

## Sıkı Zaman Pencereli Araç Rotalama Probleminin Çözümü için Yeni Bir Yöntem Önerisi ve Bir Süpermarket Zincirinde Uygulanması

Serap ERCAN CÖMERT\*<sup>1</sup>, Harun Reşit YAZGAN<sup>1</sup>, İrem SERTVURAN<sup>1</sup>, Hanife ŞENGÜL<sup>1</sup>

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

(Alınış / Received: 18.01.2017, Kabul / Accepted: 06.08.2017, Online Yayınlanma / Published Online: 11.10.2017)

### Anahtar Kelimeler

Sıkı zaman pencereli araç rotalama problemi,  
Kümeleme,  
K-Medoids kümeleme algoritması,  
DBSCAN kümeleme algoritması

**Özet:** Bu çalışmada bir süpermarket zincirinin belirli zaman aralıklarında servis gören müşterilerinin taleplerinin karşılanmasında ortaya çıkan Sıkı Zaman Pencereli Araç Rotalama Problemi (SZRARP)'nin çözümü yapılmıştır. Önce kümele sonra rotala yaklaşımına dayanan iki aşamalı hiyerarşik bir yöntem önerilmiştir. İlk aşamada müşteriler K-medoids ve DBSCAN kümeleme algoritmaları kullanılarak araçlara atanmıştır. İkinci aşamada ise rotalama problemi MILP ile çözülmüştür. Çalışmanın en önemli katkısı, önerilen yöntem büyük boyutlu gerçek problemler ele alınırken kesin çözüm yöntemlerini kullanmamıza olanak sağlamasıdır. Son olarak önerilen yaklaşım bir süpermarket zincirinde uygulanmış, iki algoritmanın sonuçları ve firmadan alınan gerçek sonuçlar ANOVA ile karşılaştırılmıştır. Test sonucuna göre DBSCAN'ın daha iyi sonuç verdiği görülmüştür.

## A New Method for the Solution of Vehicle Routing Problem with Hard Time Windows and A Supermarket Chain Application

### Keywords

Vehicle routing problem with hard time windows,  
Clustering,  
K-Medoids clustering algorithm,  
DBSCAN clustering algorithm

**Abstract:** In this study, VRPHTW (Vehicle routing problem with hard time windows) which appeared to meet demands of customers serviced within time intervals in a supermarket chain is solved. A hierarchical approach consisted of two stages as "cluster-first route-second" is proposed. In the first stage, customers are assigned to vehicles using different clustering algorithms as K-Medoids and DBSCAN. In the second stage, a VRPHTW is solved by a MILP. The main contribution of the paper is that the proposed approach enables us to deal with a large size real problem using an exact solution method. Finally, the proposed approach is employed on a supermarket chain. Two algorithms' and the actual results are compared with the ANOVA test to illustrate the feasibility of the proposed approach. The results illustrate that the DBSCAN algorithm provides a better solution than the other.

### 1. Giriş

İlk kez Dantzig ve Ramser tarafından tanımlanan araç rotalama problemi (ARP) bir yada birden fazla depodan n adet müşteriye hizmet sunmak için, kat edilen toplam süreyi veya toplam mesafeyi en aza indirecek şekilde araç rotalarının belirlenmesidir [1]. Günümüzde dağıtım sistemleri oldukça karmaşık hale gelip firmaların maliyetlerini arttırmaktadır. Etkin olarak yapılan bir araç rotalama planı lojistik maliyetlerinde önemli bir tasarruf sağlamaktadır. ARP'de amaç, bütün müşterilerin ihtiyacını karşılayan, maliyeti minimize eden rotaların

belirlenmesidir. ARP sahip olduğu kısıtlara göre farklı türlere sahiptir ve NP-Zor problem sınıfına girmektedir. Bu çalışmada zaman pencereli araç rotalama problemi (ZPARP) incelenmiştir. ZPARP'nin amacı, müşteriler tarafından belirlenen zaman aralıklarını dikkate alarak dağıtım rotalarının oluşturulmasıdır. Esnek ve sıkı zaman pencereli araç rotalama problemleri ZPARP'nin çeşitleri olup, bu çalışmada sıkı zaman pencereli araç rotalama problemi (SZPARP) ele alınmıştır.

Kümeleme yöntemi, verileri gruplara (kümelere) bölmeyi amaçlayan bir gözetimsiz öğrenme türüdür.

Veriler grup içerisindeki elemanların kendi içerisindeki benzerliği ve grupların birbirine benzememesi koşuluyla birbirinden ayrılmaktadır. Kümelemede asıl amaç sınıflar arasındaki benzerliği en aza indirmek, sınıfların kendi içerisindeki benzerliğini ise en yüksek seviyede tutmaktır. Çalışmanın birinci aşamasında K-Medoids ve DBSCAN algoritmaları kullanılarak veriler kümelendi. K-Medoids kümeleme algoritması bölme kümeleme tekniklerindedir ve kümeyi temsil edecek noktayı bulmak için kümenin merkezi noktasındaki elemanı yeni küme merkezi olarak alır ve böylece bazı verilerin küme merkezini kaydırma ihtimali ortadan kalkmış olur. DBSCAN kümeleme algoritması ise yoğunluğa dayalı kümeleme tekniklerinin temelini oluşturan bir algoritmadır. Algoritma kümeleri oluştururken nesnelere yoğunluklarını dikkate almaktadır. DBSCAN algoritmasında yoğunluk, o noktanın belirli bir mesafesinde bulunan en büyük komşuluk yarıçapı ve Eps yarıçapındaki nokta sayısına bakılarak hesaplanmaktadır [2].

Bu çalışmada, 78 şubeye sahip bir süpermarket zincirinin ana deposundan şubelerine yapacağı dağıtım problemi ele alınmıştır. Araç rotalama problemi hiyerarşik olarak çözülmüştür. Önerilen yaklaşım kümeleme ve rotalama olarak iki aşamalıdır. Kümeleme aşamasında müşteriler K-medoids ve DBSCAN gibi iki farklı kümeleme algoritması kullanılarak araçlara atanır. Rotalama aşamasında ise SZPARP toplam bekleme ve seyahat sürelerini en aza indirmeyi hedefleyen bir MILP ile çözülür. Makalenin ana katkısı önerilen yöntemin büyük boyuttaki gerçek problemi ele alırken kesin çözüm yöntemini kullanmamıza olanak sağlamasıdır.

Makalenin geri kalan kısmı şöyledir; ikinci bölümde, literatür araştırmasına yer verilmiş, üçüncü bölümde ARP detaylı bir şekilde anlatılmış ve dördüncü bölümde kümeleme analizi hakkında bilgi verilmiştir. Beşinci bölümde ise uygulama kısmına yer verilmiştir. Son olarak çalışmada elde edilen bulgular sonuç bölümünde özetlenmiştir.

## 2. Literatür Araştırması

Araç rotalama problemi, bir veya birkaç depodan belirli müşterilere yapılan ürün dağıtımı ve müşterilerden ürünlerin toplanması için gerekli olan rotaların belirlenmesi problemi olarak tanımlanmaktadır. ARP, literatürde NP-zor problemler sınıfında yer almaktadır ve klasik gezgin satıcı probleminin genelleştirilmiş bir halidir [3]. Araç rotalama probleminin ilk defa 1959 yılında Dantzig ve Ramser tarafından incelendiğini biliyoruz. Dantzig ve Ramser çalışmalarında petrol istasyonlarına benzin dağıtım problemini ele almışlar ve bu problemin çözümü için matematiksel bir model geliştirmişlerdir. Sonraki çalışmalardan en önemlisi ise aynı problemin çözümü için Clarke ve Wright'ın 1964 yılında önerdikleri sezgisel tasarruf algoritmasıdır [4].

Bu bölümde ARP türlerinden biri olan zaman pencere araç rotalama problemi (ZPARP) için bulunan bilimsel çalışmalar özet şekilde yer almaktadır. ZPARP esnek zaman pencere ve sıkı zaman pencere araç rotalama problemi olmak üzere ikiye ayrılır. Esnek zaman pencere araç rotalama probleminde (EZPARP), müşterilere belirli  $[a_i, b_i]$  zaman pencerelerinin dışında hizmet verilebilir, fakat bu durumda bir ceza maliyeti vardır. Sıkı zaman pencere araç rotalama probleminde (SZPARP), eğer bir araç müşterinin söz konusu zaman aralığının başlangıcından önce gitmişse en erken hizmete başlama zamanına kadar beklemek zorundadır, eğer en geç hizmete başlama zamanından sonra gitmişse de hizmet verememektedir. ARP'nin çözümü için kullanılan yöntemlerden biri de kesin çözüm yöntemleridir. Dumas vd. [5] zaman pencere dağıtım ve toplamalı ARP için sütun oluşturma yaklaşımını kullanarak bir algoritma geliştirmişlerdir. Aydemir [6], EZPARP için amaç programlama yaklaşımı ile bir otomotiv fabrikasının tedarikçilerinden malzeme temin etmek için kullandığı araçların rotalanması problemini ele almıştır. Boer [7], çok araçlı ZPARP için iki adet matematiksel model önermiştir. Tezer [8], bir dağıtım noktasının bulunduğu, dağıtım ve toplamalı bir problem ele almış ve çözümü için ise iki aşamalı bir kesin çözüm algoritması önermiştir. Çetin ve Gencer [9], kesin zaman pencere eşzamanlı dağıtım toplamalı bir ARP için matematiksel model geliştirmişlerdir. Önerdikleri algoritmalarını Solomon'un test verilerinde uygulamışlardır. Elde edilen sonuçlar müşteri sayısının artmasıyla optimalliğin bulunmasının zorlaştığını göstermektedir. Çetin ve Gencer [10], heterojen araç filolu zaman pencere eş zamanlı dağıtım ve toplamalı ARP için matematiksel model geliştirmişler ve elde ettikleri sonuçları Solomon test verileri ile karşılaştırmışlardır. Çetinkaya [11], iki aşamalı hat ZPARP için bir matematiksel model ve kısa sürede iyi sonuçlar veren sezgisel bir yöntem geliştirmiştir. Taş vd. [12], lineer programlama modeli yardımıyla esnek zaman pencere araç rotalama problemini (EZPARP) çözmüşlerdir. Çetinkaya [13], hat zaman pencere yer seçimi ve araç rotalama problemi üzerine matematiksel model geliştirmişlerdir. Akca [14], hammadde tedarik aktivitesi için kesin zaman pencere ARP için MILP model önermiştir.

ARP çözümü için kullanılan diğer yaklaşım ise sezgisel metotlardır. Liu ve Shen [15], Clarke ve Wright tasarruf algoritmasını kullanarak zaman pencere ve birden çok araç tipine sahip probleme çözüm geliştirmişlerdir. Demircioğlu [16], sezgisel yöntemlerden biri olan Clarke ve Wright'ın önerdiği tasarruf algoritması üzerinde bazı geliştirmeler yaparak Mersin'deki bir dağıtım firmasında ZPARP'nin uygulamasını yapmıştır. Aynı metot kullanılarak yapılan bir diğer çalışma da Şahin vd. [17] tarafından gerçekleştirilmiştir.

Schulze ve Fahle [18], çalıştıkları modelde ZPARP 'nin çözümünde paralel tabu arama algoritması uygulamışlardır. Ho ve Haugland [19], zaman pencere, bölünmüş teslimatlı ARP'yi tabu arama yöntemi yardımıyla çözümlenmişlerdir. Jiang vd. [20], heterojen filolu ZPARP 'nin çözümü için tabu arama algoritmasını kullanmışlardır.

Barán ve Schaerer [21], karınca kolonisi sistemi yöntemlerinden olan çoklu karınca kolonisi algoritmasını kullanarak ZPARP için çözüm önerisi sunmuşlardır. Tokaylı [22], ZPARP için bir karar destek sistemi (ZPARP-KDS) geliştirmiş olup, rota eliminasyon yöntemi ve karınca kolonisi sistemi algoritmasını kullanmıştır.

Bouthillier ve Crainic [23] ZPARP için kooperatif paralel bir meta-sezgisel önermişlerdir. Dondo ve Cerda [24], çok depolu ve heterojen araçlı ZPARP'yi küme tabanlı optimizasyon yaklaşımıyla çözmüşlerdir.

Dursun [25], Solomon'un zaman pencere ARP problemi için literatüre kattığı veri setini rassal sayı kodlamalı GA ile modellemiştir. Nazif ve Lee [26], ZPARP için melez bir genetik algoritma geliştirmişler ve sonuçlarının literatürdeki diğer algoritmalarla göre daha iyi olduğunu ifade etmişlerdir. Kiremitçi vd. [27] gerçek değerli genetik algoritma yaklaşımı ile zaman pencere çok araçlı dağıtım toplamalı rotalama problemini çözmüşler ve sonuçlarını literatürdeki mevcut genetik algoritma sonuçları ile karşılaştırma yapmışlardır.

Gülsoy [28], çalışması sonucunda geliştirdiği SZPARP için av arama algoritmasını literatürde sıkça kullanılan test problemlerinden Solomon'un problem setleri üzerinde çözümlenmiştir. Iqbal vd. [29], çok amaçlı EZPARP için yapay arı kolonisine dayalı bir hibrid meta-sezgisel algoritma geliştirmişlerdir. Çetin ve Gencer [30], bir sezgisel yöntem kullanarak sıkı zaman pencere eş zamanlı toplamalı ve dağıtım ARP 'nin çözümünü yapmışlardır.

Kümeleme yönteminin kullanıldığı ARP literatürü incelendiğinde, Dondo ve Cerda [24] tarafından çok depolu heterojen araç filolu zaman pencere ARP'nin çözümü için sezgisel tabanlı kümeleme algoritması geliştirmişlerdir. Hiquebran vd. [31], önce kümele sonra rotala yöntemiyle benzetimli tavlama algoritmasını ARP için çözümlenmişlerdir. Crainic vd. [32], iki aşamalı ARP için kümeleme tabanlı sezgisel algoritma kullanmışlardır. Çalışkan [33], karınca kolonisi optimizasyonu ile araç rotalama probleminin maliyetlerinin iyileştirilmesi için K-means kümeleme algoritmasını kullanmıştır. Boyzer vd. [34], kapasite kısıtlı araç rotalama problemini K-means kümeleme algoritmasına göre kümeleyip meta-sezgisel bir algoritma yardımıyla çözmüşlerdir. Şen [35], kapasite kısıtlı araç rotalama probleminin çözümü için ise DBSCAN kümeleme algoritması ile yeni geliştirdiği

genetik algoritma destekli DBSCAN kümeleme algoritmasını uygulayıp karşılaştırmıştır. Çalışmamız, Dondo ve Cerda [24], Hiquebran vd. [31], Crainic vd. [32], Çalışkan [33], Boyzer vd. [34] ve Şen [35]'nin çalışmalarına ZPARP'nin çözümü için kümeleme analizini kullanmaları açısından benzerlik göstermektedir. Bu çalışmalarda da kümeleme algoritmaları kullanılmasına rağmen araçların kapasitesini kontrol etmeden yapılan kümeleme sonrasında sezgisel bir algoritmaya geçiş yapmışlardır. Bununla birlikte, yaklaşımımızda, büyük boyutlu bir gerçek problemin çözümünde araç kapasitesini kontrol ederek gerçekleştirilen kümeleme ve kesin çözüm yöntemi kullanılmasına olanak sağlanmıştır.

Araştırmalarımıza göre, kapasite kontrollü kümeleme ve karışık tam sayılı doğrusal programlama kullanılarak ZPARP için güzergâh belirleyen hiçbir çalışma yapılmamıştır. Kümeleme tekniğinin kullanılmasının amacı, bir değerlendirme yaparken gerçek yaşam problemini göz önüne alabilmektir. Bu özelliğin kullanılması ile ileri sürdüğümüz yaklaşımın önemli bir katkı sağlayacağını düşünüyoruz.

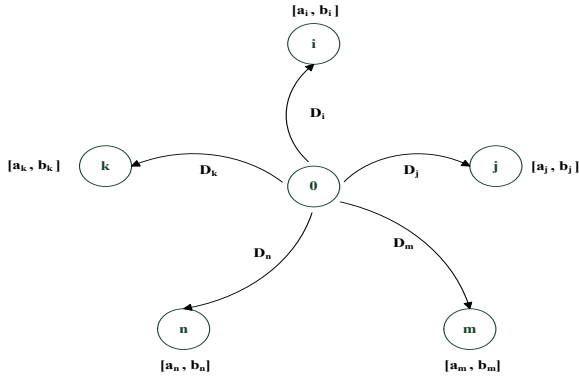
### 3. Araç Rotalama Problemi

Günümüzde lojistik sistemleri, müşterilerin sürekli artan talepleri ve değişen ürün özellikleri ile karmaşık bir hale gelmiştir. Bu da daha etkili dağıtım planları yapılmasını gerektirmektedir. Dağıtım yapan bütün işletmeler araç rotalama problemi ile karşılaşmaktadırlar. Bu problem işletmelere dağıtım maliyetine sebep olmaktadır. Bu dağıtım maliyetleri, ürün maliyetinin yaklaşık %15-20'sini oluşturmaktadır [36]. Standart bir ARP'de araçlarla farklı noktalardaki müşteri talepleri karşılanmaya çalışılır. Bu işlemde amaç müşteri ihtiyaçlarını en kısa yoldan, en kısa zamanda ve en az maliyetle karşılayan rotanın belirlenmesidir.

ARP, kapasitesi belirli araçlarla, belirli müşterilere hizmet vermek için en uygun rotanın atanmasını kapsar. Klasik ARP çözümü her rotanın depoda başlayıp depoda bittiği ve her müşteriye sadece bir kez uğrandığı rotalar kümesidir. Bunun yanında, problemin türüne göre bazı kısıtların da sağlanması gerekmektedir. Bu kısıtlara örnek olarak kapasite kısıtı, talep noktası kısıtı, bir aracın katettiği toplam süre kısıtı ve zaman penceresi kısıtı verilebilir. Farklı durumlar için geliştirilmiş çok sayıda ARP ile karşılaşılmaktadır. Dikkate alınan önemli kısıtlara göre ARP türlerinden birkaçı: kapasite kısıtlı, mesafe kısıtlı, önce dağıtım sonra toplama, eş zamanlı toplama-dağıtım, bölünmüş dağıtım, çok depolu, periyodik ve zaman pencere ARP'dir [37]. Bu çalışmada ZPARP'nin bir çeşidi olan sıkı zaman pencere araç rotalama problemi çalışılmıştır.

ZPARP ilk olarak 1984 yılında Hax ve Candea tarafından lojistik sistemlerde sınıflandırılmıştır [38]. ZPARP depodan hareket eden araçların, müşterileri

belli  $[a_i, b_i]$  zaman aralığı içinde ziyaret etme zorunluluğu olan (zaman penceresi kısıtı) özel bir araç rotalama problemi türüdür.  $[a_i]$  i. müşteri için en erken,  $[b_i]$  ise en geç servise başlama zamanıdır. Bu problem yapısında amaç, toplam maliyetlerin minimizasyonuna ek olarak müşteriye belirli zaman aralığında hizmet etmektir [39]. ZPARP'nin genel yapısı Şekil 1'de gösterilmiştir.

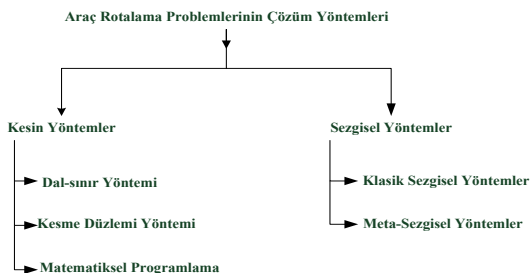


Şekil 1. ZPARP'nin genel yapısı

ZPARP esnek zaman pencere ve sıkı zaman pencere araç rotalama problemi olmak üzere ikiye ayrılır. Esnek zaman pencere ARP'de, müşterilere belirli  $[a_i, b_i]$  zaman pencerelerinin dışında hizmet verilebilir, fakat bu durumda bir ceza maliyeti vardır. Deponun zaman penceresi ise mutlaka sağlanmalıdır [25]. SZPARP ilk olarak Christofides vd. tarafından 1981 yılında çalışılmıştır. SPARP'nin çalışıldığı literatürde çok sayıda çalışma bulunmamakla birlikte, son dönemde Çetin ve Gencer, kesin zaman pencere eş zaman dağıtım toplamalı araç rotalama probleminin matematiksel modeli üzerinde çalışmışlardır [9]. Eğer bir araç müşterinin söz konusu zaman aralığının başlangıcından önce gitmişse en erken hizmete başlama zamanına kadar beklemek zorundadır, eğer en geç hizmete başlama zamanından sonra gitmişse de hizmet verememektedir.

### 3.1. Araç rotalama problemlerinin çözüm yöntemleri

NP-zor bir problem olan ARP'nin çözümü için literatürde çok sayıda yöntem bulunmaktadır. Bu yöntemler, en iyi çözümü veren kesin yöntemler ve sezgisel algoritmaları içeren yaklaşık yöntemler olmak üzere 2 ana gruba ayrılabilir. Şekil 2'de ARP'nin çözüm yöntemleri gösterilmektedir.



Şekil 2. ARP çözüm yöntemleri

Bu çalışmada ele alınan problem SZPARP'dir. SZPARP'de, eğer araç müşteriye zaman aralığının başlangıcından önce ulaşmış ise aracın  $[a_i]$  en erken hizmete başlama zamanına kadar bekleme kısıtı bulunmaktadır. Kesin çözüm yöntemlerinden düğüm tabanlı doğrusal karar modeli (DTM) bu kısıtı içerdiği için problemin çözümünde tercih edilmiştir.

#### 3.1.1. Düğüm tabanlı doğrusal karar modeli dizin kümesi

$V = \{0, 1, \dots, n\}$  düğümler (şehirler, müşteriler) kümesi, 0 ana depodur ve  $i, j \in V, i \neq j$

#### Parametreler

$a_i$  : i'inci düğümün en erken hizmete başlama zamanı  
 $b_i$  : i'inci düğümün en geç hizmete başlama zamanı  
 $[a_i, b_i]$  : i'inci düğümün zaman penceresi  
 $t_{ij}$  : i'inci düğümde j'inci düğüme geçiş süresi

#### Karar Değişkenleri

$x_{ij}$  : araç i'inci düğümden j'inci düğüme gidiyorsa  
 1, gitmiyorsa 0  
 $t_i$  : aracın i'inci düğüme geldiği an  
 $w_i$  : i'inci düğümde hizmet öncesi bekleme süresi  
 $s_i$  :  $t_i + w_i$ , başlangıçtan i'inci düğüme hizmet verilene kadar geçen süre  
 $T_1$  : toplam seyahat süresi  
 $T_2$  : beklemelerde geçen toplam süre

#### Amaç fonksiyonu

$$\text{Enk } Z = T_1 + T_2 \quad (1)$$

#### Kısıtlar

$$\sum_{i=0}^n x_{ij} = 1 \quad j=0, 1, \dots, n \quad (2)$$

$$\sum_{j=0}^n x_{ij} = 1 \quad i=0, 1, \dots, n \quad (3)$$

$$T_1 = \sum_{i=0}^n \sum_{j=0}^n t_{ij} x_{ij} \quad (4)$$

$$T_2 = \sum_{i=1}^n w_i \quad (5)$$

$$s_i + t_{i0} x_{i0} \leq T_1 + T_2 \quad i=1, \dots, n \quad (6)$$

$$s_i \geq a_i \quad i=1, \dots, n \quad (7)$$

$$s_i \leq b_i \quad i=1, \dots, n \quad (8)$$

$$t_i - t_{0i} x_{0i} \geq 0 \quad i=1, \dots, n \quad (9)$$

$$t_i + (b_i - t_{0i}) x_{0i} \leq b_i \quad i=1, \dots, n \quad (10)$$

$$s_i = t_i + w_i \quad i=1, \dots, n \quad (11)$$

$$s_i - t_j + (b_i - t_{0j} + t_{ij}) x_{ij} \leq b_i - t_{0j} \quad i, j=1, \dots, n, i \neq j \quad (12)$$

$$t_j - s_i + (b_j - a_i + t_{ij}) x_{ij} \leq b_j - a_i \quad i, j=1, \dots, n, i \neq j \quad (13)$$

$$t_i \geq 0 \quad i \neq j \quad i=1, \dots, n \quad (14)$$

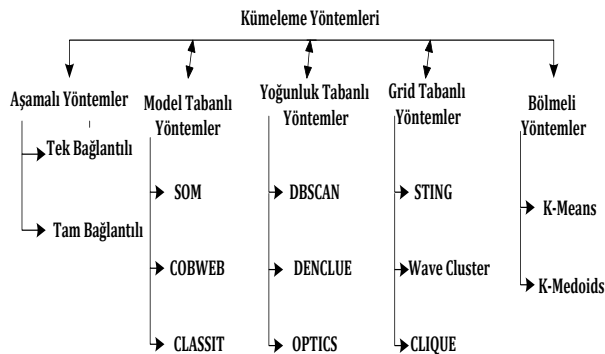
$$s_i \geq 0 \quad i=1, \dots, n \quad (15)$$

$$x_{ij} \in \{0,1\} \quad (16)$$

(2) ve (3) no'lu kısıtlar modelin atama kısıtlarıdır yani her düğüme tek bir düğümden gelinip ve bu düğümden de tek bir düğüme gidilmesini sağlamaktadır. (4) no'lu eşitlikteki T1, tur üzerindeki düğümlerde geçen sürelerin toplamını, (5) no'lu eşitlikteki T2, bekleme sürelerinin toplamını ifade eder. (6) no'lu kısıt aracın bekleme zamanları da dâhil olmak üzere depodan çıktığı zamandan tekrar depoya döndüğü zamana kadar geçen süreyi sıkıştırır. (7) ve (8) nolu kısıtlar, her düğüme hizmet verildiği an, düğüme en erken ve en geç geliş zamanları (zaman penceresi) arasında kalması gerektiğini ifade eder. Depodan ilk çıkış düğümü ve bu arada geçen süre ile tanımlanan karar değişkenleri göz önüne alınarak (9) ve (10) no'lu kısıtlar elde edilmiştir. Bu kısıtlar ZPARP için sınırlandırılan kısıtlardır. (11) no'lu kısıt aracın i'inci düğüme hizmet verene kadar geçen süre, i'ninci düğüme geliş anı ile i'nci düğüme hizmet verene kadar geçen bekleme süresinin toplamı olduğunu ifade eder. (12) ve (13) nolu kısıtlar alt turların oluşmasını engelleyen kısıtlardır. (14) ve (15) no'lu kısıtlar ( $y_i$ ) ve ( $s_i$ ) karar değişkenlerinin sıfırdan küçük olmaması gerektiğini ifade eder. (16) no'lu kısıt ( $x_{ij}$ ) karar değişkeni için 0-1 tamsayı kısıtlarıdır. Amaç fonksiyonu tur üzerindeki düğümlerde geçen süre ile bekleme sürelerinin toplamını en küçükler [40].

#### 4. Kümeleme Analizi

Kümeleme yöntemi, veriyi gruplara (kümelere) bölmeyi amaçlayan bir gözetimsiz öğrenme türüdür. Veriler grup içerisindeki elemanların kendi içerisindeki benzerliği ve grupların birbirine benzememesi koşuluyla birbirinden ayrılmaktadır. Kümelemede asıl amaç sınıflar arasındaki benzerliği en aza indirmek, sınıfların kendi içerisindeki benzerliğini ise en yüksek seviyede tutmaktır. Bu çalışmada kümeleme yöntemleri olarak K-Medoids ve DBSCAN algoritmaları kullanılmıştır. Şekil 3'te en yaygın kullanılan kümeleme yöntemleri gösterilmiştir.



Şekil 3. En yaygın kullanılan kümeleme yöntemleri

#### 4.1. K-medoids kümeleme algoritması

K-Medoids kümeleme algoritması K-Means kümeleme algoritmasının gürültülü ve aykırı değerlere karşı olan aşırı duyarlılığını gidermek amacıyla 1990 yılında Kaufman ve Rousseeuw tarafından geliştirilmiştir [41]. Algoritmada kümeyi temsil edecek noktayı bulmak için kümenin merkez noktasındaki eleman yeni küme merkezi olarak alınır ve böylece bazı verilerin küme merkezini kaydırma ihtimali ortadan kalkmış olur.

K-Medoids algoritmasının birçok farklı türü bulunmaktadır. İlk ortaya atılan K-Medoids algoritması PAM (Partitioning Around Medoids)'dır. PAM algoritmasında, rastgele seçilen k tane sayı kümenin merkezi olarak seçilir. Kümeye her yeni eleman eklendiğinde kümenin elemanları deneyerek kümenin gelişmesine en çok katkı sağlayacak noktayı tespit edince bulunan bu noktayı yeni merkez, eski merkezi ise sıradan küme elemanı olacak şekilde yer değiştirilmesi yapılır. PAM küçük veri tabanlarında iyi sonuçlar vermektedir fakat karmaşık veri tabanlarında zayıf performans gösterir ifade edilmektedir [42]. K-medoids algoritmasının sözde kodu Tablo 1'de verilmiştir.

Tablo 1. K-Medoids Algoritmasının Sözde Kodu

```

başla:
  K küme sayısını belirle
  Başlangıç medoidleri olarak k nesnelere seç
  while (değişiklik varsa)
  do
  {
  En yakın medoid x'e sahip kümeye, kalan nesnelere
  ata
  Amaç fonksiyonunu hesapla
  Tesadüfi olarak medoid olmayan y noktasını seç
  } end while

  Eğer x ile y'nin yer değiştirmesi amaç fonksiyonunu
  minimize edecekse x ile y'nin yerini değiştir
  
```

#### 4.2. DBSCAN kümeleme algoritması

DBSCAN kümeleme algoritması, Ester vd. tarafından 1996 yılında geliştirilen yoğunluk tabanlı bir kümeleme algoritmasıdır [43]. DBSCAN kümeleme algoritmasında veri kümesinde bulunan herhangi bir nokta için yoğunluk o noktanın belirli bir mesafesinde bulunan en büyük komşuluk yarıçapı ve Eps yarıçapındaki nokta sayısına bakılarak hesaplanmaktadır. Bu sayı noktanın kendisinde içermektedir [44].

DBSCAN algoritmasının Eps ve MinPts olmak üzere iki parametresi bulunmaktadır. Eps: En büyük komşuluk yarıçapını, MinPts ise Eps yarıçaplı komşuluk bölgesinde bulunan en az nesne sayısını temsil etmektedir. Algoritmada, Eps ve MinPts değerleri giriş parametresi olarak alınır. Veri tabanındaki herhangi bir nesneden başlanarak tüm nesnelere kontrol edilir. Eğer kontrol edilen nesne

daha önce bir kümeye dâhil edilmiş ise işlem yapılmadan diğer nesneye geçilir. Eğer nesne daha önce kümelanmemiş ise, bir bölge sorgusu yapılarak nesnenin Eps komşuluğundaki komşuları bulunur. Komşu sayısı MinPts'den fazla ise, bu nesne ve komşuları yeni bir küme olarak adlandırılır. Daha sonra, önceden kümelanmemiş her bir komşu için yeni bölge sorgusu yapılarak yeni komşular bulunur. Bölge sorgusu yapılan noktaların komşu sayıları MinPts'den fazla ise kümeye dâhil edilir [45].

DBSCAN, özellikle büyük veritabanları ve gürültülü nesnelere içeren veriler için oldukça kullanışlıdır. Bunun yanısıra farklı büyüklükteki ve şekillerdeki kümelerin belirlenmesinde de sıkça kullanılmaktadır [46].

DBSCAN algoritmasının sözde kodu Tablo 2'de verilmiştir.

**Tablo 2.** DBSCAN Algoritmasının Sözde Kodu

başla:

Veri setindeki her bir elemanın Eps yarıçaplı komşuluğunu araştır  
Bu bölgede MinPts'den daha fazla nesne bulunan p nesnesi çekirdek nesne olacak şekilde kümeler belirle  
Yoğunluk bağlantılı kümeleri birleştir  
Eğer hiçbir yeni nesne bir kümeye eklenmez ise işlemi sona erdir

## 5. Uygulama

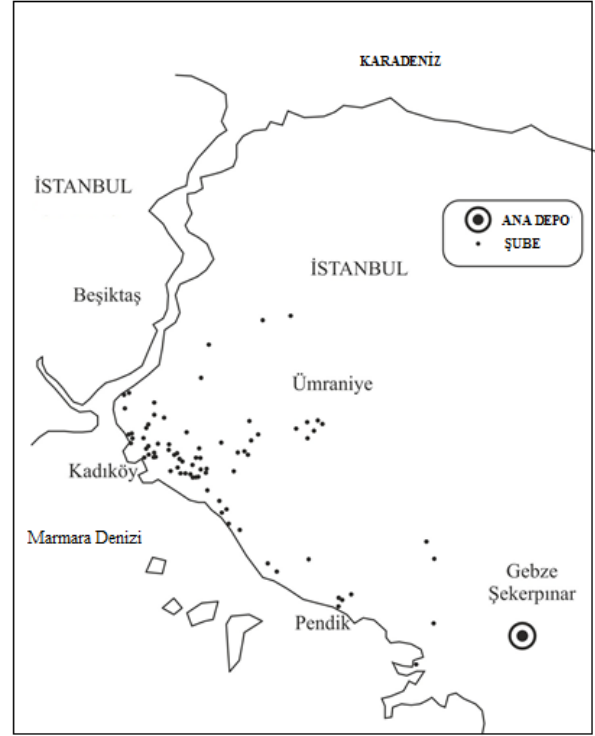
### 5.1. Problemin tanımı

Uygulama yapılan firma Türkiye genelinde faaliyet gösteren ve perakende ürün satışı yapan bir süpermarket zinciridir. Şubelere yapılan dağıtım operasyonu sadece hafta içi günlerde gerçekleştirilmektedir. Süpermarket zincirinin bir adet ana deposu ve 78 adet şubesi bulunmaktadır. Ana depo ve şubelerin iki boyutlu düzlemdeki görüntüsü Şekil 4'te gösterilmiştir. Şubelerin geçmiş üç aylık taleplerinin ortalamaları alınarak 21 haftalık talep verileri elde edilmiştir. Firmanın araç filosu homojen araçlardan oluşmaktadır ve hepsi maksimum 40 palet kapasitelidir. Bununla birlikte firma 32 paletin altındaki yüklü araçları ürün dağıtımına göndermek istememektedir. Bu nedenle şubelerden oluşturulacak kümelerin toplam kapasiteleri 32 ile 40 palet arasında olmalıdır. Şubelerin firma ile yapmış oldukları görüşmeler sonucunda her şube için en erken ve en geç teslim zamanları  $[a_i, b_i]$  belirlenmiştir. Tablo 3'te şubelere ait en erken ve en geç servise başlama zamanları verilmiştir. Şubelere erken gelme durumunda, araç en erken teslim zamanına kadar beklemektedir, geç gelme durumunda müşteri servisi kabul etmemektedir.

### 5.2. Problemin çözümü

Bu çalışmada ele alınan dağıtım problemi iki aşamalı olarak çözülmüştür. İlk aşamada şubelerin haftalık

talep yoğunlukları göz önünde bulundurularak bölmelikli yöntemlerden K-Medoids, yoğunluk tabanlı yöntemlerden ise DBSCAN kümeleme algoritması yardımıyla mesafeye göre kümelanmıştır.



**Şekil 4.** Ana Depo ve Şubelerin Buldukları Yerler

**Tablo 3.** Şubelere Ait En Erken ve En Geç Servise Başlama Zamanları

Ş	$[a_i]$ (dk)	$[b_i]$ (dk)	Ş	$[a_i]$ (dk)	$[b_i]$ (dk)	Ş	$[a_i]$ (dk)	$[b_i]$ (dk)
1	10	40	27	264	274	53	394	404
2	306	320	28	90	100	54	217	227
3	237	247	29	299	309	55	407	417
4	179	189	30	136	146	56	94	104
5	409	419	31	324	334	57	137	147
6	291	301	32	341	351	58	80	90
7	274	284	33	369	379	59	75	85
8	390	400	34	226	236	60	200	210
9	147	157	35	80	90	61	230	250
10	373	383	36	120	130	62	274	284
11	372	382	37	412	422	63	90	100
12	193	203	38	83	93	64	419	429
13	283	293	39	406	416	65	309	319
14	46	56	40	268	278	66	178	188
15	36	46	41	420	430	67	256	266
16	265	275	42	48	58	68	223	233
17	384	394	43	222	232	69	46	56
18	428	438	44	61	71	70	125	135
19	72	82	45	432	442	71	69	79
20	283	293	46	12	42	72	140	150
21	235	245	47	382	392	73	155	165
22	46	56	48	402	412	74	210	220
23	172	182	49	427	437	75	34	54
24	88	98	50	61	71	76	423	433
25	391	401	51	222	232	77	199	209
26	159	169	52	135	145	78	246	276

Ş:Şubeler

İkinci aşamada ise sıkı zaman pencereli ve homojen kapasiteli araç rotalama probleminin çözümü için

kurulan Düşüm Tabanlı Doğrusal Karar Modeli yardımıyla, oluşturulan kümeler içinden şubelerin en erken ve en geç teslim zamanları dikkate alınarak en uygun rotalar belirlenmiştir. Çalışmanın sonucunda ise bu iki algoritma ile elde edilen sonuçlar ve firmadan alınan gerçek değerler ANOVA testi yapılarak karşılaştırılmıştır.

Ele alınan problemin 1. haftadaki talep miktarları dikkate alınarak detaylı çözümü aşağıda verilmiştir. Kalan 21 hafta sonuçları ve gerçek değerler Tablo 8'de özetlenmiştir.

### 5.2.1. Kapasite kontrollü kümeleme

#### 5.2.1.1. K-medoids algoritmasının işleyişi

K-Medoids algoritmasında kümeyi temsil edecek noktayı bulmak için kümenin merkez noktasındaki eleman yeni küme merkezi olarak alınır. Uygun olarak seçilen k adet sayı küme merkezi olarak alınır. Kümeye her yeni eleman eklendiğinde kümenin elemanları denenerak kümenin gelişmesine en çok katkı sağlayacak noktayı tespit edince bulunan bu noktayı yeni merkez, eski merkezi ise sıradan küme elemanı olacak şekilde yer değiştirilmesi yapılır.

K-medoids algoritmasında ilk olarak küme sayısı belirlenmektedir. Bunun için bir kural yoktur. Elimizdeki verilere göre veya uygulamayı yapan kişiye bağlı olarak değişmektedir. Küme sayısı belirlendikten sonra, K-medoids algoritması ile kümeler oluşturulur. Oluşturulan kümelerin talep miktarları toplanır ve kapasite kısıtının sağlanıp sağlanmadığı kontrol edilir. Kapasite kısıtını sağlayan kümeler uygun küme olarak kaydedilir. Kapasite kısıtını sağlamayan kümelerdeki şubeler tüm şubelerden çıkartılır ve tüm şubeler bir kümeye ait olana kadar bu işlem devam ettirilir. Kapasite kontrollü kümeleme işlemi sonrası SZPARP'de uygun çözüm bulunmazsa kümeleme işlemi tekrarlanır. K-Medoids algoritması sonucunda oluşan kümeler, kümelerdeki şubeler ve talep miktarları Tablo 4' te gösterilmiştir.

**Tablo 4.** K-Medoids Algoritması Sonucunda Oluşan Kümeler ve Uğranacak Şubeler

Araçlar	Araçların uğrayacağı şubeler	Palet toplamları
1	30-52-53-54-56-57	34
2	4-5-6-67-69-70-71	39
3	14-72-73-75-77-78	32
4	15-16-27-40-43-47-48	39
5	13-24-26-32-34-35-37-74	37
6	17-18-22-41-42-44-45	33
7	20-23-25-28-31-33-36-39-46	39
8	1-2-3-19-29-76	40
9	49-50-58-59-61-62	38
10	7-8-9-10-11-66	38
11	12-21-38-55-68	35
12	51-60-63-64-65	35

#### 5.2.1.2. DBSCAN algoritmasının işleyişi

DBSCAN algoritmasının temel çalışma prensibi, küme içindeki her nokta merkez kabul edildiğinde, bu noktalar için, verilen bir yarıçap etrafında minimum noktadan oluşan küme oluşturmaktır [43].

DBSCAN algoritmasına göre kümedeki her nokta için, o noktanın Eps komşuluğundan en az MinPts kadar sayıda nokta olması gereklidir. MinPts değeri bir kümede olması gereken minimum nokta sayısını ifade ettiğinden talep verileri ve araç kapasitelerinin 32-40 arasında değiştiği durumu göz önüne alınarak 4 olarak hesaplanmıştır. Algoritma çalıştırılırken MinPts değeri sabit tutulup Eps değerleri 1'den başlanarak uygun küme oluşana kadar denenerak DBSCAN algoritması ile kümeler oluşturulur. Oluşturulan kümelerin talep miktarları toplanır ve kapasite kısıtının sağlanıp sağlanmadığı kontrol edilir. Kapasite kısıtını sağlayan kümeler uygun küme olarak kaydedilir. Kapasite kısıtını sağlamayan kümelerdeki şubeler tüm şubelerden çıkartılır ve tüm şubeler bir kümeye ait olana kadar bu işlem devam ettirilir. Kapasite kontrollü kümeleme işlemi sonrası SZPARP'de uygun çözüm bulunmazsa kümeleme işlemi tekrarlanır. DBSCAN algoritması sonucunda oluşan kümeler, kümelerdeki şubeler ve talep miktarları Tablo 5' te gösterilmiştir.

**Tablo 5.** DBSCAN Algoritması Sonucunda Oluşan Kümeler ve Uğranacak Şubeler

Araçlar	Araçların uğrayacağı şubeler	Palet toplamları
1	7-9-60-63-64-65	40
2	30-52-53-54-56-57	34
3	3-8-58-69-73	35
4	49-50-51-59-61	35
5	10-11-12-66-68-70	38
6	15-16-27-40-43-47-48	39
7	13-14-24-32-34-35-37-74	36
8	1-2-62-67-71-76	36
9	6-26-55-72-75-77-78	33
10	4-5-17-18-19-29	38
11	20-21-23-25-36-41-45-46	40
12	22-38-31-33-38-39-42-44	34

### 5.2.2. Rotalama

Uygulamanın ilk aşaması sonucunda elde edilen kümelerin her birinin rotaları, Tablo 2'de verilen şubelerin tanımladığı en erken ve en geç teslim zamanları dikkate alınarak ana depodan başlayıp yine ana depoda son bulacak şekilde kurulan düşüm tabanlı doğrusal karar modeli yardımıyla oluşturulur ve her rota için toplam zamanlar hesaplanır. Bu hesaplama işleminde hizmet süreleri "0" olarak alınmıştır. Rotalama işlemi Intel Core i3-3220 CPU 3,30 GHz işlemci ve 4.00 GB RAM özelliğine sahip bilgisayarda LINGO 16.0 yazılımı kullanılarak gerçekleştirilmiştir. Tablo 6 ve Tablo 7'de araçların izleyeceği rotalar ve harcadıkları toplam zamanlar verilmiştir. Bu oluşturulan rotalardaki 79 numaralı şube ana depoyu temsil etmektedir.

**Tablo 6.** K-Medoids Algoritması Sonucunda Araçların İzleyeceği Rotalar ve Toplam Zamanlar

Araçlar	Araçların izleyeceği rotalar	Toplam süre(dk)
1	79-30-52-53-54-56-57-79	430,5
2	79-4-5-6-67-69-70-71-79	445,7
3	79-14-72-73-75-77-78-79	294,1
4	79-15-16-27-40-43-47-48-79	449,5
5	79-13-24-26-32-34-35-37-74-79	459,1
6	79-17-18-22-41-42-44-45-79	475,4
7	79-20-23-25-28-31-33-36-39-46-79	448
8	79-1-2-3-19-29-76-79	468,2
9	79-49-50-58-59-61-62-79	459
10	79-7-8-9-10-11-66-79	439,7
11	79-12-21-38-55-68-79	450,1
12	79-51-60-63-64-65-79	433
Toplam süre		5252,3

**Tablo 7.** DBSCAN Algoritması Sonucunda Araçların İzleyeceği Rotalar ve Toplam Zamanlar

Araçlar	Araçların izleyeceği rotalar	Toplam süre(dk)
1	79-7-9-60-63-64-65-79	433
2	79-30-52-53-54-56-57-79	430,5
3	79-3-8-58-69-73-79	439,7
4	79-49-50-51-59-61-79	459
5	79-10-11-12-66-68-70-79	412,8
6	79-15-16-27-40-43-47-48-79	449,5
7	79-13-14-24-32-34-35-37-74-79	459,1
8	79-1-2-62-67-71-76-79	468,2
9	79-6-26-55-72-75-77-78-79	450,1
10	79-4-5-17-18-19-29-79	470,6
11	79-20-21-23-25-36-41-45-46-79	475,4
12	79-22-38-31-33-38-39-42-44-79	448
Toplam süre		5395,9

**Tablo 8.** K-Medoids, DBSCAN Algoritmalarının ve Gerçek Değerlerin Toplam Zamanı

Hafta	DBSCAN Toplam Süre (dk)	K-Medoids Toplam Süre (dk)	Gerçek Değerler
1	5395,9	5252,3	5410,5
2	5223,9	5393	5509,2
3	5110,5	5239,6	5300,7
4	5127,8	5572,1	5612,5
5	5164,7	5722,9	5803,6
6	5412,8	5839,3	5901,5
7	5159,2	5252,3	5615,4
8	5193,4	5688,4	5713,4
9	5251,4	5366,7	5445,2
10	5211,8	5521	5600,9
11	5554,5	5239,2	5614,1
12	5240,5	5657,8	5744,4
13	5074,5	5560	5714,2
14	5192,8	5299,3	5212,4
15	5103,2	5044	5000
16	5072,4	5181,1	5278,3
17	5371	5176,9	5341,2
18	5088,7	5717,7	5232,1
19	5222	5670,1	5746,8
20	5566,2	5416,6	5547,6
21	5599,5	5455,2	5423,5

**Tablo 9.** Varyansların Homojenliği Testi

Levene Statistic	df1	df2	Sig.
1,943	2	60	,152

### 5.3. ANOVA Testi

Varyans analizi (ANOVA) bir ya da daha fazla bağımsız değişkenin ikiden fazla gruptaki ortalamalarını karşılaştırmak için kullanılır. Tablo 7'deki sonuçlara ANOVA testi uygulanmış ve SPSS sonuçları aşağıda verilmiştir. Bu çalışmada;

$H_0: \mu_1 = \mu_2$  (İki algoritma ile bulunan toplam sürelerin ve gerçek değerlerin ortalaması arasında fark yoktur.)

$H_1: \mu_1 \neq \mu_2$  (İki algoritma ile bulunan toplam sürelerin ve gerçek değerlerin ortalaması farklıdır.)

Olacak şekilde hipotezler tanımlanmıştır.

ANOVA testi uygulandığında ilk olarak varyansların homojen olup olmadığı incelenmektedir. Tablo 9'daki p (sig.) değeri  $0,152 > 0,05$  olduğundan dolayı grup varyanslarının homojen olduğu sonucuna varılır. Grup varyansları homojen olduğu için Tablo 10'daki F testinin sonuçları anlamlı olacaktır. Varyans analizi tablosundaki p (sig.) değeri  $0,000 < 0,05$  olduğundan dolayı  $H_1$  hipotezi kabul edilir. Buna göre iki algoritma ile elde edilen sonuçların ve gerçek değerlerin arasında istatistiksel olarak anlamlı bir fark vardır diyebiliriz. Tablo 11'de ele alınan problemin iki algoritma ile çözümü sonucunda bulunan toplam sürelerin ve gerçek sürelerin ortalaması verilmiştir. "1" sıradaki K-Medoids, "2" sıradaki DBSCAN ve "3" sıradaki ise gerçek değerleri temsil etmektedir. Toplam sürelerin ortalaması, K-Medoids algoritması ile 5441,2381, DBSCAN algoritması ile 5254,1286 dk ve gerçek sürelerin ortalaması ise 5512,7381 dk olarak bulunmuştur. Bu sonuçlar bize DBSCAN algoritmasının daha iyi sonuç verdiğini göstermiştir.

### 6. Sonuç

Bu çalışmada, lojistik sisteminin temelini oluşturan araç rotalama problemlerinden, SZPARP ele alınmıştır, veriler K-Medoids ve DBSCAN algoritmalarıyla kümelendirilmiştir. Her kümenin sonucunda oluşan rotaların toplam beklemeleri ve seyahat sürelerini minimize etmeyi amaçlayan bir matematiksel model kurularak çözülmüştür. Firmanın gerçek verileri ve iki algoritma ile elde edilen sonuçlar ANOVA testi ile karşılaştırılmıştır. Toplam sürelerin ortalaması 5441,2381 dk (K-medoids algoritması), 5254,1286 dk (DBSCAN algoritması) ve 5512,7381 dk (firma verileri) olarak bulunmuştur. Bu sonuçlar bize DBSCAN algoritmasının daha iyi sonuç verdiğini göstermiştir. Bu olumlu sonuçlar nedeniyle, önerilen yaklaşımın uygulanması şirkete değer katacaktır.



**Tablo 10.** ANOVA Testi Sonucu

	Kareler toplamı	Df	Ortalama Karesi	F	Sig.
Gruplar arası	748969,239	2	374484,620	8,694	,000
Gruplar içi	2584458,218	60	43074,304		
Toplam	3333427,457	62			

**Tablo 11.** Tanımlayıcı İstatistikler

	N	Ortalama	Standart Sapma	Standart Hata
1	21	5441,2381	222,41203	48,53137
2	21	5254,1286	165,26380	36,06352
3	21	5512,7381	229,01883	49,97601
Toplam	63	5402,6937	231,87272	29,21322

## Kaynakça

- Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- [1] Dantzig, G. B., Ramser, J. H. 1959. The Truck Dispatching Problem. *Management Science*, 6, 80-91.
- [2] Han, J., Kamber, M. 2001. *Data Mining and Concepts Techniques*. San Francisco: Morgan Kaufmann Publishers Inc.
- [3] Laporte, G., Semet, F. 2002. Classical Heuristics for the Capacitated VRP. ss 109-128. Toth, P., Vigo, D., ed. 2002. *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia.
- [4] Clarke, G., Wright, J. W. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12, 568-581.
- [5] Dumas, J. E., Wolf, L. C., Fisman, S. N., Culligan, A. 1991. Parenting stress, child behavior problems, and dysphoria in parents of children with autism, Down syndrome, behavior disorders, and normal development. *Exceptionality*, 2, 97-110.
- [6] Aydemir, E. 2006. *Esnek Zaman Pencere Araç Rotalama Problemi ve Bir Uygulama*. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- [7] Boer, J. W. D. 2008. *Approximate Models and Solution Approaches for the Vehicle Routing Problem with Multiple Use of Vehicles and Time Windows*. Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Ankara.
- [8] Tezer, T. 2009. *Toplama ve Dağıtım Zaman Pencere Araç Rotalama Problemi için Kesin Çözüm Yaklaşımı ve Örnek Uygulamalar*. Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Balıkesir.
- [9] Çetin, S., Gencer, C. 2010. Kesin Zaman Pencere Eş Zamanlı Dağıtım Toplamalı Araç Rotalama Problemi: Matematiksel Model. *Gazi Üniv. Müh. Mim. Fak. Der.*, 25, 579-585.
- [10] Çetin, S., Gencer, C. 2011. Heterogeneous Fleet Vehicle Routing Problems with Time Windows and Simultaneous Pickup and Delivery: Mathematical Model. *International Journal of Research and Development*, 3, 19-27.
- [11] Çetinkaya, C. 2011. *İki Aşamalı Hat Zaman Pencere Araç Rotalama Problemi*. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- [12] Taş, D., Jabali, O., Woensel, T. V. 2014. A Vehicle Routing Problem with Flexible Time Windows. *Computers & Operations Research*, 52, 39-54.
- [13] Çetinkaya, C. 2014. *Terör bölgelerine yapılan askeri sevkiyatlar için yer seçimi ve hat zaman pencere araç rotalama problemi*. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Ankara.
- [14] Akca, K. 2015. *Hammadde Tedarik Aktivitesi için Kesin Zaman Pencere Araç Rotalama Optimizasyonu*. Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Bursa.
- [15] Liu, F. H., Shen, S. Y. 1999. A Method for Vehicle Routing Problem with Multiple Vehicle Types and Time Windows. *Proc. Natl. Sci. Coun. ROC(A)*, 23, 526-536.
- [16] Demircioğlu, M. 2009. *Araç Rotalama Probleminin Sezgisel Bir Yaklaşım ile Çözümlemesi Üzerine Bir Uygulama*. Çukurova Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Adana.
- [17] Şahin, M., Şahin, G., Çavuşlar, G., Özcan, T., Tüzün, D. 2010. Separable Weight Pickup and Delivery Problem Using Tabu Search Algorithm. *Operational research and Industrial Engineering 30th National Meeting*, Sabancı Üniversitesi, İstanbul.
- [18] Schulze, J., Fahle, T. 1999. A Parallel Algorithm for the Vehicle Routing Problem with Time Window Constraints. *Annals of Operations Research*, 86, 585-607.
- [19] Ho, S., Haugland, D. 2004. A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows and Split Deliveries. *Computers and Operations Research*, 31, 1947-1964.
- [20] Jiang, J., Ng, K. M., Poh, K. L., Teo, K. M. 2014. Vehicle Routing Problem with a Heterogeneous Fleet and Time Windows. *Expert Systems with Applications: An International Journal*, 41, 3748-3760.
- [21] Barán, B., Schaerer, M. 2003. A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows. *Proceeding of the 21st IASTED International Conference Applied Informatics*, 10-13 February, Innsbruck, Austria, 97-102.
- [22] Tokaylı, M. A. 2005. *Zaman Pencere Araç Rotalama Problemi için Bir Karar Destek Sistemi*.

- Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- [23] Bouthillier, A. L., Crainic, T. G. 2005. A Cooperative Parallel Meta Heuristic for the Vehicle Routing Problem with Time Windows. *Computers & Operations Research*, 32, 1685–1708.
- [24] Dondo, R., Cerda, J. 2007. A Cluster-Based Optimization Approach for the Multi-Depot Heterogeneous Fleet Vehicle Routing Problem with Time Windows. *European Journal of Operational Research*, 176, 1478–1507.
- [25] Dursun, P. 2009. Zaman Pencereli Araç Rotalama Probleminin Genetik Algoritma ile Modellenmesi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- [26] Nazif, H., Lee, L. S. 2010. Optimized Crossover Genetic Algorithm for Vehicle Routing Problem with Time Windows. *American Journal of Applied Sciences*, 7, 95-101.
- [27] Kiremitci, B., Kiremitci, S., Kesintürk, T. 2014. Zaman Pencereli Çok Araçlı Dağıtım Toplamalı Rotalama Problemi için Gerçek Değerli Genetik Algoritma Yaklaşımı. İstanbul Üniversitesi İşletme Fakültesi Dergisi, 43, 391-403.
- [28] Gülsoy, N. 2013. Av Arama Algoritması ile Sıkı Zaman Pencereli Araç Rotalama Problemlerinin Çözümü. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri.
- [29] Iqbal, S., Kaykobad, M., Rahman, M. S. 2015. Solving the Multi-Objective Vehicle Routing Problem with Soft Time Windows with the Help of Bees. *Swarm and Evolutionary Computation*, 24, 50–64.
- [30] Çetin, S., Gencer, C. 2015. A Heuristic Algorithm for Vehicle Routing Problems with Simultaneous Pick-Up and Delivery and Hard Time Windows. *Open Journal of Social Sciences*, 3,35-41.
- [31] Hiquebran, D. T., Alfa, A. S., Shapiro, J. A., Gittoes, D. H. 2007. A revised simulated annealing and cluster-first route-second algorithm applied to the vehicle routing problem. *Engineering Optimization*, 22, 77-107.
- [32] Crainic, T. G., Mancini, S., Perboli, G., Tadei, R. 2008. Clustering-based heuristics for the two-echelon vehicle routing problem. *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation*.
- [33] Çalışkan, K. 2011. Karınca Kolonisi Optimizasyonu ile Araç Rotalama Probleminin Maliyetlerinin Kümeleme Tekniği ile İyileştirilmesi. TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- [34] Boyzer, Z., Alkan, A., Fiğlalı, A. 2014. Cluster-First, Then-Route Based Heuristic Algorithm for the Solution of Capacitated Vehicle Routing Problem. *International Journal of Informatics Technologies*, 7, 29-37.
- [35] Şen, T. 2014. Kümeleme ve Genetik Algoritma Destekli Yaklaşımlarla Kapasite Kısıtlı Araç Rotalama Probleminin Çözümü: Perakende Zincirinde Uygulanması. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya.
- [36] Rushton, A., Croucher, P., Baker, P. 2006. *Handbook of Logistics and Distribution Management*. 3rd edn, Kogan Page, Limited.
- [37] Ho, W., Ho, G. T. S., Ji, P., Lau, H. C. W. 2008. A Hybrid Genetic Algorithm for the Multi-Depot Vehicle Routing Problem. *Engineering Applications of Artificial Intelligence*, 21, 548–557.
- [38] Hax, A. C., Candea, D. 1984. *Production and Inventory Management*. Prentice-Hall, Englewood Cliffs, NJ.
- [39] Tan, K. C., Lee, L. H., Zhu, Q. L., OU, K. 2001. Heuristic Methods for Vehicle Routing Problem with Time Windows. *Engineering Applications in Engineering*, 15, 281-295.
- [40] Koç, Ö. N. 2012. Zaman Pencereli Gezgin Satıcı Problemi için Yeni Karar Modelleri. Başkent Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- [41] Kaufman, L., Rousseeuw, P. 1987. Clustering by Means of Medoids. No. 87 in *Reports of the Faculty of Mathematics and Informatics*, Delft University of Technology.
- [42] Işık, M. 2006. Bölümlü Kümeleme Yöntemleri ile Veri Madenciliği Uygulamaları. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- [43] Ester, M., Kriegel, H. P., Sander, J., Xu, X. 1996. A Destiny-Based Algorithm for Discovering Clusters In Large Spatial Databases with Noise. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, 226-231.
- [44] Güven, A., Bozkurt, Ö., Kalıpsız, O. 2007. Veri Madenciliğinin Geleceği. IX. Akademik Bilişim Konferansı Bildirileri, Dumlupınar Üniversitesi, Kütahya.
- [45] Bilgin, T. T., Çamurcu, Y. 2005. Comparative Comparing of DBSCAN, OPTICS and K-Means Algorithms. *Journal of Polytechnic*, 8, 139-145.
- [46] Moreira, A., Santos, M. Y., Carneiro, S. 2005. Destiny-Based Clustering Algorithms-DBSCAN and SNN. University of Minho, Portugal.