



Kapasite kısıtlı araç rotalama probleminin çözümü için yeni bir algoritma geliştirilmesi: bir süpermarket zincirinde uygulanması

Tolga Şen^{1*}, Serap Ercan Cömert², Harun Reşit Yazgan³

08.04.2014 Geliş/Received, 29.05.2014 Kabul/Accepted

ÖZ

Bu çalışmada, bir süpermarket zincirindeki taleplerin karşılanmasında ortaya çıkan araç rotalama probleminin çözüm metotları üzerinde durulmuştur. Bu amaçla iki farklı yaklaşım geliştirilmiştir. Birincisinde bir kümeleme algoritması ile müşteriler kümelenecek ve araç rotalama problemi çözülmüştür. İkinci yaklaşımda ise, Genetik algoritma destekli bir yaklaşım geliştirilmiştir. Geliştirilen her iki metot bir örnek uygulamada test edilerek sonuçlar ANOVA testi ile karşılaştırılmıştır.

Anahtar Kelimeler: araç rotalama, kümeleme, genetik algoritma

A new developed algorithm for capacity constraint vehicle routing problem: a supermarket chain application

ABSTRACT

In this study, two approaches were developed to solve the vehicle routing problem (VRP) which were appeared from meeting demands of a supermarket chain. In the first approach, clustering algorithm was employed and then the VRP was solved within each clusters. In the second one, Genetic Algorithm (GA) was employed to classify the customers and then the problem was solved again. The approaches' results were compared by the ANOVA test to illustrate superiority of the approaches.

Keywords: vehicle routing, clustering, genetic algorithm

* Sorumlu Yazar / Corresponding Author

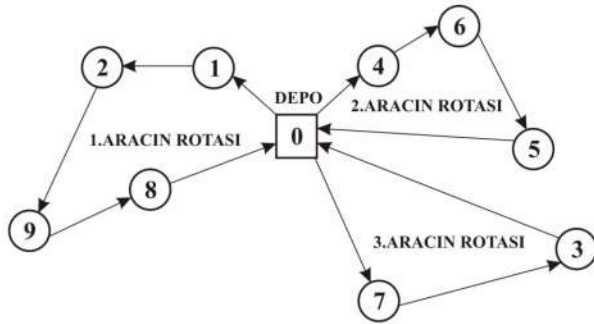
1 Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği, Sakarya - tolga_sen53@hotmail.com

2 Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği, Sakarya - serape@sakarya.edu.tr

3 Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği, Sakarya - yazgan@sakarya.edu.tr

1. GİRİŞ (INTRODUCTION)

Araç rotalama problemi (ARP), coğrafi olarak dağınık merkezlerle bir veya birden fazla depodan hizmet vermek üzere görevlendirilen araçların en iyi dağıtım/toplama rotalarının planlanması problemidir [1]. ARP gezgin satıcı problemlerinin genel halidir. İlk defa 1959 yılında Dantzig ve Ramster tarafından çalışılmıştır. ARP' de gezgin satıcı problemindeki müşterilerin yerini sipariş noktaları, satıcının yerini ise araçlar almıştır. ARP'nin genel yapısı Şekil 1'de gösterilmiştir. Şekil 1'de gösterilen 1'den 9'a kadar numaralanmış düğümler sipariş noktalarını, 0 ise ana depoyu ifade etmektedir.



Şekil 1. ARP örnek gösterimi (TSP's example notation)

ARP 'ler sahip olunan kısıtlar ile birbirlerinden farklılık göstermektedir. Müşterilere yapılacak dağıtım parçalanarak yapılabilir ya da her müşterinin hizmet kabul edebileceği belirli bir zaman aralığı olabilir. Bu gibi farklılıklar söz konusu çeşitliliğe sebep olmaktadır. Bu çalışmada uygulamasını yaptığımız ARP kapasite kısıtlı ARP'dir.

Dağıtım ağı eniyilemenin en önemli problemlerinden biri olan ve ARP'nin en yaygın türü olan kapasite kısıtlı ARP'de her aracın belirli bir kapasitesi vardır ve müşterilerin talepleri önceden bilinmektedir [2]. En basit kapasiteli araç rotalama probleminde her aracın kapasitesi eşittir, araçlar bir depodan harekete başlarlar ve en son yine aynı depoya geri dönerler. Müşterilerin talepleri tek seferde teslim edilmektedir, parçalama söz konusu değildir. Amaç araçların kat ettiği toplam mesafeyi en küçükleme [2].

Kümeleme, verinin benzer nesnelere oluşturulmuş gruplara bölünmesidir. Kümeleme işleminde küme içindeki elemanların benzerliği fazla, kümeler arası benzerlik ise az olmalıdır [3]. Bir kümeleme yönteminin kalitesi bu prensibi sağlaması ile doğru orantılıdır. Kümeleme yöntemi seçimi kullanılacak veri türüne ve uygulamanın amacına göre farklılık gösterir.

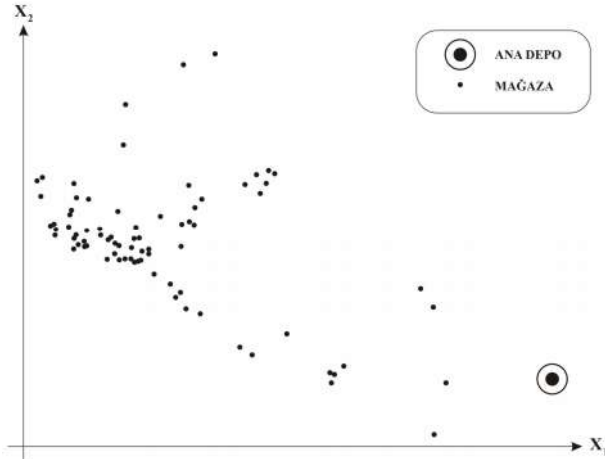
DBSCAN (Density Based Spatial Clustering of Applications with Noise) algoritması, Ester ve ark. tarafından geliştirilmiştir [4]. Algoritma, nesnelere komşuları ile olan mesafelerini hesaplayarak belirli bir bölgede önceden belirlenmiş eşik değerden daha fazla nesne bulunan alanları gruplandırarak kümeleme işlemini gerçekleştirir. DBSCAN algoritması kümeleme problemlerinde birçok yeni terim ve yaklaşım getirmiştir. Bu terimler Epsilon ve MinPoints'dir. Epsilon değeri kümedeki her bir nesnenin Epsilon yarıçapındaki komşuluğu, MinPoints terimi ise küme çevresindeki en az nesne sayısı olarak tanımlanmaktadır. Algoritma kümeleri oluştururken nesnelere yoğunluklarını dikkate alarak kümeleme yapar. Bu yaklaşımda kümeler; yüksek yoğunluklu veri nesnelere ile tanımlanmakta, düşük yoğunluklu nesnelere bulunduğu kümeler aykırı veya gürültülü noktaları göstermektedir. DBSCAN, özellikle büyük veritabanları ve gürültülü nesnelere içeren veri setleri için uygundur. Ayrıca farklı büyüklük ve şekillerdeki kümelerin tanımlanmasında da sıkça kullanılmaktadır [5].

Bu çalışmada yoğunluk tabanlı kümeleme algoritmalarından DBSCAN ve GA temelli DBSCAN'ın karşılaştırması yapılmıştır. Örnek olarak ele alınan bir süpermarketin dağıtım ağı içindeki noktalara haftalık talepleri dikkate alınarak en uygun rotanın bulunması amaçlanmıştır. Ele alınan metotlar ile farklı kümeler oluşmuş ve toplam katedilen mesafeyi en az yapacak alternatifler bulunmaya çalışılmıştır. Metotların performanslarının ölçülmesi için 20 adet farklı veri seti kullanılmıştır.

2. DENEYSEL ÇALIŞMALAR VE SONUÇLAR (EXPERIMENTAL STUDIES AND RESULTS)

2.1. Veri Setinin Tanımı (Data Set Definition)

Bu çalışmada bir ana depo ve 78 mağazadan oluşan bir işletmenin araç rotalama problemi incelenmiştir. Şekil 2'deki noktalar mağazaların, büyük nokta ise ana deponun yerinin iki boyutlu düzlemdeki görüntüsünü temsil etmektedir. Firmanın araç filosu homojen araçlardan oluşmaktadır ve her bir araç 40 palet kapasitelidir. Toplam mağaza sayısı 78 olup, bu mağazaların talep ettikleri palet sayıları tam sayı olarak 3-9 arasında dağılmakta ve toplamda 438 palet olmaktadır. Bu şekilde 20 farklı talep veri seti mevcuttur. Bu veri setlerinin varyansları 4,032 ile 4,993 arasında değişmektedir.



Şekil 1. Veri setinin iki boyutlu şekli (Two dimension figure of data set)

2.2. DBSCAN Algoritması (DBSCAN Algorithm)

DBSCAN algoritması yoğunluğa dayalı kümeleme tekniklerinin temelini oluşturmaktadır [4]. Algoritma kümeleri oluştururken nesnelerin yoğunluklarını dikkate almaktadır. Kümeler, yüksek yoğunluklu veri nesnelere ile tanımlanmaktadır. Düşük yoğunluklu nesnelerin bulunduğu kümeler ise aykırı veya gürültülü noktaları göstermektedir. DBSCAN, özellikle büyük veri tabanları ve gürültülü nesnelere içeren veri setleri için uygundur. Ayrıca farklı büyüklük ve şekillerdeki kümelerin tanımlanmasında da sıkça kullanılmaktadır [5]. ϵ ve MinPts olmak üzere iki girdi parametresine sahiptir. ϵ yarıçap (kümedeki her bir nesnenin ϵ yarıçapındaki komşuluğu) MinPts ise küme çevresindeki en az nesne sayısı olarak tanımlanmaktadır. Bu iki giriş parametresine karşı duyarlı olması algoritmanın dezavantajı olarak görülmekle birlikte ortalama hesaplama karmaşıklığı, $O(n \log n)$ olarak ifade edilmektedir. Burada n veri setindeki nesne sayısıdır [4].

Çalışmada kullanılan kümeleme algoritmalarının sonucunda oluşmuş olan kümelerin her birinin rotaları (tur uzunlukları) ana depoda başlayıp ana depoda son bulacak şekilde en iyileyici rotalama yöntemlerinden olan dal sınır algoritmasından yararlanılarak belirlenmiştir.

Algoritmanın adımları Tablo 1’de verilmiştir.

Tablo 1. DBSCAN algoritmasının aşamaları (Phases of DBSCAN algorithm)

Adımlar	Yapılan İşlem
1	Veri setindeki her bir nesnenin ϵ komşuluğu kontrol edilir.
2	Bu alanda MinPts ile belirlenen sayıdan daha fazla nesne var ise buna çekirdek nokta adı verilir.
3	Her bir kümeyi, çekirdek noktadan doğrudan yoğunluğa katılarak toplanan noktalar yardımıyla çekirdek nesne etrafında büyütülür.
4	Veri nesnelerinin bir kümeye atanma işlemi tamamlandığında yani herhangi bir kümeye artık hiçbir veri nesnesi eklenemediğinde algoritma sonlandırılır.
5	Eğer hala bir kümeye atanmamış veri nesnesi söz konusu ise bu nesnelere gürültü olarak nitelendirilir.

Öncelikle problem için DBSCAN algoritmasının parametrelerinin başlangıç değerleri belirlenmiştir. Dağıtım şirketinin politika olarak belirlediği atanabilir kümelerin 32-40 palet aralığında kapasiteye sahip olma şartını da göz önünde bulundurarak ilk olarak MinPts baz değeri sonrasında Eps baz değeri belirlenmiştir.

MinPoints’in baz değerini belirleme

En az talebi olan mağaza 3 palet, en fazla talebi olan mağaza 9 palet ve kamyon kapasiteleri 40 palet olan problemde atanabilecek kümelerin en az 4 mağazalı (örnek olarak $9+9+9+9=36$), en çok 13 mağazalı (örnek olarak $3+3+3+3+3+3+3+3+3+3+3+3=39$) olabileceği görülmektedir. Bu durumda atanabilecek kümelerinin 4-13 aralığında mağaza sayısına sahip kümeler olduğu ortaya çıkmaktadır.

Veri setine uygulanacak DBSCAN kümeleme algoritması ile oluşacak kümelerin mağaza sayılarının 4 ve 4’den büyük olmasını sağlamak için MinPoints baz değeri 4 olarak belirlenmiştir.

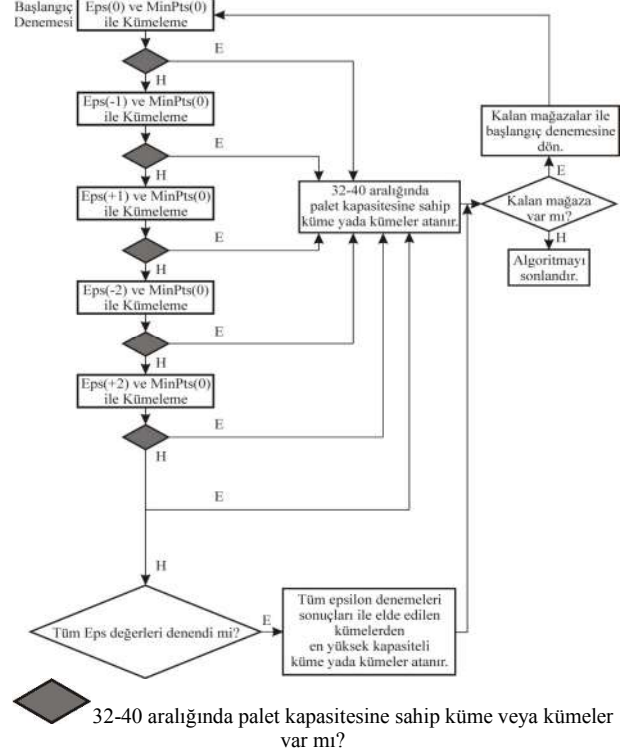
Eps’nin baz değerini belirleme:

MinPoints baz değeri 4 olarak belirlendikten sonra Eps baz değerini belirlemek için MinPts baz değeri sabit tutularak Eps değeri 1’den başlanıp her denemede bir artırılarak denemeler yapılır. Bu denemeler Eps değerinin tek küme oluşturacağı denemeye kadar tekrarlanır. Her denemede oluşan 4-13 aralığında mağazaya sahip küme sayıları belirlendikten sonra en fazla 4-13 aralığında mağazaya sahip küme sayısını veren denemedeki Eps değeri Eps baz değeri olarak belirlenir.

Veri seti ile yapılan denemeler sonucunda Eps'nin baz değeri 18 olarak tespit edilmiştir.

MinPts ve Eps'nin baz değerleri belirlendikten sonra, 78 mağaza ele alınarak DBSCAN algoritması çalıştırılır. Eğer baz değerler ile 32-40 aralığında paletle sahip küme yada kümeler oluşmuş ise direk bu kümeler atanır. Eğer baz değerler ile 32-40 aralığında paletle sahip bir küme oluşmamış ise MinPts baz değerinde kalmak şartı ile Eps baz değerinin bir eksiği alınır ve deneme tekrarlanır. Bu durumda da 32-40 aralığında paletle sahip küme yada kümeler oluşmamış ise MinPts değeri değişmeden Eps baz değerinin bir fazlası alınır ve deneme tekrarlanır. Bu şekilde her aşamada atanabilir (32-40 aralığında paletle sahip) bir küme bulununcaya kadar bu denemeler tekrarlanır. Şekil 3'de DBSCAN algoritmasının aşama işleyiş şeması verilmiştir. Bu şemada; Eps(0) Eps'nin baz değeri, MinPts(0) MinPts'nin baz değeri, Eps(-1) Eps'nin baz değerinin bir eksiği, Eps(+1) Eps'nin baz değerinin bir fazlası, Eps(-2) Eps'nin baz değerinin iki eksiği, Eps(+2) Eps'nin baz değerinin iki fazlası anlamına gelmektedir. Yani Eps baz değeri olan 18'den başlayarak 17, 19, 16, 20, 15, 21, 14, 22, 13, 23, 12, 24, 11, 25, 10, 26, 9, 27, 8, 28, 7, 29, 6, 30, 5, 31, 4, 32, 3, 33, 2, 34, 1, 35, 36, 37, ... değerlerini alır ve atanabilir küme bulunana kadar sırasıyla denemir. İlk karşılaşılan 32-40 palet aralığındaki küme ya da kümeler atanır. Sonraki aşamada kalan mağazalara aynı şekilde kümeleme yapılarak yola devam edilir.

Kapasite kısıtlı araç rotalama probleminin çözümü için yeni bir algoritma geliştirilmesi: bir süpermarket zincirinde uygulanması



Şekil 2. DBSCAN aşama işleyiş şeması (Scheme of DBSCAN stage process)

Belli bir aşamaya geldiğinde ise tüm denemeler yapılmasına rağmen atanabilir bir kümeye ulaşamadığı görülmüştür. Bu durumda tüm yapılan denemeler içerisinde oluşan en yüksek palet kapasitesine sahip küme yada kümeler atanarak yola devam edilir. Bu şekilde tüm mağazaların kümeleri belirlenene kadar bu aşamalar tekrarlanır. Tüm kümeler belirlendikten sonra her kümedeki mağazalara ana depoda eklenerek dal sınır algoritması uygulanır ve rotaları saptanır. DBSCAN algoritmasının ilerleyen aşamalarında palet kapasiteleri düşük kümeler oluşmuş ve bu durum küme sayısında artışa neden olmuştur. Bunun sonucunda kullanılacak araç sayısında ve araçların kat edeceği mesafede artışlar gözlemlenmiştir. DBSCAN algoritmasındaki bu zayıflığı azaltmak için GA destekli DBSCAN algoritması geliştirilmiştir.

2.3. GA Destekli DBSCAN Algoritması (GA-Aided DBSCAN Algorithm)

İlk olarak 1960'lı yıllarda J. Rechenberg tarafından Evrim Stratejileri (Evolution strategies) isimli çalışmada tanımlanan hesaplama [6] daha sonra John Holland'ın çabaları sonucunda genetik sürecin bilgisayar ortamında ele alınışı ilk kez gerçekleştirilmiştir [7].

Genetik algoritmaların karmaşık problemlerin eniyilenmesinde kullanılabilecek etkili teknikler olması

nedeniyle, son yıllarda önemleri hızla artmakta ve çok farklı disiplinlerde başarıyla uygulanmaktadır [8]. GA'lar klasik sezgisel yöntemler ve bazı meta-sezgisel yöntemler gibi (Tabu Arama, Tavlama Benzetimi) tek noktadan arama yapmaz, aksine çok fazla çözümün olduğu bir popülasyondan başlayarak arama yaparlar [9]. Çözüm uzayının daha çeşitli bölgelerinde arama yapıyor olması diğer yöntemlerden üstün olan taraflarından biridir. En iyi bilinen uygulama alanları; çizelgeleme ve sıralama, güvenilirlik tasarımı, araç rotalama ve çizelgeleme, grup teknolojisi, tesis tasarımı ve yerleşimi, ulaştırma gibi alanlardır [8].

Bu çalışmada DBSCAN algoritmasının önceki bölümde belirtilen zayıflığını gidermek amacıyla GA destekli DBSCAN geliştirilmiştir. Bu algoritma DBSCAN algoritmasının tüm Eps denemelerine rağmen atanabilir (32-40 palet kapasiteye sahip) küme oluşturamadığı durumda devreye girmektedir. Bu durumda en yüksek palet kapasitesine sahip küme ya da kümeler atanmaz ve tüm kalan mağazalar ana depo ile birlikte gezgin satıcı problemi gibi düşünülerek genetik algoritma yardımıyla rotalanır. Genetik algoritmanın aşamaları Tablo 2'de belirtilmiştir.

Tablo 2. Genetik algoritmasının aşamaları (Phases of genetic algorithm)

Adımlar	Yapılan İşlem
1	Başlangıç popülasyonunun oluşturulması
2	Popülasyondaki çözümlere göre toplam mesafelerin hesaplanması ve minimumun seçilmesi
3	Çaprazlama işlemi
4	Mutasyon işlemi
5	Yeni oluşan çözümlere göre toplam mesafelerin hesaplanması ve minimumun seçilmesi
6	Algoritmanın durdurulması

Bu aşamada GA yapılırken permütasyon kodlama kullanılarak rassal çözümlerden meydana gelen bir başlangıç popülasyonu oluşturulmuştur. Kromozomların rassal olarak türetilmesi işlemi, popülasyon büyüklüğü kadar kromozom oluşturulunca sonlandırılmaktadır. Sonrasında kromozomların seçim yöntemi olarak rulet çemberi yöntemi kullanılmıştır. Çaprazlama işlemi sıra tabanlı çaprazlama yöntemi kullanılarak %80 çaprazlama oranı ile yapılmıştır. Mutasyon işlemi ikili değişim yöntemi kullanılarak %0,1 mutasyon oranı ile gerçekleştirilmiştir. Algoritmayı durdurma kriterleri iterasyon sayısına göre belirlenmiş ve iterasyon sayısı 10000 olarak belirlenmiştir. Bu şekilde GA algoritması Intel Core 2 Duo CPU T5450 1.67 GHz işlemci ve 3.00 GB RAM özelliğine sahip bilgisayarda oluşturulmuş ve 20 farklı veri seti ile ortalama 17 dakikalık bir sürede

her bir koşum gerçekleşmiştir. GA sonucunda bulunan rotada ana depodan sonra gelen mağazadan başlanarak her birinin palet kapasiteleri kamyon kapasitesine (40 palete) yakınsayacak şekilde kümeler oluşturulmuştur. Bunun sonucunda, düşük kapasiteli kümelerin oluşmadığı buna istinaden kullanılacak araç sayısında ve araçların kat edeceği mesafede azalmalar sağlandığı gözlemlenmiştir.

3. TARTIŞMA VE DEĞERLENDİRME (ARGUMENT AND ASSESSMENT)

Bu çalışmada, farklı olarak yeni bir algoritma önerilmiş olup, yalnızca bilgilendirme içerikli bir karşılaştırma ile yetinilmemiştir. Çalışmada belirlenen algoritmalar, her bir algoritma için yazılım platformu hazırlanarak karşılaştırılmıştır. Karşılaştırma, bilinenlerden yararlanılarak yalnızca yorum getirme şeklinde değil, incelenen her algoritmanın ürettiği sonuçlar gözlenerek yapılmıştır.

Böylece DBSCAN ve GA destekli DBSCAN algoritmalarının uygulanması ile elde edilen Tablo 3'deki sonuçlar kullanılarak ANOVA testi yapılmıştır.

Tablo 3. GA Destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması (Comparison of GA-aided DBSCAN and DBSCAN algorithms)

Deneme	GA Destekli	Küme Sayısı	DBSCAN	Küme Sayısı
	DBSCAN		DBSCAN	
	Toplam		Toplam	
	Mesafe (km)		Mesafe (km)	
1	1196.1	12	1395.0	14
2	1171.6	12	1318.4	14
3	1225.9	13	1318.2	14
4	1200.8	12	1331.7	14
5	1160.7	12	1223.6	13
6	1180.9	12	1357.5	14
7	1235.0	13	1337.0	14
8	1155.3	12	1345.6	14
9	1143.4	12	1220.3	13
10	1198.2	12	1378.4	14
11	1180.6	12	1265.4	13
12	1235.1	13	1342.6	14
13	1168.3	12	1356.9	14
14	1207.7	13	1299.4	14
15	1210.4	12	1407.0	14
16	1152.7	12	1348.3	14
17	1174.3	12	1309.2	14
18	1205.9	13	1297.1	14
19	1165.1	12	1340.4	14
20	1208.7	12	1275.1	13

ANOVA testinde öncelikli olarak varyansların eşit olup olmadığı incelendiğinde sig. değeri 0,05 küçük olduğu için grup varyanslarının eşit olduğu sonucuna varılmıştır (Tablo 4). Bundan dolayı da F testinin sonuçları anlamlı olabilecektir (Tablo 5). ANOVA tablosundaki sig.

değeri 0,05'den küçük olduğu için iki metot ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir farkın olduğu sonucuna varılabilir. Bu durumda toplam katedilen mesafelerin ortalamasına baktığımızda genetik algoritma destekli kümeleme algoritması ile ortalama yol uzunluğu 1188.835 olur iken diğer kümeleme algoritması ile 1323.355 olmuştur. Bu da yeni geliştirilen metodun daha iyi sonuç ürettiğini göstermektedir (Tablo 6).

GA destekli DBSCAN algoritmasının elde ettiği sonuçların ortalaması daha küçük olduğu için GA destekli DBSCAN algoritması tercih edilmelidir. Yapılan gözlemler sonucu GA destekli DBSCAN algoritmasında hem düşük kapasiteli kümeler oluşmamış hem de kullanılacak araç sayısında ve araçların kat edeceği toplam mesafede azalmalar gözlemlenmiştir.

Tablo 4. Varyansların homojenliği testi (Test of homogeneity of variances)

Levene Statistic	Df1	Df2	Sig.
3,947	1	38	,054

Tablo 5. ANOVA testi sonucu (Result of ANOVA test)

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	180956,304	1	180956,304	111,622	,000
Within Groups	61604,055	38	1621,159		
Total	242560,359	39			

Tablo 6. Betimleyici sonuçlar (Results of descriptive)

	N	Mean	Std. Deviation	Std. Error
1	20	1188,835	27,4785	6,1444
2	20	1323,355	49,8724	11,1518
Total	40	1256,095	78,8638	12,4695
Model			40,2636	6,3662
Fixed Effects				
Random Effects				67,2600

Bu çalışmada sezgisel bir algoritma olan GA ya başvurmamızın nedeni denemelerde kalan mağaza sayılarının en iyileyen algoritmalar için yüksek bir sayıda seyir etmesidir. Uygulama aşamasında veri setindeki mağazaların kuşbakışı uzaklıkları yerine gerçek uzaklıkları hesaba katılmış ve bu uzaklıklara göre kümeleme yapılmıştır. Kümeler belirlendikten sonra oluşturulan her bir kümedeki mağazalara ana depoda eklenerek dalsınır algoritması ile her bir kümenin rotaları (tur uzunlukları) belirlenmiştir.

KAYNAKLAR (REFERENCES)

- [1] G. Laporte, Y. Nobert, ve S. Taillefer, «Solving a family of multi-depot vehicle routing and location-routing problems», *Transportation Science*, sayı 22, konu 3, Ağustos, s. 161-172, 1987.
- [2] S. W. Lin, Z. J. Lee, K. C. Ying, ve C. Y. Lee, «Applying hybrid meta-heuristic for capacitated vehicle routing problem», *Expert Systems with Applications*, sayı 36, konu 2, bölüm 1, Ağustos ve Eylül, s.1505-1512, 2009.
- [3] J. Han ve M. Kamber, «Data Mining Concepts and Techniques», San Francisco: Morgan Kaufmann Publishers Inc, 2001.
- [4] M. Ester, H. P. Kriegel, J. Sander ve X. Xu, «A density based algorithm for discovering clusters in large spatial databases with noise», *Int. Conference of Knowledge Discovery and Data Mining (KDD'96)*, Portland, USA, s. 226-231, 1996.
- [5] A. Moreira, M. Y. Santos ve S. Cameiro, «Density-Based Clustering Algorithms-DBSCAN and SNN», *Portugal, University of Minho*, Temmuz, s. 1-18, 2005.[5]
- [6] M. Kurt ve C. Semetay, «Genetik Algoritma ve Uygulama Alanları», *Mühendis Makine Dergisi*, Ekim Sayısı, 2001.
- [7] M. Mitchell, «An Introduction to Genetic Algorithms», *London: The MIT Press*, 1999.
- [8] M. Gen ve R. Cheng, «Genetic Algorithms and Engineering Design», New York: John Wiley and Sons, Inc,1997.
- [9] L. V. Snyder ve M. S. Daskin, «A random-key genetic algorithm for the generalized traveling salesman problem», *European Journal of Operational Research*, sayı 174, konu 1, s. 38-53,2006.