grubu için bir TSP çözüme ile çalışan iki aşamalı bir algoritma önermişlerdir. Konum Rotalama Problemi (LRP)'nin ARP'nin bir uzantısı olduğunu bilerek, Barreto vd. [46], müşterileri ARP ve türevlerinde gruplara ayıran başka bir çalışma yapmışlardır.

### 3. KAPASİTELİ ARAÇ ROTALAMA PROBLEMİ (CAPACITATED VEHICLE ROUTING PROBLEM)

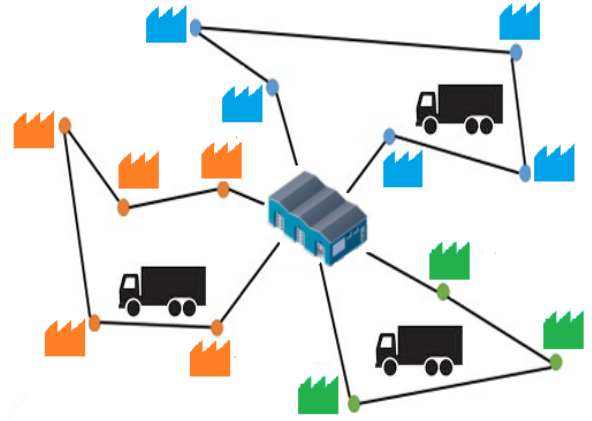
ARP, dağıtım yönetiminin kalbini ifade eder. İlk olarak 1959'da Dantzig ve Ramsey tarafından duyurulmuştur. O günlerdeki asıl problem, şirketlerin ulaşmak zorunda oldukları durumların, literatürden çıkmış gerçek sorunlardan daha karmaşık olması ve zamanla değişmeseydi. Bu sorunların çözülebilmesi için, karmaşık ve pratik araç rotalama problemleri çözecek entegre bir modelleme ve optimizasyon tekniği sunmaktı. ARP, hizmetlerin ve malların müşteriye sunumundaki temel sorundur. Uygulamada ARP, taşınan malların türüne, hizmet kalitesi seviyesine ve müşterilerin ve araçların özelliklerine göre değişir. Bu nedenle ana kısıtlardan bazıları, farklı depo lokasyonlarına sahip araçlar, aynı araç tipleri ile çatışan müşteriler, müşterilere belirli zaman aralığında hizmet verebilmek, farklı günlerde teslimat süreleri ve karışık araç rotalarıdır. Çok sayıda kısıt olmasına rağmen, ARP'de minimum maliyetle bu rotayı elde etmeye odaklanılmıştır.

Algoritmalar ve yeni yazılım türleri hakkında yapılan araştırmalar, esnek optimizasyonları sistemlerin çeşitli pratik bağlamlara uyum sağlamak için değişmesine yardımcı olan sınırlı sayıda prototip problemine odaklanır [1]. ARP, fiziksel dağıtım ve lojistik kısmında temel bir özelliğe sahiptir ve bir grup müşteriye hizmet verilen deponun başlangıç ve bitiş noktası olan daha az maliyetli teslimat rotasını bulmak için hedef olarak belirlenir. Daha sonra, bir grup araç tarafından müşterilere veya tedarikçilere hizmet etmek için kullanılan optimal bir rota setinin belirlenmesini

gerektiren verilerin zor bir kombinasyonu oluşturulur. Bu çözümün zor kısmı, her müşteriye hizmet verebilmek için sadece bir aracın kullanılması gerektiğidir.

ARP'de firmaların sahip olduğu müşteri sayısı, müşteri lokasyonu, talep dağılımı ve her aracın belli bir kapasite ile tamamladığı ortalama rota sayısı gibi ARP'yi tanımlayan kısıtlar vardır. Bu kısıtların probleme göre farklılık göstermesi ile otomatik olarak farklı örnek türlerine ulaşabiliriz.

Standart bir ARP, aynı zamanda kapasiteli araç rotalama problemi (KARP) olarak da adlandırılan yükleme kapasitesi sınırına sahip bir araç rotalama problemidir. Her araç maksimum bir kapasiteye sahiptir. Aşağıdaki Şekil 1'de gösterildiği gibi en temel araç rotalama problemidir. Orta depo; yerel depo olabilir ve her aracın gerçek hayattaki uygulamalarda olduğu gibi kapasitesi vardır.



Şekil 1. Araç rotalama örnek şeması (Vehicle routing example diagram)

Diğer araç rotalama problemleri KARP'ye dayanmaktadır. Araç akış formülasyonları, bir rotanın optimal çözüme dâhil edilip edilmediğini belirtmek için tamsayı değişkenleri kullanan formülasyonların daha yaygın olanıdır. Standart bir araç rotalama problemi için matematiksel model aşağıda verilmiştir. Parametreler, toplam araç sayısı ( $M$ ), toplam müşteri sayısı ( $N$ ),  $i$ . ve  $j$ . noktaları arasındaki mesafe ( $d_{ij}$ ),  $i$ . müşterinin talep miktarı ( $q_i$ ), araç kapasitesi ( $C$ ), ( $y_i$ , alt turları engellemek için kullanılan rastgele deęişkendir). Karar deęişkeni  $x_{ijk}$  aşağıdaki şekilde tanımlanır:

$$x_{ijk} = \begin{cases} 1, & k \text{ nolu araç } i \text{ noktasından } j \text{ noktasına giderse} \\ 0, & \text{diğer durumlarda} \end{cases}$$

Amaç Fonksiyonu:

$$\min z = \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^M d_{ij} x_{ijk} \quad i \neq j \quad (1)$$

Kısıtlar:

$$\sum_{k=1}^M \sum_{j=1}^N x_{0jk} = M \quad (2)$$

$$\sum_{k=1}^M \sum_{j=0}^N x_{ijk} = 1 \quad i \neq j \quad i = 1, \dots, N \quad (3)$$

$$\sum_{k=1}^M \sum_{i=0}^N x_{ijk} = 1 \quad i \neq j \quad i = 1, \dots, N \quad (4)$$

$$\sum_{i=1}^N x_{i0k} \leq 1 \quad k = 1, \dots, M \quad (5)$$

$$\sum_{i=1}^N q_i \sum_{j=0}^N x_{ijk} \leq C \quad i \neq j \quad k = 1, \dots, M \quad (6)$$

$$y_i - y_j + N \sum_{k=1}^M x_{ijk} \leq N - 1 \quad i \neq j \quad i, j = 1, \dots, M \quad (7)$$

Amaç fonksiyonu (Eş. 1) toplam mesafenin minimize edilmesini ifade etmektedir. (Eş. 2), başlangıç noktasından harekete geçecek toplam araç sayısının M olduğunu, (Eş. 3) ve (Eş. 4) kısıtlar her bir müşterinin sadece bir araç tarafından ziyaret edilmesini ve depodan müşteriye gelen veya müşteriden depoya gelen rotalardan yalnızca bir tanesinin kullanılması gerektiğini ifade etmektedir. (Eş. 5), bir aracın yalnızca bir rotalamada kullanılacağını ifade eder. (Eş. 6) ise araç kapasitesinin (C) aşılamayacağını ifade eder. (Eş. 7)'de verilen kısıt ile alt turların oluşması engellenir.

### 3.1. Tavlama Benzetimi (Simulated Annealing)

Yerel bir arama yönteminin aksine, TB rastgele aramalar yapan bir yöntemdir. Optimizasyonda benzetilmiş tavlama kavramını anlamak için ilk olarak Kirkpatrick ve ark. tarafından sunulan fiziksel tavlama sisteminin benzetmesine bakmak gerekir [47]. Bir katının, örneğin çeliğin temel durumu, atomlarının veya parçacıklarının minimum enerji konfigürasyonunda düzenlendiği durumdur (katının en kararlı hâli). Bir metalin temel durumu, fiziksel tavlama işlemi ile elde edilebilir. Metal, katıdan sıvıya dönüşümünü sağlamak için önce yüksek bir sıcaklığa ısıtılır. Bu sıcaklığa metalin erime noktası denir. Sıvı halde metal kararsızdır, parçacıklar herhangi bir set konfigürasyonunda düzenlenmediklerinden yüksek enerji sergileyerek serbestçe hareket ederler. Daha sonra parçacıkların minimum enerji düzenlenmesine yavaş yavaş yerleşmesine izin vermek için sıcaklık dikkatlice düşürülür ve temel durum elde edilir.

Benzer şekilde TB, katının temel durumuna karşılık gelen bir optimizasyon probleminin amaç fonksiyonunun minimum değerini elde etmeyi amaçlamaktadır [48]. Katı maddenin herhangi bir başka hâli, optimizasyon problemi için uygun bir çözüme karşılık gelir ve katının bir hâlinin enerjisi, bir çözümün amaç fonksiyon değerine eşdeğerdir. Fiziksel sistemin sıcaklığına benzer bir kontrol parametresi  $q$ , amaç fonksiyon değerini bozan hareketlerin kabulünü düzenleyerek TB algoritmasının kademeli olarak global optimuma yakınsamasını kontrol etmek için kullanılır.

TB meta-sezgisel, bir  $x_0$  başlangıç çözümü ve bir durdurma kriteri gerektirir. Alfa vd. [49], iyi çözümler bulmak için gereken hesaplama süresinin başlangıç çözümlerinin kalitesine duyarlı olduğunu belirtmektedir. TB algoritmasının başlatılması, hem yineleme sayımının hem de sıcaklık kontrol sayısının sıfıra ayarlanmasını, sıcaklığın

başlangıç sıcaklığına ayarlanmasını ve başlangıç çözümünün yerleşik  $x$  olarak atanmasını gerektirir. Komşuluk, herhangi bir permütasyon yoluyla kurulabilir.

Yineleme sayım sınırı  $t^{\max}$ 'a ulaşırsa veya mevcut çözüm  $x_t$  için komşu hareket kümesi  $M^{x_t}$ 'de  $\Delta x$  daha fazla uygulanabilir hareket yoksa algoritma sona erer. Aksi takdirde hareket kabul için test edilir. Hareket, amaç fonksiyon değerini iyileştiriyorsa 1 olasılıkla kabul edilir. Hareket kötüleşiyorsa yine de olasılıkla kabul edilebilir.

$$p = e^{\left(\frac{c(x) - c(x')}{q}\right)} \quad (8)$$

Bir katının fiziksel tavlama ile TB arasındaki analogiye dönersek, TB algoritması için kabul kriteri Metropolis kriterinden çıkarılır. Metropolis ve ark. tarafından sunulan Metropolis algoritması [50] bir katının fiziksel tavlama simüle etmek için basit bir algoritmadır.  $E_i$  enerjili katının mevcut hâli  $i$  verildiğinde,  $E_j$  enerjisine sahip bir  $j$  durumunun, katının atomlarının küçük bir yer değiştirmesi yoluyla üretildiğini belirtir. Ortaya çıkan enerji farkı,  $E_j - E_i$ , sıfırdan küçük veya sıfıra eşitse,  $j$  yeni mevcut durum olarak kabul edilir. Bununla birlikte, enerji farkının sıfırdan büyük olması gerekiyorsa,  $j$  durumu yalnızca olasılıkla kabul edilecektir.

$$p = e^{\left(\frac{E_i - E_j}{k_B T}\right)} \quad (9)$$

$T$ , katının mevcut mutlak sıcaklığıdır ve  $k_B$ , fiziksel Boltzmann sabiti olarak bilinir. Kirkpatrick ve ark. [47], sıcaklığın yalnızca bir kontrol parametresi olduğu için Boltzmann sabitinin ihmal edilebileceğini belirtirler. Kontrol parametresi, algoritmanın ilk aşamalarında neredeyse tüm bozulan hareketlere izin verecek şekilde formüle edilmiştir. Kontrol parametresi kademeli olarak azaldığından, kötüleşen hareketleri kabul etme olasılığı da azalır ve algoritma global optimuma yakınsar.

Robuste vd. [51] TB uygulamasında, çözümün kalitesi açısından büyük problemler için algoritmadan daha iyi performans gösterebileceğini belirtmektedir. TB'nin gelişimi, Van Breedam'ın [52] TB'nin varyantlarını gözden geçirmesi ve karşılaştırmasıyla devam etmiştir. Tan vd. [53] TB meta-sezgisinin bir dizi avantajı olduğunu ifade eder:

- Keyfi sistemler ve maliyet fonksiyonları ile ilgilenir.
- İstatistiksel olarak en uygun çözümü garanti eder (yeterli işlem süresi sağlandığında).
- Karmaşık problemler için bile nispeten kolay kodlama.
- Genellikle makul işlem süresi içinde iyi bir çözüm sunar.

Van Breedam [52] tarafından TS ve TB arasındaki çözüm kalitesindeki farkın değerlendirilmesinde hiçbir zaman %4'ü aşmadığını belirtmiştir. Üç meta-sezgiselin karşılaştırmalı analizinde, Tan vd. [53], TB'nin hesaplama çabası ve çözüm kalitesi arasında benzerlik olduğu sonucuna varmıştır.



### 1.1. Bulanık C-ortalama (Fuzzy C means)

FCM algoritması, yaygın olarak kullanılan bulanık kümeleme algoritmasıdır [54, 55]. İlk geliştirme Dunn [56] tarafından yapılmış ve Bezdek [54] tarafından ikinci kez geliştirilmiştir. Sert kümeleme ile bulanık kümeleme arasındaki en önemli fark, bulanık kümelemenin her nesneyi kalıcı olarak bir kümeye atamamasıdır. Bulanık kümeleme, nesnelerin kümelere nasıl ait olduğunu ifade etmek için bir "derece" kavramı kullanır. Her nesne için, derecelerin toplamı (bulanık katsayılar olarak da adlandırılır) 1'dir [54, 56].

FCM'nin amaç işlevi Üyelik Fonksiyonu olarak adlandırılmıştır. FCM'nin "iyi" bir kümeleme çözümünün hedef işlevi en aza indirmesi beklenir. Üyelik Fonksiyonu (Eş. 10), (Eş. 11) ve (Eş. 12) denklemleri olarak tanımlanır [54, 56]:

$$F_m = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^m \|x_i - C_j\|^2 \quad (1 \leq m < \infty) \quad (10)$$

$$u_{ij} = \frac{1}{\sum_{q=1}^k \left( \frac{\|x_i - C_j\|}{\|x_i - C_q\|} \right)^{\frac{2}{m-1}}} \quad (11)$$

$$C_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m} \quad (12)$$

Burada n, belirli bir veri kümesindeki nesnelerin sayısıdır. k, küme sayısıdır, m, 1'den büyük veya 1'e eşit herhangi bir gerçek sayı olabilir.  $u_{ij}$ , j kümesine ait olan  $x_i$  nesnesinin derecesini gösteren bulanık katsayıdır.  $C_j$ , j kümesinin merkezidir.  $\|x_i - C_j\|$ ,  $x_i$  nesnesi ile  $C_j$  küme merkezi arasındaki mesafeyi belirtir.

FCM'nin temel adımları şu şekilde özetlenebilir [54, 56]:

- Küme sayısının tahmini sayısına göre her nesneye bulanık katsayılar atanır.
- Bulanık katsayılar göre küme merkezlerini hesaplanır.
- Yeni küme merkezlerine dayalı olarak bulanık katsayıları yeniden hesaplanır.
- Bulanık katsayıların varyansını değerlendirilir (önceki bulanık katsayılardan biriyle karşılaştırarak).
- Varyansı önceden tanımlanmış bir duyarlılık eşiğiyle karşılaştırılır.
- Bulanık katsayıların varyansı duyarlılık eşiğinden az olana kadar 2, 3, 4 ve 5. adımları tekrarlanır.

FCM'nin ana avantajları aşağıdaki gibidir [55]:

- FCM, belirsiz verilerle başa çıkmak için sağlam ve esneklerdir.

FCM'nin temel sınırlamaları ve sorunları aşağıdaki gibidir [55]:

- FCM, kenar ve kabuk gibi şekillerle verileri analiz edemez.

- Farklı mesafe metrikleri kullanmak, FCM sonuçlarını saptrabilir.
- Verilerdeki gürültü ve aykırı değerlerle başa çıkamaz.
- FCM, küme sayısı hakkında önceden bilgi gerektirir.
- FCM'nin sonuçları yerel minimumlar içinde sıkışmış olabilir.

### 3.3. Önerilen Yönteminin Önemi ve Literatüre Katkısı (Importance of the Proposed Method and Contribution to the Literature)

Bu çalışmada, TB algoritmasında başlangıç çözümü Bulanık c-ortalama ile iyileştirilen bir yaklaşım önerilmektedir. TB, optimizasyon problemlerini çözmek için iyi bilinen bir stokastik yerel arama algoritmasıdır. TB, araç yönlendirme problemlerini çözmeye en başarılı yöntemlerden biri olarak kabul edilir. Literatürdeki çalışmalarda, kapasite kısıtlı araç rotalama çalışmasında TB algoritmasının diğer tüm algoritmalarından daha iyi performans gösterdiği ve birçok örnek için optimum çözümleri sunduğu bilinir [57, 58]. Öte yandan bulanık c-ortalama kümeleme yöntemi ise literatürde bulunan en başarılı kümeleme tekniklerinden biri olarak kabul edilir. Veriler aracılığı ile örtüşen kümeler bulmakta kullanılan bu metod, esnek bir yapıda ve diğer kümeleme algoritmalarına göre daha iyi sınıflandırma yeteneğine sahiptir.

TB'nin klasik yerel arama yöntemlerine göre en büyük avantajı, bir uygun çözüm ararken yerel minimumda sıkışıp kalmamasıdır. TB güçlü ve sağlam bir teknik olarak doğrusal olmayan modeller, kaotik ve gürültülü veriler ve birçok kısıtla başedebilme gücüne sahiptir. Pratik bir bakış açısıyla, bu yaklaşım genellikle zor optimizasyon problemlerine mükemmel çözümler üretir. Esnekliği ve optimallğe yaklaşma yeteneği diğer yerel arama yöntemlerine göre başlıca avantajlarıdır. Belirli bir algoritmayı birden fazla problemde kullanılmak üzere ayarlama becerisi bir algoritmanın önemli bir özelliği olarak kabul edildiğinden, TB kolayca ayarlanabilir bir özelliğe sahiptir. Ancak farklı kısıtlamaları hesaba katmak ve algoritmanın parametrelerinde ince ayar yapmak için gereken ayarlama çalışması oldukça hassas olabilir. TB'nin uygulamasında kullanılan parametrelerin kesinliği, sonucun kalitesi üzerinde önemli bir etkiye sahip olabilmektedir.

Zayıf yönleri ise TB bir meta-sezgisel olduğundan, çözümlerin kalitesi ile hesaplamak için gereken süre arasında bir ödünleşme yapmak gerekmektedir. Olasılıklı bir arama yöntemi olan TB algoritması, simülasyon tabanlı ve yinelemeli bir stratejiye dayanır. Yinelemelerinin her birinde tamamen rastgele kurallar uyguladığından, stokastik hafızasız bir yaklaşım olarak kabul edilir. Özellikle karmaşık problemlerde yavaş yakınsama, onun zayıf yönü ve dezavantaj olarak kabul edilir [59]. Bu nedenle, yerel bir optimumdan kaçmak genellikle uzun zaman alır. Bu durum hesaplama süresinin uzamasına neden olur ve performansına olumsuz etki yapar.

Bu çalışmanın amacı, standart TB algoritmasının performansının artırılmasına yönelik olarak geliştirilen

çözümlerin etkinliğini ve doğruluğunu geliştirmekle birlikte çözümlerin hazırlanma hızını artırmaktır. Yöntemin etki alanını ve daha iyi sonuçlar elde edilmesine yönelik katkı sağlanmaktadır. Aşağıda bu kapsamda son zamanlarda literatürde yapılan çalışmaların bir değerlendirilmesi yapılmaktadır. Literatürde görüldüğü üzere bu çalışmada önerilen şekilde bir çalışma yapılmamıştır. TB algoritmasının belirtilen dezavantajlarının önüne geçerek, bu çalışmada geliştirilen başlangıç çözümü iyileştirme sayesinde, TB algoritmasında hesaplama süresinin kısaltılmasında önemli bir katkı sunarak hem çözüm kalitesi hem de hesaplama gerekliliği açısından iyi bir performans artışı sağlanmaktadır. Çalışma, başlangıç çözümü iyileştirme sonucunda rassal başlangıç çözümüne göre optimum çözüme daha yakın bir durum elde edilmektedir. Dolayısıyla daha az iterasyon veya hesaplama yapılarak, daha kısa sürede optimum çözüme ulaşılabilmektedir. Önerilen yöntem, KARP için son derece rekabetçidir. TB algoritmasının en büyük sorunu, algoritmayı başlatmak için kullanılan başlangıç çözümünü rassal olarak oluşturduğundan dolayı optimum çözüme ulaşmak için kullanılan arama uzayı büyük olduğu zaman, çözüm süresi büyük oranda artar. Bu çalışma ile daha iyi bir başlangıç çözümüyle optimum çözüme ulaşma daha kısa zamanda gerçekleşir. Ulaşmak istediğimiz optimum çözüm minimum mesafe olduğundan, başlangıç çözümünü iyileştirmek için Bulanık c-ortalama yöntemi kullanılarak rotalar kümelenir. Bulanık mantık gereği, her verinin 0-1 arasında birden fazla kümeye dahil olabilmesi sayesinde, algoritmanın her çözümünde başlangıç çözümünü değiştireceğinden dolayı optimum çözüme yaklaşma durumu daha hızlı olmaktadır.

Aşağıda literatürde son zamanlarda literatürde konu ile ilgili çalışmaların bir değerlendirilmesi yapılmaktadır. Literatürde TB algoritmasının KARP üzerindeki başarımını arttırmak için genellikle diğer yöntemler ile birleştirilmesi veya diğer sezgisellerden operatörler alarak hibrit yapılar oluşturulması yoluna başvurulmaktadır. Bu yönde önerilen bir dizi algoritma, TB'ni diğer optimizasyon yöntemleriyle birleştirmenin avantajlarını kullanır. Osman [60], KARP probleminin çözümünde başlangıç çözümü geliştirmek için tasarruf algoritmasının kullanarak tabu arama ve tavlama benzetiminin birleştiren hibrit bir algoritma önerdi. Lin vd. [61], aynı şekilde TB ve tabu arama yöntemlerinin avantajlarını kullanan bir yaklaşım önerdi, burada önerilen algoritmayı doğrulamak değişik boyutlarda çeşitli karşılaştırma örnekleri kullanılmıştır. Sigauke ve Talukder [62], heterojen bir araç filosu kullanarak KARP'yi optimize etmede Osman'ın hibrit yönteminin değiştirilmiş bir versiyonunu önerdi. Chiang ve Russell [63], problemi çözmeye tabu aramanın kısa süreli bellek işlevi üstlendiği zaman pencereli kısıtlı bir TB uygulaması gerçekleştirdi. Tavakkoli-Moghaddam vd. [64], maliyetleri minimize etmek ve kapasite kullanımını maksimize etmek için bağımsız bir rota uzunluğuyla ve en yakın komşu ilkesine dayalı hibrit bir TB yaklaşımı önerdi. Yu ve Lin [65], ARP'nin bir varyantı olan açık konum rotalama problemini çözmek için TB tabanlı sezgisel bir yöntem önerdiler. Vincent ve ark. [66], TB algoritmasının basit ama güçlü bir

versiyonunu geliştirdi. Burada araştırmacılar problemin çözümü için çevreci bir bakış açısıyla Boltzmann ve Cauchy fonksiyonlarına dayalı yeniden başlatma stratejisiyle birlikte TB'nin hibrit bir yaklaşımını kullandılar. Rabbouch vd. [67] KARP'yi çözmek için arama işlemi sırasında geçmiş verileri hesaba katan ampirik tip tavlama benzerimi adını verdikleri bir TB algoritması önerdiler. Büyük ölçekli bir optimizasyon problemlerinde TB'nin etkinliğini artırmak için El Afia vd. [68] karmaşık sayıları kullanan bir sayı sistemi olan kuaterniyonu kullanarak yöntemi iyileştirmeye çalıştılar ve böylece orijinal arama alanını genişleterek yerel optima sıkışmaktan kurtarmaya çalıştılar. Burada kuaterniyon gösterimi, TB algoritmasının uygunluk ortamını yumuşatmaktadır. Leung vd. [69], iki boyutlu yükleme kısıtlı bir KARP'ni çözmek için bir TB yaklaşımı geliştirdi; burada problemin yükleme bileşeni ayrı bir paketleme buluşsal ile çözülmektedir. Wei vd. [57] İki boyutlu yükleme kısıtlamaları olan kapasiteli araç yönlendirme problemi için bir tavlama benzetim algoritması önerdi. Burada tekrar tekrar soğuma ve sıcaklık artırma temeline dayalı iki aşamalı bir yöntem önerdi. Afifi vd [25], ARP'yi zaman pencereli ve senkronizasyon kısıtları ile iyileştirmek için farklı yerel arama tekniklerini içeren bir TB algoritması önerdi. Sajid vd. [70], toplam mesafeyi optimize etmede optimal çözümleri aramak için genetik algoritma ve TB'ni birleştiren bir yöntem önerdi. Burada önerilen yaklaşım en yakın komşu geçiş operatörünü kullanmaktadır. Mar'i vd. [71] minimum mesafeyi bulmak için rota modelinde parçacık sürü optimizasyonu ve TB'den oluşan hibrit bir yaklaşımını önerdi. Kayman [72] parçacık sürü optimizasyonu algoritmasının performansını bulanık c-ortalama yöntemini ile iyileştiren bir çalışma yaptı. Fisher ve ark, iki aşamalı bir yöntemi önerdiler [73]. Bu yaklaşımın ilk aşaması, müşterileri birkaç kümeye ayıran kümeleme yöntemi ve ikinci aşaması, her rota için minimum maliyeti elde etmek amacıyla her aracın rotasını oluşturmaktadır. Beasley ise bu yaklaşımın tam tersi yönde, önce rota oluşturdu ve ardından kümeleme yaptı [74]. Bu kapsamda Alfa vd. [49] ilk önce rotayı tasarlayan, ardından ilk çözümü oluşturmak için kümeleme yöntemini kullanan, ardından da TB'da aramayı gerçekleştiren 3-opt yöntemini kullandıkları bir yaklaşım önerdiler. Mohammed vd. [76] K-En Yakın Komşu Algoritmasını kullanarak ARP için en uygun rotayı bulmaya yönelik toplam mesafeyi ve zamanı azaltarak, müşterilerin hedeflerine hızlı ulaşımını sağlayan bir model önerdi. Bozyer vd. [76] önce bulanık c-ortalama ile kümeleme yapıldıktan sonrada tabu arama sezgiseli ile çözüm arandığı iki aşamalı bir yaklaşım önerdi. Ewbank vd. [77] bulanık - c kümeleme tekniği ve genetik algoritmasının bir uzantısı olan memetik bir algoritma kullanarak ARP'ye çözüm önerisi geliştirmişlerdir. Memetik bir algoritma, küme sayısını belirler ve gelişmiş bir bulanık c - ortalama algoritma, müşterileri rotalara tahsis eder. Görüldüğü üzere literatürde bulunan yöntemler rota yapılandırma yöntemleri, rota geliştirme yöntemleri ve iki aşamalı yöntemler şeklinde sınıflandırılmaktadır [78]. Rota yapılandırma yöntemleri çözümü yapılandırmada problem verisini adım adım kullanırlar. Rota geliştirme yöntemleri yada yerel arama geliştirme yöntemleri yinelemeli arama prosedürleridir.



Uygun bir başlangıç çözümünden süreci başlatarak çözüm kalitesini yerel modifikasyonlar ile artırır. İki aşamalı sezgisel yöntemler ya önce kümeleme yaparak sonra rotalamanın yapıldığı biçimde ya da ilk önce rotalama yapılarak daha sonra kümeleme yapma biçiminde gerçekleştirilmektedirler.

Literatür araştırmasından anlaşıldığı üzere bu çalışmada gerçekleştirilen şekilde TB algoritmasının performansını artırıcı tarzda bir çalışma yapılmamıştır. Bu yönüyle önerilen yöntem literatüre yeni bir katkı sağlamaktadır. Başlangıç çözümü TB algoritmasının performansında önemli bir rol oynamaktadır. Kümelenebilirliğe dayalı başlangıç çözümünün kullanılmasının KARP'ni çözmede TB'nin performansını nasıl etkilediğini görmek açısından önemli bir katkı sağlamaktadır. Yöntem, hem çözüm kalitesi hem de hesaplama gerekliliği açısından iyi bir performans gösterir. Hesaplama sonuçları, kümeleme tabanlı bir başlangıç çözümünün kullanılmasının, literatürdeki önceki algoritmalara kıyasla daha iyi çözümlere ulaşmada etkili olduğunu göstermektedir.

Önerilen model küçük çaplı problemlerden kapsamlı ve zor problemlere kadar çok çeşitli zorluk seviyelerinde kullanılabilir. Önerilen model ile beraber problem çözüldüğünde sonuçlar ümit vericidir. Bu model ile birlikte daha kısa sürelerde optimum çözüme daha yakın sonuçlar elde edilmiştir. Bu durum, daha karmaşık problemlerde daha iyi sonuçlar vereceğinin kanıtıdır. Bu da önerilen modelin ölçeklenebilirliğinin yüksek olduğunu gösterir.

#### 4. DENEYSEL SONUÇLAR VE TARTIŞMALAR (COMPUTATIONAL RESULTS AND DISCUSSIONS)

Bu çalışma, tüm sistem kısıtlarını göz önünde bulundurarak müşteri taleplerini karşılamak için araç filosu için en uygun rotayı bulmayı amaçlamaktadır. Kısıtlar, araç kapasitesi ve müşteri talebidir. Araç filosu bir depodan başlar ve araç

kapasitesi kısıtlarını ihlal etmeden yolundaki maksimum teslimatları karşılamaya çalışır ve ardından depoya geri döner. Tüm müşteriler sadece bir kez ziyaret edilir ve bu araçlar teslimattan sonra tekrar depoya geri döner. Bu modelin çözülmesinde MATLAB R2017A Programının Fuzzy seti kullanılmıştır. Bulanık c-ortalamalar yönteminin uygulanması için gerekli olan şehirlerin enlem ve boylam koordinatları Google Haritalar üzerinden belirlenerek Tablo 3'de listelenmiştir. TB algoritmasının çalışabilmesi ve başlangıç çözümünü optimum çözüme ulaştırılabilmesi için gereken program, modüller hâlinde el ile yazılmıştır.

Aşağıdaki tablolarda rassal başlangıç çözümü ve bulanık c-ortalamalar ile kümelenebilir başlangıç çözümü için sırasıyla 12, 13 ve 14 kümelili olacak şekilde problem çözülmüştür. Her yöntem ve küme sayısı için 15 adet deney yapılmıştır. Burada yalnızca 13 kümelili Deney sonuçları aşağıdaki tablolarda gösterilmiştir (Tablo 2 ve Tablo 4). En iyi çözümün amaç fonksiyonu değişimi ise Şekil 6 da görülmektedir. Problem 13 kümelili rassal başlangıç ile çözüldüğünde en iyi sonucu 10 numaralı deney vermiştir. Ortalama başlangıç çözümü 49933,08 kilometre, ortalama en iyi çözüm 15950,91 kilometre, ortalama minimum mesafe 483,73 kilometre, ortalama maksimum mesafe 2336,16 kilometre, ortalama toplam mesafe 15898,36 kilometre, ortalama çözüm süresi 34,65 saniye olarak bulunmuştur. 15 çözümden 11 tanesi uygulanabilir çözümdür. Şekil 2 başlangıç çözümünü; Şekil 3 nihai çözümü göstermektedir.

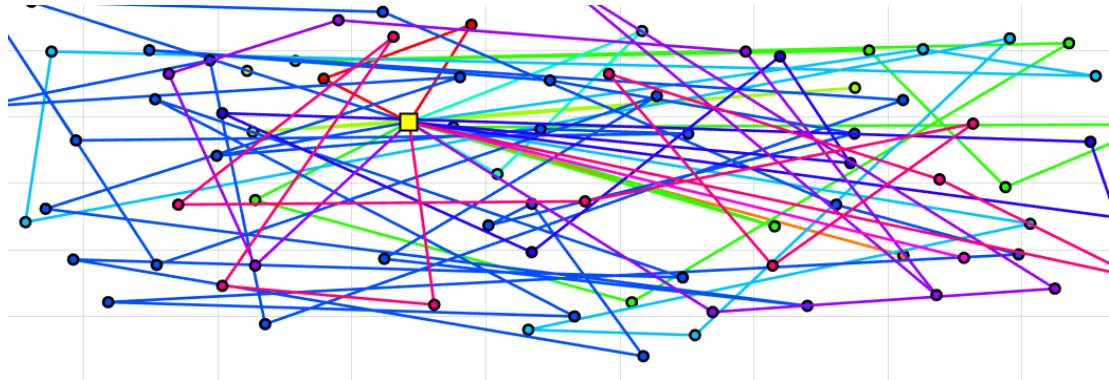
Problem 13 kümelili iyileştirilmiş başlangıç ile çözüldüğünde en iyi sonucu 5 numaralı deney vermiştir. Ortalama başlangıç çözümü 21203,97 kilometre, ortalama en iyi çözüm 15549,88 kilometre, ortalama minimum mesafe 452,74 kilometre, ortalama maksimum mesafe 2275,5 kilometre, ortalama toplam mesafe 15539,69 kilometre, ortalama çözüm süresi 34,58 saniye olarak bulunmuştur. 15 çözümden 14 tanesi uygulanabilir çözümdür. Şekil 4 başlangıç çözümünü; Şekil 5 nihai çözümü göstermektedir.

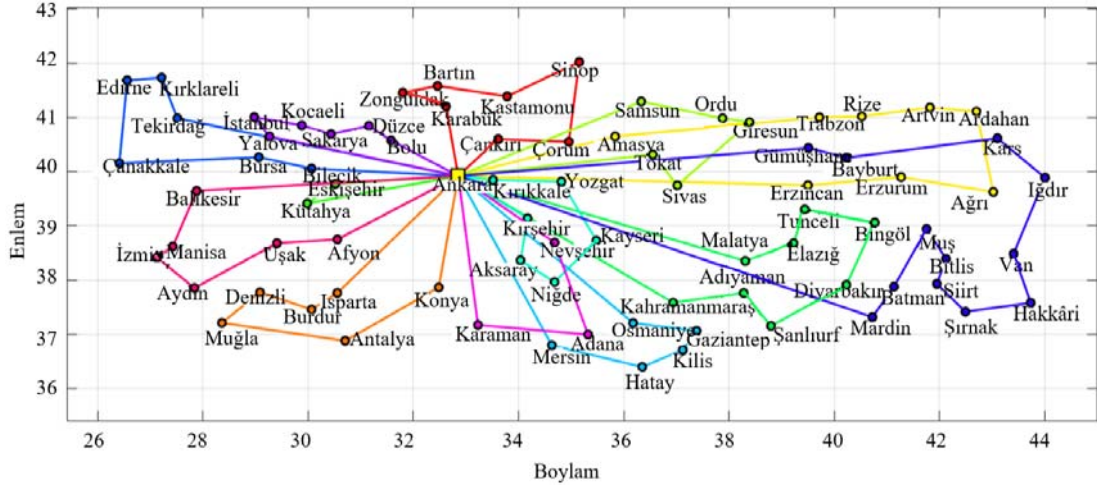
**Tablo 2.** 13 kümelili ve Standart TB ile elde edilen sonuçları (Results with 13 cluster and Standard SA)

Deney No	Başlangıç Çözümü	En İyi Çözüm	Küme Sayısı	Minimum Mesafe	Maksimum Mesafe	Toplam Mesafe	Toplam Fazla Kapasite	Uygulanabilir Çözüm	Çözüm Süresi
1	49317,17	15833,97	13	428,7202	2378,7921	15833,97	0	1	34,906
2	52090,7	15899,82	13	417,556	2400,3257	15742,39	1	0	34,45
3	50413,92	15840,94	13	417,556	2308,6088	15840,94	0	1	34,723
4	50575,31	16019,47	13	474,9832	2149,9403	16019,47	0	1	34,669
5	48613,26	15892,07	13	505,64	2423,1411	15892,07	0	1	34,771
6	49264,25	15965,33	13	428,7202	2265,4637	15965,33	0	1	34,73
7	51860,19	15947,15	13	487,4924	2363,618	15789,26	1	0	34,47
8	47109,15	15834,81	13	539,7087	2356,8804	15834,81	0	1	34,539
9	53526,07	15732,05	13	450,1598	2369,7732	15732,05	0	1	34,725
10	48217,26	15800,67	13	512,8669	2511,1849	15800,67	0	1	34,425
11	51913,68	16125,1	13	399,7	2532,5885	16125,1	0	1	34,751
12	46184,96	16370,81	13	474,9832	2267,8834	16370,81	0	1	34,464
13	50333	15520,45	13	525,8825	2166,2095	15366,78	1	0	34,871
14	49490,56	16280,35	13	487,4924	2383,9985	15961,13	2	0	34,754
15	50086,68	16200,58	13	704,4885	2163,9992	16200,58	0	1	34,464

**Tablo 3.** Şehirlerin enlem ve boylam koordinatları (Latitude and longitude coordinates of cities)

Sıra Kod	İl Adı	Enlem	Boylam	Talep	Sıra Kod	İl Adı	Enlem	Boylam	Talep		
0	6	Ankara	39,92077	32,85411	0	41	46	K. Maraş	37,585831	36,937149	1
1	1	Adana	37	35,321333	12	42	78	Karabük	41,2061	32,62035	3
2	2	Adıyaman	37,764751	38,278561	2	43	70	Karaman	37,17593	33,228748	4
3	3	Afyon	38,750714	30,556692	2	44	36	Kars	40,616667	43,1	2
4	4	Ağrı	39,626922	43,021596	4	45	37	Kastamonu	41,38871	33,78273	3
5	68	Aksaray	38,36869	34,03698	2	46	38	Kayseri	38,73122	35,478729	6
6	5	Amasya	40,64991	35,83532	2	47	71	Kırkkale	39,846821	33,515251	3
7	7	Antalya	36,88414	30,70563	2	48	39	Kırklareli	41,733333	27,216667	2
8	75	Ardahan	41,110481	42,702171	2	49	40	Kırşehir	39,14249	34,17091	5
9	8	Artvin	41,18277	41,818292	2	50	79	Kilis	36,718399	37,12122	4
10	9	Aydın	37,856041	27,841631	2	51	41	Kocaeli	40,85327	29,88152	1
11	10	Balıkesir	39,648369	27,88261	2	52	42	Konya	37,866667	32,483333	4
12	74	Bartın	41,581051	32,460979	6	53	43	Kütahya	39,416667	29,983333	5
13	72	Batman	37,881168	41,13509	1	54	44	Malatya	38,35519	38,30946	7
14	69	Bayburt	40,255169	40,22488	2	55	45	Manisa	38,619099	27,428921	8
15	11	Bilecik	40,056656	30,066524	7	56	47	Mardin	37,321163	40,724477	2
16	12	Bingöl	39,062635	40,76961	3	57	33	Mersin	36,8	34,633333	2
17	13	Bitlis	38,393799	42,12318	2	58	48	Muğla	37,215278	28,363611	7
18	14	Bolu	40,575977	31,578809	2	59	49	Muş	38,946189	41,753893	3
19	15	Burdur	37,461267	30,066524	2	60	50	Nevşehir	38,69394	34,685651	9
20	16	Bursa	40,266864	29,063448	10	61	51	Niğde	37,966667	34,683333	4
21	17	Çanakkale	40,155312	26,41416	1	62	52	Ordu	40,983879	37,876411	5
22	18	Çankırı	40,601343	33,613421	1	63	80	Osmaniye	37,213026	36,176261	9
23	19	Çorum	40,550556	34,955556	5	64	53	Rize	41,02005	40,523449	2
24	20	Denizli	37,77652	29,08639	4	65	54	Sakarya	40,693997	30,435763	7
25	21	Diyarbakır	37,91441	40,230629	2	66	55	Samsun	41,292782	36,33128	5
26	81	Düzce	40,843849	31,15654	2	67	56	Siirt	37,933333	41,95	3
27	22	Edirne	41,681808	26,562269	2	68	57	Sinop	42,02314	35,153069	3
28	23	Elazığ	38,680969	39,226398	5	69	58	Sivas	39,747662	37,017879	3
29	24	Erzincan	39,75	39,5	3	70	63	Şanlıurfa	37,159149	38,796909	2
30	25	Erzurum	39,9	41,27	9	71	73	Şırnak	37,418748	42,491834	1
31	26	Eskişehir	39,776667	30,520556	7	72	59	Tekirdağ	40,983333	27,516667	3
32	27	Gaziantep	37,06622	37,38332	5	73	60	Tokat	40,316667	36,55	4
33	28	Giresun	40,912811	38,38953	9	74	61	Trabzon	41,00145	39,7178	2
34	29	Gümüşhane	40,438588	39,508556	2	75	62	Tunceli	39,307355	39,438778	3
35	30	Hakkâri	37,583333	43,733333	2	76	64	Uşak	38,682301	29,40819	6
36	31	Hatay	36,401849	36,34981	5	77	65	Van	38,48914	43,40889	3
37	76	Iğdır	39,887984	44,004836	3	78	77	Yalova	40,65	29,266667	2
38	32	Isparta	37,764771	30,556561	3	79	66	Yozgat	39,818081	34,81469	5
39	34	İstanbul	41,00527	28,97696	12	80	67	Zonguldak	41,456409	31,798731	4
40	35	İzmir	38,41885	27,12872	3						

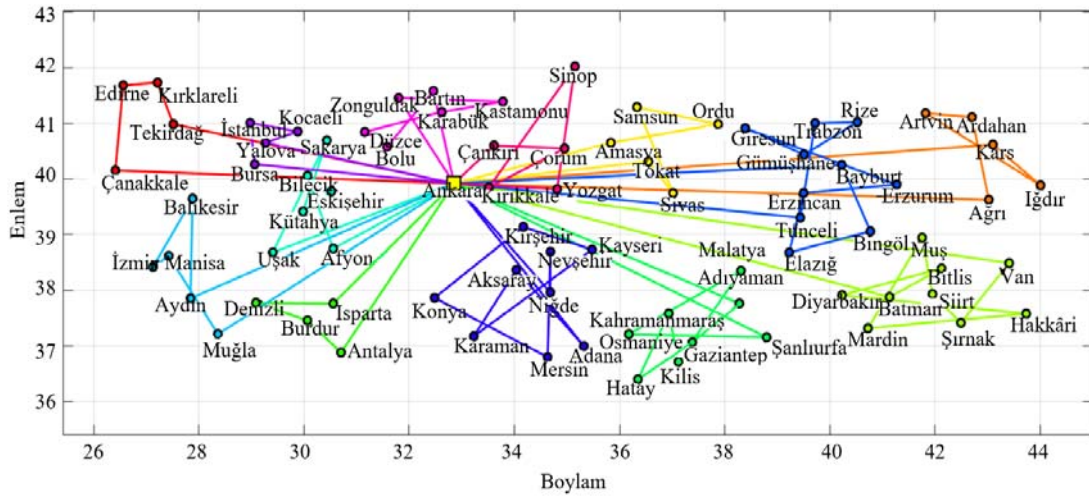
**Şekil 2.** Standart TB ve 13 kümeli yaklaşımın başlangıç çözümü (Initial solution of standard SA and 13 cluster approach)



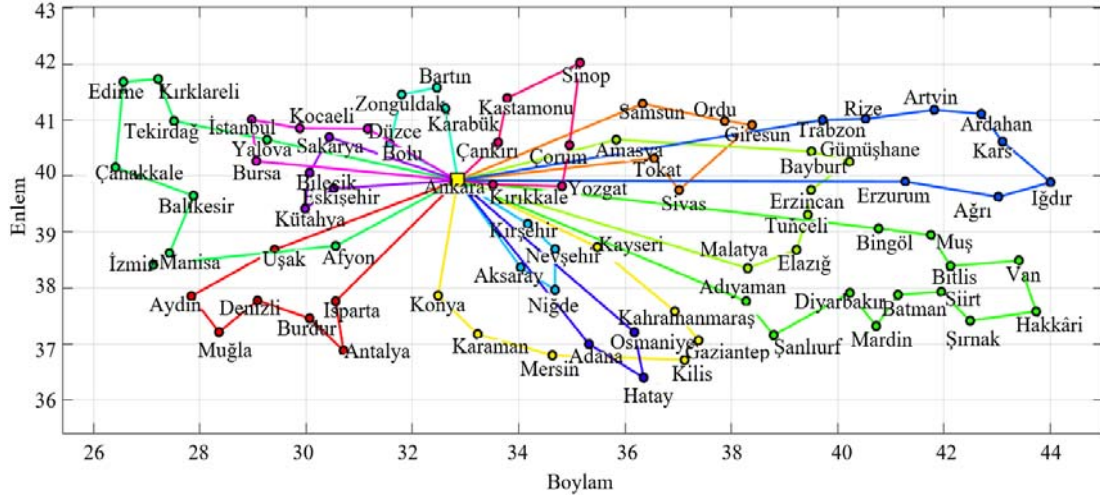
Şekil 3. Standart TB ve 13 kümeli yaklaşımın nihai sonucu (Final result of standard SA and 13 cluster approach)

Tablo 4. 13 kümeli ve iyileştirilmiş başlangıçlı sonuçlar (Results with 13 clusters and improved initialization)

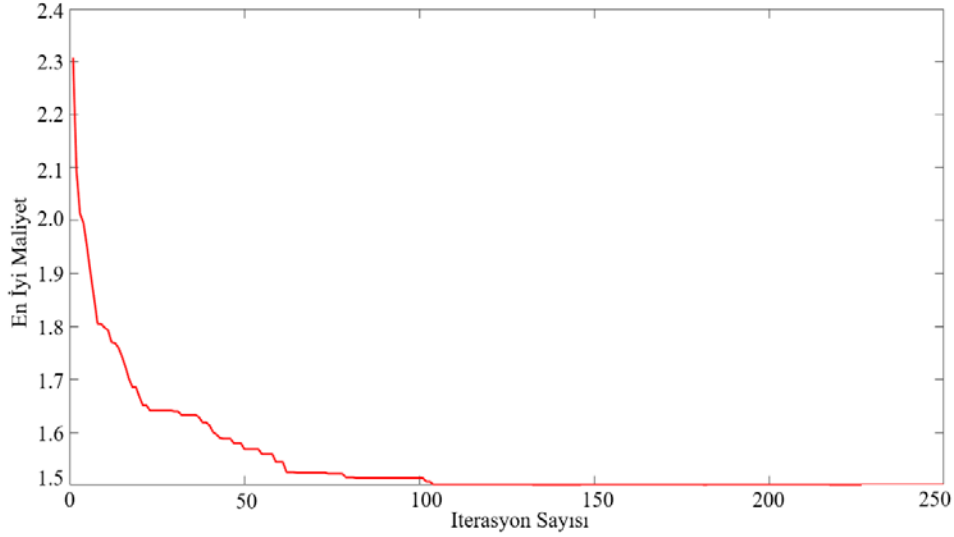
Deney No	Başlangıç Çözümü	En İyi Çözüm	Küme Sayısı	Minimum Mesafe	Maksimum Mesafe	Toplam Mesafe	Toplam Fazla Kapasite	Uygulanabilir Çözüm	Çözüm Süresi
1	20806,95	15832,4	13	437,4377	2346,2753	15832,4	0	1	34,469
2	21476,36	15589,93	13	441,9619	2364,5582	15589,93	0	1	34,585
3	20806,95	15441,87	13	417,556	2371,5039	15441,87	0	1	34,59
4	21255,83	15721,4	13	417,556	2475,6356	15721,4	0	1	34,561
5	21351,41	15008,19	13	474,9832	2226,8662	15008,19	0	1	34,905
6	20806,95	15491,48	13	417,556	2226,8662	15491,48	0	1	34,636
7	21130,08	15819,58	13	437,4377	2170,9131	15819,58	0	1	34,503
8	20806,95	15563,24	13	428,7202	2168,3869	15563,24	0	1	34,436
9	22092,89	15794,72	13	541,9302	2222,4666	15794,72	0	1	34,895
10	21118,83	15484,17	13	417,556	2150,4097	15484,17	0	1	34,557
11	21572,44	15605,05	13	555,2454	2367,1425	15605,05	0	1	34,493
12	21536,05	15440,78	13	568,371	2339,3172	15287,9	1	0	34,508
13	21340,09	15658,85	13	417,556	2347,6152	15658,85	0	1	34,372
14	20806,95	15495,43	13	399,7	2157,0576	15495,43	0	1	34,386
15	21150,86	15301,07	13	417,556	2197,445	15301,07	0	1	34,844



Şekil 4. 13 küme ve başlangıç çözümü optimize edilen yöntemin başlangıç durumu (Initial state of the method with 13 clusters and initial solution optimized)



**Şekil 5.** 13 küme ve başlangıç çözümü optimize edilen yöntemin final sonucu  
(Final result of the method with 13 clusters and initial solution optimized)



**Şekil 6.** Amaç fonksiyonundaki değişme (Changes in the objective function)

#### 4.1. Deneysel Sonuçlarının Değerlendirilmesi (Evaluation of Test Results)

Problemi çözmek için kullanılan TB algoritmasında üst iterasyon sayısı 250 ve alt iterasyon sayısı 500 olmak üzere, arama uzayında toplamda 125000 adet rota oluşturacak kombinasyon test edilmiştir. İlk sıcaklık değeri 10, sıcaklık düşürme oranı 0,97 olarak belirlenmiştir. Problemden, 26 birim kapasiteli özdeş araçlar kullanılmıştır. Problemden müşterilerin toplam talebi 311'dir. Problemi minimum araç sayısı ile çözmek istersek, 12 ( $311/26 \approx 12$ ) araca ihtiyaç vardır. Farklı araç sayıları için sonuçları gözlemleyebilmek için 12, 13 ve 14 araçlı olacak şekilde modeller oluşturulmuştur. 3 farklı araç sayısı için rassal başlangıç çözümü ve iyileştirilmiş başlangıç çözümü olmak üzere toplamda 6 adet model oluşturulmuştur. Her bir modelde 15 adet olmak üzere toplamda 90 deney yapılmıştır. Aşağıda sadece uygulanabilir çözümlerin istatistikleri verilmiştir.

12 araç için rassal çözüm oluşturulduğunda ortalama arama uzayı 50834,62 kilometre, ortalama en iyi çözüm değeri 16919,2 kilometre, tüm çözümlerde oluşan rota (küme) sayısı 12, minimum rotaların ortalaması 779,94 kilometre, maksimum rotaların ortalaması 2393,42 kilometre, ortalama çözüm süresi 33,55 saniye olarak bulunmuştur. 15 deneyden 6 tanesi uygulanabilir çözümdür. Uygulanabilir çözüm bulunma ihtimali %40'tır. 12 araç için iyileştirilmiş çözüm oluşturulduğunda ortalama arama uzayı 20802 kilometre, ortalama en iyi çözüm değeri 16199,96 kilometre, tüm çözümlerde oluşan rota (küme) sayısı 12, minimum rotaların ortalaması 730,81 kilometre, maksimum rotaların ortalaması 2242 kilometre, ortalama çözüm süresi 33,75 saniye olarak bulunmuştur. 15 deneyden 8 tanesi uygulanabilir çözümdür. Uygulanabilir çözüm bulunma ihtimali %53'tür. 13 araç için rassal çözüm oluşturulduğunda ortalama arama uzayı 49565,61 kilometre, ortalama en iyi çözüm değeri 15965,08 kilometre, tüm çözümlerde oluşan rota (küme) sayısı 13,

minimum rotaların ortalaması 485,23 kilometre, maksimum rotaların ortalaması 2338,93 kilometre, ortalama çözüm süresi 34,65 saniye olarak bulunmuştur. 15 deneyden 11 tanesi uygulanabilir çözümdür. Uygulanabilir çözüm bulunma ihtimali %73'tür. 13 araç için iyileştirilmiş çözüm oluşturulduğunda ortalama arama uzayı 21180,25 kilometre, ortalama en iyi çözüm değeri 15557,67 kilometre, tüm çözümlerde oluşan rota (küme) sayısı 13, minimum rotaların ortalaması 444,48 kilometre, maksimum rotaların ortalaması 2270,94 kilometre, ortalama çözüm süresi 34,59 saniye olarak bulunmuştur. 15 deneyden 14 tanesi uygulanabilir çözümdür. Uygulanabilir çözüm bulunma ihtimali %93'tür.

14 araç için rassal çözüm oluşturulduğunda ortalama arama uzayı 49538,56 kilometre, ortalama en iyi çözüm değeri 16105,94 kilometre, tüm çözümlerde oluşan rota (küme) sayısı 13, minimum rotaların ortalaması 512,63 kilometre, maksimum rotaların ortalaması 2290,99 kilometre, ortalama çözüm süresi 34,17 saniye olarak bulunmuştur. 15 deneyden 13 tanesi uygulanabilir çözümdür. Uygulanabilir çözüm bulunma ihtimali %87'dir. 14 araç için iyileştirilmiş çözüm oluşturulduğunda ortalama arama uzayı 21842,15 kilometre, ortalama en iyi çözüm değeri 15768,01 kilometre, oluşan rota (küme) sayısı bir deney için 12, on dört deney için 13, minimum rotaların ortalaması 442,94 kilometre, maksimum rotaların ortalaması 2253,71 kilometre, ortalama çözüm süresi 34,9 saniye olarak bulunmuştur. 15 deneyden 14 tanesi uygulanabilir çözümdür. Uygulanabilir çözüm bulunma ihtimali %93'tür.

Tüm modeller için ortalamalara bakıldığında en iyi sonucu 13 araç için iyileştirilmiş çözümlü model vermiştir. Tüm çözümlere genel olarak bakıldığında yine aynı model içerisindeki 5 numaralı deney en iyi sonucu vermiştir. Bu deneyin sonuçları: Arama uzayı 21351,41 kilometre, en iyi çözüm 15008,19 kilometre, rota (küme) sayısı 13, minimum rota 447,98 kilometre, maksimum rota 2226,87 kilometre, çözüm süresi 34,9 saniyedir. En iyi çözümün 12 araç için uygulanan modeller olması beklenmiştir. Ama bu modellerde müşterilerin toplam taleplerini karşılayabilmek için toplam araç kapasitelerinin %99,68 oranında kullanılacağından uygulanabilir bir çözüm bulamadan algoritma çıkmaza girer. 14 araç için uygulanan modeller de amaç fonksiyonu optimum çözüme ulaşmaya çalıştığında algoritmanın en rahat çalışacağı araç sayısına düşmüştür. Karar vericiler 12 araç için fazla mesafe kat etmeyi göze alırlarsa bu model kullanılabilir. Bir aracın, rotasını tamamlaması için gereken maliyet, fazladan kat edilen mesafeden büyük ise 12 araçlı model; diğer durumlarda 13 araçlı model kullanılabilir.

Başlangıç çözüm üzerinde bulanık c-ortalamlar yöntemin etkisini daha iyi görebilmek için bulanık c-ortalamlar ile kümeleme yöntemlerinin rassal başlangıç çözümüne karşı iyi sonuçlar verip vermediği test edildi. Bu test için eşlenik t testi kullanıldı. Bu test başlangıç çözümü ve toplam mesafe değerleri için ayrı ayrı uygulandı. En iyi sonuçlar 13 kümelili modelde elde edildiğinden test için de yine 13 kümelili model kullanıldı. Sonuçlar aşağıdaki tablolarda görülmektedir.

Başlangıç çözümü için test sonuçları şu şekilde bulunmuştur: Rassal Başlangıç için Ortalama, Varyans ve Gözlem değerleri sırasıyla 49933,07, 3839999,49 ve 15 dir. Bulanık c-ortalamlar için ise Ortalama, Varyans ve Gözlem değerleri de sırasıyla 21203,97, 139787,04 ve 15 dir. Pearson Korelasyonu değeri 0,4652 ve t Stat değeri 61,2678 dir. Hem  $P(T \leq t)$  tek-uçlu hem de  $P(T \leq t)$  iki-uçlu test için elde edilen değer 0,0000 dir. t Kritik tek-uçlu değer 1,7613 ve t Kritik iki-uçlu test değeri 2,1448 dir. Başlangıç çözümü için bu test sonuçlarına bakıldığında Rassal başlangıç çözümü ve bulanık c-ortalamlar başlangıç çözümü için  $P < 0,05$  olduğundan değerler arasında fark yoktur hipotezi reddedilir. Yani değerler arasında fark vardır. Bulanık c-ortalamlar başlangıç çözümü rassal başlangıç çözümüne göre daha iyi sonuçlar vermiştir.

Aynı şekilde toplam mesafe için test sonuçları ise şu şekilde bulunmuştur: Rassal Başlangıç için Ortalama, Varyans ve Gözlem değerleri sırasıyla 15898,35, 54041,80 ve 15 dir. Bulanık c-ortalamlar için ise Ortalama, Varyans ve Gözlem değerleri sırasıyla 15539,68, 50242,40 ve 15 dir. Pearson Korelasyonu değeri -0,4254 ve t Stat değeri 3,6034 dir.  $P(T \leq t)$  tek-uçlu test için elde edilen değer 0,0014 ve  $P(T \leq t)$  iki-uçlu test için elde edilen değer 0,0029 dur. t Kritik tek-uçlu değer 1,7613 ve t Kritik iki-uçlu test değeri 2,1448 dir. Toplam mesafe (nihai çözüm) için test sonuçlarına bakıldığında rassal başlangıç çözümü ve bulanık c-ortalamlar başlangıç çözümü için  $P < 0,05$  olduğundan değerler arasında fark yoktur hipotezi reddedilir. Yani değerler arasında fark vardır. Bulanık c-ortalamlar başlangıç çözümü rassal başlangıç çözümüne göre daha iyi sonuçlar vermiştir.

## 5. SONUÇLAR (CONCLUSIONS)

Başlangıç çözümü TB algoritmasının performansında önemli bir rol oynamaktadır. Bu makalede, KARP'ni çözmek için başlangıç çözümü iyileştirilmiş bir tavlama benzetimi algoritması geliştirildi. Kümeleneleme dayalı başlangıç çözümünün kullanılmasının KARP'ni çözmeye TB'nin performansını nasıl etkilediğini görmek açısından önemlidir. TB algoritmasının performansının artırılmasına yönelik geliştirilen yöntem başlangıç çözümü iyileştirme sonucunda TB'nin hesaplama süresini kısaltmada önemli bir katkı sağlar. Yöntem, hem çözüm kalitesi hem de hesaplama gerekliliği açısından iyi bir performans gösterir. Geliştirilen yöntem basittir ve çok az hesaplama çabası gerektirir: Çözümlerin etkinliğini ve doğruluğunu artırmakla birlikte çözümlerin hazırlanma hızını da önemli ölçüde artırmaktadır. TB rastgele aramalar yapan yapısından dolayı özellikle karmaşık problemlerde yavaş yakınsamadan mustarip bir yöntemdir. Çalışmada geliştirilen yöntem başlangıç çözümü iyileştirme sonucunda TB'nin rassal yapısından kaynaklanan yavaş yakınsamanın neden olduğu zamanını kısaltarak önemli katkı sağlar. Bu çalışma ile başlangıç çözümü iyileştirme sonucunda rassal başlangıç çözümüne göre optimum çözüme daha yakın bir durum elde edilmektedir. Dolayısı ile daha az iterasyon veya hesaplama yapılarak, daha kısa sürede optimum çözüme ulaşılmaktadır.



Çalışmada rassal başlangıç çözümü ve bulanık c-ortalamlar ile iyileştirilmiş başlangıç çözümü durumları için 1200 iterasyon kullanılarak testler yapıldı. Bulanık c-ortalamlar ile iyileştirilmiş başlangıç çözümü, diğer duruma göre optimuma daha yakın bir çözüm oluşturmuştur. Bulanık mantık gereği, her verinin 0-1 arasında birden fazla kümeyle dahil olabilmesi durumu, algoritmanın her çözümünde başlangıç çözümünü değiştireceğinden dolayı optimum çözüme yaklaşıma durumunu artırmıştır. Test sonuçları ile ilgili tablo ve grafiklerinde görüldüğü gibi bulanık c-ortalamlar ile iyileştirilmiş başlangıç çözümü rassal başlangıç durumuna göre daha iyi bir çözümle başlamıştır ve 1200 iterasyonda optimum çözüme daha erken ulaşmıştır. Hesaplama sonuçları, kümeleme tabanlı bir başlangıç çözümünün kullanılmasının, literatürdeki önceki algoritmalara kıyasla daha iyi çözümlere ulaşmada etkili olduğunu göstermektedir. Bu model ile her başlangıçta, başlangıç çözümü farklı olduğundan ve noktalar, mesafe olarak birbirine en yakın olacak şekilde kümelendiğinden ve arama alanında ortalama olarak %57 oranında azaltma yapıldığından dolayı çıkan sonuçlar oldukça ümit vericidir. Meta sezgisel, her bir sınıf için iyi bir performans gösterir ve sınırlı bir hesaplama çabası gerektirir. Dahası, algoritma KARP için son derece rekabetçi olduğunu da kanıtladı. Kümeleme algoritması parametrelerini ayarlama ve çözümün kalitesine nasıl katkıda bulduklarını analiz etme gibi bazı muhtemel ileri araştırma alanları da vardır. TB ile Genetik Algoritma veya Tabu Arama gibi diğer iyi bilinen sezgisel yöntemler ve meta-sezgisellerin karşılaştırılması da mümkündür. Başlangıç çözümünün oluşturulmasında diğer kümeleme algoritmalarının etkisi de analiz edilebilir.

#### KAYNAKLAR (REFERENCES)

1. Dantzig G.B., Ramser J.G., The truck dispatching problem, *Management Science*, 6 (1), 80-91, 1959.
2. Jaegere N.D., Defraeye M., Nieuwenhuysse I.V., The Vehicle Routing Problem: State of the Art Classification and Review, FEB Research Report KBI1415, Leuven, Belçika, 2014.
3. Borcinova Z., Two models of the capacitated vehicle routing problem, *Croatian Operational Research Review*, 8 (2), 463-469, 2017.
4. Ropke S., Heuristic and exact algorithms for vehicle routing problems, DTU library, Kopenhagen, Danimarka, 2006.
5. Kechmane L., Nsiri B., Baalal A., A hybrid particle swarm optimization algorithm for the capacitated location routing problem, *International Journal of Intelligent Computing and Cybernetics*, 11 (1), 106-120, 2018.
6. Goel R., Maini R., A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems, *Journal of Computational Science*, 25, 28-37, 2018.
7. Khambampati S., Calyam P., Zhang X., A tabu search algorithm for a capacitated clustering problem, *International Journal of Operational Research*, 33 (1), 387-412, 2018.
8. Karagul K., Sahin Y., Aydemir E., Oral A., A simulated annealing algorithm based solution method for a green vehicle routing problem with fuel consumption, in Paksoy T., Weber G.W., Huber S. (Eds), *Lean and Green Supply Chain Management*, Springer, 161-187, 2019.
9. Ghannadpour S.F., Noori S., T.-Moghaddam R., Ghoseiri K., A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application, *Applied Soft Computing*, 14, 504-527, 2014.
10. Toth P., Vigo D., Models, relaxations and exact approaches for the capacitated arc routing problem, *Discrete Applied Mathematics*, 123 (1-3), 487-512, 2002.
11. Min H., Jayaraman V., Srivastava R., Combined location-routing problems: A synthesis and future research directions, *European Journal of Operational Research*, 108 (1), 1-15, 1998.
12. Balinski M.L., Quandt R.E., On an integer program for a delivery problem, *Operations Research*, 12 (2), 300-304, 1964.
13. Ghiani G., Laporte G., Musmanno R., *Introduction to logistics systems management*, John Wiley & Sons, Chichester, Birleşik Krallık, 2013.
14. Lenstra J.K., Kan A.H.G.R., Some simple applications of the travelling salesman problem, *Operational Research Quarterly* (1970-1977), 26 (4), 717, 1975.
15. Christofides N., Mingozzi A., Toth P., Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations, *Mathematical Programming*, 20 (1), 255-282, 1981.
16. Burrows C., Eilon S., Watson-Gandy C., Christofides N., *Distribution management: Mathematical modelling and practical analysis*, *Applied Statistics*, 21, 337, 1972.
17. Christofides N., Vehicle Routing. In Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys, D.B., editors, *The traveling salesman problem. a guided tour of combinatorial optimization*, chapter Vehicle Routing, Wiley, Chichester, Birleşik Krallık, 431-448, 1985.
18. Laporte G., The vehicle routing problem: An overview of exact and approximate algorithms, *European Journal of Operational Research*, 59 (3), 345-358, 1992.
19. Clarke G., Wright J.W., Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12 (4), 568-581, 1964.
20. Paessens H., The savings algorithm for the vehicle routing problem, *European Journal of Operational Research*, 34 (3), 336-344, 1988.
21. Wren A., Holliday A., Computer scheduling of vehicles from one or more depots to a number of delivery points, *Journal of the Operational Research Society*, 23 (3), 333-344, 1972.
22. Gendreau M., Hertz A., Laporte G., A Tabu Search heuristic for the vehicle routing problem, *Management Science*, 40 (10), 1276-1290, .
23. Zhang H., Zhang Q., Ma L., Zhang Z., Liu Y., A hybrid ant colony optimization algorithm for a multi-objective



- vehicle routing problem with flexible time windows, *Information Sciences*, 490, 166-190, 2019.
24. El-Sherbeny N.A., Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods, *Journal of King Saud University - Science*, 22 (3), 123-131, 2010.
  25. Afifi S., Dang, D.C. Moukrim. A Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. *Optimization Letters*, 10 (3), 511–525, 2016.
  26. Ünsal Ö., Yiğit T., Yapay Zekâ ve Optimization of school bus routing problem by using a method with artificial intelligence and clustering techniques, *Journal of Engineering Sciences and Design*, 6 (1), 7-20, 2018.
  27. Swersey A.J., Ballard W, Scheduling School Buses, *Management Science*, 30 (7), 844-853, 1984.
  28. Baker E.K., Schaffer J.R., Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints, *American Journal of Mathematical and Management Sciences*, 6 (3-4), 261-300, 1986.
  29. Solomon M.M., Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research*, 35 (2), 166-324, 1987.
  30. Landeghem H., A bi-criteria heuristic for the vehicle routing problem with time windows, *European Journal of Operational Research*, 36 (2), 217-226, 1988.
  31. Potvin J.Y., Rousseau J.M., An exchange heuristic for routing problems with time windows, *Journal of the Operational Research Society*, 46 (12), 1433-1466, 1995.
  32. González O.M., Segura C., Peña S.I.V., A parallel memetic algorithm to solve the capacitated vehicle routing problem with time windows, *International Journal of Combinatorial Optimization Problems and Informatics*, 9 (1), 35-45, 2018.
  33. Puspita F.M., Cahyono E.S., Rahayu S., Sintia B.L., Model of demand robust counterpart open capacitated vehicle routing problem (DRC-OCVRP) Simplification by applying pre-processing techniques in rubbish controlling in Sematang Borang District, *E3S Web of Conferences*, EDP Sciences, Palembang, 68, 01025, 2018.
  34. Xia Y., Fu Z., Pan L., Duan F., Tabu Search Algorithm for The Distance-Constrained Vehicle Routing Problem with Split Deliveries by Order, *PloS One*, 13 (5), 1-19, 2018.
  35. Novoa-Flores G.I., Carpena L., Lorenzo-Freire S., A vehicle routing problem with periodic replanning, *Multidisciplinary Digital Publishing Institute Proceedings*, 2 (18), 1192, 2018.
  36. Li F., Golden B., Wasil E., The open vehicle routing problem: Algorithms, large-scale test problems, and computational results, *Computers & Operations Research*, 34 (10), 2918-2930, 2007.
  37. Liu C., Lai M.Y., The vehicle routing problem with uncertain demand at nodes, *Transportation Research Part E Logistics and Transportation Review*, 45 (4), 517-524, 2009.
  38. Bayrak A., Özyörük B., Comparative mathematical models for split delivery simultaneous pickup and delivery vehicle routing problem, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32 (2), 469-479, 2017.
  39. Atasagun G.C., Karaoğlan İ., A mathematical model for the time dependent vehicle routing problem with simultaneous pick-up and delivery, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (4), 1743-1755, 2019.
  40. Küçükaydın H., Column generation based metaheuristics for a vehicle routing problem with time Windows and variable start time, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (4), 2061-2078, 2019.
  41. Aydoğdu B., Özyörük B., Mathematical model and heuristic approach for solving dynamic vehicle routing problem with simultaneous pickup and delivery: Random iterative local search variable neighborhood descent search, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35 (2), 563-580, 2020.
  42. Cil I., Erdil N. O., Kilic T., Kosar B., Lean logistic network design and analysis with Anylogic. In *LM-SCM 2016 XIV. International Logistics and Supply Chain Congress*, 523, 2016.
  43. Cil I., Demir H. I., Yaman B., Lean Logistics in the 2020s and a case study about logistics and supply chain management in Toyota Boshoku Turkey. In *Global Perspectives on Green Business Administration and Sustainable Supply Chain Management*, 276-315, IGI Global, 2020.
  44. Rashid N. M., Ngalawa F., Cil I., Comparative study of logistic industry of Tanzania and Turkey. *International Journal of Operations and Logistics Management*, 5 (2), 74-82, 2016.
  45. Yoshiike N., Takefuji Y., Solving vehicle routing problems by maximum neuron model, *Advanced Engineering Informatics*, 16 (2), 99-105, 2002.
  46. Barreto S., Ferreira C., Paixao J., Santos B.S., Using clustering analysis in a capacitated location-routing problem, *European Journal of Operational Research*, 179 (3), 968-977, 2007.
  47. Kirkpatrick S., Gelatt C., Vecchi M., Optimization by simulated annealing, *Science*, 220 (4598), 671-680, 1983.
  48. Demir A.S., Tavlama benzetimi algoritması ile tesis yerleşim düzenlemesi ve TÜVASAŞ Fabrikası'nda bir uygulaması, Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya, 2002.
  49. Alfa A.S., Heragu S.S., Chen M., A 3-opt based simulated annealing algorithm for vehicle routing problems, *Computers & Industrial Engineering*, 21 (1-4), 635-639, 1991.
  50. Aarts E., Korst J., *Simulated Annealing and Boltzmann Machines*. Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, Chichester, Birleşik Krallık, 1989.

51. Robust'e F., Daganzo C.F., Souleyrette R.R.I., Implementing vehicle routing models, *Transportation Research Part B*, 24 (4), 263-286, 1990.
52. Van Breedam A., Comparing descent heuristics and metaheuristics for the vehicle routing problem, *Computers & Operations Research*, 28 (4), 289-315, 2001.
53. Tan K.C., Lee L.H., Zhu Q.L., Ou K., Heuristic methods for vehicle routing problem with time windows, *Artificial Intelligence in Engineering*, 15 (3), 281-295, 2001.
54. Bezdek J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, ABD, 1981,
55. Pandey G., Kumar V., Steinbach M., *Computational Approaches for Protein Function Prediction, Supported by the National Science Foundation under Grant Nos. IIS-0308264 and ITR-0325949*, 2007.
56. Dunn J.C., A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, *Journal of Cybernetics*, 3 (3), 32-57, 1973.
57. Wei L., Zhang, Z., Zhang D., Leung S.C.H., A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints, *European Journal of Operational Research* 265, 843–859, 2018.
58. Wang J., Jagannathan A.K.R., Zuo X., Murray C.C., Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries, *Computers & Industrial Engineering*, 112, 84–98, 2017.
59. Johnson R. K., and Sachin F., Particle swarm optimization methods for data clustering. *IEEE fifth international conference soft computing, computing with words and perceptions in system analysis, decision and control, Famagusta, Cyprus*, 1–6, 2009.
60. Osman I.H., Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, 41, 421-451, 1993.
61. Lin S. W., Lee Z. J., Ying K. C., Lee C. Y., Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications*, 36 (2), 1505-1512, 2009.
62. Sigauke C., Talukder H. M. A modified Osman's simulated annealing and tabu search algorithm for the vehicle routing problem. *Australian Society For Operations Research*, 22 (3), 9–14, 2003.
63. Chiang W.-C., Russell R. A., Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63 (1), 3–27, 1996.
64. Tavakkoli-Moghaddam R., Safaei N., Gholipour Y., A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation*, 176 (2), 445-454, 2006.
65. Yu V. F., Lin S. Y., A simulated annealing heuristic for the open location-routing problem. *Computers and Operations Research*, 62, 184–196, 2015.
66. Vincent F. Y., Redi A. P., Hidayat Y. A., Wibowo O. J., A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, 53, 119-132, 2017.
67. Rabbouch B., Saâdaoui F., Mraïhi R., Empirical-type simulated annealing for solving the capacitated vehicle routing problem. *Journal of Experimental Theoretical Artificial Intelligence*, 32 (3), 437-452, 2020.
68. El Afia A. Lalaoui M. Talbi E., Quaternion Simulated Annealing. In: Yalaoui F., Amodeo L., Talbi EG. (eds) *Heuristics for Optimization and Learning. Studies in Computational Intelligence*, 906. Springer, Cham, 2021.
69. Leung S. C., Zheng J., Zhang D., Zhou X., Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible services and manufacturing journal*, 22 (1), 61-82, 2010.
70. Sajid M., Jafar A., Sharma S., Hybrid Genetic and Simulated Annealing Algorithm for Capacitated Vehicle Routing Problem. In *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 131-136, IEEE, 2020, November.
71. Mar'i F., Mahmudy W. F., Santoso P. B., Hybrid Particle Swarm Optimization and Simulated Annealing for Capacitated Vehicle Routing Problem. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, 66-71, IEEE, 2019, September.
72. Kayman A. Y., Gezgin satıcı probleminin çözümünde parçacık sürü optimizasyonu algoritması performansının bulanık c-ortalamalar yöntemi ile iyileştirilmesi. *Doktora Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, 2015.
73. Fisher M.L. Jaikumar R. A generalized assignment heuristic for vehicle routing, *Networks*, 11, 109-124, 1981.
74. Beasley J.E., Route first-cluster second methods for vehicle routing, *Omega*, 11., 403-408, 1983.
75. Mohammed M. A., Abd Ghani M. K., Hamed R. I., Mostafa S. A., Ibrahim D. A., Jameel H. K., Alallah A. H., Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. *Journal of Computational Science*, 21, 232-240, 2017.
76. Bozyer Z., Alkan A., Fiğlalı A., Kapasite kısıtlı araç rotalama probleminin çözümü için önce grupla sonra rotala merkezli sezgisel algoritma önerisi. *Bilişim Teknolojileri Dergisi*, 7 (2), 2014.
77. Ewbank H., Wanke P., Correa H. L., Figueiredo O., The capacitated vehicle routing problem revisited: using fuzzy c-means clustering. *International Journal of Logistics Systems and Management*, 34 (4), 411-430, 2019.
78. Cordeau J.F., Laporte G., Savelsbergh M.W.P., Vigo D., *Handbook in OR & MS, Chapter 6: Vehicle Routing*, Elsevier, 14, 369, 2007.