

Kapasite Kısıtlı Araç Rotalama Probleminin Çözümü için Önce Grupla Sonra Rotala Merkezli Sezgisel Algoritma Önerisi

Zafer Bozyer¹, Atakan Alkan², Alpaslan Fıglalı³

¹Yönetim Bilişim Sistemleri Bölümü, İİBF, Bartın Üniversitesi, Bartın, Türkiye

^{2,3}Endüstri Mühendisliği, Mühendislik Fakültesi, Kocaeli Üniversitesi, Kocaeli, Türkiye

zaferbozyer@hotmail.com, aalkan@kocaeli.edu.tr, figlalia@kocaeli.edu.tr

(Geliş/Received: 21.02.2014; Kabul/Accepted: 05.05.2014)

DOI: 10.12973/bid.2018

Özet- Lojistik, insanoğlunun uzun zamandır üzerinde çalıştığı en temel konulardan birisidir. Son yüzyıl içerisinde müşteri taleplerinin kısa zamanda ve verimli bir şekilde ulaştırılabilmesi ihtiyacı, lojistik faaliyetlerinin içerisinde yer alan araç rotalama problemlerinin (ARP) öneminin artmasına neden olmuştur. Gün geçtikçe artan önemi nedeniyle birçok araştırmaya konu olan araç rotalama problemlerinin çözülebilmesi için farklı yöntemler geliştirilmiştir. En iyi sonuca ulaşmayı mümkün kılan yöntemlerde problem boyutu arttıkça daha fazla hesaplama süresine ihtiyaç duyulmaktadır. Bundan dolayı daha kısa sürede kabul edilebilir sonuçlara ulaşmayı sağlayan sezgisel yöntemler hakkında da oldukça fazla çalışma yapılmıştır. Bu çalışmada da, kapasite kısıtlı araç rotalama problemlerinin (KKARP) çözümüne yönelik önce grupla sonra rotala prensibine dayanan sezgisel bir yöntem önerilmiştir. Gruplandırma adımında, talep noktalarının bulanık c-ortalama kümeleme yöntemi ile olası tüm rotalara 0-1 arasında üyelik dereceleri hesaplanmıştır. Rotalama adımında ise sezgisel bir algoritma olan tabu arama prensiplerine dayanan bir arama algoritması ile rotalar iyileştirilmeye çalışılmıştır. Sonuç olarak KKARP'lerinin gezgin satıcı problemine dönüştürülerek çözülebileceği görülmüştür. Önerilen yöntem literatürde yer alan veri kümelerine uygulanmış ve elde edilen sonuçlar tartışılmıştır.

Anahtar Kelimeler- Araç rotalama problemi, önce grupla sonra rotala, tabu arama, bulanık c-ortalama, kümeleme

Cluster-First, Then-Route Based Heuristic Algorithm for the Solution of Capacitated Vehicle Routing Problem

Abstract- Logistics has important effects on human life throughout the history. Within the past century the need to convey customer demands shortly and efficiently has resulted in the increasing importance of the vehicle routing problems (VRPs) that included in logistics activities. Different methods have been developed in order to solve vehicle routing problems that are subject of various researchs due to its increasing importance day by day. More computation time is required to the methods that provides to find the optimal results while the problem size increases. Therefore, pretty much work has been done about heuristic methods that provide access to acceptable results in a shorter time. In this paper, a "cluster-first, then-route" based heuristic algorithm is proposed for the solution of Capacitated Vehicle Routing Problem (CVRP). The membership degrees of demand points to each potential route are calculated by using fuzzy c-means in clustering step; and then the routes are improved by using a search procedure based on Tabu Search algorithm in the routing step. As a result of, the capacitated vehicle routing problems can be solved by conversion to the traveling salesman problem has been observed. The proposed method is applied on literature problems and the results are discussed.

Keywords- Vehicle routing problem, cluster-first then-route, tabu search, fuzzy c-means, clustering

1. GİRİŞ (INTRODUCTION)

İnsan, ürün ve hizmetin olduğu tüm alanlarda lojistik faaliyetler kaçınılmaz olarak gereklidir. Dolayısıyla lojistik faaliyetlerin verimli yönetimi işletmeler arasında ciddi bir avantaj farkı yaratmaktadır. Lojistik faaliyetleri temel olarak işletmenin yerinin seçilmesi, ürününün depolanması ve paketlenmesi, stok yönetimi, taleplerin ve siparişlerin yönetimi, nakliye ve dağıtım faaliyetleri olarak beş gruba ayrılmıştır [1]. Araç rotalama problemi, bir veya birden fazla depodan çıkan araçların, talep noktalarına ait talep miktarlarının, araçların taşıma kapasitesinin, depo kapasitesini dikkate alınarak taleplerin tamamını karşılamak ve başladıkları noktaya geri dönmek suretiyle mümkün olan en kısa yol kat edilerek çizelgelenmesidir [2]. Araç rotalama problemi tüm lojistik faaliyetleri içerisinde maliyetin yaklaşık % 50'sini oluşturan nakliye ve dağıtım faaliyetleri başlığı altında yer almaktadır [3]. Son yarım asır içerisinde araç rotalama probleminin çözümüne yönelik birçok çalışma yapılmış ve dönemin ihtiyaçlarına göre çeşitli çözüm yöntemleri geliştirilmiştir. Günümüzde rotalama problemleri daha karmaşıklaşmış ve çözüm üretilmesi zorlaşmıştır. Bu nedenle büyük boyutlu araç rotalama problemleri, en iyi sonucu veren kesin çözüm yöntemleri ile çözülememekte ve literatürde kabul edilebilir sonuçları kısa sürelerde veren sezgisel algoritmalar üzerinde daha fazla durulmaktadır. Günümüze kadar araç rotalama problemleri için çeşitli çözüm önerileri sunulsa bile optimal sonuç veren algoritmalarla en fazla 135 talep noktası içeren problemler çözülebilmektedir [4, 5]. Sezgisel algoritmalar ile büyük boyutlu problemler çözülebilmekte ancak algoritmanın komşu çözümleri arayacağı bir başlangıç çözümüne ihtiyaç duyulmaktadır. Kümeleme yöntemlerinden birisi olan bulanık kümeleme yönteminin, başlangıç çözümleri üretilmesinde kullanıldığı çalışmalara rastlamak mümkündür. Zarandi ve diğerleri simülasyon ile tümleşik benzetilmiş tavlama algoritması önermiş, noktalar arasındaki uzaklıkları ve talepleri bulanık kümelerle ifade etmişler ve ayrıca çözüm algoritmasının kullanacağı başlangıç çözümlerini bulanık kümelemeye dayalı bir yöntemle üretmişlerdir [6]. Sheu ise talep noktalarını gruplandırırken noktaların coğrafik konumları yerine talep ile ilgili özelliklerini dikkate almış ve bulanık kümeleme teknikleri ile üç aşamalı yönteminin ilk aşamasındaki başlangıç rotalarını elde etmiştir [7]. Wang ve diğerleri talep noktalarının birden fazla kriterini dikkate alarak lojistik ağının optimizasyonunun amaçlandığı bulanık tabanlı bir gruplandırma yaklaşımı önermişlerdir [8].

Bu çalışmada önerilen iki aşamalı yöntemin ilk aşamasında talep noktaları; bulanık c-ortalama (BCO) algoritması kullanılarak birbirlerine olan yakınlıklarına göre araç sayısı kadar kümeye, araç kapasiteleri dikkate alınmadan bölünmüş; ardından araçların kapasite kısıtını sağlamak amacıyla noktalar üyelik derecelerine göre oluşan kümeler arasında takaslanmıştır. İkinci aşamada ise elde edilen başlangıç çözümünden yola çıkarak

kapasite kısıtlarının ihlal edilmediği yeni komşu çözümler tabu arama sezgiseli ile aranmış, bir çözümden komşu çözüme geçilirken önceki takasların bir tabu listesine belirli bir süre eklenmesi sağlanarak daha önce yapılan değişikliklerin geri alınmasının önüne geçilmeye çalışılmıştır. Önerilen yöntem, Christofides ve Eiloon ile Augerat ve diğ.'nin geliştirdiği deneysel problemlerden en fazla yedi araç ve elli talep noktası içeren on yedi farklı problem [4] ile test edilmiş ve sonuçlar değerlendirilmiştir. Yapılan çalışma ile araç rotalama problemlerine ait talep noktaları, yakınlıklarına ve uygunluklarına göre gruplandırılarak her bir kümenin diğerlerinden bağımsız olarak çözülebilmesi sağlanmıştır. Böylece araç rotalama problemlerinin nispeten çözümü daha kolay olan gezgin satıcı problemine indirgenerek çözüme ulaşılabileceği gösterilmiştir.

2. ROTALAMA PROBLEMLERİ VE ÇÖZÜM YÖNTEMLERİ (ROUTING PROBLEMS AND SOLUTION METHODS)

Ürünlerin, insanların ve hizmetin üretildiği noktadan tüketildiği noktaya en verimli şekilde nasıl taşınacağı rotalama problemlerinin konusudur. Bir satıcının belirli bir noktadan yola çıkarak uğraması gereken tüm noktalara sadece bir defa uğramak ve ilk hareket noktasına tekrar dönmek kaydıyla en kısa sürede veya en kısa yol kat ederek turlaması için uygun rotanın tayini, Gezgin Satıcı Problemi (GSP) olarak literatürde yerini almıştır [9]. Araç rotalama problemi (ARP) ise aslında birden fazla satıcı içeren m-GSP'nin daha karmaşık bir türevi niteliğindedir. ARP'ye, m-GSP'de her satıcının farklı bir rota oluşturmasına ek olarak satıcıların belirli bir miktarda yük taşıyabileceği ve her noktanın talebinin farklı olabileceği kısıtları eklenmiştir. ARP, üzerinde çok çalışılan ve algoritma geliştirilen problemlerden birisidir [10]. ARP, fiziksel dağıtım ve lojistik içerisinde merkezi bir role sahip, en az bir depodan hareket edilerek müşteriler için en iyi ulaştırma veya toplama rotalarının oluşturulmasıdır [11]. ARP, çevre faktörlerinin ve taleplerin zamana bağlı olarak değişip değişmediğine, depodan hareket eden araçların rota durumlarına, kısıt ve amaç fonksiyonlarına göre üç şekilde sınıflandırılmıştır. Bunlar; çevrenin durumuna göre statik ve dinamik, rota durumuna göre açık uçlu ve kapalı uçlu, problem kısıtlarına göre zaman, maliyet, talep noktalarının birbirlerine olan uzaklıkları, araçların kapasitesi, kısıtların dinamikliği, vb. kısıtların bir tanesini veya birkaçını birlikte içeren araç rotalama problemleri olarak alt gruplara ayrılmaktadırlar [12,13]. ARP türlerinden birisi olan kapasite kısıtlı araç rotalama probleminin matematiksel modeli aşağıda verilmiştir [14].

$$enk \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \quad (1)$$

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in N \quad (2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq q_k, \quad \forall k \in V \quad (3)$$

$$\sum_{j \in N} x_{0jk} = 1, \quad \forall k \in V \quad (4)$$

$$\sum_{i \in N} x_{i0k} = 1, \quad \forall k \in V \quad (5)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in N, \forall k \in V \quad (6)$$

$$x_{ijk} \in \{0,1\}, \quad i \neq j, \forall i, j \in N, \forall k \in V \quad (7)$$

Eşit kapasiteye sahip (homojen) V adet araç, N noktanın olduğu bir sistemde x_{ijk} , k aracı ile (i,j) bağlantısının kurulup kurulmadığını belirleyen ikili değişkendir. d_i , i noktasının talebini ifade eder. Denklem (1) modelin amaç fonksiyonudur ve k adet araç ile (i,j) arasındaki bağlantıların sahip olduğu c_{ij} maliyetlerinin en küçülenmesine çalışılmaktadır. Denklem (2) her noktaya sadece bir araç ile hizmet verilebilmesini sağlayan kısıttır. Rotaların araç kapasitesi olan q_k değerini aşmaması Denklem (3) ile modele dâhil edilmiştir. Denklem (4) ve Denklem (5) ile her k aracı için depo noktasına bir gelen birde giden bağlantının yapılması sağlanmış olur. Bir diğer ifadeyle depoya yapılan bağlantı sayısı araç sayısının iki katı kadar olmalıdır. Denklem (6) ise akışı düzenleyen kısıtı ifade eder. Bir noktaya gelen araç varsa, aynı araç mutlaka o noktadan sadece bir diğer noktaya doğru ayrılmalıdır. Farklı kısıtlar eklenmesi ve amaç fonksiyonun değiştirilmesiyle zaman pencereli araç rotalama problemi, topla-dağıt araç rotalama problemi, çoklu depo araç rotalama problemi, parçalı dağıtım araç rotalama problemi, periyodik araç rotalama problemi, vb. problem türlerinin matematiksel modelleri elde edilebilmektedir.

ARP'nin çözümü için önerilen yöntemler kesin ve sezgisel yöntemler olarak iki gruba ayrılmış durumdadır. Araç rotalama problemleri kombinatoriyal optimizasyon problemleridir. Yani, en iyi çözüm, üretilecek olası rota kombinasyonlarından birisidir. Eğer tüm kombinasyonlar test edilebilirse, en iyi çözüme ulaşmak mümkündür. Bu nedenle tamsayılı programlama ARP'ne uygulanmıştır. En çok bilinen kesin yöntemler dal-sınır, dal-kesme, dal-değer algoritmaları ve tamsayılı programlama olarak karşımıza çıkmaktadır [10,14-17]. Öte yandan büyük boyutlu problemlerin tüm kombinasyonların denenmesi ve umut vadeden çözüm bölgesinin aranması süper bilgisayarla bile mümkün değildir. Çözümlere ulaşmak için çok fazla zamana ihtiyaç duyulmaktadır. Bu nedenle sonuçlara kabul edilebilir çözümler ile daha hızlı ulaşılması ihtiyacından dolayı sezgisel yöntemler geliştirilmiştir. Sezgisel yöntemler, klasik sezgiseller ve metasezgiseller olarak iki gruba ayrılmışlardır. Klasik sezgiseller; tur kurucu, tur geliştirici ve iki aşamalı sezgiseller olarak üç grupta [10], metasezgiseller ise esin kaynaklarına göre doğadan esinlenen, popülasyon tabanlı, dinamik ve komşuluk yapılarına göre dört grupta sınıflandırılırlar [18]. Tur kurucu sezgisellerden en çok

bilineni Clark ve Wright tarafından geliştirilen tasarruf algoritmasıdır [19]. Tasarruf algoritmasında, tüm talep noktalarının depodan doğrudan ziyaret edildiği varsayımından yola çıkılarak sırasıyla elde edilen tasarruf değerlerine göre rotalara yeni talep noktalarının eklenmesiyle problem çözülmeye çalışılır. Çalışmalarda başlangıç çözümlerinin oluşturulması işleminde sıklıkla kullanılmaktadır. Tur geliştirici sezgiseller rotalama işleminin ardından rotaların iyileştirilmesi amacıyla kullanılmaktadır. Rotalar arasında talep noktalarının veya bağlantıların değişmesi ile yeni çözümler elde edilir. Rota iyileştirme sezgiselleri noktaların takaslanması, noktaların aktarılması, kenarların takaslanması ve karma yöntem olarak dört başlık altında sınıflandırmıştır. İki aşamalı sezgisellerde ise, önce grupla sonra rotala veya önce rotala sonra grupla prensibi hâkimdir. Klasik sezgiseller üzerine literatürde yapılmış birçok çalışma ve literatür özeti bulunmaktadır [20-25]. Metasezgisel kavramı ilk defa Glover tarafından 1986 yılında kullanılmasıyla literatüre girmiştir. Klasik sezgisellere göre, daha iyi çözümler üretirler. Kombinatoriyal optimizasyon problemlerinde kullanılabilirler nedeniyle özelleşmiş ve hatta diğer algoritmalarla birleşerek hibritleşmiş yüzlerce türevleri bulunmaktadır. Ele alınan problemin kısıtları iyi bir şekilde adapte edilir ve problem net olarak tanımlanabilirse, metasezgisel algoritmalar iyi sonuçlar üretebilirler [26]. Metasezgiseller, çözüm uzayının araştırılması için klasik sezgisellerle farklı yöntemlerin zekice bir araya getirildiği ve optimal değere yakın sonuçların verimli bir şekilde elde edilebilmesi için öğrenme stratejilerinin kullanıldığı yinelemeli geliştirme süreçleri olarak tanımlanabilir [18]. Metasezgiselleri, klasik sezgisellerden ayıran en büyük özelliği, çözüm uzayının farklı noktalarında arama yapabileceği kabiliyetine sahip olması ve böylece yerel optimum değerlerden uzaklaşabilmesidir. En çok bilinen metasezgiseller listesiyle kısıtlı olmamakla birlikte, Adaptif Hafıza Tabanlı Arama (HTA), Tabu Arama (TA), Genetik Algoritma (GA), Benzetilmiş Tavlama (TB) ve Karınca kolonisi Optimizasyonu (KKO)'dur. Uygulandıkları problemlere göre farklı parametreler tanımlanabilmesi ve özelleştirilebilmeleri sayesinde uygulamalarda oldukça fazla yer bulmuşlardır. Metasezgiseller ve araç rotalama problemlerinde uygulamaları hakkında bilgi veren kitap bölümü ve literatür özeti türünde birçok çalışma bulunmaktadır [27-30].

3. ÖNERİLEN YÖNTEM (THE PROPOSED METHOD)

Çalışmada, sırasıyla talep noktalarının gruplandırılması, rotalanması ve iyileştirmesi adımlarının uygulandığı sezgisel bir algoritma önerilmiştir. Gruplandırma adımında bulanık c-ortalama algoritması kullanılarak, talep noktalarının birden fazla kümeye üyelikleri bulanık sayılarla ifade edilmiştir. Gruplar arasında araç kapasite kısıtları dikkate alınarak nokta takasları yapılmış ve her grup için dağıtım rotaları oluşturularak başlangıç çözümü üretilmiştir. Son adımda ise başlangıç çözümünden hareketle bir metasezgisel olan tabu arama prensiplerine

dayalı bir arama sezgiseli ile komşu çözümler üretilmiş ve rotalar iyileştirilmeye çalışılmıştır.

3.1. Bulanık C-Ortalama (Fuzzy C-Means)

BCO, kümeleme işlemlerinde yaygın olarak kullanılan, k-ortalama gibi katı kümeler algoritmalarının aksine küme elemanlarının birden fazla kümeye ait olduğu bir kümeleme algoritmasıdır [31]. 1981 yılında ise Bezdek tarafından geliştirilmiştir [32]. Talep noktaları gruplandırılırken her noktanın her kümeye üyeliği hesaplanmış, ancak üyeliği en büyük olan küme ile rotalanmıştır. Bulanık sayılar daha sonra noktaların gruplar arasında takas edilmesi esnasında en küçük üyelik değerlerine veya belli bir üyelik değerinin altında olanların başka komşu bir kümeye atanmasında kullanılmıştır. BCO, BCO, Denklem (8)'de verilen amaç fonksiyonunun minimize edilmesine dayanır [31].

$$J(X;U,V) = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m \|x_j - v_i\|^2, 1 < m < \infty \quad (8)$$

X değişkeni, $X = \{x_1, x_2, \dots, x_n\}$ noktalar kümesi iken $V = \{v_1, v_2, \dots, v_c\}$ grupların merkez noktalarının kümesidir. m parametresi noktaların bulanıklık derecesine etki eden bir parametredir. u_{ik} değişkeni, k . verinin i . kümeye üyelik derecesini gösterir ve 0 ile 1 arasında değer alır (Denklem 9). Ayrıca bir talep noktasının tüm kümelerle olan üyelikler toplamı 1 olmalıdır (Denklem 10). c , küme sayısını ifade eden bir değerdir. u_{ik} ve v_i değişkenleri aşağıda verilen Denklem 11 ve Denklem 12 ile hesaplanmaktadır. Bulanık c-ortalama kümeleme algoritmasına durma ölçütü ϵ , bulanıklaştırma parametresi m ve küme sayısı c belirlenerek başlanabilir. m kümenin bulanıklık derecesini belirlemektedir ve büyük m değeri fonksiyonun bulanıklığını artırmaktadır. Kümeleme algoritması ancak ϵ gibi bir değere ulaşıldığında durmaktadır. Aksi halde sürekli yeni v_i , u_{ij} değerlerini hesaplanarak sonuç iyileştirilmeye çalışılacaktır. Durma ölçütü 0,01 gibi küçük bir sayıdır ve r değeri algoritmanın iterasyon sayısını ifade eder. BCO yönteminin adımları Tablo 1'de verildiği gibidir [31].

$$u_{ij} \in [0,1], \quad 1 \leq i \leq c, \quad 1 \leq j \leq N \quad (9)$$

$$\sum_{i=1}^c u_{ij} = 1, \quad 1 \leq j \leq N \quad (10)$$

$$u_{ik}^{r+1} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_j - v_i^{r+1}\|}{\|x_j - v_k^{r+1}\|} \right)^{\frac{2}{m-1}}}, \quad 1 \leq i \leq c, \quad 1 \leq j \leq N \quad (11)$$

$$v_i^{r+1} = \frac{\sum_{j=1}^N (u_{ij}^r)^m (x_j)}{\sum_{j=1}^N (u_{ij}^r)^m}, \quad 1 \leq i \leq c \quad (12)$$

3.2 Tabu Arama (Tabu Search)

1989 yılında Glover tarafından önerilen ve adı yasak anlamındaki "taboo" kelimesinden gelen Tabu Arama (TA) algoritmasının bu isimle anılmasının nedeni içerdiği yasaklama prosedürüdür. TA kombinatoriyal optimizasyon problemlerinde iyi sonuçlar vermektedir ve araç rotalama problemlerinin çözümünde genetik algoritma, benzetilmiş tavlama, karınca kolonisi, yapay sinir ağları gibi yöntemlerden daha iyi sonuçlar vermektedir [33]. Bu nedenle çalışmada rotaların iyileştirilmesinde tabu arama prensiplerine dayanan bir komşu çözüm arama algoritması kullanılmıştır. Tabu arama algoritmasında başlangıç çözümünün tüm komşuları öncelikle komşuluk dizisine kaydedilir. Komşu çözümlerden en iyi sonuç veren çözüm seçilir ve mevcut çözümden komşu çözüme geçilebilmesi için yapılacak hareketin yasak (tabu) olup olmadığını kontrol ettikten sonra eğer hareket tabu değilse komşu çözümü mevcut çözüm olarak değiştirilir. Hareket tabu olduğu halde, elde edilen en iyi sonuca ulaşıyorsa tabu durumu göz ardı edilir. Bu göz ardı etme olayına tabu yıkma (aspirasyon) adı verilir. Tabu yıkma ve yasaklama kriterleri TA algoritmasının başında tanımlanmaktadır. Bu döngü iteratif bir şekilde durma kriteri sağlanana kadar devam ettirilir. TA adımları Tablo 2'de verilmiştir [34].

Tabu arama algoritmasında S_{eni} yi çözümü geliştirilemediğinde aramaya amaç fonksiyonunu en az etkileyen kötü çözüm ile devam edilmektedir. Böylece yerel optimumlardan kurtulmak amaçlanmaktadır. Öte yandan TA'da komşu çözümler aranırken daha önce ulaşılmış çözümlere tekrar ulaşılabilir. Bir kısır döngü oluşmasını engellemek için önceki çözüme götüren hareketler tabu listesine eklenir ve bu çözüme ulaşılması engellenmiş olur. Çözüm problemin elemanlarının farklı dizilimleri arasında aranmaktadır. Her bir dizilim $N(s)$ komşu çözümler kümesinin elemanı ise, $(N_{syeni}) = H$ ($N_{smevcut}$) arasında geçiş H ile verilen bir hareket ile gerçekleşir. Komşuluk çözümlerinin elde edilmesi için yapılacak H hareketi için çeşitli yaklaşımlar mevcuttur. Çalışmanın önceki bölümlerinde bu yöntemlerden tur geliştirici sezgiseller (nokta takası, kenar takası, nokta atama, karma yöntemler) olarak bahsedilmiştir. TA'da, bazı sezgiseller gibi komşu çözümleri rassal olarak üretmek yerine verimliliği ispatlanmış yöntemler kullanılmaktadır.

Tabu arama algoritmasındaki bir diğer önemli farklılık ise tabu listesinin varlığıdır. Tabu listesi daha önce üretilmiş çözümlere tekrar dönmeyi veya çözüm uzayının iyi sonuçlar vermeyen bölgelerinde arama yapmayı

engellemek amacıyla kullanılır. H hareketi eğer tabu listesinde yer alıyorsa bu hareket aspirasyon kriteri adı verilen şartın sağlanması dışında gerçekleştirilmez.

Aspirasyon kriteri, bir H hareketinin hangi şartlar dâhilinde tabu listesinden çıkartılacağını belirler. Ancak, tabu listesinin çok uzun veya çok kısa olmaması gerekir.

Tablo 1. Bulanık c-ortalama algoritmasının adımları
(The steps of fuzzy c-means algorithm)

Adım 1	Küme sayısı (c), ağırlık değeri (m), ve durma ölçütü ε belirle.
Adım 2	$r = 0$ iterasyonda $U_{(c \times N)}$ üyelik matrisini Denklem (9) ve (10) dikkate alarak rassal olarak oluştur.
Adım 3	İterasyon değerini artır $r = r + 1$.
Adım 4	V_i^{r+1} değerini hesapla.
Adım 5	u_{ij}^{r+1} değerini hesapla.
Adım 6	$\ U^{r+1} - U^r\ \leq \varepsilon$ ise dur. Aksi halde Adım 3'e git.

Tablo 2. Tabu arama algoritmasının adımları.
(The steps of tabu search algorithm)

Adım 1	Bir başlangıç çözümü (S) al. Algoritmada kullanılacak tabu listesi, tabu süresi, Tabu yıkma (aspirasyon kriteri), vb. parametreleri belirle.
Adım 2	Bir komşu çözüm üretme yöntemini kullanarak S'ye ait komşu çözümleri üret ve tabu listesinde olmayan tüm (S1 N(S)) komşu çözümler arasından en iyi kabul edilebilir olan ($S_{eni\tilde{y}i}$) olarak seç.
Adım 3	Mevcut çözümü (S), $S_{eni\tilde{y}i}$ ile yer değiştir ve S'den $S_{eni\tilde{y}i}$ 'ye geçmek için yapılan hareketi tabu listesine ekle.
Adım 4	Durdurma kriteri sağlanıncaya kadar Adım 2 ve 3'ü tekrar et.

Çok uzun bir liste tanımlanmışsa belirli bir süre iterasyondan sonra birçok hareket tabu listesine girecek ve farklı komşu çözümlere ulaşamayacaktır. Kısa olduğu durumlarda ise döngülerin oluşması ve işlem süresinin uzaması söz konusu olabilecektir. Tabu arama algoritmasında var olan stratejiler aşağıda verilmiştir [34]:

- Tabu listesine eklenecek çözümleri belirleyen yasaklama stratejisi,
- Tabu listesinden çözümlerden hangisinin ne zaman çıkacağını belirleyen serbest bırakma stratejisi
- Denenecek hareketlerin seçimi için yukarıdaki stratejiler arasında ilişkiyi kontrol edecek kısa dönem stratejisi.

Tabu arama algoritması araç rotalama problemlerinin çözümünde çok fazla çalışmada kullanılmıştır. TA prensipleri kullanılarak yapılan çalışmalar ve geliştirilen en önemli yöntemler, Taburoute algoritması [35], Taillard'ın algoritması [36], Rochat ve Taillard'ın Adaptif Hafıza Prosedürü [37], Xu and Kelly algoritması [38], Barbarosoğlu ve Özgür algoritması [39] ve Bütünleşik Tabu Arama algoritmasıdır [40].

3.3. Önerilen Yöntemin Adımları (Steps of Proposed Method)

Önerilen yöntem talep noktalarının gruplandırılması, başlangıç çözümünün elde edilmesi ve başlangıç çözümünün iyileştirilmesi adımlarını içermektedir. Noktaların gruplandırılması ile, N nokta içeren bir

kümede rota araması yapmak yerine, noktaları c adet alt kümeye ($n_1 \in N, n_2 \in N, \dots, n_c \in N$) bölerek daha az nokta üzerinden hızlı ve etkili bir rota araması yapılması amaçlanmıştır. İlk olarak, BCO ile noktaların koordinat düzlemindeki (x,y) değerleri kullanılarak, noktalar birbirlerine olan uzaklıklarına göre gruplandırılırlar. Kümeleme işleminde kapasite kısıtları dikkate alınmadığından, her bir grupta yer alan noktaların talepleri toplanır ve araç kapasitesinin aşıldığı gruplarda, gruba en az ait olan nokta veya noktalar, diğer gruplar arasından en yüksek ait olduğu diğer komşu gruplara aktarılır. Bu takas işlemi tüm gruplardaki noktaların talepler toplamı araç kapasitesinden küçük veya eşit olana kadar devam edecektir. Takas işlemi bittiğinde gruplardaki noktaların toplam talepleri her bir gruba atanacak olan aracın toplam taşıma kapasitesinden büyük olmayacaktır. Gruplandırma adımından sonra bir başlangıç çözümü üretmek için tur kurucu sezgisel algoritmalarla yararlanılarak başlangıç turu elde edilir. Çalışmada tur kurucu sezgisel olarak tasarruf algoritmasından yararlanılmıştır. Her grup için depodan başlayan ve depoda sonlanan bir tur elde edildikten sonra iyileştirme algoritmaları ile her rota kendi içerisinde iyileştirilir. Bu adımda 2-opt, 3-opt, gibi iyileştirme yöntemleri kullanılmaktadır. Burada amaç tasarruf algoritması ile elde edilen rotadan daha kısa bir rota olup olmadığı araştırılmaktadır.

Başlangıç rotası elde edildikten sonra rotanın iyileştirilmesi adımına geçilecektir. Bu adımda tabu

arama sezgiselinden yararlanılmıştır. İlk adımda elde edilen çözüm tabu arama sezgiselinin kullanacağı başlangıç çözümü ve en iyi çözüm olarak atanmaktadır. Ayrıca tabu arama algoritmasında kullanılacak durma kriteri, yasaklama stratejisi, serbest bırakma stratejisi gibi parametreler ve hangi noktanın diğer gruplara aktarılacağını belirleyen üyelik değer sınırı tanımlanır. Gruplanmış talep noktaları, nokta takası, kenar takası gibi çoklu rota geliştirme algoritmaları ile gruplar arasında takaslanarak ve araçların taşıma kapasitelerinin aşılmasına dikkat edilerek komşu çözümler üretilir. Komşu çözümler üretilirken belirlenen üyelik değer sınırının altında bir üyelik derecesiyle gruba ait olan noktalar arasından seçilen noktalar veya her bir grupta en az üyelik değerine sahip olan nokta komşu gruplardan birisine aktarılmaktadır. Bir gruptan ayrılan veya gruba dâhil olan noktalar tur kurucu sezgisellerden sıralı ekleme

sezgiseli ile rotaya eklenerek rotalar güncellenir. Aktarılan noktalar yapılan değişikliğin geri alınmasını önlemek amacıyla tabu listesine alınır ve serbest bırakma stratejisinde belirtildiği şekilde tabu listesinde tutulur. Her bir takas işleminden sonra tabu listesi güncellenir. Üretilen komşu çözüm eğer mevcut çözümden iyi ise mevcut çözüm olarak atanır ve durma kriteri sağlanana kadar yeni çözümler aranır. Eğer elde edilen sonuç mevcut çözümden kötü ise geçici çözüm olarak atanır ve belirlenmiş bir iterasyon sayısı kadar kötü çözümden yeni komşu çözümler aranır. Belirlenmiş iterasyon sayısına kadar mevcut çözüm geçilemezse elde edilen çözümler silinerek mevcut çözüme dönlür ve yeni bir komşu çözüm üretilerek işlem devam ettirilir. Durma kriterine ulaşıldığında uzun dönem hafıza içerisinde yer alan en iyi çözüm, sonuç olarak aktarılacaktır. Tablo 3 ve Tablo 4’de önerilen yöntemin algoritmaları verilmiştir.

Tablo 3. BCO ile başlangıç çözümünün oluşturulması adımları
(The steps of generating initial solution by FCM)

Adım 1	Parametreleri belirle. (küme sayısı; c , araç kapasitesi; q_k ve m değeri)
Adım 2	BCO ile noktaları kümele. Üyelik değerlerini $A[i][j]$, matrisine ekle (i kümeler j talep noktaları).
Adım 3	Her i kümesindeki noktaların taleplerinin q_k değerini aşıp aşmadığını kontrol et. Aşmıyorsa Adım 6’ya geç.
Adım 4	Talepler toplamı ile q_k farkı en fazla olan kümenin üyeliği en az olan noktayı veya belli bir üyelik değerinin altında (f) olan noktaları bul ve takas edilebilir olarak işaretle.
Adım 5	İşaretlenen noktalardan seçilenleri, üyeliği yüksek olduğu diğer bir kümeye ata. İşaretlenen noktalardan depoya uzak köşelerde olan noktaların aktarılmasını engellemek amacıyla aday noktalarının aktarılacağı küme merkezi ile depo arasındaki tasarruf değerlerini hesapla. Tasarruf değeri en küçük olanlara öncelik ver.
Adım 6	Yeni durumu güncelle. Adım 3’e dön.
Adım 7	Kümeleme işlemini sonlandır.
Adım 8	Her küme için tur kurucu sezgiseller (tasarruf, en yakın komşuluk, vb.) ile rota oluştur. Rotayı λ -opt yöntemi ile iyileştir. Elde edilen çözümü başlangıç çözümü olarak belirle.

Tablo 4. Tabu arama ile rotanın iyileştirilmesi adımları
(The steps of route improvement by tabu search)

Adım 1	Başlangıç çözümünü mevcut rota ve en iyi rota olarak ata (M_{Rota} ve E_{Rota}). Parametreleri belirle. (durma kriteri, yasaklama stratejisi, geçici çözüm iterasyon sayısı, takaslanacak noktaları için f üyelik değer sınırı, $K_{sınır}$)
Adım 2	Durma kriteri sağlanıyorsa Adım 11’e geç. Rassal olarak f değerinin altında bir nokta seç ve üyeliğin yüksek olduğu diğer bir kümeye ata. Tabu matrisini güncelle (küme i den j ye yapılan hareketi tut).
Adım 3	Takas edilen noktayı sıralı ekleme sezgiseli ile yeni kümenin mevcut rotasına ekle. Kümenin q_k değerini aşıp aşmadığını kontrol et. Aşmıyorsa noktanın aktarıldığı eski kümeyi tasarruf algoritması ile güncelle ve Adım 7’ye atla.
Adım 4	Kapasite kısıtını sağlamayan kümenin f üyelik değerinin altında olan noktalarını bul ve takas edilebilir olarak işaretle.
Adım 5	Kapasite kısıtı sağlanana kadar üyelik değeri en düşük olan noktaları komşu kümelere ata.
Adım 6	Yasaklama stratejisine göre tabu listesini güncelle. Durma kriteri sağlanmıyorsa Adım 3’e dön. Aksi halde Adım 11’e atla.
Adım 7	Çözümü komşu çözüm olarak kabul et. Rotaları kendi içerisinde λ -opt yöntemi ile iyileştir.
Adım 8	Rota uzunluklarını M_{Rota} ile kıyasla. Eğer daha iyi bir çözüme E_{Rota} ve M_{Rota} ‘yı yeni rota ile değiştir. Adım 2’ye dön.

Adım 9	Kötü çözümü MRota olarak belirle ve kötü çözüm deneme sayacını $K_{iter} = K_{iter} + 1$ olarak güncelle. K_{iter} değeri kötü çözüm deneme sınırından $K_{sınır}$ küçük ise Adım 2'ye dön.
Adım 10	M_{Rota} değerini E_{Rota} olarak değiştir. Tabu listesini boşalt. $K_{iter} = 0$ olarak güncelle. Adım 2'ye dön.
Adım 11	İşlemi durdur. E_{Rota} 'yı sonuç olarak yansıt.

4. UYGULAMA VE BULGULAR (IMPLEMENTATION AND FINDINGS)

Önerilen yöntem 1969 yılında Christofides ve Eilon ve 1995 yılında Augerat ve ark. tarafından sunulmuş KKARP veri setlerinden [4] elli talep noktasına ve yedi araca kadar olanlarından on yedi problem ile test edilmiştir. Bu çalışmada, C++ programlama dili kullanılarak geliştirilen yazılım ile problemlerde yer alan talep noktaları BCO ile kümelenecek ve başlangıç rotaları oluşturulmuştur. Ardından komşu çözümlerin aranmasıyla başlangıç rotası iyileştirilmiş ve yeni rotalar elde edilmiştir. Elde edilen sonuçlar Tablo 5'de sunulmuştur.

Yöntem ile ele alınan on yedi problemde on tanesinin en iyi çözümüne ulaşılmış kalan yedi tanesinde ise yaklaşık % 0.8 ile %10.7 oranında en iyi çözümlerden uzak çözümler elde edilmiştir. Küme (araç) sayısı arttıkça en iyi çözümden sapma oranında da artış gerçekleşmiştir. Özellikle depo merkezlerinin kümelerin ortasında yer alması durumunda optimum ya da optimuma yakın çözümler elde edilmektedir. Depo merkezleri küme sınırlarına doğru kaydıka, çözüm kalitesinde düşüş başlamaktadır. Bu durumun daha net anlaşılabilmesi için Şekil 1'de temsili bir örnek verilmiştir. Örnekte depo merkezi sol üst köşede yer almaktadır. Noktaların üç kümeye dâhil olacak şekilde BCO ile kümelendiğini, 9-10-11, 1-2-3-4 ve 5-6-7-8 nokta gruplarının aynı kümeyi paylaştıklarını ve ikinci kümedeki 1-2-3 noktalarının araç kapasitesini doldurduğunu varsayalım. Bu durumda 4. noktanın başka bir kümeye aktarılması gerekecektir. Önerilen yöntemin bulanık üyelik derecesine göre

kümeler arası takas yapılması mantığından dolayı 4. noktanın aktarılacağı küme kendisine en yakın olan ve üyelik derecesinin yüksek olacağı 1. küme olacaktır. Hâlbuki 4. nokta 3. kümeye uzak olmasına rağmen, kümeye servis veren aracın rotası noktanın oldukça yakınından geçmektedir. 4. noktanın 3. kümeye aktarılmasıyla iyi bir çözüm elde edilebilir iken önerilen yöntemde üyelik derecesine göre takas yapılması nedeniyle bu çözüme ulaşılamamaktadır. Bu durum, araç rotalama problemlerinin çözümünde kullanılan pek çok yöntemde olduğu gibi, önerilen yöntemde de talep noktalarının sadece birbirlerine olan uzaklıklarının değerlendirilerek çözüm aranması nedeniyle ortaya çıkmıştır. Yöntemde, talep noktaları arasındaki uzaklıklar dikkate alınmakta ve yakın noktalar aynı grupta toplanmaktadır. Hâlbuki her bir araç rotasının diğer tüm rotalara göre konumu da önemlidir. Bu nedenle bazı problemlerde en iyi sonuca ulaşılmış, geri kalanında ise en iyi sonuçtan % 10'a kadar daha uzun rotalar önerilmiştir. İlgili küme ile deponun tam arasında konumlanmış olan noktaların kümeye uzak (aitliği düşük) olsa dahi çözüme dâhil edilmesiyle daha iyi rotaların elde edilebileceği fark edilmiştir. Ayrıca, BCO kümeleme adımında yer alan durma ölçütü (ϵ) ve rota iyileştirme adımında takas edilecek talep noktalarını belirlemek için kullanılan üyelik sınırı (f) parametreleri kümeleme algoritmasının çözüm süresini doğrudan etkilemektedirler. ϵ değeri azaltıldıkça daha hassas kümeleme yapılabilen, ancak işlem süresi uzamaktadır. Yine 0-1 arasında değer alan f parametresi 1 değerine yaklaştıkça daha fazla nokta arasından komşu çözüm üretilmekte, ancak çözüme daha uzun sürelerde ulaşılmaktadır.

Tablo 5. Önerilen yöntem ile elde edilen sonuçlar
(The results by the obtained method)

Problem	Talep Noktası Sayısı	Araç Sayısı	En İyi Sonuç	Çalışmada Elde Edilen Sonuç	Sapma Oranı (%)
A-n32-k5	31	5	784	810	3,32
A-n33-k5	32	5	661	661	0
A-n33-k6	32	6	742	742	0
A-n34-k5	33	5	778	778	0
A-n36-k5	35	5	799	806	0,88
A-n37-k5	36	5	669	669	0
A-n37-k6	36	6	949	975	2,74
A-n39-k5	38	5	822	822	0
A-n39-k6	38	6	831	863	3,85
A-n44-k6	43	6	937	937	0
A-n46-k7	45	7	914	1012	10,72

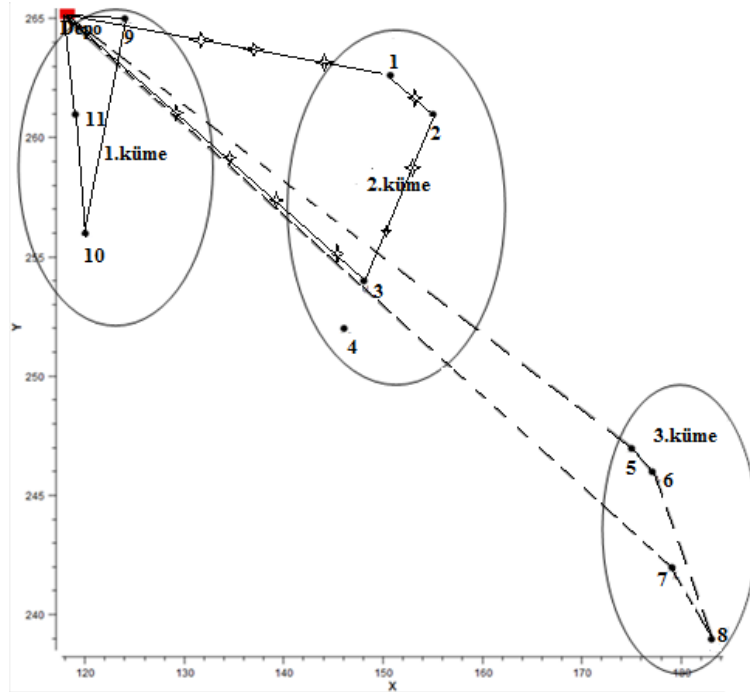
A-n48-k7	47	7	1073	1158	7,92
E-n22-k4	21	4	375	375	0
E-n23-k3	22	3	569	569	0
E-n30-k4	29	4	503	503	0
E-n33-k4	32	4	835	880	5,38
E-n51-k5	50	5	521	521	0

5. SONUÇLAR VE GELECEK ÇALIŞMALAR (CONCLUSIONS AND FUTURE WORKS)

Kombinatoriyal optimizasyon problemleri ile ilgilenilen erken dönemlerde problem boyutlarının nispeten küçük olması nedeniyle kesin sonuçlar elde edilebilirken günümüzde problemlerin büyümesiyle karmaşıklıkları artmıştır. Bu problemlerden araç rotalama problemi iyi bilinen ve çözümü zor bir problemidir. Sezgisel algoritmaların araç rotalama problemlerini çözmedeki başarısı nedeniyle bu çalışmada yeni bir iki aşamalı sezgisel çözüm yaklaşımı önerilmiştir. ARP'nin talep noktaları birbirlerine olan yakınlıkları dikkate alınarak daha küçük gruplara ayrılmış ve her bir araç rotası tabu arama sezgiseliyle optimize edilmiştir. Talep noktaları BCO ile araç kapasite kısıtları dikkate alınarak gruplara ayrıldıktan sonra her bir aracın rotası GSP gibi çözülebilir hale gelmekte ve kapasite kısıtlarının dikkate alınması gerekmemektedir. Önerilen yöntemde BCO kullanılmasının getirdiği diğer bir avantaj ise komşu

çözümlerin aranması esnasında ortaya çıkmaktadır. Komşu çözümleri rassal olarak oluşturmak yerine noktaların üyelik derecesi yüksek olduğu kümelere atanmasıyla, çözüm kümesinde daha zeki bir arama yapılabilmektedir.

Önerilen yöntem, literatürde yer alan kapasite kısıtlı araç rotalama problemleri için önerilmiş test problemleri ile test edilmiş ve bazı problemler için en iyi sonuçlara ulaşılmıştır. Bazı problemlerde en iyi sonuca ulaşılama nedeni, sadece noktalar arası uzaklıkların dikkate alınması ve bazı noktaların araç rotasına yakın olduğu halde, küme üyelik değerlerinin düşük olması nedeniyle rotaya eklenememesinden kaynaklanmaktadır. Gelecekteki çalışmalarda bu durumu ortadan kaldırmak üzere rotaların geçtiği bölgelere yakın olan noktalarında rota iyileştirme adımı dikkate alınması sağlanabilir. Çalışma zamanını etkileyen ε ve f parametrelerinin optimum değer aralıklarının belirlenerek çözüm süresini uzatmadan sonuçlar elde edilmesi üzerine de çalışmalar yapılabilir.



Şekil 1. Önerilen yöntemin zayıf noktası
(The weakness of the proposed method)

KAYNAKLAR (REFERENCES)

- [1] Ö.B. Tek, E. Ozgul, **Modern Pazarlama İlkeleri: Uygulamalı Yönetimsel Yaklaşım**, 3.Baskı, Birleşik Matbaacılık, İzmir, 2010.
- [2] K.C. Tan, "A Framework of Supply Chain Management Literature", *European J. of Purchasing & Supply Management*, 7(1), 39-48, 2001.
- [3] A. Rushton, P. Croucher, P. Baker, **The Handbook of Logistics and Distribution Management**, 4rd ed., Kogan Page, London, 2006.
- [4] M.H.F. Zarandi, M. Hossein, S. Davari, I.B. Turksen, "Capacitated Location-Routing Problem with Time Windows Under Uncertainty." *Knowledge-Based Systems*, 37, 480-489, 2013.
- [5] J.B. Sheu, "A Hybrid Fuzzy-Optimization Approach to Customer Grouping-Based Logistics Distribution Operations", *Applied Mathematical Modelling*, 31(6), 1048-1066, 2007.
- [6] L.T. Hu, J.B. Sheu, "A fuzzy-based customer classification method for demand-responsive logistical distribution operations", *Fuzzy Sets and Systems*, 139(2), 431-450, 2003.
- [7] D. Applegate, R. Bixby, V. Chavátal, W. Cook, "On the Solution of Traveling Salesman Problems", *Documenta Mathematica*, 645-656, 1998.
- [8] P. Toth, D. Vigo, "An Overview of Vehicle Routing Problems", **The Vehicle Routing Problem**, Editör: P. Toth, D. Vigo, SIAM, Philadelphia, 1-26, 2002.
- [9] Ç. Alabaş, B. Dengiz, "Yerel Arama Yöntemlerinde Yöre Yapısı: Araç Rotalama Problemine Bir Uygulama", **Yöneyim Araştırmaları ve Endüstri Mühendisliği XXIV. Ulusal Kongresi**, Çukurova Üniversitesi, Adana, 333-335, 15-18 Haziran, 2004.
- [10] Toth, P., Vigo, D, **The Vehicle Routing Problem**, Society for Industrial and Applied Mathematics, SIAM Monographs On Discrete Mathematics And Applications, Philadelphia, 2002.
- [11] E. Vural, **Araç Rotalama Problemleri İçin Popülasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı Ve Uygulanması**, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2006.
- [12] P. Toth, D. Vigo, "Exact Solution of the Vehicle Routing Problem", **Fleet Management and Logistic**, Editör: T.G. Crainic, G. Laporte, Kluwer Academic, Boston, 1-31, 1998.
- [13] M. Dell'Amico, G. Righini, M. Salani, "A Branch-And-Price Approach to the Vehicle Routing Problem with Simultaneous Distribution and Collection", *Transportation Science*, 40(2), 235-247 2006.
- [14] P. Toth, D. Vigo, "An Exact Algorithm for the Vehicle Routing Problem with Backhauls", *Transportation Science*, 31(4), 372-385, 1997.
- [15] Z. Başkaya, B. Avcı Öztürk, "Tamsayılı Programlamada Dal Kesme Yöntemi ve Bir Ekmek Fabrikasında Oluşturulan Araç Rotalama Problemine Uygulanması", *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 24(1), 101-114, 2005.
- [16] C.D. Tarantilis, G. Ioannou, G. Prastacos, "Advanced Vehicle Routing Algorithms for Complex Operations Management Problems", *Journal of Food Engineering*, 70(3), 455-471, 2005.
- [17] H. Paessens, "The Savings Algorithm for the Vehicle Routing Problem", *European Journal of Operational Research*, 34(3), 336-344, 1988.
- [18] G. Clarke, J.W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, 12(4), 568-581, 1964.
- [19] A.V. Breedam, "Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem", *Computers & Operations Research*, 28(4), 289-315, 2001.
- [20] G. Laporte, M. Gendreau, J.Y. Potvin, F. Semet, "Classical and Modern Heuristics for the Vehicle Routing Problem", *International Transactions in Operation Research*, 7(4-5), 285-300, 2000.
- [21] I.K. Altinel, T. Öncan, "A New Enhancement of the Clarke and Wright Savings Heuristic for the Capacitated Vehicle Routing Problem", *Journal of the Operational Research Society*, 56(8), 954-961, 2005.
- [22] G. Laporte, F. Semet, "Classical Heuristics for the Vehicle Routing Problem", **The Vehicle Routing Problem**, Editör: P. Toth, D. Vigo, SIAM, Philadelphia, 109-128, 2002.
- [23] A. Hertz, M. Widmer, "Guidelines for the Use of Meta-Heuristics in Combinatorial Optimization", *European Journal of Operation Research*, 151(2), 247-252, 2003.
- [24] M. Gendreau, G. Laporte, J.Y. Potvin, "Metaheuristics for the Capacitated VRP", **The Vehicle Routing Problem**, Editör: P. Toth, D. Vigo, SIAM, Philadelphia, 129-154, 2002.
- [25] B. Eksioğlu, A.V. Vural, A. Reisman, "The Vehicle Routing Problem: A Taxonomic Review", *Computers & Industrial Engineering*, 57(4), 1472-1483, 2009.
- [26] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, A. Chatterjee, **Metaheuristics for Hard Optimization: Methods and Case Studies**, 1th ed., Springer, Berlin, 2006.
- [27] T. Vidal, T.G. Crainic, M. Gendreau, C. Prins, "Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis", *European Journal of Operational Research*, 231(1), 1-21, 2013.
- [28] R. Kruse, C. Borgelt, D. Nauck, "Fuzzy Data Analysis: Challenges and Perspectives", **IEEE International Conference on Fuzzy Systems**, Seoul, South Korea, 22-25 Ağustos, 1211-1216, 1999.
- [29] W.C. Chen, M.S. Wang, "A Fuzzy C-Means Clustering-Based Fragile Watermarking Scheme for Image Authentication", *Expert System with Application*, 36(2), 1300-1307, 2009.
- [30] J.F. Cordeau, M. Gendreau, G. Laporte, J.Y. Potvin, F. Semet, "A Guide to Vehicle Routing Heuristics", *Journal of the Operational Research Society*, 53(5), 512-522, 2002.
- [31] D. Karaboğa, **Yapay Zekâ Optimizasyon Algoritmaları**, 2. baskı, Nobel Yayın Dağıtım, Ankara, 2011.
- [32] M. Gendreau, A. Hertz, G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem", *Management Science*, 40(10), 1276-1290, 1994.
- [33] E.D. Taillard, "Parallel Iterative Search Methods for Vehicle Routing Problem", *Networks*, 23(8), 661-673, 1993.
- [34] Y. Rochat, E.D. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing", *Journal of Heuristics*, 1(1), 147-167, 1995.
- [35] J. Xu, J.P. Kelly, "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem", *Transportation Science*, 30(4), 379-393, 1996.
- [36] G. Barbarosoğlu, D. Özgür, "A Tabu Search Algorithm for the Vehicle Routing Problem", *Computers & Operations Research*, 26(3), 255-270, 1999.
- [37] J.F. Cordeau, M. Gendreau, G. Laporte, "A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems", *Networks*, 30(2), 105-119, 1997.
- [38] Internet: Vehicle Routing Data Sets, <http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old#E>, 14.01.2014.