



## Çok Bölмелі Araç Rotalama Problemi için Bir Melez Genetik Algoritma

### A Hybrid Genetic Algorithm for Multi-Compartment Vehicle Routing Problem

Kazım Erdoğdu<sup>1\*</sup> 

<sup>1</sup> Yaşar Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, İzmir, TÜRKİYE  
Sorumlu Yazar / Corresponding Author \*: [kazim.erdogdu@yasar.edu.tr](mailto:kazim.erdogdu@yasar.edu.tr)

Geliş Tarihi / Received: 27.05.2020

Araştırma Makalesi/Research Article

Kabul Tarihi / Accepted: 10.07.2020

DOI:10.21205/deufmd.2021236706

Atıf Sekili / How to cite: ERDOĞDU, K.(2021). Çok Bölмелі Araç Rotalama Problemi için Bir Melez Genetik Algoritma. DEUFMD, 23(67), 71-80.

#### Öz

Bu çalışmada, Çok Bölмелі Araç Rotalama Problemi (ÇB-ARP) ele alınmıştır. Günlük hayatı marketler, firmalar ve kurumlar bazı ürünlerini müşterilerine teslim ederken ya da belirli noktalardan toplarken, bu ürünleri araç içinde farklı bölmelere koymaları gerekmektedir. Bazı ürünlerin oda sıcaklığında, bazlarının soğuk olarak taşınması gerekmektedir. Bazı atıkların, kimyasal ürünlerin ya da yakıtların diğer ürünlerle karıştırılmadan taşınması gerekmektedir. Bu yüzden dağıtım ya da toplama yapan araç filosundaki her bir aracın birden fazla bölmeye sahip olması ve dağıtılan ya da toplanan ürünlerin ilgili bölmelerde taşınması gerekmektedir. Bu makalede çalışılan ÇB-ARP, bir, iki ve üç bölmeli araç senaryoları dahilinde ayrı ayrı ele alınmıştır. Çözüm yöntemi olarak melez bir Genetik Algoritma (GA) kullanılmış ve bu algoritma Araç Rotalama Problemi (ARP) literatüründe sıkılıkla kullanılan bir problem örnek seti üzerinde uygulanmıştır. Sonuç olarak bu çalışmada ÇB-ARP modeli için yeni referans sonuçları üretilmiş ve sonuçlar yorumlanmıştır.

**Anahtar Kelimeler:** Çok Bölмелі Araç Rotalama Problemi, Melez Genetik Algoritma, Sezgisel Arama

#### Abstract

In this study, a Multi-Compartment Vehicle Routing Problem (MC-VRP) was studied. In daily life, markets, companies and organizations need to load their certain products into different compartments in vehicles during the delivery or collecting of these products. Some of the products need to be carried in cold temperature while others in ambient temperature. Some wastes, chemical products or fuel types need to be transported in the same vehicle separately. Therefore, the transportation fleet must have vehicles with separate compartments and the different products must be either delivered or collected in related compartments of these vehicles. The MC-VRP studied in this paper was taken into consideration with three scenarios of vehicles with one-, two-, or three-compartments. As the solution method, a hybrid Genetic Algorithm (GA) was used and applied on a well-known problem instance set in Vehicle Routing Problem (VRP) literature. As a result, new benchmark results were obtained for the version of MC-VRP in this study and these results were evaluated.

**Keywords:** Multi-Compartment Vehicle Routing Problem, Hybrid Genetic Algorithm, Heuristic Search

## 1. Giriş

Araç Rotalama Problemi (ARP)'nin bir türü olan Çok Bölмелii Araç Rotalama Problemi (ÇB-ARP), lojistik, nakliye, dağıtım, toplama ve tedarik zinciri alanlarında sıkça karşılaşılan bir problemdir. ÇB-ARP'de, aynı klasik ARP'de olduğu gibi, belirli sayıda müşterilerin talepleri mevcut araç filosu aracılığıyla tedarik edilmeye çalışılır. Klasik ARP'ye göre fark, araçların farklı türde talepleri birbirinden ayrı şekilde taşıyabildikleri bölmelere sahip olmasıdır. Araçlar depodayken, müşterilerin farklı türdeki talepleri, araçların ilgili bölmelerine yüklenir. Yüklü araçlar depodan yola çıkarak sırasıyla müşterilerine taleplerini dağıtır ve dağıtma işini bitirdiklerinde depoya geri dönerler. ÇB-ARP'de amaç, tüm araçların neden olduğu toplam maliyeti en aza indirgemektir. Bu maliyet, klasik ARP'deki gibi toplam dolaşilan mesafe olabildiği gibi, harcanan zaman, tüketilen yakıt miktarı, finansal maliyetler (ücretler, vergiler, masraflar, vb.) veya çevre kirliliğine neden olan gazlar da olabilmektedir. Problemin amaç fonksiyonu, üzerinde çalışılan problemin amacına göre belirlenmektedir.

ÇB-ARP'nin oldukça geniş uygulama alanı vardır. Örneğin bazı market zincirleri, ana depodan bayilerine veya müşterilerine ürünlerini taşırken, farklı sıcaklıklarda (özellikle soğuk ve dondurulmuş olarak) taşınması gereken ürünleri aracın ayrı bölmelerine koyarak taşımaları gerekmektedir. Ya da, benzin, mazot, vb. farklı türden yakıtların merkezden diğer yakıt istasyonlarına nakliyesinde kullanılan araçlar her bir yakıt türü için ayrı bölmeye içermektedir. Diğer bir örnek olarak ise tıbbi atıkların ya da geri dönüşüm atıklarının ayrı bölmelere sahip araçlarla toplanması gösterilebilir. Bunlara benzer birçok alanda ÇB-ARP önemli rol oynamakla birlikte yaygın uygulama alanına sahiptir.

### 1.1. Literatür taraması

ÇB-ARP'in ilk çalışmaları Christofides ve ark. [1] tarafından 1979 yılında ortaya atılmış olmasına rağmen, problemin bu alanda çalışan araştırmacıların dikkatini çekmesi daha geç dönemde olmuştur. 2008 yılında Fallahi ve ark. [2] problemin matematiksel modelini ortaya koymuş ve probleme iki bölümlü araçlar için çözüm önerilerinde bulunmuştur. Çalışılan problemde müşterilerin sadece iki tür ürün talebinde bulundukları varsayılmış ve araçların

her bir bölmesi ilgili ürün tipi için ayrılmıştır. Falahi ve ark. problemlerinde iki senaryo ele almışlardır. İlkinde araçların iki bölmesi de eşit kapasitedeyken, ikinci senaryoda bu kapasiteler rastgele belirlenmiştir. Problemde, araçların aynı müşteriye farklı türdeki taleplerini farklı zamanlarda iletmesine izin verilmiştir. Ancak, müşteriye teslim edilen ürün türünün tamamının tek seferde yapılması koşulu eklenmiştir. Çözüm yöntemi olarak da iki adet sezgisel yöntem kullanılmıştır: melez bir memetik algoritma ve tabu arama algoritması. Bu yöntemler ARP'de iyi bilinen Christofides problem örnekleri [3] üzerinde uygulanmıştır.

Mendoza ve ark. [4] ÇB-ARP çalışmalarına müşterilerin stokastik ürün talep miktarlarını da dahil etmişlerdir. Bu tür ÇB-ARP'de her araç her bölmesi tam dolu olacak şekilde tura başlamaktadır. Ziyaret ettiği müşterinin hangi ürününden ne kadar talep ettiği ise tur sırasında rastgele belirlenmektedir. Stokastik taleplerinden dolayı bu problem Stokastik Talepli ÇB-ARP (ST-ÇB-ARP) olarak adlandırılmıştır. Mendoza ve ark. önerileri melez bir memetik algoritmay, kendi üretikleri problem örnekleri ve kendi problemlerine göre uyarladıkları Christofides problem örnekleri üzerinde iki ayrı senaryo ile uygulanmıştır. İlk senaryoda, her müşterinin bütün ürünleri tek bir araç tarafından tek seferde teslim edilirken, ikinci senaryoda müşteriler her bir ürün için ayrı seferlerde ziyaret edilebilmektedir. Ancak bu senaryoda her bir müşterinin talep ettiği ürün türü tek bir seferde teslim edilmektedir.

Reed ve ark. [5] cam ve kağıt dönüşüm atıklarının toplandığı bir ÇB-ARP üzerinde çalışmıştır. Araçlarda kağıt ve cam için eşit kapasiteli iki ayrı bölmeye bulunmaktadır ve her bir müşteri tek bir araç tarafından tek seferde ziyaret edilmektedir. Çözüm yöntemi olarak da Karınca Kolonisi Optimizasyonu (KKO) kullanılmıştır. Reed ve ark. da Christofides problem örnekleri üzerinde deneysel çalışmalarını yapmışlardır.

Goodson [6] da aynı Mendoza ve ark. [4] gibi ST-ÇB-ARP üzerinde çalışmıştır. Çalışılan probleme farklı ürün sayısı üç olarak belirlenmiş ve araçların bölmeleri eşit olacak şekilde bir tam sayıla sabitlenmiştir. Her bir müşterinin sadece bir araç tarafından tek seferde ziyaret edilmesine izin verilmiştir. Çözüm yöntemi olarak da önermiş olduğu melez Benzetimli Taylama (BT) kullanmıştır.

Abdulkader ve ark. [7] çalışmaları ÇB-ARP'yi Christofides problem örneklerini baz alarak ürettikleri kendi problem örnekleri üzerinde, bir melez KKO algoritması kullanarak çözümüsterdir. Önceki çalışmaların çoğunda olduğu gibi, onlar da her bir müşterinin sadece bir araç tarafından tek seferde ziyaret edilmesine izin vermişlerdir. Araçların bölmeleri ise iki adettir ve birisi diğerinin 3 katı büyülüktedir.

Henke ve ark. [8] dört farklı renkte cam atıklarının toplandığı bir ÇB-ARP üzerinde çalışmışlardır. Bu problemin farklı yanı, araçların bölmelerinin sabit değil esnek olduğunu. Yani, araçların bölmeleri hareketli bir duvar ile birbirinden ayrılmıştır ve aracın taşıdığı cam çeşidi ve miktarına göre bu bölmeler büyüp küçülebilirler. Bu yüzden de çalışmaları bu probleme Esnek ÇB-ARP (E-ÇB-ARP) adını vermişlerdir. Geliştirdikleri bir Değişken Komşu Arama (DKA) yöntemini, özel bir şirketten aldıkları verilerle oluşturdukları problem örnekleri üzerinde uygulamışlardır.

Hübner ve ark. [9] market zincirlerinin depolarından şubelerine farklı sıcaklıkta taşınması gereken malları ele alan bir ÇB-ARP üzerinde çalışmışlardır. Amaç fonksiyonu olarak toplam maliyetin (mesafe + yükleme + boşaltma maliyetlerinin) en aza indirmesini ele almışlardır. Problemlerde iki bölmeye sahip araçlar üzerinde durulmuştur. Bölmelerin birisi soğuk saklanması gereken gıdalari içeren dondurucu bölüm ve diğeri ise oda sıcaklığına sahip bölümde. Özel bir market zincirinden aldıkları veri ile oluşturdukları ve de rastgele oluşturdukları iki problem örnek kümesi üzerinde, Büyük Komşuluk Arama (BKA) algoritmasını kullanarak problemlerinin deneyel sonuçlarını elde etmişlerdir.

Chen ve ark. [10] çalışmalarında, klasik ÇB-ARP'ye müşterilerin zaman penceresi kısıtlarını da ekleyerek ZK-ÇB-ARP'yi (Zaman Kısıtlı ÇB-ARP) ele almışlardır. Problemlerde kullandıkları amaç fonksiyonları araçların toplam tur uzunlukları, yaktı tüketim masrafları ve sabit masrafların hepsini içermektedir. Araçlar eş kapasiteli iki bölmeye sahiptir. Her bir müşteri tek bir araç tarafından bir kez ziyaret edilmektedir. Chen ve ark. gerçek hayat veri setinden oluşturdukları problem örnekleri üzerinde, CPLEX yöntemini ve melezledikleri Uyarlanabilir Büyük Komşuluk Arama (UBKA) algoritmasını uygulayarak bu algoritmaların performanslarını karşılaştırmışlardır.

Chen ve ark. [11] da bir ZK-ÇB-ARP üzerinde çalışmışlardır. Problemlerde üç farklı tür ürün için esnek bölünmüş üç bölmeli araçlar yer almaktadır. Her bir müşterinin tek bir araç tarafından tek bir seferde ziyaret edilmesi gerekmektedir. Chen ve ark. Parçacık Sürü Optimizasyonu (PSO) algoritmasını Benzetimli Tavlama (BT) algoritması ile melezlemişler ve önerdikleri bu çözüm yöntemini Solomon problem örnekleri [12] üzerinde uygulamışlardır.

Kaabachi ve ark. [13] ÇB-ARP problemini çalışmışlardır. İki bölmeli araçların bulunduğu probleme, çözüm olarak önerdikleri iki yöntemi kıyaslamışlardır: melez Yapay Ari Kolonisi (YAK) algoritması ve melez Özayarlamalı Genel Değişken Komşu Arama (ÖGDKA) algoritması. Problem örnekleri olarak da Abdulkader ve ark. [7]'nin çalışmasında ortaya konulan ve yazarların kendilerinin ürettiği problem örnekleri kullanılmıştır.

Wang ve ark. [14] farklı türdeki fosil yakıtların ana depodan benzin istasyonlarına taşıdığı bir ÇB-ARP çalışmışlardır. Çindeki bir yakıt nakliyesi şirketinden aldıkları veriler üzerinde UBKA algoritması kullanarak çözüm üretmiş ve elde ettikleri sonuçları, oluşturdukları KTDP (Karışık Tamsayılı Doğrusal Programlama) sonuçlarıyla karşılaştırmışlardır. Küçük boyutlu problem setlerinde KTDP'den daha hızlı bir şekilde optimum sonuçları bulmuşlardır.

Eştehadi ve ark. [15] çalışmaları ÇB-ARP'de genel olarak şehir için lojistik problemini göz önünde bulundurmuştur. Genişlettikleri UBKA algoritmasını, Londra haritası üzerinde yazarların farklı boyutlarda üretikleri problem örnekleri üzerinde uygulamışlardır.

ÇB-ARP alanındaki çalışmalar gün geçtikçe artmaktadır ve problemin değişik türleri üretilmektedir. Bu çalışmada, klasik ÇB-ARP ele alınmıştır. 2 ve 3 bölmeli araçlar için iki farklı senaryo kullanılmıştır. Yöntem olarak da iyi bilinen popülasyon tabanlı bir sezgisel yöntem olan Genetik Algoritma (GA), bilinen ve bu çalışmada geliştirilen yerel arama yöntemleriyle melezlenerek kullanılmıştır. Problemde müşterilerin her bir ürün için farklı araçlar tarafından ziyaret etmelerine izin verilmiş ancak her bir ürünün tek seferde teslim edilmesi koşulu eklenmiştir.

## 2. Materyal ve Metot

### 2.1. Matematiksel model

Klasik ÇB-ARP matematiksel olarak yönşüz bir tam çizge olarak tanımlanmaktadır,  $G = (N, A)$ . Burada  $N' = \{1, 2, \dots, n\}$  müşterilerin kümesi ve 0 depo olmak üzere  $N = N' \cup \{0\}$ , ve A da çizgedeki düğümler arasındaki kenarların kümesidir. Müşterilerin talep ettiği ürün türleri  $P = \{1, 2, \dots, m\}$ , araçların kümesi  $V = \{1, 2, \dots, v\}$ , i ile j düğümleri arasındaki mesafe  $c_{ij}$ , araçların p. ürün için ayrılan kapasiteleri  $Q_p$  ve her bir aracın en fazla seyahat edebileceği mesafe L olmak üzere, problemin amaç fonksiyonu ve kısıtları aşağıda verilmiştir [2]:

Amaç Fonksiyonu:

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

Kısıtlar:

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

$$\sum_{i \in N} x_{ijk} = \sum_{i \in N} x_{jik} \quad \forall j \in N', \forall k \in V \quad (3)$$

$$\sum_{i \in N} x_{ijk} \leq |S| - 1 \quad \forall k \in V, \quad \forall S \subseteq N', |S| \geq 2 \quad (4)$$

$$y_{jkp} \leq \sum_{i \in N} x_{ijk} \quad \forall j \in N', \forall k \in V, \quad \forall p \in P \quad (5)$$

$$\sum_{k \in V} y_{jkp} = 1 \quad \forall j \in N', \forall p \in P \quad (6)$$

$$\sum_{j \in N'} y_{jkp} q_{jp} \leq Q_p \quad \forall k \in V, \forall p \in P \quad (7)$$

$$\sum_{i,j \in N} c_{ij} x_{ijk} \leq L \quad \forall k \in V \quad (8)$$

$$x_{ijk} \in \{0,1\} \quad \forall i \in N, \forall j \in N, \quad i \neq j, \forall k \in V \quad (9)$$

$$y_{jkp} \in \{0,1\} \quad \forall j \in N', \forall k \in V, \quad \forall p \in P, q_{jp} \neq 0 \quad (10)$$

Burada (2) numaralı kısıt aynı tur içinde her bir müşterinin tek bir kez ziyaret edilmesini garantilemektedir. (3) numaralı kısıt aracın turunu kendi müşterilerini atlamanadan ardışık bir şekilde yapmasını sağlamaktadır. (4) numaralı kısıt aracın kendi içinde alt turlar yapmasını engellemektedir. (5) numaralı kısıt, eğer j numaralı müşterinin p numaralı talebi k aracının tarafından getirilmiyorsa talebin yanlışlıkla aracın bölmeye kapasitesini azaltmasını engellemektedir. (6) numaralı kısıt ise müşterinin her bir talep ettiği ürünün tek seferde bir araç tarafından getirilmesini garantilemektedir. (7) ve (8) numaralı kısıtlar, araçların bölmeye kapasitelerini aşmalarını ve araçların maksimum seyahat sınırlarını aşmalarını engellemektedir. (9) numaralı kısıt müşterinin o araç turunda seçiliş seçilmediğini belirten karar değişkenidir. (10) numaralı kısıt ise müşterinin talebi olan p ürünün o araçta olup olmadığını gösteren karar değişkenidir.

### 2.2. Metotlar

ÇB-ARP, NP-Zor sınıfından bir problem olduğu için [2], bu çalışmada yaygın çözüm metodlarından biri olan Genetik Algoritma (GA) kullanılmıştır.

#### 2.2.1 Çözüm temsili

Problemden kullanılan çözüm tüm müşterilerin indislerinin oluşturduğu bir permütasyon ile temsil edilmiştir. ÇB-ARP'nin yapısı gereği bir müşterinin birden fazla talebi olduğu için, permütasyondaki her bir genin hangi müştereye ve ürüne karşılık geldiğinin hesaplanması, Fallahi ve ark. [2] çalışmasındaki gibi (11) ve (12) numaralı formüller aracılığıyla yapılmıştır.

$$\text{müşteri}(\pi_i) = \lfloor (\pi_i + m - 1)/m \rfloor \quad (11)$$

$$\text{ürün } (\pi_i) = ((\pi_i - 1) \bmod m) + 1 \quad (12)$$

Burada  $\pi$  permütasyonu,  $\pi_i$  permütasyon içindeki  $i$ . geni,  $m$  ise toplam ürün çeşit (ya da

araçtaki bölüm) sayısını temsil etmektedir.  $n$  toplam müşteri sayısını göstermek üzere,  $\pi$ 'nin boyutu  $\rho = n \cdot m$ 'dir.

Örneğin, toplam 5 müşterinin ve 2 farklı ürünün olduğu bir ÇB-ARP'de,  $n = 5$ ,  $m = 2$ ,  $\rho = 10$  olmaktadır. Müşterilerin ve ürünlerin indislenmesi ise şu şekilde: birinci müşterinin birinci ürün talebi 1, birinci müşterinin ikinci ürün talebi 2, ikinci müşterinin birinci ürün talebi 3, ikinci müşterinin ikinci ürün talebi 4, vb. Örneğin  $n = 5$ ,  $m = 2$  için şöyle bir permutasyon olsun,  $\pi = (2, 1, 8, 4, 10, 5, 7, 10, 3, 6)$ . Bu permutasyondaki üçüncü genin hangi müşterinin hangi ürün talebine karşılık geldiği sırasıyla (11) ve (12)'deki formüller aracılığıyla bulunmaktadır.

$$\text{müşteri}(\pi_3) = \text{müşteri}(8) = \lfloor (8 + 2 - 1) / 2 \rfloor = 4.$$

$$\text{ürün}(\pi_3) = \text{ürün}(8) = ((8 - 1) \bmod 2) + 1 = 2.$$

Yani, permutasyondaki 8 değeri, 4. müşterinin 2. tür ürün talebi olduğunu belirtmektedir.

Kullanılan GA'daki kromozomlar çözüm temsilindeki permutasyonlardır. Bu permutasyonun araçlara ayrılması da literatürde en çok kullanılan Bellman Split algoritmasının ÇB-ARP'ye uyarlanmış haliyle [2] gerçekleştirilmektedir.

### 2.2.2 Kullanılan sezgisel yöntemler

Kullanılan GA, beş adet tur oluşturan sezgisel yöntem ve dört adet yerel arama sezgisel yöntem ile melezlenmiştir. Kullanılan beş adet tur oluşturan sezgisel yöntem şunlardır: Clark & Wright Tasarruf Algoritması, En Yakın Komşu Algoritması, Rastgele Görev Algoritması, Rastgele Görev - Çiçek Algoritması, Tarama Algoritması.

Clark & Wright Tasarruf Algoritması [16] en kötü senaryoya, yani her bir müşteriye bir araç atamaya başlar. Ardından bu araçların turlarını sırasıyla birleştirmeye devam eder. Gidilen mesafede en çok tasarruf edilen araçlar, problemin kısıtları dahilinde birleştirilir. Birleştirilebilecek araç kalmayana kadar bu işlem devam eder.

En Yakın Komşu Algoritması ise çözüm için birbirlerine yakın müşterilerden oluşan araç turları oluşturmaya çalışır. Her bir aracın turu oluşturulurken, ilk olarak depoya en yakın

müşteri tura dahil edilir. Ardından geriye kalan müşteriler arasında turun en son müşterisine en yakın müşteri bulunur ve o da turun sonuna eklenir. Eğer kapasite, maksimum tur uzunluğu, vb. kısıtlar aşılıyorsa, yeni bir araç eklenir ve bir sonraki müşteri yine aynı en yakın komşuluk kriterine göre bu yeni aracın turuna dahil edilir. Bu işlem tüm müşteriler araç turlarına eklenene kadar devam eder.

Rastgele Görev Algoritması [17] En Yakın Komşu Algoritması ile aynı prensipte çalışmaktadır. Tek farkı ise, eğer birden fazla en yakın müşteri varsa Rastgele Görev Algoritması bunları bir listeye dahil eder ve seçim yaparken bu listeden rastgele birisini seçer. Bu seçim işlemi tüm müşterilerin araç turlarına eklenmesiyle son bulur.

Rastgele Görev - Çiçek Algoritması [17] Rastgele Görev Algoritmasının geliştirilmiş halidir. Bu algoritmda her bir yeni müşteri seçilmeden önce, geriye kalan müşteriler turu depodan uzaklaştırılanlar ve turu depoya yaklaştırılanlar olarak iki gruba ayrılır. Araç turuna eklenecek bir sonraki müşteri seçilirken, eğer aracın kapasitesi yarısına kadar dolmamışsa ilk gruptaki en yakın müşteri, değilse ikinci gruptaki en yakın müşteri seçilir. Bu çalışmada ÇB-ARP ele alındığı için, aracın tüm kapasitesi yerine seçilen müşteri ve talebinin karşılık geldiği araç bölmesinin kapasitesinin yarıya ulaşılıp ulaşmadığına bakılmıştır. Bu sayede araçların etkin bir şekilde müşterileri turlarına dahil etmesi sağlanmaktadır. Tüm bu işlem, tüm müşteriler araçlara dahil edilene kadar devam eder.

Tarama Algoritması [18] tüm müşterilerin depoya olan kutupsal koordinatlarına göre açlarını belirleyerek başlar. Ardından açılarına göre müşterileri küçükten büyüğe doğru sıralar. Bu sıralama doğrultusunda ve kısıtlar dahilinde kutupsal koordinat açısı en küçük olanın başlayarak müşterileri sırasıyla araçların turlarına dahil eder. Tüm bu işlem bütün müşteriler araçlara atanana kadar devam eder.

GA'da kullanılan yerel arama yöntemleri de şunlardır: 2-Opt, Yer Değiştirme, Araya Ekleme ve Araç Sayısını Azaltma.

2-Opt yönteminde [19] amaç turda iki müşteri arasındaki yolların belirli bir kurala göre yer değiştirilmesiyle mevcut permutasyonu iyileştirmektir. Tur içindeki müşteriler sırasıyla ikişer olarak yer değiştirilir ve bu iki müşterinin

arasında kalan diğer müşterilerin sırası tersten yazılır. Eğer yeni oluşan tur daha iyi sonuç üretirse, aynı işlem baştan bu yeni tur üzerinden yapılır, değilse turun orijinal permütasyonu üzerinden devam edilir. Tüm bu süreç daha iyi bir sonuç üretilememesi durumuna kadar devam eder.

Yer Değiştirme sezgiseli ise, sırasıyla tur içindeki iki müşterinin yer değiştirdiği yerel aramadır. Turun birinci müşterisinden başlayarak her bir müşteri sırasıyla seçilir ve seçilen her bir müşteri turdaki diğer tüm müşterilerle yer değiştirilir. Her bir yer değiştirmede eğer tur daha da iyileşiyorsa, tur güncellenmektedir ve işlem kaldığı yerden bu güncellenmiş tur üzerinden yapılmaktadır. Tüm bu süreç mevcut turdaki tüm yer değiştirmeler bitince son bulmaktadır.

Araya Ekleme sezgiseli, Yer Değiştirme sezgiseline benzer şekilde uygulanmıştır. Farklılık olarak, her arama adımımda iki müşterinin yer değiştirmesi değil, seçilen müşterinin mevcut turdaki her bir konuma eklenmesi gerçekleştirilmiştir.

Araç Sayısını Azaltma sezgiseli ise yazarın geliştirmiş olduğu bir yerel arama yöntemidir. Bu yöntemde, tüm araçlar toplam yüklerine göre küçüğen büyüğe doğru sıralanır. Ardından, en az yüke sahip araçtan başlanarak, araç içindeki müşteriler sırasıyla problem kısıtları dahilinde diğer araçlarda olabilecek ve toplam tur uzunluğunu en aza indirgenen konuma yerleştirilirler. Bu işlem tüm araçlar üzerinde uygulandıktan sonra yerel arama sonlandırılır. Bu yerel aramada amaç, araçların sayısının azaltılarak toplam tur uzunluğunu azaltmaya çalışmaktadır.

2-Opt ve Yer Değiştirme yerel arama yöntemleri hem çözümdeki permütasyon üzerinde hem de alt araç turları üzerinde ayrı ayrı uygulanmıştır. Bir yerel arama yönteminin ürettiği en iyi sonuç bir sonraki yerel arama için girdi olacak şekilde uygulanmıştır. Bu çalışmada kullanılan Yerel Arama yönteminin algoritması Şekil 1'de verilmektedir.

```

procedure YerelArama( $\text{Çözüm } \varsigma$ )
     $\varsigma \leftarrow 2\text{-OptAnaTur}(\varsigma)$ 
     $\varsigma \leftarrow \text{YerDeğiştirmeAnaTur}(\varsigma)$ 
     $\varsigma \leftarrow \text{ArayaEklemeAnaTur}(\varsigma)$ 
     $\varsigma \leftarrow \text{AraçSayısınıAzaltma}(\varsigma)$ 
     $\varsigma \leftarrow 2\text{-OptTümAraçTurları}(\varsigma)$ 
     $\varsigma \leftarrow \text{YerDeğiştirmeTümAraçTurları}(\varsigma)$ 
     $\varsigma \leftarrow \text{ArayaEklemeTümAraçTurları}(\varsigma)$ 
end procedure

```

**Şekil 1.** YerelArama yönteminin algoritması

### 2.2.3 GA'nın uygulanması

Bu çalışmada kullanılan melez GA yöntemi şu şekilde uygulanmıştır. Başlangıç topluluğunun beş bireyi 2.2.2'de bahsedilen tur oluşturma sezgiselleriyle, geri kalan bireyler ise rastgele permütasyonlar oluşturularak elde edilmiştir. Bu başlangıç topluluğundaki tüm bireyler (kromozomlar) yerel arama prosedüründen geçirildikten sonra, melez GA önceden belirlenmiş tekrarlama sayısı kadar çalıştırılmıştır. Her bir tekrarda, topluluktaki bireyler çaprazlama, mutasyon ve yerel arama işlemlerinden geçirilmiş ve topluluğa dahil edilmiştir. Çaprazlama işlevi olarak PMX [20], mutasyon işlevi ise permütasyonda rastgele seçilen iki gen arasını ters çevirmeye kullanılmıştır. Çaprazlama için gerekli olan iki birey ikili turnuva seçimi ile belirlenmiştir. Seçilen bu bireylerden çaprazlama ile iki yeni birey daha üretilmiş ve sonrasında bu iki yeni birey sırasıyla mutasyon ve yerel arama işlemlerinden geçirilmiştir. Bu süreçlerden geçen bu yeni iki birey topluluğa eklenmiştir. Tüm bu işlemler topluluktaki birey sayısının yarısı kadar tekrarlanmıştır. Bir tekrar sonunda iki katına çıkan topluluk, bir sonraki tekrara geçmeden önce başlangıçta belirlenen birey sayısına indirgenmiştir. Bu indirgenmede, elitizm kullanılmış ve en iyi bireyler saklanırken geriye kalanlar nüfustan elenmiştir. Tüm bu tekrarlar esnasında elde edilen en iyi birey (en düşük toplam tur uzunluğunu veren birey) global bir değişkende tutulmuş ve her tekrar sırasında güncellenmiştir. Melez GA'nın çıktısı da bu global en iyi bireydir. Bu çalışmada kullanılan melez GA, Şekil 2'de verilmiştir.

```

 $\pi \leftarrow$  BaşlangıçTopluluğunuOluştur()
for each  $i$  in  $\pi$ 
    YerelArama( $\pi_i$ )
    enİyiBireyiGüncelle( $\pi_i$ )
end for
tekrar  $\leftarrow$  0
while (tekrar  $<$  maxTekrar)
     $i \leftarrow 0$ 
    while ( $i <$  ToplulukLimiti / 2)
         $p_1, p_2 \leftarrow$  ikiliTurnuvaSeçimi( $\pi$ )
         $c_1, c_2 \leftarrow$  çaprazla( $p_1, p_2$ )
        mutasyon( $c_1$ )
        mutasyon( $c_2$ )
        YerelArama( $c_1$ )
        YerelArama( $c_2$ )
        enİyiBireyiGüncelle( $c_1$ )
        enİyiBireyiGüncelle( $c_2$ )
        topluluğaEkle( $c_1$ )
        topluluğaEkle( $c_2$ )
         $i \leftarrow i + 1$ 
    end while
    bireySayısınıToplulukLimitineİndirge( $\pi$ )
    tekrar  $\leftarrow$  tekrar + 1
end while

```

**Şekil 2.** Kullanılan melez GA

### 3. Bulgular

Bu çalışmada bahsedilen yöntem ve algoritmalar C++ dilinde kodlanmış ve Intel Core i5-7200 CPU 2.50 GHz işlemcili ve 8GB RAM donanımlı bilgisayarda çalıştırılıp sonuçlar elde edilmiştir.

GA'da kullanılan parametreler ise şu şekildedir:

Nüfustaki Kromozom Sayısı: 100

Tekrar Sayısı: 100

Çaprazlama Yüzdesi: %70

Mutasyon Yüzdesi: %10

Problem örnekleri olarak, vrpncXX olarak adlandırılan ve literatürde yaygın bir şekilde kullanılan Christofides problem örnekleri seti [3] kullanılmıştır. Set içerisinde müşteri sayıları 50-199 arasında değişen toplam 14 adet problem örneği bulunmaktadır. Her bir örneğin müşteri sayısı Tablo 1'de "n" başlıklı sütunda belirtilmiştir. Örneklerde yer alan müşterilerin talepleri tekdeuze dağılıma uymaktadır. Müşterilerin x ve y koordinatları, ilk 10 örnekte

(vrpnc01-vrpnc10) tekdeuze bir dağılıma, son 4 örnekte (vrpnc11-vrpnc14) kümelenmiş bir dağılıma sahiptir. Ayrıca, 7 problem örneğinde (vrpnc01-vrpnc05, vrpnc11-vrpnc12) tüm araçların kat edebileceği toplam tur uzunluğu sınırı (Tmax) bulunmazken, geri kalanlarda (vrpnc06-vrpnc10, vrpnc13-vrpnc14) bu sınır bulunmaktadır. Bu çalışmada, bu problem örneklerinin her biri, rastgele sayı ve permütasyon üretimi için 10 farklı başlangıç çekirdek değeri ile çalıştırılmış ve bu 10 çalıştırmanın ürettiği sonuçların en iyisi, melez GA'nın o problem için bulduğu en iyi sonuç olarak kaydedilmiştir.

Bu çalışmada kullanılan melez GA, aynı problem örnekleri için, sırasıyla bir, iki ve üç bölmeli araç senaryoları için ayrı ayrı çalıştırılmıştır. Her bir örnek üzerinde bir bölmeli ARP için elde edilen toplam tur uzunluğu sonuçları, literatürdeki optimum toplam tur uzunluğu değerleri ile karşılaştırılmıştır (Tablo 1). Yazının bildiği kadariyla, iki ve üç bölmeli ARP için mevcut problem örneklerinin literatürde optimum toplam tur uzunluğu değerleri bulunmamaktadır. Bazı çalışmalarında müşterilerin talepleri eşit bir şekilde gerçek sayı olarak bölünmüş ve bu düzenleme için sonuçlar üretilmiştir. Bazı çalışmalarında ise bu bölmeler rastgele yapılmış ancak çalışmada açık bir şekilde paylaşılmamıştır. Bu çalışmada ise müşterilerin talepleri ve araçların depoları tam sayı olarak (13) ve (14) formüllerine göre belirlenmiştir.

$$Q_i = \begin{cases} \lfloor Q/m \rfloor, & 1 \leq i < m \\ Q - \sum_{i=1}^{m-1} Q_i, & i = m \end{cases} \quad (13)$$

$$q_{ij} = \begin{cases} \lfloor D_i/m \rfloor, & 1 \leq i < m \\ D_i - \sum_{j=1}^{m-1} q_{ij}, & i = m \end{cases} \quad (14)$$

Burada  $m$  araçtaki bölme (ya da ürün çeşit) sayısı,  $Q$  aracın toplam kapasitesi,  $Q_i$   $i$ . bölmeyin kapasitesi,  $D_i$   $i$ . müşterinin toplam talebi,  $q_{ij}$   $i$ . müşterinin  $j$ . ürün çeşidine dair talep miktarıdır. Örneğin,  $m = 3$  için eğer aracın toplam kapasitesi 100 ise, (13) numaralı formüle göre her bir bölmeli sırasıyla 33, 33 ve 34 olmaktadır. Yine  $m = 3$  için, eğer  $i$ . müşterinin toplam talebi 11 ise, (14) numaralı formüle göre her bir ürün çeşidi için bu müşterinin talebi sırasıyla 3, 3 ve 5

olmaktadır. Bu tür bir düzenleme, yazarın bildiği kadariyla, literatürdeki çalışmalarında uygulanmamıştır. Bu tür bir araç ve talep bölünmeleri literatürdeki çalışmalarında kullanılmadığı için, bu çalışmada elde edilen sonuçlar literatürdeki çalışmalarındaki sonuçlarla kıyaslanamamıştır. Bu çalışmada  $m = 1$  için sonuçlar Tablo 1'de  $m = 2$  ve  $3$  için elde edilen sonuçlar ise Tablo 2'de verilmiştir.

**Tablo 1.** Bir bölmeli ( $m = 1$ ) ARP için melez GA sonuçları.

Problem	n	Opt.	melez GA	Süre (sn.)	Fark
vrpnc01	50	524.61	<b>524.61</b>	86.43	%0.00
vrpnc02	75	835.26	<b>835.77</b>	39.51	%0.06
vrpnc03	100	826.14	828.67	150.57	%0.31
vrpnc04	150	1028.42	1034.34	732.52	%0.58
vrpnc05	199	1291.29	1314.48	1623.89	%1.80
vrpnc06	50	555.43	<b>555.43</b>	44.98	%0.00
vrpnc07	75	909.68	<b>909.68</b>	231.38	%0.00
vrpnc08	100	865.94	866.87	110.65	%0.11
vrpnc09	150	1162.55	1176.69	86.98	%1.22
vrpnc10	199	1395.85	1440.15	1935.30	%3.17
vrpnc11	120	1042.12	1042.42	310.33	%0.03
vrpnc12	100	819.56	<b>819.56</b>	7.71	%0.00
vrpnc13	120	1541.14	1548.71	549.58	%0.49
vrpnc14	100	866.37	<b>866.37</b>	215.16	%0.00
<b>Ortalama</b>	<b>113</b>	<b>976.03</b>	<b>983.13</b>	<b>437.50</b>	<b>%0.55</b>

**Tablo 2.** İki ( $m = 2$ ) ve üç ( $m = 3$ ) bölmeli ARP için melez GA sonuçları

Problem	<b>m = 2</b>		<b>m = 3</b>	
	melez GA	Süre (sn.)	Melez GA	Süre (sn.)
vrpnc01	542.43	228.20	553.80	355.52
vrpnc02	856.94	95.58	866.83	492.03
vrpnc03	834.93	548.36	864.54	1614.57
vrpnc04	1075.27	219.90	1107.60	6408.27
vrpnc05	1379.04	1117.15	1414.86	8437.63
vrpnc06	555.43	9.08	557.49	136.12
vrpnc07	926.10	34.75	932.02	1369.49
vrpnc08	870.08	1017.65	879.29	2280.97

vrpnc09	1198.64	1115.37	1209.31	3436.55
vrpnc10	1467.51	4209.12	1472.83	12512.70
vrpnc11	1083.53	1620.32	1242.12	1112.22
vrpnc12	819.56	165.91	954.98	178.20
vrpnc13	1547.09	2068.90	1545.04	4186.10
vrpnc14	866.53	976.21	960.95	2809.08
<b>Ortalama</b>	<b>1001.65</b>	<b>959.04</b>	<b>1040.12</b>	<b>3237.82</b>

#### 4. Tartışma ve Sonuç

Bu çalışmada ÇB-ARP çalışılmış ve sırasıyla bir, iki ve üç bölmeli ARP senaryoları ele alınmıştır. İyi bilinen bir sezgi üstü yöntem olan GA, literatürde bulunan sezgisel arama yöntemleriyle ve de yazar tarafından geliştirilen bir sezgisel arama yöntemiyle melezlenip problemin çözümünde kullanılmıştır. Deneysel çalışmalar, ARP çalışmalarında literatürde yaygın kullanılan Christofides [3] problem örnekleri üzerinde gerçekleştirilmiştir.

Bu çalışmada melez GA, kullanılan problem örnekleri üzerinde, öncelikle bir bölmeli ARP olarak çalıştırılmış ve elde edilen toplam tur uzunluğu sonuçları literatürdeki optimum toplam tur uzunluğu sonuçlarıyla kıyaslanmıştır. Bu sonuçlara göre, melez GA, üzerinde çalışılan problem örneklerinin beş tanesinde (vrpnc01, vrpnc06, vrpnc07, vrpnc12 ve vrpnc14) optimum toplam tur uzunluğu sonucunu bulmuş, tüm problem örneklerinde ise optimum toplam tur uzunluğu sonucuna ortalama %0.55'lik bir farkla yaklaşmıştır. Bu başarılı sonuçlarından ardından, melez GA aynı problemin iki ve üç bölmeli senaryolarında uygulanmıştır. Bu makalede ayarlandığı şekilde, bu problem örneklerinin iki ve üç bölmeli ARP sonuçları literatürde bulunmadığından, bu çalışmada sonuçlar herhangi başka bir sonuç ile karşılaştırılamamıştır.

Bir, iki ve üç bölmeli ARP'ler için bu çalışmada elde edilen sonuçlara bakıldığından, araçlardaki bölmeye (müşterilerin ürün çeşitlerinin) sayısı arttıkça, amaç fonksiyonunun değerinin ve algoritmaların işlem sürelerinin arttığı gözlemlenmiştir. Bunun nedeni araçtaki bölmeye (müşterilerin talep ettiği ürün çeşidinin) sayısının artmasını çözüm temsilindeki permütasyon boyutunu da artırıyor olmasıdır. Ayrıca, araçların toplam kapasiteleri de bölmelerden dolayı küçüldüğü için, bazı

durumlarda araç sayısı artmakta bazı durumlarda ise aynı müşterinin farklı ürün talepleri kapasite sınırlarından ötürü farklı araçlara denk gelebilmektedir. Bu da amaç fonksiyonu olan toplam tur uzunluğunu artırmaktadır. Bölme sayısının artması aynı zamanda ana permütasyonun ve araç turlarındaki permütasyonların boyutunu artmasına neden olduğu için, genel ve yerel arama yöntemleri daha fazla sayıda adımda arama yapmaktadır. Bu da algoritmanın çalışma süresini uzatmaktadır.

Bu çalışmanın ilgili alana yaptığı katkılar birkaç maddede özetlenebilir. İlk olarak, yazarın bildiği kadariyla, Türkçe alanda ÇB-ARP alanında yapılan ilk deneysel çalışmındır. İkincisi, literatürde mevcut ÇB-ARP çalışmalarından farklı olarak, araç bölmelerinin ve müşterilerin farklı türdeki ürün taleplerinin tamamıyla tam sayı olarak hesaplanmasıdır ((13) ve (14) numaralı formüller). Üçüncüsü, melez GA'da kullanılan yerel arama yöntemlerin birisinin yazar tarafından geliştirilmiş olmasıdır. Dördüncüsü, ÇB-ARP'nin bu çalışmada ele alınan şekli için elde edilen sonuçlar, bir, iki ve üç bölgeli ARP için bir referans seti oluşturmaktadır.

Bu makaledeki çalışma, farklı algoritmaların dahil edilip kıyaslanmasıyla, ya da başka amaç fonksiyonlarına (örneğin toplam yaktın azaltılmasına) göre çözümler yapılmasıyla ve yahut zaman penceresi kısıtlamalarının eklenmesi ile geliştirilebilir ve ileriki çalışmalar olarak ele alınabilir. Ya da gerçek hayatı elde edilen veriler kullanılarak özel örnekler üzerinde çalıştırılabilir.

## Kaynakça

- [1] Christofides, N., Mingozi, A., and Toth, P., "The Vehicle Routing Problem," *Rev. française d'automatique, informatique, Rech. opérationnelle. Rech. opérationnelle*, vol. 10, 1979, doi: 10.1051/ro/197610V100551.
- [2] El Fallahi, A., Prins, C., and Wolfler Calvo, R., "A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem," *Comput. Oper. Res.*, vol. 35, no. 5, pp. 1725–1741, 2008, doi: 10.1016/j.cor.2006.10.006.
- [3] Christofides, N. and Eilon, S., "An Algorithm for the Vehicle-Dispatching Problem," *OR*, vol. 20, no. 3, p. 309, 1969, doi: 10.2307/3008733.
- [4] Mendoza, J. E., Castanier, B., Guéret, C., Medaglia, A. L., and Velasco, N., "A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands," *Comput. Oper. Res.*, vol. 37, no. 11, pp. 1886–1898, 2010, doi: 10.1016/j.cor.2009.06.015.
- [5] Reed, M., Yiannakou, A., and Evering, R., "An ant colony algorithm for the multi-compartment vehicle routing problem," *Appl. Soft Comput. J.*, vol. 15, pp. 169–176, 2014, doi: 10.1016/j.asoc.2013.10.017.
- [6] Goodson, J. C., "A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands," *Eur. J. Oper. Res.*, vol. 241, no. 2, pp. 361–369, 2015, doi: 10.1016/j.ejor.2014.09.031.
- [7] Abdulkader, M. M. S., Gajpal, Y., and Elmekkawy, T. Y., "Hybridized ant colony algorithm for the Multi Compartment Vehicle Routing Problem," *Appl. Soft Comput. J.*, vol. 37, pp. 196–203, 2015, doi: 10.1016/j.asoc.2015.08.020.
- [8] Henke, T., Speranza, M. G., and Wäscher, G., "The multi-compartment vehicle routing problem with flexible compartment sizes," *Eur. J. Oper. Res.*, vol. 246, no. 3, pp. 730–743, 2015, doi: 10.1016/j.ejor.2015.05.020.
- [9] Hübner, A. and Ostermeier, M., "A multi-compartment vehicle routing problem with loading and unloading costs," *Transp. Sci.*, vol. 53, no. 1, pp. 282–300, 2019, doi: 10.1287/trsc.2017.0775.
- [10] Chen, L., Liu, Y., and Langevin, A., "A multi-compartment vehicle routing problem in cold-chain distribution," *Comput. Oper. Res.*, vol. 111, pp. 58–66, 2019, doi: 10.1016/j.cor.2019.06.001.
- [11] Chen, J. and Shi, J., "A multi-compartment vehicle routing problem with time windows for urban distribution – A comparison study on particle swarm optimization algorithms," *Comput. Ind. Eng.*, 2019, doi: 10.1016/j.cie.2019.05.008.
- [12] Solomon, M. M., "Algorithms for the Vehicle Routing And Scheduling Problems with Time Window Constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987, doi: 10.1287/opre.35.2.254.
- [13] Kaabachi, I., Yahyaoui, H., Krichen, S., and Dekdouk, A., "Measuring and evaluating hybrid metaheuristics for solving the multi-compartment vehicle routing problem," *Meas. J. Int. Meas. Confed.*, 2019, doi: 10.1016/j.measurement.2019.04.019.
- [14] Wang, L., Kinable, J., and van Woensel, T., "The fuel replenishment problem: A split-delivery multi-compartment vehicle routing problem with multiple trips," *Comput. Oper. Res.*, vol. 118, 2020, doi: 10.1016/j.cor.2020.104904.

- [15] Eshtehadi, R., Demir, E., and Huang, Y., "Solving the vehicle routing problem with multi-compartment vehicles for city logistics," *Comput. Oper. Res.*, vol. 115, 2020, doi: 10.1016/j.cor.2019.104859.
- [16] Clarke, G. and Wright, J. W., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, 1964, doi: 10.1287/opre.12.4.568.
- [17] Prins, C., Labadi, N., and Reghioui, M., "Tour splitting algorithms for vehicle routing problems," *Int. J. Prod. Res.*, vol. 47, no. 2, pp. 507–535, 2009, doi: 10.1080/00207540802426599.
- [18] Gillett, B. E. and Miller, L. R., "A Heuristic Algorithm for the Vehicle-Dispatch Problem," *Oper. Res.*, 1974, doi: 10.1287/opre.22.2.340.
- [19] Croes, G. A., "A Method for Solving Traveling-Salesman Problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958, doi: 10.1287/opre.6.6.791.
- [20] Goldberg, D. E. and Lingle, R. J., "Alleles, Loci and the Traveling Salesman Problem," in *Proceedings of the 1st International Conference on Genetic Algorithms*, 1985, pp. 154–159, [Online]. Available: <http://dl.acm.org/citation.cfm?id=645511.657095>.