

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**KÜMELEME VE GENETİK ALGORİTMA DESTEKLİ YAKLAŞIMLARLA
KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMİNİN ÇÖZÜMÜ:
PERAKENDE ZİNCİRİNDE UYGULANMASI**

YÜKSEK LİSANS TEZİ

Tolga ŞEN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ
Tez Danışmanı : Doç. Dr. Harun Reşit YAZGAN

Haziran 2014

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KÜMELEME VE GENETİK ALGORİTMA DESTEKLİ YAKLAŞIMLARLA
KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMİNİN ÇÖZÜMÜ:
PERAKENDE ZİNCİRİNDE UYGULANMASI

YÜKSEK LİSANS TEZİ

Tolga ŞEN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 27 / 06 /2014 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



Prof. Dr. İsmail Hatice Cadıoğlu
Jüri Başkanı



Doç. Dr. Harun R. Yenger
Üye



Doç. Dr. Cemil Öz
Üye

ÖNSÖZ

Öncelikle, bu çalışma sırasında yardımlarını ve desteklerini benden esirgemeyen tüm sevdiklerime çok teşekkür ediyorum. Danışmanım ve sevgili hocam H. Reşit Yazgan'a, konu ile ilgili çok değerli tavsiyeleri ve yönlendirmeleri için teşekkürü bir borç bilirim. Bu zorlu süreçte her anlamda yanımda olan çok değerli Sakarya Üniversitesi endüstri mühendisliği öğretim üyelerine, araştırma görevlilerine ve tüm dostlarıma çok teşekkür ediyorum. Son olarak tüm eğitimim süresince yanımda olan aileme teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	vii
TABLolar LİSTESİ	vii
ÖZET	x
SUMMARY	xi
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
GEZGİN SATICI VE ARAÇ ROTALAMA PROBLEMİ.....	4
2.1. Gezgin Satıcı Problemi (GSP)	4
2.2. Araç Rotalama Problemi (ARP)	5
2.3. Araç Rotalama Problemi En İyileme Ölçütleri.....	6
2.4. Araç Rotalama Problemi Çeşitleri	7
2.4.1. Kapasite kısıtlı araç rotalama problemi	8
2.4.2. Mesafe kısıtlı araç rotalama problemi	10
2.4.3. Önce dağıt sonra topla araç rotalama problemi	10
2.4.4. Eş zamanlı topla-dağıt araç rotalama problemi	11
2.4.5. Bölünmüş dağıtımlı araç rotalama problemi	11
2.4.6. Çok depolu araç rotalama problemi.....	11
2.4.7. Periyodik araç rotalama problemi.....	12
2.4.8. Zaman pencereleli araç rotalama problemi.....	12

BÖLÜM 3.

ARAÇ ROTALAMA PROBLEMLERİ ÇÖZÜM YÖNTEMLERİ	14
3.1. Araç Rotalama Problemi Matematiksel Modeli	14
3.2. Araç Rotalama Problemi Çözüm Algoritmaları	16
3.2.1. Eniyilemeye dönük algoritmalar.....	16
3.2.2. Klasik sezgisel algoritmalar.....	16
3.2.3. Meta-sezgisel algoritmalar.....	18
3.2.3.1. Yasaklı arama algoritması	18
3.2.3.2. Genetik Algoritmalar	18
3.2.3.3. Tavlama Benzetimi Algoritması.....	19

BÖLÜM 4.

KÜMELEME ANALİZİ VE ARAÇ ROTALAMADA KÜMELEME TEKNİKLERİ	20
4.1. Kümeleme Analizi Teknikleri.....	22
4.1.1. Hiyerarşik teknikler	23
4.1.2. Hiyerarşik olmayan teknikler.....	24
4.2. Araç Rotalamada Kümeleme Teknikleri.....	25
4.2.1. Model esaslı kümeleme teknikleri	26
4.2.2. Merkeze dayalı bölümleyici kümeleme teknikleri.....	26
4.2.3. Hiyerarşik kümeleme teknikleri.....	27
4.2.3.1. Toplayıcı kümeleme teknikleri.....	28
4.2.3.2. Ayırıcı kümeleme teknikleri.....	28
4.2.4. Yoğunluğa dayalı kümeleme teknikleri	29
4.2.4.1. DBSCAN algoritması	31
4.2.5. Izgara tabanlı kümeleme teknikleri.....	33

BÖLÜM 5.

GENETİK ALGORİTMA.....	34
5.1. Genetik Algoritma'daki Kavramlar	34
5.2. Genetik Algoritma Aşamaları	35

BÖLÜM 6.

UYGULAMA	40
----------------	----

6.1. Uygulamanın Tanıtılması ve Amacı	40
6.2. Uygulamada Kullanılan Algoritmaların İşleyişleri.....	44
6.2.1. DBSCAN algoritması	44
6.2.2. Genetik Algoritma Destekli DBSCAN Algoritması	48
6.3. Alt Problemler ve Çözümleri.....	50
6.3.1. Kuru yoğun gün problemi ve çözümleri.....	50
6.3.2. Taze yoğun gün problemi ve çözümleri.....	53
6.3.3. Kuru sakin gün problemi ve çözümleri.....	57
6.3.4. Taze sakin gün problemi ve çözümleri	60
BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER	65
KAYNAKLAR.....	67
EKLER.....	72
ÖZGEÇMİŞ	84

SİMGELER VE KISALTMALAR LİSTESİ

ANOVA	: Analysis of variance
ARP	: Araç rotalama problem
BIRCH	: Balanced iterative reducing and clustering using hierarchies
CHAMELEON	: Hierarchical clustering using dynamic modeling
CLARA	: Clustering large applications
CLARANS	: Clustering large applications based upon randomized search
CLIQUE	: Clustering high-dimensional space
CURE	: Clustering using representative
DBSCAN	: Density-based spatial clustering of applications with noise
DENCLUE	: Density-based clustering
GA	: Genetik algoritma
GSP	: Gezgin satıcı problem
NP-zor	: Non-deterministic polynomial-time hard
OPTICS	: Ordering points to identify the clustering structure
PAM	: Partitioning around medoids
ROCK	: Robust clustering using links
STING	: Statistical information grid
WAVECLUSTER	: Clustering using wavelet transformation

ŞEKİLLER LİSTESİ

Şekil 3.1. Kazanç algoritması.....	17
Şekil 4.1. Kümeleme tekniklerinin genel görünümü	23
Şekil 4.2. Yoğunlaşmış kitle (contiguity) tabanlı kümeleme	30
Şekil 4.3. Yoğunluk tabanlı kümeleme	30
Şekil 5.1. Allel-gen-kromozom ilişkisi	35
Şekil 6.1. Veri setinin harita üzerindeki görüntüsü.....	43
Şekil 6.2. DBSCAN aşama işleyiş şeması	47

TABLolar LİSTESİ

Tablo 4.1. Kümeleme analizi aşamaları	21
Tablo 5.1. Sıra tabanlı çaprazlama operatörü.....	38
Tablo 5.2. İkili değişim yöntemi ile mutasyon	38
Tablo 6.1. Mağaza noları ve türleri.....	41
Tablo 6.2. Günlerin çeşitleri ve ürün talep eden mağaza sayıları	41
Tablo 6.3. DBSCAN algoritmasının aşamaları.....	44
Tablo 6.4. Genetik algoritmasının aşamaları	48
Tablo 6.5. Kuru yoğun gün için GA destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması.....	52
Tablo 6.6. Kuru yoğun gün için varyansların homojenliği testi	53
Tablo 6.7. Kuru yoğun gün için ANOVA testi sonucu.....	53
Tablo 6.8. Kuru yoğun gün için sonuçlar.....	53
Tablo 6.9. Kuru yoğun gün için welch ve brown-forsythe testi.....	53
Tablo 6.10. Taze yoğun gün için GA destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması.....	56
Tablo 6.11. Taze yoğun gün için varyansların homojenliği testi.....	56
Tablo 6.12. Taze yoğun gün için ANOVA testi sonucu	57
Tablo 6.13. Taze yoğun gün için sonuçlar	57
Tablo 6.14. Kuru sakin gün için GA Destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması.....	59
Tablo 6.15. Kuru sakin gün için varyansların homojenliği testi.....	60
Tablo 6.16. Kuru sakin gün için ANOVA testi sonucu	60
Tablo 6.17. Kuru sakin gün için sonuçlar	60
Tablo 6.18. Taze sakin gün için GA Destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması.....	63
Tablo 6.19. Taze sakin gün için varyansların homojenliği testi	63
Tablo 6.20. Taze sakin gün için ANOVA testi sonucu.....	64

Tablo 6.21. Taze sakin gün için sonuçlar.....	64
Tablo 6.22. Taze sakin gün için welch ve brown-forsythe testi.....	64

ÖZET

Anahtar kelimeler: Araç Rotalama, Kümeleme, Genetik Algoritma

Küreselleşmenin rekabeti hızla arttırdığı son yıllarda müşteri memnuniyeti bu rekabetin en belirleyici faktörlerinden biri olmuştur. Tedarik zincirinde, ürünün ya da hizmetin müşteriye ulaştığı son aşama olan lojistik ve dağıtım şirketlerin üzerinde daha dikkatli durduğu bir alan haline gelmiştir. Ne kadar hızlı ve çok sayıda müşteriye ulaşırsa, o kadar müşteri memnuniyeti artacaktır. Fakat bunun yanısıra şirketler, kendi maliyetlerini de azaltmaya çalışmaktadır. Sözkonusu rekabetin içerisindeki büyük lojistik ve dağıtım şirketlerinin en iyilemeye çalıştıkları problemler de büyük ve karmaşık olacaktır. Aynı anda hem müşterilerine hızlı ve kaliteli hizmet sağlamak hem de maliyetleri en aza indirmek bu şirketlerin hedefidir. Bu çalışmada, bu şirketlerin problemlerinden biri olan araç rotalama probleminin özel bir hali ele alınmıştır. Müşterilerin belirli taleplere sahip olduğu kapasite kısıtlı araç rotalama probleminin çözümünde hızlı ve maliyetleri en aza indirgeyen bir mağaza kümeleme ve araç rotalama ortaya konulmaya çalışılmıştır.

Bu uygulama tezinde ele alınan problem için iki aşamalı bir çözüm yöntemi önerildi. Problemin ilk kısmında kümeleme, ikinci kısmı ise ARP çözümü yapılmıştır. Ele alınan bu problemde bir süpermarket zincirindeki taleplerin karşılanmasındaki ortaya çıkan araç rotalama ve kümeleme probleminin çözüm metotları üzerinde durulmuştur. Bu amaçla iki farklı yaklaşım geliştirilmiştir. Birincisinde bir kümeleme algoritması olan DBSCAN ile müşteriler kümelenecek ve araç rotalama problemleri çözülmüştür. İkinci yaklaşımda ise, GA destekli DBSCAN algoritmasıyla kümeleme geliştirilmiştir ve araç rotalama problemleri çözülmüştür. Her iki yöntemin ARP kısmında belirlenmiş mağazalardan oluşan kümelere ana depo da eklenerek kesin çözüm veren dalsınır algoritması ARP çözümü için uygulanmıştır. Geliştirilen her iki metot bir örnek uygulamada test edilerek sonuçlar ANOVA testi ile karşılaştırılmıştır.

SOLUTION OF THE CAPACITY CONSTRAINT VEHICLE ROUTING PROBLEM WITH CLUSTER AND GENETIC ALGORITHM BASED APPROACH: A RETAIL CHAIN APPLICATION

SUMMARY

Key Words: Vehicle Routing, Clustering, Genetic Algorithm

In the last few years in which the globalization increases the competition, customer satisfaction has been one of the most decisive factors in this competition. In supply-chain, logistics and distribution which is the last process reaching the product or service to customers has become the important field for factories. When it is reached more customers and more quickly, then the satisfaction of the customers will be more. However, factories has an objective to reduce their expenditures. In this competition, the problems that are tried to make them better by big logistic and distribution factories will be major and complicated. Providing fast and high-quality service and decreasing the expenditures are the aims of these factories. In this study, a specific situation of vehicle routing problem was discussed. In the solution of vehicle routing problem which was restricted with certain demands of customers, a store clustering and vehicle routing that decreases the expenditure have been tried to handle.

In this study, a solution approach which has two progressive stages was suggested. Clustering and VRP problems were solved respectively. The proposed approach was implemented on a supermarket to handle vehicle routing and clustering problems under certain demand and capacity constraints. The proposed approach consists of two methods to compare effectiveness of the methods. At the first one, clustering approach was implemented by employing DBSCAN to solve vehicle routing problem. Later, the DBSCAN improved with GA was implemented in order to solve same problem. After completing clustering among branches of the company, travelling salesmen problem was solved with employing branch and bound algorithm for each cluster included main store. Both of these methods have been tested on a sample data and the results have been compared with ANOVA test.

BÖLÜM 1. GİRİŞ

Küreselleşme ile birlikte ortaya çıkan rekabet ortamı, şirketleri sürekli daha iyi olmaya zorlamaktadır. Büyük şirketlerde karşılaşılan problemlerin boyutu da büyük olmakta ve problemin çözülme süreci daha karmaşık bir hal almaktadır. Bu noktada klasik en iyileme yöntemleri yetersiz kalacaktır, çünkü bu yöntemlerle sözkonusu problemlerin çözülmesi çok uzun süre gerektirecektir, halbuki rekabet etmenin gerekliliklerinden biri de hızlı olmaktır. Bitmiş ürünün depolardan müşterilere dağıtımını lojistik yönetiminin önemli bir parçasıdır. Günümüzde hızla önem kazanmakta olan dağıtım sektöründe, lojistik şirketleri, kuryeler ve büyük çaplı şirketlerin kendi dağıtım departmanları, müşterilerine hızlı ve kaliteli hizmet vermek durumundadırlar. Daha iyi rotalama ve kümeleme kararları, kısa zamanda daha çok müşteriye hizmet edebilmeyi sağlayacağından daha fazla müşteri memnuniyeti ve tasarruf sağlayacaktır. Fakat şirketler bir yandan da kendi performanslarını ve ekonomik çıkarlarını gözönünde bulundurmaları gerekmektedir. Literatürde sıkça yer alan araç rotalama ve kümeleme probleminin çözülmesi ile sözkonusu şirketler gerçek hayatta dağıtım ağlarını en iyileme yoluna gideceklerdir.

Ele alınan problemde perakende zincirinin mağazalarına dağıtılan ürünler iki çeşit olup iki farklı kamyon tipi ile dağıtılmakta bunun yanısıra mağazalar da iki farklı çeşide ayrılarak bazılarında belirli günlerde teslimat yapılmamaktadır. Bunun sonucunda problem dört adet alt problemden oluşan bir hal almıştır. Bu problem için açıklama ayrıntılı olarak uygulama bölümünde verilecektir. Bu uygulama problemi bir araç rotalama ve kümeleme problemi olduğundan bu problemin çözümü için önerilen yöntem iki aşama olarak oluşturulmuştur. Bunlardan ilki kümeleme, ikinci aşaması ise ARP'dir. Buna istinaden bu problemin çözümü için iki farklı yaklaşım geliştirilmiştir. Birincisinde bir kümeleme algoritması olan DBSCAN ile müşteriler kümelenecek ve sonrasında ARP'ler dalsınır algoritması ile çözülmüştür. İkinci yaklaşımda ise, GA destekli DBSCAN ile kümeleme yapılmış ve sonrasında ARP'ler dalsınır algoritması

ile çözülmüştür. Geliştirilen her iki metot bu örnek uygulamada test edilerek sonuçlar ANOVA testi ile karşılaştırılmıştır.

Çok sayıda çözüme sahip bir problem olan ve en iyi çözümün zor elde edilebildiği kümeleme problemleri için çok sayıda kümeleme algoritması ortaya atılmıştır. Bu algoritmalar arasından ele alınan problemdeki verilerin yapısına uygun olan bir kümeleme algoritmasından yararlanılması sonuç açısından önem arz etmektedir. Yoğunluğa dayalı kümeleme tekniklerinden olan DBSCAN algoritmasının ele aldığımız problem için uygun bir algoritma olduğu tespit edilmiştir ve probleme uygulanması sağlanmıştır. DBSCAN algoritmasının tespiti aşamasında önceki yıllarda yapılan algoritmaların karşılaştırıldığı çalışmalardan örnekler verilmiştir. Bu algoritmanın tercih nedenleri ilerleyen bölümlerde ayrıntılı olarak belirtilmiştir. Literatürde DBSCAN'ın kullanıldığı ve kümeleme gibi diğer problemlerin çözümü için geliştirilmiş çok sayıda modelin olduğu çalışma yer almaktadır. Sezgisel yöntemlerin bir sınıfına giren ve meta-sezgisel bir yöntem olan Genetik Algoritma (GA), doğadaki evrim sürecini bilgisayar ortamında taklit ederek bu tür problemler için en iyi çözümü ya da yakın çözümü elde edebilmektedir. Uygulamada ele alınan problemin çözüm uzayında çok fazla sayıda çözüm olduğundan bu problem NP-Zor problem sınıfına girmektedir. Bu çalışmada çözüm uzayının birden fazla noktasından arama yapmaya başlayan, hızlı ve etkili bir yöntem olan GA, ele alınan problemde ARP için kullanılmış ve belirlenen rota yardımıyla kümeleme yapılmıştır. GA problemin kümeleme aşamasında DBSCAN algoritmasını desteklemek amacıyla kullanılmıştır. GA'da bir çözümü temsil eden bireyin nasıl kodlandığı modelin etkinliği açısından önemlidir. Ele aldığımız NP-zor problem için çoğunlukla permütasyon kodlama kullanılmaktadır. Bu çalışmada klasik ARP'den daha zor bir problem olan Kapasite Kısıtlı ARP'nin büyük boyutlarının çözümü için, permütasyon kodlamanın söz konusu olduğu GA kullanılarak, hızlı ve verimli bir model oluşturulmaya çalışılmıştır. Söz konusu problem için geliştirilen model ile şirketlerin rekabet gücünü arttırmak çalışmanın amacıdır.

Ele aldığımız problem için önerdiğimiz çözüm yönteminin ikinci aşamasında ARP ele alınmıştır. Çözüm uzayı büyük olan bir problem olan ve en iyi çözümün zor elde edilebildiği türleri olmasına rağmen bazı ARP problemlerinin boyutu en iyilemeye

dönük algoritmaların çözebileceği yapıdadır. ARP sahip olduğu kısıtlara göre değişkenlik gösteren çok fazla çeşide sahiptir. Kapasite Kısıtlı ARP bunlardan biridir ve bu problemde müşteriler belirli taleplere sahip olup dağıtım filomuzun da belli bir kapasiteye sahip dağıtım araçları vardır. Klasik ARP gibi NP-zor olan bu problemin de büyük boyutlarının çözümü kesin yöntemlerle çok uzun sürede gerçekleştirilmektedir. Fakat ele aldığımız problemde ortaya atılan çözüm yönteminin ilk aşamasında oluşturulan kümelerdeki mağaza sayıları en iyilemeye dönük çözüm sunan algoritmalar için uygun bir boyuta sahip olduğundan ARP aşamasında en iyilemeye dönük algoritmalarından olan dalsınır algoritmasından yararlanılmıştır. Bu uygun boyutun tespiti uygulamayı yaptığımız bilgisayarda yapılan denemeler sonucu tespit edilmiştir. Yapılan bu denemeler esnasında 10 noktaya yada daha fazla noktaya sahip kümelerin olması durumunda ileriki bölümlerde özelliği belirtilecek olan bilgisayarın yetersiz kaldığı tespit edilmiştir.

Tezin şu an anlatılan giriş bölümünden sonra ikinci bölümünde GSP ve ARP, ARP en iyileme ölçütleri ve ARP çeşitleri hakkında bilgi verilecektir. Ayrıca ikinci bölümde Kapasite Kısıtlı ARP'nin matematiksel modeli açıklanacaktır. Üçüncü bölümde ARP'nin genel matematiksel modeli ve ARP çözüm algoritmaları açıklanacaktır. Dördüncü bölümde Kümeleme Analizi ve Araç Rotalamada Kümeleme Teknikleri açıklanacaktır. GA'nın detaylı bir şekilde ele alınacağı beşinci bölümde, GA'daki kavramlar ve GA'nın aşamaları anlatılacaktır. Daha sonra altıncı bölümde ele alınan problem tanıtılarak bu problem için oluşturulan kümeleme algoritmaları adım adım çözülecektir. Ardından yine bu bölümde kullanılmış olan kümeleme algoritmalarının sonuçları verilecek ve ANOVA testi ile karşılaştırılmaları yapılacaktır. Sonuçlar ve öneriler kısmında sonuçlar tartışılarak, öneriler ve konuyla ilgili gelecekte yapılabilecek çalışmalardan bahsedilecektir.

BÖLÜM 2. GEZGİN SATICI VE ARAÇ ROTALAMA PROBLEMİ

Ürünlerin müşterilere sevk edilmesi durumunda, hangi noktaların hangi sıralama ile ziyaret edileceğinin araştırılması ve en az maliyet oluşturacak rotanın tespit edilmesi, lojistik problemleri kapsamında ele alınır. Litaratürde bu konuda, gezgin satıcı ve araç rotalama problemleri ön plandadır. İlerleyen bölümlerde sırasıyla bu iki problem türü açıklanacak sonrasında araç rotalama problemi eniyileme ölçütlerine ve araç rotalama problemi çeşitlerine değinilecektir.

2.1. Gezgin Satıcı Problemi (GSP)

Gezgin satıcı problemi, bir ağdaki bir düğümden yola başlayan gezginin tüm düğümlere yalnızca bir kez uğrayarak başlangıç düğümüne geri dönmesi ile biten yolun en kısa halinin tespit edilmesi problemidir. Gezgin satıcı problemi, NP-zor sınıfına ait problem türlerindedir fakat basit olanları yani çözüm uzayındaki çözüm sayısı az olan türleri kolaylıkla çözülebilir.

GSP, ARP ile yakın bir bağlantıya sahiptir. Belirli sayıda noktayı birer defa ziyaret ederek başladığı noktaya dönen gezgin gibi belirli noktalara dağıtım yapması gereken araçlar da güncel dağıtım problemleri içerisinde söz konusu olmaktadır. Günümüzde başlangıçtaki noktaya dönüp dönmeme, araç sayısı vb. faktörlere göre problemin farklı tipleri ortaya çıkmaktadır. Buna bir örnek olarak, rota dahilinde talep edilebilecek hizmet miktarının limitli olduğu kapasite kısıtlarının varlığı GSP' i kapasite kısıtlı araç rotalama problemine dönüştürür. Burada belirtilen kısıtlar, rota üzerindeki şehirlere dağıtım yapacak olan araçların yükleme kapasiteleridir. ARP' nin çözümünde GSP' den farklı olarak, toplam mesafeyi en küçüklemenin yanısıra kullanılan araçların en yüksek doluluk oranıyla dağıtım yapacağı güzergahları oluşturmak da hedeflenir. Araç rotalama problemlerine ilerleyen bölümlerde ayrıntılı olarak yer verilecektir.

GSP çözümü için en iyileyci yöntemlerin yanısıra sezgisel yöntemler, meta-sezgisel yöntemler kullanılabilir.

2.2. Araç Rotalama Problemi (ARP)

Araç rotalama problemi, belli bir coğrafyaya dağılmış olan merkezlere bir veya birden fazla sayıda depodan hizmet vermek amacıyla görevlendirilen araç filosunun en avantajlı biçimde dağıtım/toplama güzergahlarının planlanması problemidir [1].

ARP, gezgin satıcı problemlerinin genel halidir. İlk olarak 1959 yılında Dantzig ve Ramster tarafından ortaya atılmıştır. ARP’de gezgin satıcı problemindeki müşteriler yerine sipariş noktaları, satıcı yerine ise araçlar kullanılmaktadır.

Araç rotalama problemleri, ele alınan problemdeki faktörlere, kısıtlara, hedeflere ve amaçlara göre farklılıklar göstermektedir. Problemin çeşidini belirleyebilmek amacıyla birtakım soruların cevaplanması gerekmektedir. Bu sorular şunlardır: [2]

- Problemin faaliyet konusu dağıtımı mı, toplamayı mı veya her ikisini birden mi içermektedir? Dağıtım ve toplama arasında bir öncelik var mıdır?
- Dağıtım faaliyetleri tek depodan mı yoksa birden fazla depodan mı gerçekleştirilmektedir?
- Araç filosunda kaç araç kullanılmaktadır? Bu sayı sabit bir değer mi yoksa değişkenlik göstermekte midir? Araç filosundaki araçların özellikleri aynı mıdır yoksa değişkenlik göstermekte midir? Kullanılan araçların kapasitesi, hızı ve taşıma maliyetleri nasıldır?
- Sürücülerin çalışma koşulları nelerdir? Normal çalışma gününün süresi kaç saattir? Fazla mesai şartları ve koşulları nelerdir?
- Talepler bilinmekte mi yoksa tahmin mi edilmektedir?
- Müşterilere planlama periyodu içinde hangi sıklıkta veya ne zamanlar uğranabilmektedir? Belirli bir zamanda yada zaman aralığında müşteriye uğranması istenmekte midir?

Bir ARP sisteminin temel elemanlarını; talep yapısı, dağıtım/toplama noktaları ve araç filosu meydana getirir.

Talep Yapısı: ARP' de dağıtım/toplama noktalarının talepleri önceden belirli veya değişken özellikte olabilir. Belirli talep durumunda talep önceden bilinir. Taleplerin değişken olması durumunda ise dağıtım noktalarındaki talepler araç rotasında hareket halindeyken belirlenmektedir [3].

Dağıtım/Toplama Noktaları: ARP'de, dağıtım noktaları genellikle müşterilerin bulunduğu yeri, toplama noktaları ise depoların bulunduğu yeri temsil eder. Depo genellikle aracın rotasına başladığı ve sonlandığı noktadır. Depo sayısına göre problem, tek depolu ve çok depolu olarak çeşitlenebilir. Çok depolu problemlerde, depoların her biri kendi araçlarıyla faaliyetlerini yürütebilir, bunun sonucunda problem birbirinden ayrı olarak çözülen bağımsız tek depolu ARP'leri halini alır.

Araç Filosu: ARP'lerin çoğunda araçların kapasite özelliklerinin bilindiği ve araçların homojen (aynı kapasitede) olduğu görülmektedir. Araç filosu heterojen ise filodaki araçların taşıma kapasiteleri birbirinden farklıdır. Bu durumda hangi araç tipinin, hangi rotaya hizmet vereceğinin belirlenmesinin ayrıca çözülmesi gerekmektedir.

2.3. Araç Rotalama Problemi En İyileme Ölçütleri

ARP'de karşılaşılan eniyileme ölçütlerinden en yaygın olarak kullanılanları aşağıda belirtilmiştir.

Rota Sayısı: Genellikle her rotaya bir aracın atandığı kabul edilir. Bu durumda rota sayısının en küçükleme, araç sayısının da en küçükleme manasına gelir. Araçların Yatırım maliyetleri yüksek olduğundan araç sayısının en küçükleme avantajdır [4].

Toplam Rota Uzunluğu: ARP sonucunda belirlenen rotaların uzunlukları toplamıdır. Her bir rotanın uzunluğu, rota üzerindeki dağıtım/toplama noktaları arasındaki mesafelerin toplamıdır.

Rota Süresi: Hareket halinde geçen sürenin yanısıra yükleme-boşaltma ve dinlenme halindeki sürelerin toplamıdır.

Müşteri Memnuniyeti: Perakende satılan tüketim mallarının nakliyesinde müşteri memnuniyeti, malın müşteriye gerektiği sürede ulaştırılmasıyla sağlanır.

Yukarıda açıklanan ARP en iyileme ölçütlerinden rota sayısı ve toplam rota uzunluğu en çok kullanılan en iyileme ölçütlerindedir.

2.4. Araç Rotalama Problemi Çeşitleri

Lojistikte önemli bir yere sahip dağıtım problemlerinden biri olan ARP, gerçek hayat koşullarından dolayı ortaya çıkan kısıtlar nedeniyle çeşitlenebilmektedir. Zaman kısıtlarına örnek olarak; araçların hizmet verebilecekleri zaman kısıtı, araç şoföründen kaynaklanabilecek zaman kısıtı, deponun açılış kapanış saatlerinin olması ile oluşan zaman kısıtı, müşterilerin hizmet kabul edebileceği bir zaman aralığından kaynaklanan zaman kısıtı verilebilir. Mesafe kısıtlarına örnek olarak; araçların toplamda kat edebileceği mesafe kısıtı verilebilir. Kullanım kısıtlarına örnek olarak; kullanılacak toplam araç sayısı kısıtı, şoför sayısı kısıtı verilebilir. Klasik ARP’de, maliyetler, müşteri talepleri, araç seyahat süreleri gibi problem ile alakalı tüm parametreler belirlidir.

ARP çeşitlerini öncelikle iki sınıfa ayırmak mümkündür. Bunlardan ilki problemin çözümünden önce tüm bilgilerin (kapasiteler, talep miktarları, mesafeler, talep noktası konumları, zaman pencereleri) bilindiği ve çözüm süresince bu bilgilerin sabit kaldığı statik ARP’dir. İkincisi ise bu bilgilerin değişkenlik gösterebildiği dinamik ARP’dir. Gerçek hayattaki dinamik koşulların etkisi ile verimlilik ve hizmet seviyesinde ortaya çıkan ihtiyaçlar artmıştır. Bu nedenle taşımacılık sistemlerinin geliştirilmesi gerekliliği kaçınılmaz bir hal almış ve statik ARP modelleri gerçek uygulamaların basite indirgenmiş hali haline gelmiştir [5]. Stokastik ARP problemlerinde ise yukarıda değinilen tüm parametreler rassal olmaktadır. Gündelik hayatta karşılaşılabilecek aksamaların çözüme yansıtılmasıyla çözüme ulaşmanın hedeflendiği stokastik ARP modellerinin çözümü daha zordur [6]. Bu nedenle sıkça çalışılan bir konu olan

ARP'nin çok sayıda çeşidi vardır. Aşağıda anlatılacak olan ARP çeşitleri problemin sahip olduğu kısıtlar nedeniyle ortaya çıkmaktadır.

2.4.1. Kapasite kısıtlı araç rotalama problemi (Kapasite kısıtlı ARP)

Kapasite kısıtlı ARP bu tez çalışmasında ele alınan ARP çeşididir. Dağıtım ağını en iyilemenin en önemli problemlerden biri olan ve ARP'nin en çok karşılaşılan türü olan kapasite kısıtlı ARP'de her aracın belli bir kapasitesi vardır ve müşterilerin talepleri önceden bilinmektedir [7]. En basit kapasiteli araç rotalama probleminde araç filosunda yer alan her aracın kapasitesi aynıdır. Araçlar bir depodan harekete başlayıp en son yine depoya geri dönerek rotalarını tamamlarlar.

Literatürde çok karşılaşılan bu problem türünde, her müşterinin talep miktarı bellidir ve her müşteriye yalnızca bir araç tarafından hizmet verilir ve sadece bir kez uğranarak bu hizmet gerçekleştirilir. Depo ve müşterilerin birbirlerine olan uzaklıkları simetriktir ve araçların seyahat süreleri bu mesafelerle doğru orantılı değişmektedir. Bu problemde de genellikle amaç, araçların kat ettiği toplam mesafeyi en aza çekmektir [7].

NP-zor sınıfında yer alan ARP için makul bir sürede en iyi çözümü bulabilen etkili bir algoritma yoktur. Matematiksel yöntemlerin kullanılması ile en iyi çözüme ulaşılır, fakat bu kesin yöntemler çözüm uzayının tamamında arama yapacağından ARP için problemin büyüklüğü arttıkça çözüm için gereken süre de üssel olarak artış gösterecektir [5].

ARP için oluşturulmuş çok sayıda matematiksel model mevcuttur, literatürde yer alan Christofides ve diğ. 1981 yılında yaptığı çalışmasındaki tamsayılı modelde M araç sayısını ve N müşteri sayısını göstermektedir [8]. x_{ijk} değişkeni, i müşterisinden j müşterisine k aracı ile gidilip gidilmediğini belirtir.

$$x_{ijk} = \begin{cases} 1, & \text{eğer } k \text{ nolu araç } i \text{ müşterisinden } j \text{ müşterisine gidiyorsa} \\ 0, & \text{aksi takdirde} \end{cases} \quad (2.1)$$

c_{ij} = i müşterisi ile j müşterisi arasındaki mesafe (maliyet)

Q = Bir araca yüklenebilecek toplam ürün miktarı

q_i = i müşterisinin talep miktarı

y_i = Alt turları engellemek için kullanılan rastgele değişken

Model şu şekildedir;

Amaç fonksiyonu:

$$\text{En az } z = \sum_{i=0}^N \sum_{j=0}^N (c_{ij} \sum_{k=1}^M x_{ijk}) \quad (2.1)$$

Öyle ki:

$$\sum_{i=0}^N \sum_{k=1}^M x_{ijk} = 1, \quad j = 1, \dots, N \quad (2.2)$$

$$\sum_{i=0}^N x_{ipk} - \sum_{j=0}^N x_{pjk} = 0, \quad k = 1, \dots, M, \quad p = 0, \dots, N \quad (2.3)$$

$$\sum_{i=0}^N (q_i \sum_{j=0}^N x_{ijk}) \leq Q, \quad k = 1, \dots, M \quad (2.4)$$

$$\sum_{i=0}^N x_{ojk} = 1, \quad k = 1, \dots, M \quad (2.5)$$

$$y_i - y_j + N \sum_{k=1}^M x_{ijk} \leq N - 1, \quad i \neq j = 1, \dots, N \quad (2.6)$$

$$x_{ijk} \in \{0,1\}, \quad \forall i, j, k \quad (2.7)$$

$$y_i \text{ işareti serbest} \quad (2.8)$$

Amaç fonksiyonunda amaç toplam yol mesafesini en küçükmektir. (2.2) numaralı kısıta göre her müşteri yalnızca bir kez ziyaret edilmelidir. (2.3) numaralı kısıt ise eğer bir araç bir müşteriyi ziyaret ediyorsa ziyaret sonrası yine o müşteriden başlayarak hareketine devam etmesi gereğini sağlayan kısıttır. Her aracın taşıyabileceği toplam ürün miktarının belli bir sınırı vardır ve bununla ilgili kapasite kısıtı (2.4) numaralı kısıt ile ifade edilmiştir. (2.5) numaralı kısıtta ise her araç yalnızca bir kez kullanılmalı

şartı sağlanır. Problemin çözümünde alt turların engellenmesi için (2.6) numaralı kısıt modele eklenmiştir. Bu kısıt her rotanın depodan geçmesini sağlamaktadır. Christofides ve diğ. çalışmasında bu kısıtların yanısıra her aracın toplam maliyet kısıtı da belirtilmektedir, bu kısıt aşağıda anlatılacak olan mesafe kısıtının da olduğu ARP çeşidinde yer bulunmaktadır [8].

2.4.2. Mesafe kısıtlı araç rotalama problemi (Mesafe kısıtlı ARP)

Mesafe Kısıtlı ARP'yi Kapasite Kısıtlı ARP'den farklılaştıran kısıt, rotaları belirlenmiş her aracın katedebileceği belirli bir toplam mesafenin olması kısıtıdır. Bu durum gerçek bir dağıtım probleminde taşınan ürünün cinsinden, araç veya sürücü kısıtlarından dolayı kaynaklanabilir. Eğer taşınan ürünün uzun süre taşınması ile bozulma riski ortaya çıkıyorsa, ya da araç sürücüsünün sürekli olarak belirli bir süreden daha fazla yolculuk yapamaması durumu varsa bu kısıt eklenmelidir. Bu nedenle yukarıda ifade edilen modele bir kısıt daha eklenecektir:

T: Her bir aracın katedebileceği toplam mesafe

$$\sum_{i=0}^N \sum_{j=0}^N (c_{ij} x_{ijk}) \leq T \quad (2.9)$$

2.4.3. Önce dağıt sonra topla araç rotalama problemi (Önce dağıt sonra topla ARP)

ARP'nin bu çeşidinde müşteriler, ürün teslim edilecek ve ürün teslim alınacak müşteriler olmak üzere iki çeşitten oluşmaktadır. Bir rotaya her iki müşteri çeşidinden de atanabilmektedir, fakat öncelikle ürün dağıtımı yapılacak olan müşterilere uğranacak, daha sonra toplama işlemi yapılacak müşterilere uğranacaktır [5]. Dağıtımın ve toplamanın rastgele yapılması durumunda aracın yükleme yapılan tarafının yeniden düzenlenmesi zaman ve olurluluk yönünden avantajlı değildir.

Teslim edilecek ve toplanılacak ürün miktarları önceden bilinmekte ve araç filosundaki tüm araçlar aynı miktarda taşıma kapasitesine sahiptir. Burada önemli olan diğer bir husus da araçların hem dağıtım yapacağı ürün miktarının, hem de teslim alacağı ürün miktarının araç kapasitesini geçemeyeceğidir [9].

2.4.4.Eş zamanlı topla-dağıt araç rotalama problemi (Eş zamanlı topla-dağıt ARP)

Eş Zamanlı Topla-Dağıt ARP’de ise, eş zamanlı olarak bir müşteriye hem ürün teslim edilebilir ya da müşteriden ürün teslim alınabilir. Bu problem türünde müşterileri iki ayrı gruba ayırma işlemi yapılmaz. Yani öncelikle dağıtım yapılacak sonrasında ise toplama yapılacak gibi bir kısıt yoktur. Araçlar rotaları boyunca dağıtım ve toplama bir arada yaptıklarından dolayı araç kapasitesinin daima korunması gibi zor bir durumla karşılaşmaktadır. Her müşteriye yapılacak olan ürün miktarı, yada toplama yapılacak ürün miktarı önceden bilinmektedir. Bianchessi ve Righin’in çalışmalarındaki tanıma göre, Eş Zamanlı Topla-Dağıt ARP, yapılacak işlemler arasında bir öncelik ilişkisi olmadan aynı anda ürün dağıtımının ve atık toplamanın uygun biçimde birleştirilmesidir [10]. ARP’nin bu türü, hem ürün dağıtımının yapıldığı hemde kullanılmış ürünlerin geri dönüşüm tesislerinde tekrar işlenmesi için geri toplanması ile ilgili olan ters lojistik uygulamaları için kullanışlıdır.

2.4.5.Bölünmüş dağıtımlı araç rotalama problemi (Bölünmüş dağıtımlı ARP)

Bu problem çeşidinde bir müşterinin talep ettiği ürünler birden fazla araç tarafından temin edilebilmektedir. Bu dağıtım şeklini, ortalama müşteri talebinin çok büyük olduğu durumlarda kullanmak yararlıdır. Bölünmüş Dağıtımlı ARP araçların gittiği toplam mesafeyi en küçüklemeyi amaçlarken, klasik ARP problemine eklenen şu kısıt ile farklı bir hal almaktadır: Bir müşterinin talebi bir veya daha fazla araçla teslim edilir. Bir diğer farklılığı da yukarıda belirtildiği üzere ortalama müşteri talebinin klasik ARP’dekinden çok fazla olmasıdır [11].

2.4.6.Çok depolu araç rotalama problemi (Çok depolu ARP)

Daha önce değinilen ARP problemlerinde tek deponun kullanıldığı kabul edilmiştir. Bu problem türünde ise araçların harekete başlayabileceği birden fazla depo bulunmaktadır. Depoların ve müşterilerin yerleri önceden bilinmektedir ve her depo tüm müşterilerin toplam taleplerini karşılayabilecek özelliklere sahiptir. Bu problem çeşidinde her araç harekete başladığı depoya geri dönmek durumundadır. Birden fazla

deposu bulunan dağıtım şirketlerinin araç rotalaması yapılmakta ise çok depolu olma durumunu yapılan modele ilave etmek gerekecektir. Bu ARP türü de NP-zor problem türlerindedir ve en iyi çözümün elde edilebileceği kullanışlı bir yöntem bulunmamaktadır [12].

2.4.7. Periyodik araç rotalama problemi (Periyodik ARP)

Periyodik ARP’de belirli bir sürenin planı başlangıçta yapılmaktadır ve müşteriler bu süreçte birden fazla hizmet almaktadır. Müşterilere gerçekleştirilecek servis sayısı müşterilerin talep ettikleri ürün miktarlarına, depo alanlarına göre değişmektedir. Eğer bir müşterinin talep miktarı fazla ise az miktarda talebi olan müşteriye göre daha fazla ziyaret edilecektir. Bu problem çeşidi bakkaliye, alkolsüz içki endüstrisi, atık toplama gibi alanlarda kullanılmaktadır [13].

2.4.8. Zaman pencereci araç rotalama problemi (Zaman pencereci ARP)

Bu problem çeşidinde Kapasite Kısıtlı ARP’de olduğu gibi belirli sayıda ve aynı kapasiteye ve özelliklere sahip homojen araçlardan oluşmuş araç filosu vardır, talep ettikleri ürün miktarları ve yerleri bilinen müşteriler ve yine konumu bilinen bir adet depo bulunmaktadır. Bu problem tipini diğerlerinden ayıran ve problemi daha da zorlaştıran kısıt, her müşteri için servis yapılabilecek belirli bir zaman aralığının olmasıdır. Bu zaman aralığı içerisinde sözkonusu müşteriye hizmete başlamak zorunludur. Her müşteri için ürün dağıtımı yada toplaması için belirli bir servis süresi vardır ve araç müşteriye ulaştıktan sonra bu hizmet süresi kadar müşterinin olduğu yerde kalacaktır, daha sonra rotasındaki bir sonraki müşteriye yada depoya doğru yoluna devam edecektir. Gerçek hayatta müşteriler istedikleri ürünlerin kendilerine ulaşmasını belirli zaman aralıklarında isteyebilirler, bu durumlarda bu problem çeşidi için geliştirilen yöntemleri kullanmak kullanışlıdır.

Zaman Pencereci ARP iki alt çeşide ayrılmaktadır:

- Sıkı Zaman Pencereci ARP (Hard Time Windows VRP)

Kullanılan araçlar müşterilerin hizmet alabileceği zaman aralığının dışında hizmet verememektedir.

- Esnek Zaman Pencereci ARP (Soft Time Windows VRP)

Kullanılan araçlar müşterilere ilgili zaman aralığının dışında hizmet verilebilir, fakat bu durumda bir ceza maliyeti meydana gelmektedir. Bu tip problemlerde deponun zaman penceresini sağlaması gerekmektedir [14].

BÖLÜM 3. ARAÇ ROTALAMA PROBLEMLERİ ÇÖZÜM YÖNTEMLERİ

ARP' nin çözümünde sezgisel algoritmaların çoğunlukla kullanıldığı bir önceki bölümde açıklanmıştı. Bugüne kadar yapılan çalışmalar incelendiğinde, en iyi çözüm yöntemleri yalnızca çok iyi tanımlanmış bazı ARP' ler için kullanılabilir.

Bu yöntemler için oluşturulan matematiksel modellerde problemin çözümünü zorlaştıran atama ve GSP alt-tur engelleme kısıtları kullanılmaktadır. Çoğu durumda amaç fonksiyonu, doğrusal olmayan ve/veya çok amaçlı bir yapı halini alır. Bu durum problemin çözümünü zorlaştırır. Gerçek hayat problemleri bu şekilde büyük ölçekli bir yapıdadır.

Sonuçta bu çeşit problemlerin en iyi çözümlerini bulmak oldukça çok zaman almakta ve mevcut yazılımlarla belirli büyüklüklerden sonra bu problemlerin çözümünü bulmak imkansız olabilmektedir.

3.1. Araç Rotalama Problemi Matematiksel Modeli

ARP' nin genel matematiksel modeli aşağıda verilmektedir [15]:

Parametreler

C_{ij} = i ile j düğümü arasındaki mesafe

m = araç sayısı

n = düğüm sayısı

S = düğümler alt kümesi

$|S|$ = S alt kümesindeki düğüm sayısı

Karar Değişkenleri

$$x_{ijk} = \begin{cases} 1, & k \text{ aracı } i \text{ düğümünden hemen sonra } j \text{ düğümüne giderse} \\ 0, & \text{diğer durumda} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & k \text{ aracı } i \text{ düğümünü ziyaret ederse} \\ 0, & \text{diğer durumda} \end{cases}$$

Amaç Fonksiyonu

Toplam katedilen mesafenin en küçüklenmesi

$$\text{Enaz } z = \sum_{i,j} c_{ij} \sum_k x_{ijk} \quad (3.1)$$

Kısıtlar

Her müşteri (düğüm) bir araca atanmalıdır;

$$\sum_k y_{ik} = \begin{cases} 1, & i = 2, \dots, n \\ m, & i = 1 \end{cases} \quad (3.2)$$

Bir araçla bir şehre en fazla bir kez uğranabilmelidir (i düğümüne k aracı ile gelindiyse ve i düğümünden de j düğümüne geçilecekse, bu geçiş yine k aracı ile olmalıdır);

$$\sum_k x_{ijk} = y_{ik} \quad \forall (i, k) \quad i = 1, \dots, n \quad k = 1, \dots, m \quad (3.3)$$

Alt tur oluşması engellenmelidir;

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \quad \forall (k) \quad S \subseteq \{2, \dots, n\} \quad k = 1, \dots, m \quad (3.4)$$

Karar değişkenleri 0-1 tamsayıli değişken olmalıdır;

$$y_{ik} \in \{0,1\} \quad i = 1, \dots, n \quad k = 1, \dots, m \quad (3.5)$$

$$x_{ijk} \in \{0,1\} \quad i, j = 1, \dots, n \quad k = 1, \dots, m \quad (3.6)$$

Bu tanımlamalar altında ARP' nin genel matematiksel modeli aşağıdaki gibidir;

$$\sum_k y_{ik} = \begin{cases} 1, & i = 2, \dots, n \\ m, & i = 1 \end{cases} \quad (3.7)$$

$$\sum_k x_{ijk} = y_{ik} \quad \forall(i, k) \quad i = 1, \dots, n \quad k = 1, \dots, m \quad (3.8)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \quad \forall(k) \quad S \subseteq \{2, \dots, n\} \quad k = 1, \dots, m \quad (3.9)$$

$$y_{ik} \in \{0,1\} \quad i = 1, \dots, n \quad k = 1, \dots, m \quad (3.10)$$

$$y_{ik} \in \{0,1\} \quad i, j = 1, \dots, n \quad k = 1, \dots, m \quad \text{kısıtları altında,} \quad (3.11)$$

$$\text{Enaz } z = \sum_{i,j} c_{ij} \sum_k x_{ijk} \quad (3.12)$$

3.2. Araç Rotalama Problemi Çözüm Algoritmaları

ARP' nin çözümü için kullanılan algoritmalar, en iyiyi amaçlayan ve sezgisel olarak iki çeşide ayrılabilir. Sezgisel algoritmalar da kendi içinde klasik sezgisel ve meta-sezgisel olarak iki gruba ayrılabilir.

3.2.1. Eniyilemeye dönük algoritmalar

Bölüm 3.1' de yer alan matematiksel modele, talep edilen durumları sağlayacak kısıtlar eklenerek çeşitli en iyi çözüm algoritmaları oluşturulmuştur [1]. Doğrudan ağaç arama, dinamik programlama ve tamsayılı programlama en iyilemeye dönük çözüm algoritmalarına örnek olarak verilebilir.

3.2.2. Klasik sezgisel algoritmalar

ARP için kullanılan klasik sezgisel çözüm algoritmaları üç çeşit altında incelenmiştir [16]:

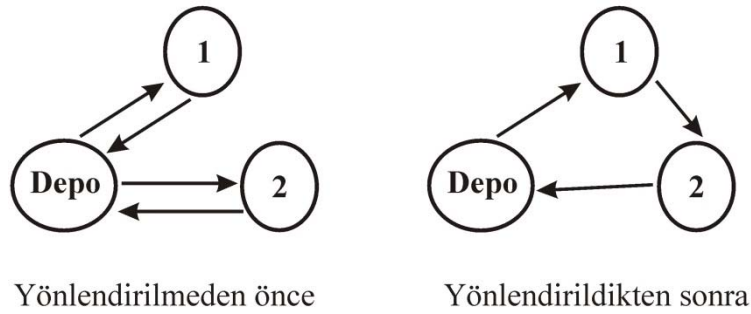
1. Tur kurucu yöntemler

2. İki aşamalı yöntemler
3. İyileştirici yöntemler

Tur kurucu sezgiseller, uygun olmayan atamalarla çözüme başlayıp sonrasında her seferde iki düğüm arasına bir dal ekleyerek uygun çözüme ulaşırlar. Dal eklerken araç kapasite kısıtının sağlanıp sağlanmadığı takip edilir.

Tur kurucu yöntemler içinde en çok kullanılanı Clarke ve Wright tarafından, Dantzing ve Ramser' in çalışmalarını temel alarak geliştirilen kazanç algoritması'dır [17].

Algoritma, bir noktanın başlangıç noktası (depo) olarak belirlenmesi ile başlar. Öncelikle bütün durakların depodan ziyaret edildiği farzedilir. Sonrasında iki alt turun birleştirilmesiyle elde edilen kazanç hesaplanır. En yüksek çıkan kazanç değerinden başlayarak rotalar birleştirilir. Algoritmanın temel işleyişi Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Kazanç algoritması

İki aşamalı yöntemlerin ilk aşamasında, düğümler araçlara kapasiteyi aşmayacak biçimde atanır. Ardından ikinci aşamada her bir araç için GSP sezgiselleri kullanılarak rotalar tespit edilir [18]. Önce gruplayıp sonra rotalayan türdeki algoritmalar, iki aşamalı tekniklere örnek olarak verilebilir.

İyileştirici yöntemleri ise araç rotalarının birer birer veya birden çok sayıda rotanın aynı anda iyileştirilmelerini sağlayan iki farklı çeşide sahiptir. Tek rota iyileştirmelerinde, gezgin satıcı algoritması için kullanılan tekniklerden yararlanılmaktadır.

3.2.3. Meta-sezgisel algoritmalar

Bütünleşik eniyileme problemlerinin mantığında, problemin bütün mümkün çözümlerinin çözülüp en iyi olan çözümün çözüm olarak tespit edilmesi vardır. Teorik olarak bir problemi böyle bir yöntemin yaklaşımıyla çözmek mümkündür. Ancak pratikte çözüm uzayının çok fazla boyuta sahip olması durumunda bu tür yöntemlerle çözüme ulaşmak genellikle mümkün olmamaktadır.

Bu nedenlerden dolayı son dönemde yeni çözüm yöntemlerinin ortaya konulduğu görülmektedir. Yasaklı arama, genetik algoritma ve tavlama benzetimi algoritmalarını bu tip yöntemlere örnek olarak verebiliriz. En iyi yerine çoğunlukla en iyiye yakın olan çözümü daha kısa sürede tespit edebilmeleri bu tür yöntemlerin ortak özelliğidir.

3.2.3.1. Yasaklı arama algoritması

Yasaklı arama algoritması, bütünleşik eniyileme problemlerini çözümüne ulaşmak için için oluşturulmuş sezgisel bir yöntemdir. Diğer yöntemlerle birlikte kullanıldığında yerel en iyiye takılmayı önleyebilen bir tekniktir.

Yasaklı arama algoritmasının başarılı bir biçimde kullanıldığı alanlara; ulaştırma, çizelgeleme, tesis yerleşimi, telekomünikasyon, araç rotalama gibi pek çok alan örnek olarak verilebilir [19].

3.2.3.2. Genetik Algoritmalar

Genetik Algoritmalar sezgisel biçimde uygulanabilecek olasılıklı arama tekniklerindedir. İlk ortaya çıkışı 1960 'lı yılların sonlarındaki Holland' ın uygulamalarına dayanır. Öğrenebilen makinelerin tasarımını araştıran Holland öğrenme işleminin; tek bir organizmadan ziyade türlerin nesilleri boyunca ortaya çıkan evrimsel uyumu ile ortaya çıktığını fark eder [20]. Genetik algoritmalar da buna benzer olarak doğadaki gelişim sürecini temel alır. Bu tip algoritmalar, hayatta kalabilen ve özelliklerini yeni nesillere yansıtabilen organizmaların özelliklerini taklit ederek

özümü bulmaya alışırlar. Bir problem için bu sürecin taklit edilmesinin sebebi, iyi özümün iftleştirilerek bireylerin güçlü özelliklerinin kullanımının sağladığı avantajdır. Bu şekilde problemler için daha iyi sonuçlara ulaşılmaktadır. İleriki bölümlerde bu algoritma daha ayrıntılı şekilde anlatılacaktır.

3.2.3.3. Tavlama Benzetimi Algoritması

Tavlama benzetimi algoritması eniyileme problemleri için iyi özümün oluşturan olasılıklı bir arama yöntemidir. Metallerin fiziksel tavlama süreci ile olan benzerliğinden dolayı ‘Tavlama Benzetimi’ ismini almıştır. Algoritma, Kirkpatrick vd. tarafından ortaya konmuş ve geliştirilmiştir [21]. Bilgisayar tasarımı, görüntü işleme, moleküler fizik ve kimya, çizelgeleme ve araç rotalama gibi pek çok problemde bu algoritma kullanılabilir.

BÖLÜM 4. KÜMELEME ANALİZİ VE ARAÇ ROTALAMADA KÜMELEME TEKNİKLERİ

Araç Rotalama Problemlerinin çözüm aşamalarında çok sayıda veriye çözümün tutarlılığı açısından gerek duyulmaktadır. Ancak verilerin fazla olması durumunda birçok ARP yönteminin uygulanmasında problemler oluşmaktadır. Bu durumda yöntemlerin uygulanabilirliğini sağlamak amacıyla verilerin sınıflandırılması ihtiyacı oluşmaktadır. Kümeleme analizi, verileri sınıflandırmada faydalı olan çok değişkenli bir analiz yöntemidir. Kümeleme analizinde amaç, sınıflanmamış verileri benzer özelliklerine göre sınıflandırmak ve araştırmacıya özetleyici bilgiler sağlamada yardımcı olmaktır.

Kümeleme analizi farklı sorunların giderilmesinde kullanılan önemli bir analizdir. Özellikle, doğru grupların bulunmasında iyi sonuçlar meydana getirir. Örneğin, uzaklık matrisine göre bir perakende zincirinin mağazalarını kümeleme probleminde kullanılabilir. Bunun yanısıra kümeleme analizi veri indirgemesi içinde kullanılır. Örneğin, mağazaların çoğu yeni bir ürün için test pazarları olarak kullanılması gerekebilir. Eğer mağazalar benzer mağazaları içeren küçük sayıdaki gruplara ayrılırsa her bir gruptan bir mağaza seçilerek test pazarını daha az sayıda mağazaya indirebiliriz. Böylece uygulamada gerek zaman gerekse maliyet ve emek açısından önemli tasarruflar elde edilmiş olur.

‘Küme, birbirine benzer nesnelerin çok boyutlu uzayda oluşturdukları bulutlar benzetmesi’ biçiminde açıklanabilir [22]. Bu kümeleri oluşturma işlemi ise kümeleme analizidir.

Kümeleme analizi ile alakalı yapılabilecek diğer bir tanımlama ise; gruplandırma problemlerini çözmek amacıyla kullanılan tanımlayıcı veri analizi, homojen grupların

düzenlenmesi için değişkenler hakkındaki bilgileri sıralama amacıyla yapılan çok değişkenli istatistiksel analiz tekniği şeklinde yapılabilir [23] .

Hemen hemen tüm analitik problemlerde karşılaşılan birçok özellik nedeniyle sınıflandırmayı gerektirecek durumlar ile karşılaşılır [24]. Kümeleme analizi bu gibi durumlarda sınıflandırmaya destek olan bir analiz yöntemidir.

Kümeleme analizi bağımlı ve bağımsız değişkenleri eşdeğer ağırlıkta hesaba katar. Birbirine bağımlı olan tüm durumları araştırır. Kümeleme analizi, nispeten homojen olarak oluşturulmuş gruplardaki nesnelere belirleyip sınıflandırmaktadır. Aynı grup içinde yer almış olan nesnelere; değişkenleri açısından genellikle birbirlerine benzerlik gösterirken diğer gruptaki nesnelere ile özellikleri bakımından farklılaşmaktadırlar [25].

Kümeleme analizi; birkaç aşamadan meydana gelen bir çözüm sürecidir. Analizin ilk aşamasını veri girişi oluşturur. Yani ilk olarak doğal sınıflamaları hakkında kesin bilgilerin bulunmadığı anakütlelerden alınan n sayıda birimin incelenen p sayıda değişkenle alakalı inceleme sonuçları değerleri oluşturulur. Böylece veri matrisi oluşturulmuş olur. Daha sonra verinin ölçüm tipine uygun bir benzerlik ölçüsü ile nesnelere uzaklıklar matrisi meydana getirilir. Uygun kümeleme tekniği seçilir ve kullanılır. Tekniğin kullanılması sonucu nesnelere kümelerine ayrılmış olur. Kümeleme sonuçlarının tutarlılığının yorumlandığı aşama ise analizin son aşamasıdır [26].

Aşağıda şekil 4.1.' de kümeleme analizinin aşamaları işlem sırasına göre ayrıntılı olarak görülmektedir.

Tablo 4.1. Kümeleme analizi aşamaları

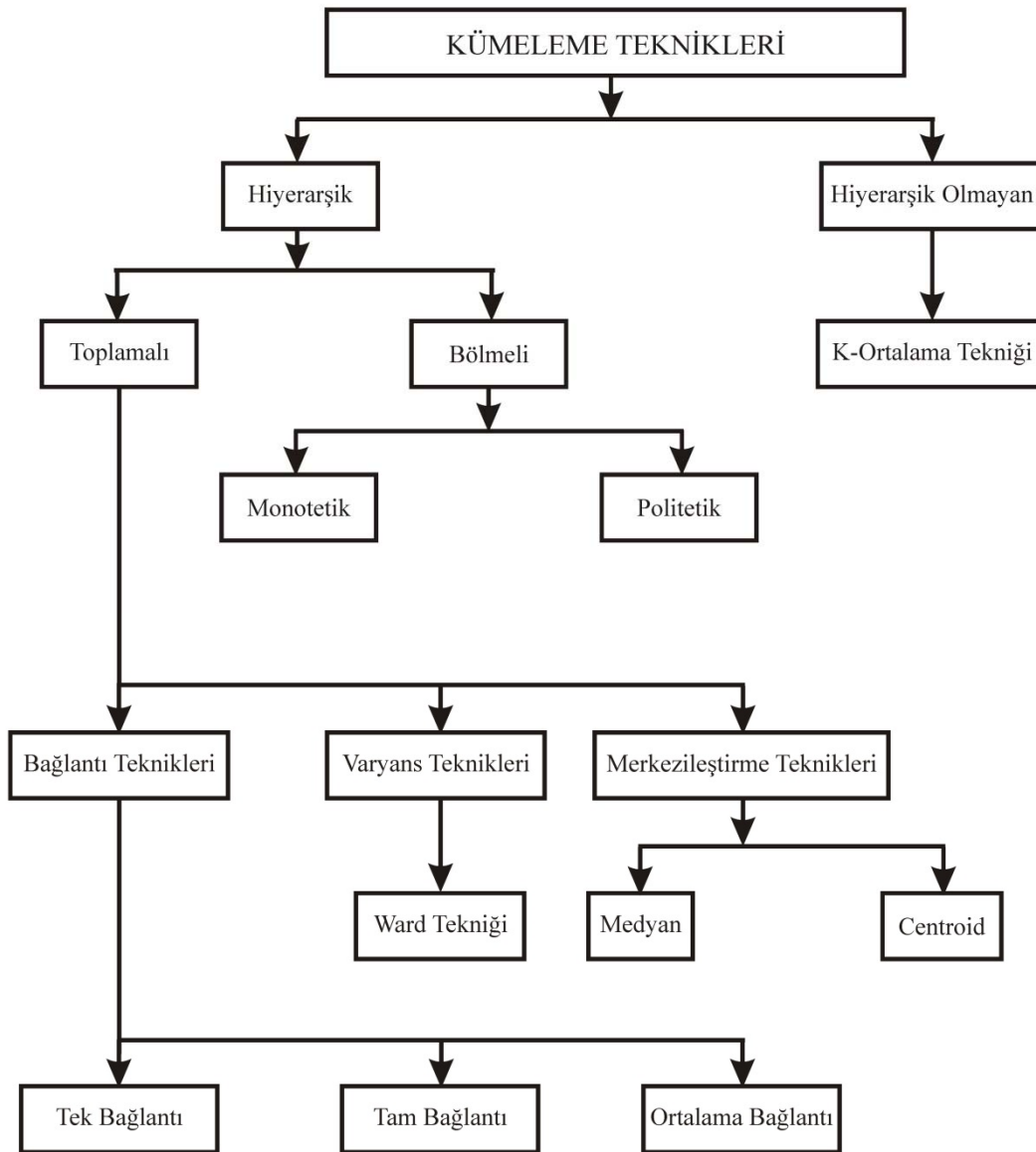
Adımlar	Yapılan İşlem
1	Problemin Tanımlanması
2	Analiz için veri girişinin sağlanması
3	Uzaklık matrisinin elde edilmesi
4	Kümeleme tekniğinin seçilmesi
5	Elde edilen kümelerin yorumlanması
6	Sonuçların anlamlılığının tartışılması

4.1. Kümeleme Analizi Teknikleri

Kümeleme teknikleri; uzaklık matrisini kullanarak nesnelere kendi içinde benzer ve kendi aralarında benzer olmayan gruplar haline getirmeye imkan veren tekniklerdir.

Kümeleme analizi için çok sayıda algoritma ortaya atılmıştır. Ancak literatürde bu algoritmalar iki ana başlık altında toplanmıştır. Bunlardan birincisi, hiyerarşik kümeleme teknikleri, diğeri ise hiyerarşik olmayan kümeleme teknikleri olarak adlandırılmaktadır. Her iki tekniğin de ortak amacı kümeler arasındaki farklılıkları ve kümeler içi benzerlikleri en yüksek düzeye çıkarmaktır. Hangi tekniğin kullanılacağı küme sayısı ile ilişkili olmakla birlikte her iki tekniğin beraber kullanılması daha faydalıdır. Bu şekilde hem sonuçların hangisinin hem de iki tekniğin hangisinin daha iyi sonuçlar verdiğini tespit etmek mümkün olmaktadır.

Kümeleme tekniklerinin genel yapısı Şekil 4.2.'de verilmiştir [27].



Şekil 4.1. Kümeleme tekniklerinin genel görünümü

4.1.1. Hiyerarşik teknikler

Hiyerarşik teknikler, nesnelere arasındaki uzaklıkların matrisi ele alınarak işlemlere başlamaktadır. Bütün gruplar tek bir nesnedен oluşacak şekilde işlemlere başlanmakta yani analizin başında her bir nesne tek başına bir küme olarak düşünülmektedir. Sonrasında birbirine yakın kümeler birleştirilmekte, birleştirilme işlemi ard arda tekrarlanarak devam ettirilmektedir.

Bu tekniklerden kabul görmüş olanları; tek bağlantı tekniği (En Yakın Komsuluk), tam bağlantı tekniği (en uzak komşuluk), ortalama grup bağlantı tekniği, ward tekniği, medyan tekniği ve centroid tekniğidir [28].

4.1.2. Hiyerarşik olmayan teknikler

Küme sayısı hakkında bilgi mevcutsa veya araştırmacı küme sayısına karar vermiş ise bu durumda hiyerarşik teknikler yerine hiyerarşik olmayan kümeleme tekniklerini kullanmak gerekir.

Küme sayılarının önceden belirlenmesi ve küme sayısı seçimlerinin rasgele olması hiyerarşik olmayan kümeleme tekniklerinin dezavantajıdır. Hiyerarşik olmayan kümeleme teknikleri, hiyerarşik tekniklere oranla daha hızlı sonuca ulaşırlar ve nesne sayısı arttıkça o kadar tutarlı sonuçlar çıkarmaktadırlar. Fakat bütün bu avantaj ve dezavantajlarına rağmen hiyerarşik tekniklerle hiyerarşik olmayan tekniklerin beraber kullanılması önerilmektedir.

Hiyerarşik kümeleme tekniği olan k- ortalama tekniği daha çok büyük sayıdaki nesnelere küçük sayıda kümeler oluşturmak için kullanılmaktadır. Bu tekniğin amacı, kümeler içi kareler toplamını minimize ederek yani küme içi değişkenliği en aza indirgeyerek p boyutlu değişkene sahip nesneyi k kümeyle bölmektir. Mümkün bölünmelerin sayısı çok yüksek olduğundan en iyi çözümü beklemek doğru değildir. Bu algoritma bir lokal (yerel) optimum çözümü bulmaktadır. Bu çözüm bir nesnenin bir kümeden diğerine geçmesiyle kümeler içindeki kareler toplamını azaltmaktadır. Algoritmada çeşitli başlangıç bölünmeleri ile çok sayıda tekrarlama yapılarak kümeleme sonuçlarından en iyiye yakın olanının tespit edilmesi amaçlanır. Burada farklı başlangıç bölünmelerinin tekrarlanmasındaki neden, ya bütünden başlanarak ayırtırmak ya da nesneden başlanarak nesnelere birleştirmektir.

Kümelemede başlangıç yöntemi, sonuçta meydana gelen küme çözümünü etkilemektedir. K-ortalama algoritması tekniğinde, öncelikle nesnelere her birinin rasgele biçimde kümelere atanmaları yapılır. Sonrasında yapılan bu atamalar k-ortalama algoritması yardımı ile optimize edilir. Birçok rasgele başlangıç ataması

denenerek kümelerin belli bir sayısı için lokal optimum çözümün olasılığı artmaktadır [29].

K-ortalama tekniği büyük veri setlerinde kullanılabilirliği nedeniyle yaygın olarak uygulanmaktadır. Bu algorithmada arařtırmacının bulunacak küme sayısını belirlemesi algoritmanın vereceđi sonuç için önem taşımaktadır.

K-ortalama tekniđi büyük veri setleri için iyi ve hızlı sonuç veren bir teknik olduđu için çođu ticari veri için kullanışlı bir tekniktir. Optimal kümeleme tam olarak gerçekteşmemesine rağmen pratik kullanımı açısından bu teknik oldukça kullanışlıdır. Bunların yanısıra birçok veri madenciliđi yazılımları k-ortalama tekniđine dayanmaktadır [28].

4.2. Araç Rotalamada Kümeleme Teknikleri

Yukarıda da belirtildiđi üzere kümeleme analizi birçok alanda uygulanmaktadır. Fakat kümeleme analizi uygulamalarında ölçeklenebilirlik (scalability) göz ardı edilmektedir. Arařtırmacılar daima ana bellekte bulunan verinin tam ve eksiksiz olmasını arzu etmekte ve tutarlı kümeler bulmayı amaçlamaktadırlar. Ancak bilinen klasik kümeleme teknikleri ile büyük veri setleri ölçülememektedir. Bu nedenle son yıllarda ölçeklenebilirliđe önem veren yeni algoritmalar geliştirilmiřtir [30]. Yeni geliştirilen bu kümeleme algoritmaları ařađıda açıklanacaktır.

Literatürde çok sayıda kümeleme algoritması olmasına rağmen araç rotalamada kümeleme teknikleri başlıca beř bölüme ayrılabilir. Bunlar;

Model Esaslı Kümeleme Teknikleri: Her bir küme için bir modelin hipotezini kurarak en uygun modeli tespit etmeye çalışırlar.

Merkeze Dayalı Bölümleyici Kümeleme Teknikleri: Yapılan çeřitli bölünmelerden sonra bazı özelliklere göre bu bölünmeler deđerlendirilir ve küme sayısı tespit edilir. Burada her bir bölme bir küme anlamına gelmektedir. Bařlangıç kümesi ise çođunlukla rassal olarak seçilir.

Hiyerarşik Kümeleme Teknikleri: Bu teknikler merkeze dayalı ayırıcı tekniklerde olduđu gibi bazı özellikleri kullanarak nesnelerin hiyerarşik yapı içinde kümelene-mesi temeline dayanır.

Yoğunluğa Dayalı Kümeleme Teknikleri: Bu teknikler, kümeleri oluştururken, veri seti içindeki noktaların yoğunluğunu temel alarak kümeleri belirlemektedirler [31]. Dağınmık şekle sahip kümelerin oluşacağı yani gürültülü verilerin çok olduđu durumlarda oldukça kullanışlıdır [32].

Izgara Tabanlı Kümeleme Teknikleri: Veri setini hücelere ayırarak ızgaralı bir yapı oluşturulması temeline dayanır ve kümeleme bu ızgaralı yapı göz önüne alınarak gerçekleşir [33].

4.2.1. Model esaslı kümeleme teknikleri

Bu tekniklere, model esaslı kümeleme teknikleri denmesinin sebebi; kullanılan algoritmaların veri seti ile uyumlu bir model seçerek kümeleme yapıyor olmasıdır. Özetle bu algoritmalar, veriler ile bazı matematiksel modeller arasındaki uyumu optimize etmeyi amaçlarlar. Çoğunlukla verilerin mevcut olasılık dağılımları tarafından oluşturulduđu varsayılmaktadır. Model esaslı kümeleme teknikleri iki çeşide ayrılmaktadır. Bunlardan ilki, istatistiksel yaklaşım, diğeri ise yapay sinir ağlarıdır.

4.2.2. Merkeze dayalı bölümleyici kümeleme teknikleri

Merkeze dayalı bölümleyici kümeleme teknikleri, belli bir sayıda nesneden oluşan veri setini başlangıçta belirlenen sayıda kümeye ayırmaktadırlar. Kümeleme sonucunda meydana gelen kümelerde, küme içi benzerlik yüksek iken kümeler arası benzerlik düşüktür. Fakat başlangıçta tespit edilen küme sayısı doğru tahmin edilmediğinde elde edilen sonuçlarda tutarsızlıklar oluşmaktadır [34].

Uygulamada en çok karşılaşılan merkeze dayalı bölümleyici kümeleme tekniklerini k-means, k-medoid, PAM, CLARA ve CLARANS algoritmaları olarak belirtmek mümkündür.

4.2.3. Hiyerarşik kümeleme teknikleri

Hiyerarşik kümeleme tekniklerinde, başlangıçta küme sayısı belirtilmemektedir. Algoritma, veri seti ve uzaklıklar matrisi girdileri ile tanımlanmaktadır. Sonuçta çıktı olarak meydana getirilen kümeler hiyerarşiktir. Hiyerarşik kümeleme tekniklerinin birçoğunda uygulanan süreç optimizasyon temelli değildir. Bu tekniklerin amacı, birleşme tamamlanıncaya kadar bölmenin ilerlemesi için tekrarlamalar yaparak bazı yaklaşımları tespit etmektir [35]. Hiyerarşik kümeleme teknikleri ağaca benzer (dendogram) bir grafik meydana getirirler.

Hiyerarşik kümeleme tekniklerini, hiyerarşik birleşmenin aşağıdan yukarıya ve yukarıdan aşağıya yapılmasına bağlı olarak toplayıcı ve ayırıcı hiyerarşik kümeleme teknikleri olarak iki çeşide ayırmak mümkündür.

Hiyerarşik kümeleme tekniklerinin bazı dezavantajları bulunmaktadır. Bunlar, veri nesneleri bir kez birleştiğinde veya bölündüğünde ikinci adım artık yeni oluşturulan kümeler üzerinden devam edecektir. Yani daha önce yapılan herhangi bir işlemi geri almak mümkün olmamaktadır. Bu nedenle bu teknikler gürültülü veriye karşı oldukça duyarlıdır. Ayrıca konveks olmayan kümelerin bulunduğu veri setleri üzerinde uygulandıklarında da bazı sorunlar ile karşılaşmaktadır. Yine de bu teknikler büyük kümeleri bölme eğilimine sahiptir [36].

Geleneksel hiyerarşik kümeleme teknikleri sadece küçük ve orta büyüklükteki veri setleri için kullanılmaktadır. Ancak günümüz şartlarında artık çok büyük boyutlara sahip olan ve bununla birlikte gürültülü veriler içeren veri setleri söz konusudur. İşte sözü edilen bu veri setlerini ölçeklendirebilmek için kümeleme algoritmaları geliştirilmiş ve birçok farklı algoritma ortaya atılmıştır.

Hiyerarşik kümeleme algoritmaları, bölümleyici algoritmalara göre daha esnektirler. Bu algoritmaların başlangıç parametrelerine ihtiyacı olmamaktadır. Ancak, yukarıda anlatılan bölümleyici kümeleme algoritmalarına oranla hesaplama karmaşıklıkları daha fazladır.

Hiyerarşik kümeleme tekniklerinin avantajları, veri nesnelерinin yoğunluğu konusunda esnek olması, uzaklık ölçülerinin tümünü değerlendirebiliyor olması ve her ölçek türü ile uygulanabilir olmasıdır. Tekniğin dezavantajları ise daha önce belirtildiği gibi, iterasyonların bitiş kriterlerinde belirsizliklerin olması ve oluşturulan kümelere iyileştirme amacı ile bile geri dönülememesidir [37].

4.2.3.1. Toplayıcı kümeleme teknikleri

Toplayıcı kümeleme algoritması olarak bilinen algoritma 1990 yılında Kaufman ve Rousseeuw tarafından ortaya atılmıştır. Algoritmada temel mantık oldukça basittir. Kümelenecek veri seti n sayıda nesne içeriyorsa, algoritma n sayıda küme ile başlar. Kümelerin her bir çifti için belirlenen uzaklık ölçüleri kullanılarak $(n \times n)$ boyutunda bir uzaklık matrisi oluşturulur. Daha sonra algoritma birbirine en yakın kümeleri birleştirir ve $(n - 1) \times (n - 1)$ boyutunda yeni bir uzaklık matrisi meydana getirir. Bu süreç veri setindeki tüm nesnelер bir küme içinde yer alıncaya kadar tekrarlanır [38]. Kümeleme süreci, aşağıdan yukarıya doğru çalışan bir süreçtir [39].

Bu tekniklerin hesaplama karmaşıklığı yüksektir. Bu nedenle süreci hızlandırmak için bazı yöntemlerden yararlanılmadır. Bu yöntemler BIRCH, CURE, ROCK ve CHAMELEON gibi algoritmalarıdır.

4.2.3.2. Ayırıcı kümeleme teknikleri

Ayırıcı kümeleme algoritması olarak bilinen algoritma 1990 yılında Kaufman ve Rousseeuw tarafından ortaya atılmıştır. Algoritma veri setinin tüm nesnelерini tek bir küme içerisinde kabul ederek başlar sonrasında veri setini alt kümelere ayırarak süreci başlatır. Daha sonra alt kümeler daha da küçük kümelere ayrılarak algoritma devam

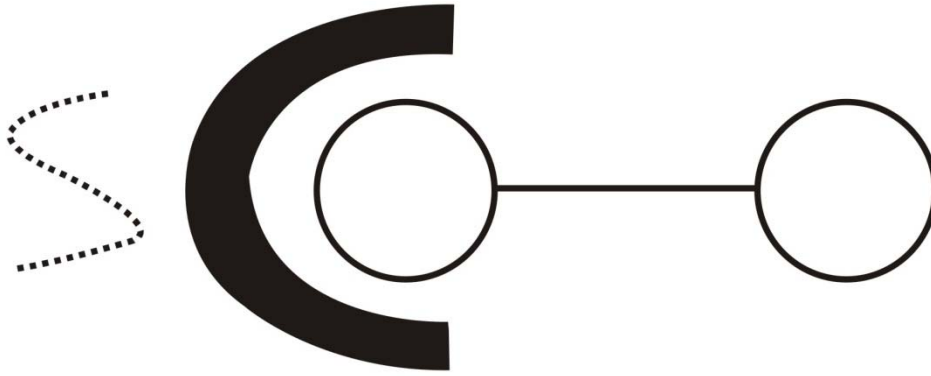
ettirilir. Bu teknikte kümeleme toplayıcı kümeleme tekniğinin tersine yukarıdan aşağıya doğru bir süreç ile yapılır [40].

Ayrııcı kümeleme tekniği toplayıcı kümeleme tekniği ile karşılaştırıldığında; ayrııcı kümeleme teknikleri daha fazla işleme ihtiyaç duyduğundan daha yüksek oranda tutarsız sonuç oluşturmaktadır. Bu nedenle uygulamalarda öncelikle toplayıcı kümeleme tekniklerinden yararlanılmalıdır [27].

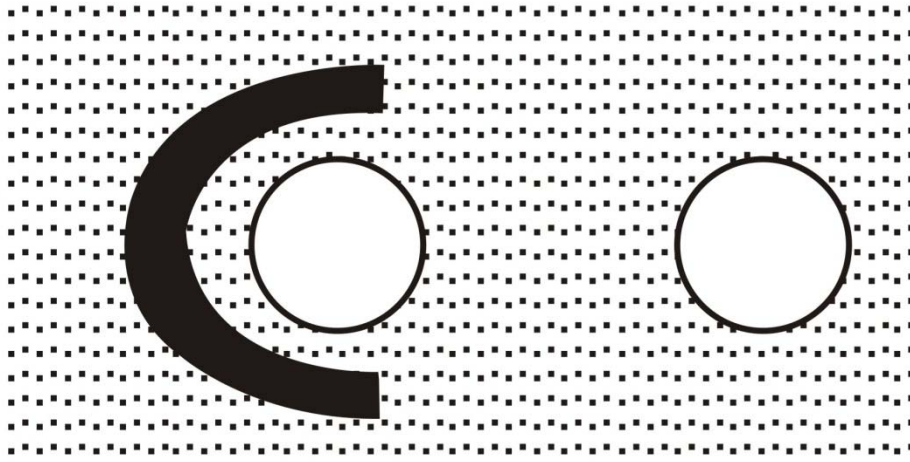
4.2.4. Yoğunluğa dayalı kümeleme teknikleri

Yoğunluğa dayalı kümeleme teknikleri, yukarıda bahsedilen teknikler gibi kümeleri meydana getirirken nesnelere arasındaki uzaklıklardan yararlanmamakta; bu teknikler nesnelere göre yoğunluklarına göre kümeleri belirlemektedir.

Aşağıdaki şekil 4.2. ve şekil 4.3.'de küme nesnelere gürültülü nesnelere eklenerek oluşturulan veriler için yoğunluğa dayalı kümeleme gösterilmiştir. Şekil 4.2.'de her bir noktanın kendi kümesinin dışında kalan noktalara olan uzaklığının kendi bulunduğu kümedeki noktalara olan uzaklığından daha uzun kaldığı görülmektedir. Şekil 4.3.'de ise kümeler düşük yoğunluklu bölgelerden yüksek yoğunluklu bölgelere doğru ayrılmaktadır. Gürültülü veriler mevcutsa iki dairesel kümeyi şekil 4.2'deki gibi birleştirmek imkansızdır. Ayrıca, şekil 4.2'de gürültülü veriler, kümelerin dışında bir eğri oluşturmakta; bu aykırı durumun şekil 4.3 incelendiğinde ortadan kalkmış olduğu görülmektedir. Kısaca belirtmeye çalışılan yoğunluğa dayalı kümeleme tekniklerinin, gürültülü ve aykırı değerler varsa sıkça kullanılmasıdır [41].



Şekil 4.2. Yoğunlaşmış kitle (contiguity) tabanlı kümeleme



Şekil 4.3. Yoğunluk tabanlı kümeleme

DBSCAN, OPTICS, DENCLUE yoğunluğa dayalı kümeleme tekniklerinin en yaygın olanlarıdır [42]. Yapılan bir araştırmada DBSCAN ve OPTICS algoritmaları küresel olmayan kümelenmeleri bulma konusunda merkeze dayalı bölümleyici kümeleme algoritmalarından olan K-ortalama algoritmasından çok daha başarılı olduğu görülmüştür [43]. Yoğunluğa dayalı kümeleme tekniklerini karşılaştırsak OPTICS algoritması görselleştirme tabanlı olduğu için DBSCAN algoritmalarının aksine, büyük boyutlu veritabanlarına uygulandığında elde edilen grafikleri yorumlamak çok zorlaşmaktadır [43]. Perakende sektöründe market zincirleri her geçen gün zincirlerine yeni mağazalar eklemekte olduğundan çok sayıda markete dağıtımlarını gerçekleştirmek durumundadırlar. Yapılan bir araştırma sonucunda DBSCAN algoritması sonucundaki küme kalitesinin DENCLUE algoritmasındakine göre daha iyi olduğu görülmüştür [44]. Bu nedenlerden ötürü yoğunluğa dayalı kümeleme

tekniklerinden DBSCAN algoritmasının uygulama kısmı için avantajlı olacağı tespit edilmiştir.

Uygulama kısmında ele aldığımız problemimizde gürültülü ve aykırı değerler çokça yer aldığından dolayı tezin uygulama kısmında yoğunluğa dayalı kümeleme tekniklerinden biri olan DBSCAN algoritması kullanılmıştır. Uygulama kısmında DBSCAN algoritması kullanıldığından DBSCAN algoritması ayrıntılı olarak açıklanacaktır.

4.2.4.1. DBSCAN algoritması

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algoritması, 1996 yılında ortaya atılmış ve yoğunluğa dayalı kümeleme tekniklerinin temelini meydana getirmiştir [45].

Algoritma kümeleri oluştururken nesnelerin yoğunluklarını hesaba katar. Kümeler, yüksek yoğunluklu veri nesneleri ile tanımlanmakta; düşük yoğunluklu nesnelerin bulunduğu kümeler ise aykırı veya gürültülü noktaları göstermektedir. DBSCAN, özellikle büyük veritabanları ve gürültülü nesneler içeren veri setleri için oldukça kullanışlıdır. Bunun yanısıra farklı büyüklük ve şekillerdeki kümelerin belirlenmesinde de sıkça kullanılmaktadır [46].

DBSCAN algoritması, ϵ ve MinPts olmak üzere iki adet girdi parametresine ihtiyaç duyar. Burada; ϵ : Yarıçap (kümedeki her bir nesnenin ϵ yarıçapındaki komşuluğu) MinPts: Küme çevresindeki en az nesne sayısı olarak tanımlanmaktadır.

Yukarıda açıklanan bu iki giriş parametresine karşı algoritmanın duyarlı olması DBSCAN algoritmasının dezavantajı olarak görülmektedir. Algoritmanın ortalama hesaplama karmaşıklığı, $O(n \log n)$ dir. Burada n veri setindeki nesne sayısıdır [45]. Algoritmanın adımlarının daha iyi anlaşılması için bazı kavramların tanımlanması yararlı olacaktır.

Bir kümenin nesnelere ele alınır eğer her biri için ϵ yarıçaplı alanında en az MinPts değeri kadar nesne yer alıyorsa bu kümedeki nesnelere yoğunluklarına göre tanımlanırsa yoğun nesnelere olarak tanımlanmaktadır [47].

Ele alınan bir nesnenin ϵ yarıçapı ile komşuluğu; nesnenin ϵ –komşuluğu (ϵ - neighborhood) olarak adlandırılmaktadır.

Eğer bir nesnenin ϵ komşuluğu en az MinPts sayıda nesneyi kapsıyorsa, bu nesne çekirdek nesne (core object) olarak adlandırılmaktadır.

Verilen bir veri setinde, eğer p nesnesi, q nesnesinin ϵ komşuluğu içinde ise p nesnesi q nesnesinin doğrudan yoğunluğa katılan nesnesi (directly density reachable) olarak tanımlanmaktadır ve burada q nesnesi çekirdek nesneyi temsil etmektedir.

Algoritma şu şekilde çalışmaktadır:

Adım 1: Veri setindeki her bir nesnenin ϵ komşuluğunu kontrol et.

Adım 2: Bu alanda MinPts ile belirlenen sayıdan daha fazla nesne var ise buna çekirdek nokta adını ver.

Adım 3: Her bir kümeyi, çekirdek nesne etrafında büyüt.

Adım 4: Veri nesnelere bir kümeye atanma işlemi tamamlandığında yani herhangi bir kümeye artık hiçbir veri nesnesi eklenemediğinde algoritmayı sonlandır.

Adım 5: Eğer hala bir kümeye atanmamış veri nesnesi söz konusu ise bu nesnelere gürültü olarak nitelendir.

4.2.5. Izgara tabanlı kümeleme teknikleri

Izgara Tabanlı Kümeleme Teknikleri, veri setini ızgaralı bir yapı haline getirerek belli sayıda hücreye bölmekte ve tüm kümeleme işlemlerini bu ızgaralı yapı üzerinde gerçekleştirmektedir. Izgara hücrelerin yoğunlaştığı yerleri hesaba katarak küme sayısının belirlenmesinden sonra veri nesnesi olmayan ızgara hücreleri ile küme merkezleri arasındaki uzaklığın hesaplanması ile bu teknikte yer alan algoritmaların hızlı çalışması sağlanır ki bu durum ızgara tabanlı tekniklerin en önemli avantajı olarak görülmektedir [48].

Izgara Tabanlı Kümeleme Teknikleri en etkili algoritmalar olmasına rağmen, algoritmaların etkisi önceden belirlenmiş ızgaraların büyüklüğünden ve belirli hücrelerin eşik değerlerinden ciddi biçimde etkilenmektedir. Bu durum bu tekniklerin dezavantajı olarak kabul edilmektedir [31].

Izgara Tabanlı Kümeleme Tekniklerine ızgara hücrelerinde depolanmış istatistiksel bilgiyi belirten STING algoritması, dalga dönüşüm yöntemi kullanarak nesnelere kümeleyen WaveCluster algoritması ve çok boyutlu veri uzayında kümeleme için hem ızgara tabanlı hem de yoğunluğa dayalı yaklaşımı temsil eden CLIQUE algoritması örnek olarak verilebilir.

BÖLÜM 5. GENETİK ALGORİTMA

Bugüne kadar ortaya atılan bütün algoritmaların temel amacı belirli bir problemi çözmektir, yani potansiyel çözümlerden meydana gelen bir çözüm uzayını araştırmaktır. Küçük boyutlu çözüm uzayına sahip problemlerde tüm uzayını eksiksiz arayan kesin yöntemler kullanılmaktadır; fakat arama uzayının boyutunun büyük olduğu problemlerde bazı özel yapay zeka tekniklerini kullanmak gerekmektedir. Genetik Algoritma (GA) böyle bir yöntemdir; arama işlemini genetik kalıtım gibi bazı doğal olayları kullanarak yapan stokastik bir arama algoritmasıdır [49].

5.1. Genetik Algoritma'daki Kavramlar

GA'nın nasıl çalıştığı açıklanmadan önce algoritmada geçen kavramlar açıklanacaktır.

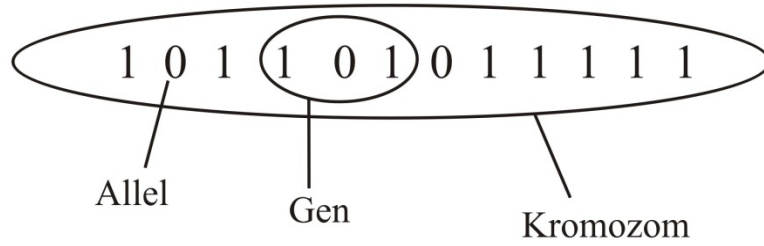
Genotip ve fenotip: Genotip bireyin mevcut gen yapısı özellikleri iken fenotip ise organizmanın gözlenen özellikleridir [50].

Allel: Alleller bir kromozomdaki en küçük parçalardır. Doğada, alleller XY gibi iki tane sembolden meydana gelmektedir, fakat genetik algoritma uygulamalarında genellikle tek sembol ile belirtilirler.

Gen: Gen, bir kromozom (birey) bünyesinde belirli bir fenotip özelliği temsil eden allel ya da alleller dizisidir.

Kromozom (Birey): Genetik algoritmada bir çözümü temsil eden, belirli uzunluğa sahip ve bireyin tüm genetik bilgisini barındıran kromozom yani birey alleller dizisidir.

Şekil 5.1'de yukarıda açıklanan allel, gen ve kromozom arasındaki ilişki belirtilmiştir.



Şekil 5.1. Allel-gen-kromozom ilişkisi

Popülasyon (Nüfus): Popülasyon çözüm uzayındaki birden fazla sayıda noktadan oluşan çözüm kümesidir, yani birden fazla sayıda kromozomdan meydana gelir.

5.2. Genetik Algoritma Aşamaları

Aşama 1. Genetik Kodlama

Bir bireyin, genlerinin deşifre edilmiş fenotipinin kodlanmış hali bireyin gen yapısını verir. En iyileme probleminde de benzer şekilde, problemin bir çözümü için yapılan genetik kodlama, bu bireyin gen yapısını meydana getirmektedir.

Belirlenen kodlama ile en iyileme probleminin olası tüm olurlu çözümlerinin şifrelenebilmesi ve çaprazlama, mutasyon (dönüşüm) gibi genetik operatörlerin, bu kodlama ile meydana getirilmiş bireylere uygulanabilir olması gerekmektedir. Genetik kodlamadaki ufak değişiklikler sonucunda algoritmanın performansı büyük ölçüde değişebilmektedir, çoğunlukla araştırmacının sezgileri ile belirlenen sözkonusu kodlamanın doğru seçilmesi algoritmanın performansını ciddi biçimde değiştirmektedir [51]. Varolan çok sayıda genetik kodlama çeşidi aşağıda açıklanmaktadır. Başlıca kodlama yöntemlerine ikili kodlama, gri kodlama, permütasyon kodlama, tamsayı kodlama, gerçek sayılı kodlama, rassal sayılı kodlama örnek verilebilir. Uygulama kısmında permütasyon kodlama kullanıldığından permütasyon kodlama açıklamalı olarak verilmiştir.

Permütasyon Kodlama: Gezgin satıcı veya araç rotalama gibi, sıralamanın önem arzettiği problem türlerinde kullanılması elverişlidir. Satıcı tarafından talep noktalarının ziyaret edilme sırasına göre belirlenen kodlamadır. Küçük bir örnek

üzerinde gösterecek olursak 9 talep noktası olan bir GSP için uğranılacak müşteriler sırasıyla 3-6-5-4-7-1-2-9-8 ise kodlama da [3 6 5 4 7 1 2 9 8] şeklinde olacaktır [52].

Aşama 2. Başlangıç popülasyonu oluşturma

Problemin yapısına göre tespit edilecek olan boyutta, rassal çözümlerden oluşan bir başlangıç popülasyonu meydana getirilir. Çözüm uzayındaki farklı bölgelerden çözümlerle meydana getirilen bu küme sayesinde, diğer meta-sezgisel yöntemlerden farklı olarak, farklı bölgelerde aynı anda arama yapılabilecektir.

Aşama 3. Uygunluk değerlendirme (evaluation of fitness) fonksiyonu

Uygunluk değerlendirme fonksiyonu, GA'da problem ile algoritma arasında bağlantı sağlayan bir işlemdir. Bu fonksiyon bireylerin popülasyon içerisindeki değerini uygunluk değerlerine göre belirler, daha iyi bireyler daha büyük şansla hayatta kalacaktır ve yeniden üreyecektir. Dolayısıyla uygunluk değerlendirme fonksiyonu tanımlamak çok önemlidir. Olurlu olmayan çözümlerin bu fonksiyon sayesinde hesaba katılmaması gerekmektedir [49]. En iyileme probleminde tanımlanacak amaç fonksiyonu bu görevi üstlenecektir. Kodlanmış birey algoritma esnasında deşifre edilerek belirlenen amaç fonksiyonuna sokulur ve amaç fonksiyon değeri hesaplanır. Uygulama kısmında ele alınan problem bir ARP olması dolayısıyla öncelikle oluşturulan rotalar için toplam kat edilecek mesafenin en küçüklenmesi amaçlanmaktadır. Litaratürde benzer problemler için;

D_{ij} : i ile j şehirleri arasındaki mesafe olmak üzere;

$$f(x) = \frac{1}{\sum_i \sum_j D_{ij}} \quad (5.1)$$

uyum fonksiyonunun kullanıldığı görülmüştür.

Aşama 4. Seçim operatörü

Seçim popülasyon içerisinde bir sonraki nesile aktarılacak olan bireylerin tespit edilmesini sağlar. Seçim operatörü GA'da yeniden üreme işleminin en önemli kısmını oluşturur. Bu operatör ile algoritmadaki birey çeşitliliği artacak ve böylece çözüm uzayında farklı bölgelerde arama yapılabilecektir [52]. Çok sayıda seçim yöntemi ortaya atılmıştır. Bunlardan en önemlileri; rulet çemberi, sıralamalı, turnuva ve elitizm

seçim yöntemleridir. Bunlar arasından uygulamada kullanacağımız rulet çemberi yöntemi açıklanacaktır.

Rulet Çemberi Seçim Yöntemi: Uygunlukla Orantılı Seçim yöntemlerinde biri olan bu yöntem en yaygın olarak kullanılan seçim yöntemidir. Bireyler uygunluk değerlerine göre belirli olasılık değerlerine sahip olurlar. Bu olasılıklar, her bir bireyin uygunluk değerinin toplam uygunluk değerine oranlanması ile elde edilir.

Toplam popülasyon büyüklüğü m olsun ve en büyükleme problemi olsun.

Adım 1. Her bir i bireyinin uygunluk değeri f_i hesaplanır.

Adım 2. Tüm bireylerin uygunluk değerleri toplanır ve toplam uygunluk değeri

$$F = \sum_{i=1}^m f_i \text{ hesaplanır.}$$

Adım 3. Her bir bireyin seçilme olasılığı

$$p_i = f_i / \sum_{i=1}^m f_i \text{ hesaplanır.}$$

Adım 4. Bu oranlar sırasıyla toplanarak her bir birey için $(0,1)$ aralığında ve seçilme üst sınırını veren değerler elde edilir.

Adım 5. Popülasyondaki birey sayısı kadar $(0,1)$ aralığında rassal sayı atılır ve her rassal sayıdan bir sonra gelen birikimli olasılık değerine karşılık gelen birey seçilir [52].

Aşama 5. Çaprazlama

Genetik algoritmanın performansını etkileyen önemli parametrelerden biridir. Tek noktalı, iki noktalı, düzenli (üniform), kısmi planlı, sıra tabanlı, pozisyon bazlı ve dairesel çaprazlama gibi çeşitleri vardır. Çaprazlamada amaç, ebeveyn kromozom genlerinin yerini değiştirerek çocuk kromozomlar üretmek ve böylece varolan uygunluk değeri daha yüksek olan kromozomlar üretmektir.

Ele alınan problemimiz ARP olduğundan bu tür problemlerde sıkça kullanılan sıra tabanlı çaprazlama yöntemi kullanılmıştır. Çaprazlama işleminin diğer bir önemli parametresi de çaprazlama oranıdır. Uygulama kısmında ele alınan problem için yapılan denemeler sonucunda çaprazlama oranı %80 olarak belirlenmiştir.

Sıra tabanlı çaprazlamada, öncelikle iki kesim noktası tespit edilir. Daha sonra kesim noktaları arasında kalan genler konumları korunarak yeni kromozoma aktarılır. Kalan boş konumlar ise eşleştirilen diğer kromozomdan (Ebeveyn 2) kullanılan genler çıkarılarak kalan genleri soldan başlayarak sırayla yeni oluşturulan kromozoma soldan başlanarak doldurulur. Buna ait bir örnek Tablo 5.1.' de verilmektedir.

Tablo 5.1. Sıra tabanlı çaprazlama operatörü

Ebeveyn 1	1 2 3 4 5 6 7 8 9
Ebeveyn 2	4 5 3 6 8 9 7 2 1
Çocuk	3 8 9 4 5 6 7 2 1

Aşama 6. Mutasyon (dönüşüm) operatörü

Genetik işlemlerden biri olan mutasyon sayesinde GA yerel en iyi çözüme yakalanmaktan kurtulabilmektedir [12]. Mutasyon işlemi, arama uzayında çaprazlama işlemi ile ulaşılamayan bölgelere ulaşmayı sağlamaktadır [53]. Amaç, varolan bir kromozomun genlerinin bir ya da birkaçının yerlerini değiştirerek yeni kromozom meydana getirmektir. Özellikle permütasyon kodlamalı kromozomlarda kullanılan ters çevirme, ekleme, yerdeğiştirme, ikili değişim ve sezgisel mutasyonlar gibi çeşitleri vardır [52]. Bunların içinde en çok kullanılan ikili değişim yöntemidir. Tablo 5.2.' de ikili değişim yönteminin permütasyon kodlama ile hazırlanmış iki kromozom üzerinde uygulanışı gösterilmektedir.

Tablo 5.2. İkili değişim yöntemi ile mutasyon

Mutasyon işlemi uygulanmadan önceki kromozom	6 5 1 2 7 3 4
Mutasyon işlemi uygulandıktan sonraki kromozom	2 5 1 6 7 3 4

İkili değişim yöntemi ile kromozomdan rassal olarak seçilen iki gen karşılıklı yer değiştirir. Hangi genlere mutasyon işlemi uygulanacağı, mutasyon oranına bağlı olarak belirlenir. Uygulama kısmında ele alınan problem için yapılan denemeler sonucunda mutasyon oranı % 0,1 olarak belirlenmiştir. Algoritmadaki mutasyon işleminde

öncelikle her gen için bir rassal sayı oluşturulmaktadır. Bu rassal sayılar içinde mutasyon oranından küçük bir sayı oluştuğunda ilgili olan gen seçilmekte ardından yine rassal olarak başka bir gen daha seçilmekte ve bu seçilen iki genin karşılıklı olarak yerleri değiştirilmektedir.

Aşama 7. Bitirme Koşulu

Algoritmada, yeniden üreme aşamasından sonra eğer belirlenen bitirme koşulu sağlanıyorsa elde edilen popülasyondaki en iyi birey algoritma sonucundaki en iyi çözüm olacaktır. Bu bitirme koşulu; belirli bir iterasyon sayısı, belirli bir eşik değeri ya da belirli sayıda iterasyon sonucunda değişme gerçekleşmiyorsa durdurmak olabilir. Ele alınan uygulamadaki problemde bitirme koşulu olarak iterasyon sayısı kullanılmış olup, yapılan denemeler sonucunda iterasyon sayısının 10000 olması uygun görülmüştür.

BÖLÜM 6. UYGULAMA

6.1. Uygulamanın Tanıtılması ve Amacı

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

Uygulamada mağazalara olan dağıtım operasyonu sadece haftaiçi günlerde gerçekleştirilmektedir.

Bu çalışmada dağıtım filomuzda iki çeşit kamyonumuz mevcuttur. Birinci çeşit kamyonlar soğuk bir ortamda dağıtılması gereken taze ürün gruplarını taşıyan taze ürün kamyonlarıdır. İkinci çeşit kamyonlar ise kuru ürün gruplarını taşıyan kuru ürün kamyonlarıdır. Bu iki çeşit kamyon arasındaki fark taze ürün kamyonunda soğutucu ve yalıtımın olmasıdır.

Süpermarket zincirindeki mağazalar iki çeşittir. Birinci çeşit mağazalar haftaiçi her gün taze ve kuru ürün talep eden mağazalar olup günlük mağazalar olarak adlandırılmaktadır. İkinci çeşit mağazalar ise haftaiçi günlerden sadece pazartesi, çarşamba ve cuma günleri taze ve kuru ürün talep eden mağazalar olup yedek mağazalar olarak adlandırılmaktadır. Uygulamada toplam 78 mağaza olup bunlarda 66 adedi günlük, 12 adedi yedek mağazadır. Tablo 6.1.'de 78 mağazanın türü belirtilmiştir.

Tablo 6.1. Mağaza noları ve türleri

Mağaza No	Mağaza Türü	Mağaza No	Mağaza Türü	Mağaza No	Mağaza Türü
1	Günlük	27	Yedek	53	Günlük
2	Günlük	28	Günlük	54	Günlük
3	Günlük	29	Günlük	55	Günlük
4	Günlük	30	Yedek	56	Günlük
5	Günlük	31	Günlük	57	Günlük
6	Yedek	32	Günlük	58	Günlük
7	Günlük	33	Yedek	59	Günlük
8	Günlük	34	Günlük	60	Günlük
9	Günlük	35	Günlük	61	Günlük
10	Günlük	36	Yedek	62	Günlük
11	Günlük	37	Günlük	63	Günlük
12	Günlük	38	Günlük	64	Günlük
13	Günlük	39	Yedek	65	Günlük
14	Günlük	40	Günlük	66	Günlük
15	Günlük	41	Günlük	67	Günlük
16	Günlük	42	Yedek	68	Günlük
17	Günlük	43	Günlük	69	Günlük
18	Günlük	44	Günlük	70	Günlük
19	Günlük	45	Yedek	71	Yedek
20	Günlük	46	Günlük	72	Günlük
21	Günlük	47	Günlük	73	Günlük
22	Günlük	48	Günlük	74	Yedek
23	Günlük	49	Günlük	75	Günlük
24	Yedek	50	Günlük	76	Günlük
25	Günlük	51	Günlük	77	Yedek
26	Günlük	52	Günlük	78	Günlük

Uygulamada tüm mağazalara (günlük ve yedek mağazalar) dağıtım yapılan günler yoğun gün olarak ifade edilmiş, Sadece günlük mağazalara dağıtım yapılan günler ise sakin gün olarak ifade edilmiştir. Tablo 6.2.'de haftaiçi günlerin her birinin çeşitleri ve ürün talep eden mağaza sayıları belirtilmiştir.

Tablo 6.2. Günlerin çeşitleri ve ürün talep eden mağaza sayıları

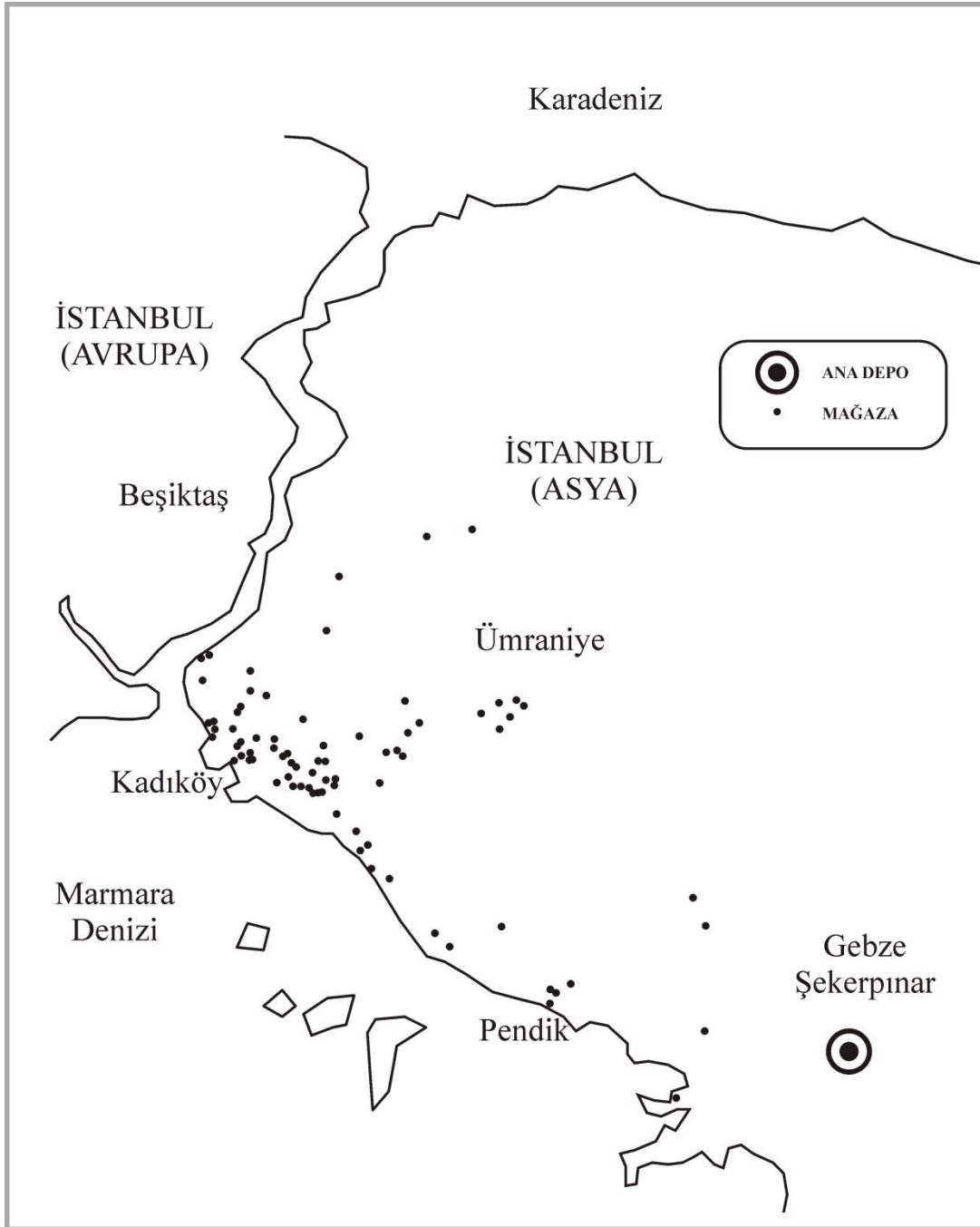
Gün	Gün Çeşidi	Dağıtım yapılacak mağaza sayısı
Pazartesi	Yoğun	78
Salı	Sakin	66
Çarşamba	Yoğun	78
Perşembe	Sakin	66
Cuma	Yoğun	78

Bu çeşitlere istinaden 4 adet alt problem meydana gelmiştir. Bunlar sırasıyla;

- Kuru ürün, yoğun gün dağıtım problemi,

- Taze ürün, yoğun gün dağıtım problemi,
- Kuru ürün, sakin gün dağıtım problemi,
- Taze ürün, sakin gün dağıtım problemidir.

Uygulamada bu 4 alt problem ayrı ayrı ele alınıp çözülecektir. Şekil 6.1'deki noktalar mağazaların, büyük nokta ise ana deponun yerinin harita üzerindeki temsil etmektedir. Firmanın araç filosu homojen araçlardan oluşmaktadır ve taze araçlar ile kuru araçların hepsi 40 palet kapasitelidir.



Şekil 6.1. Veri setinin harita üzerindeki görüntüsü

Uygulamada her bir alt problem için yoğunluk tabanlı kümeleme algoritmalarından DBSCAN ve yeni geliştirilen GA destekli DBSCAN'ın karşılaştırması yapılmıştır. Bu iki yöntemin karşılaştırılması için 20 adet farklı veri seti her bir alt problemde uygulanmıştır. Bu veriler EK-1'de verilmiştir. Dağıtım şirketi DBSCAN algoritması sonucunda oluşan kümelerden toplamda 32-40 palet aralığında talebi olan kümeleri atanabilir küme olarak kabul etmektedir.

6.2. Uygulamada Kullanılan Algoritmaların İşleyişleri

Karşılaştırılacak kümeleme algoritmalarının sonuçları, oluşturmuş oldukları kümelerin her birinin rotaları (tur uzunlukları) ana depoda başlayıp ana depoda son bulacak şekilde en iyileyici rotalama yöntemlerinden olan dal sınır algoritmasından yararlanılarak belirlenmiştir.

Bu çalışmada, 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. Bu çalışmada karşılaştırılan algoritmalar; DBSCAN Algoritması ve GA destekli DBSCAN algoritmasıdır.

Uygulama aşamasında veri setimizdeki mağazaların kuşbakışı uzaklıkları yerine gerçek uzaklıkları hesaba katılmış ve buna göre kümelemeler ve rotalamalar belirlenmiştir.

6.2.1. DBSCAN algoritması

Öncelikle her bir alt problem için DBSCAN algoritmasının parametrelerinin baz 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.

Algoritmanın adımları aşağıdaki Tablo 6.3.'de verilmiştir.

Tablo 6.3. DBSCAN algoritmasının aşamaları

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 algoritmayı sonlandır.
5	Eğer hala bir kümeye atanmamış veri nesnesi söz konusu ise bu nesnelere gürültü olarak nitelendir.

Yoğunluk tabanlı bir kümeleme algoritması olan DBSCAN'ın iki adet kümeleme parametresi vardır. Bunlar MinPoints ve Epsilondur. İlk aşamada MinPoints değeri sonrasında ilk aşamada belirlenen MinPoints baz değerine bağlı olarak Epsilon baz değeri tespit edilecektir.

MinPoints baz değerini belirleme:

Adım 1. Veri setinde yer alan mağazaların en düşük sayıda ve en yüksek sayıda palet ürün talep eden mağazaları belirlenir.

Adım 2. En düşük sayıda palet talep eden mağazanın palet sayısının katları alınır ve kamyon kapasitesi olan 40 palete yakınsayacak şekilde kaç kat olduğu hesaplanır. Bu kat sayısı problemde atanabilecek kümelerin en çok kaç mağazalı kümeler olabileceğini gösterir.

Adım 3. En yüksek sayıda palet talep eden mağazanın palet sayısının katları alınır ve kamyon kapasitesi olan 40 palete yakınsayacak şekilde kaç kat olduğu hesaplanır. Bu kat sayısı problemde atanabilir kümelerin en az kaç mağazalı kümeler olabileceğini gösterir.

Bu adımlar sonucunda tespit ettiğimiz atanabilecek küme olmayı sağlayan en az mağaza sayılı durum MinPoints baz değeri kabul edilir. Bu MinPoints değeri ile algoritma çalıştırıldığında kümeler bu MinPoints değeri kadar yada daha fazla mağaza sayısına sahip olarak meydana geleceklerdir.

Epsilon baz değerini belirleme:

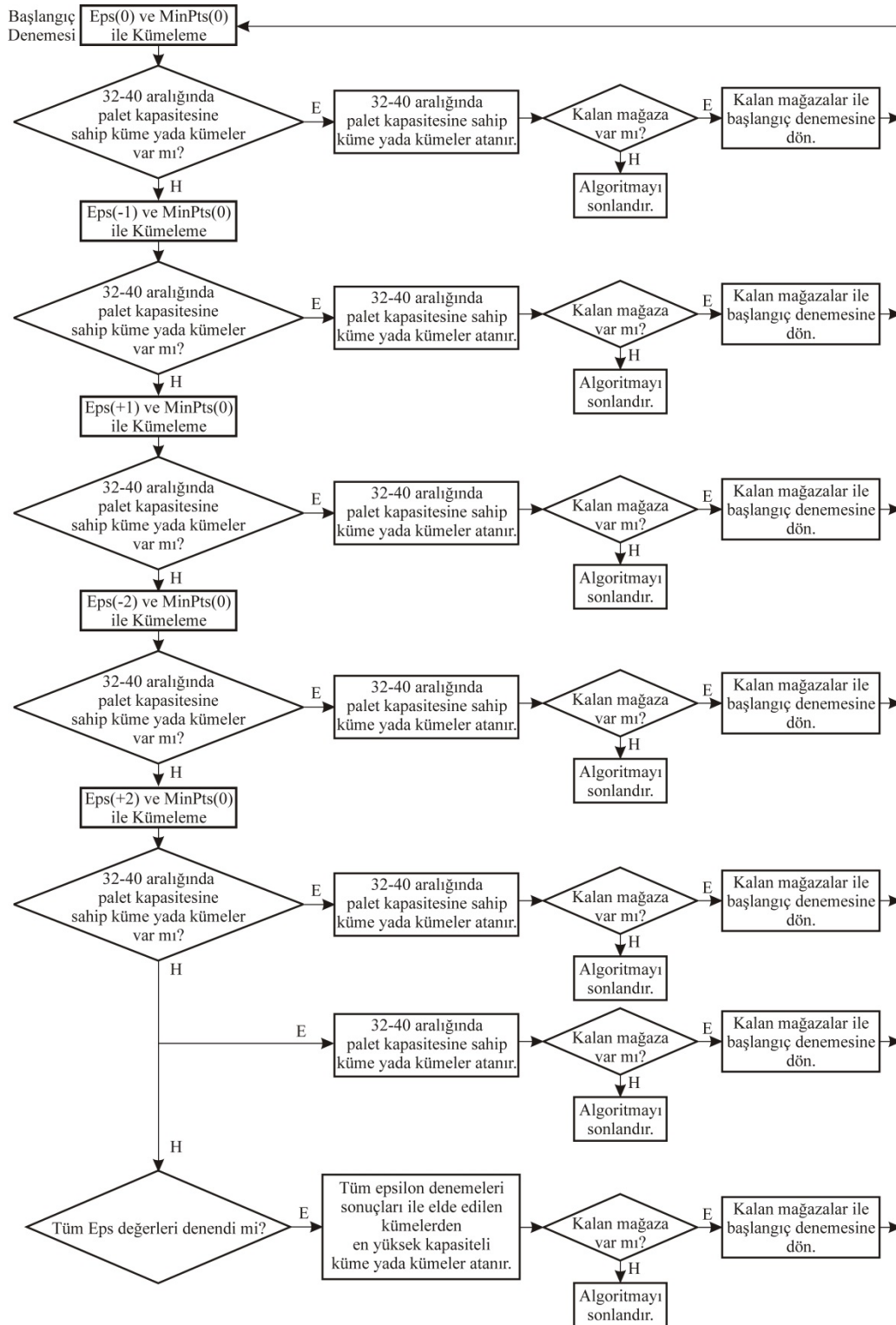
MinPoints değeri belirlendikten sonra Epsilon değerini belirlemek için DBSCAN algoritması denemeleri yapılır.

Adım 1. İlk aşamada belirlenen MinPoints baz değeri sabit tutularak Epsilon değeri 1 den başlanarak tek küme oluşturacağı Epsilon değerine kadar tekrarlanır.

Adım 2. Her deneme sonucunda meydana gelen kümelerden atanabilecek kümelerin mağaza sayısı aralığında mağaza sayısına sahip kümelerin sayıları tespit edilir.

Adım 3. Adım 2’de yapılan denemelerde, mağaza sayısı aralığında oluşan küme sayısı en yüksek olan denemedeki Eps değeri, Eps baz değeri olarak kabul edilir.

MinPts ve Eps’nin baz değerleri belirlendikten sonra, 78 mağaza ele alınarak DBSCAN algoritması çalıştırılır. Eğer baz değerler ile 32-40 aralığında palete sahip küme yada kümeler oluşmuş ise direk bu kümeler atanır. Eğer baz değerler ile 32-40 aralığında palete 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 palete 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 palete sahip) bir küme bulununcaya kadar bu denemeler tekrarlanır. Şekil 3’de 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 gelerek DBSCAN algoritmasının aşama işleyiş şemasıdır. Yani Eps değeri baz değerinden yani Eps(0) değerinden başlayarak Eps(-1), Eps(+1), Eps(-2), Eps(+2), Eps(-3), Eps(+3), Eps(-4), Eps(+4), ... değerleri atanabilir küme bulunana kadar sırasıyla denenir. İlk karşılaşılan 32-40 palet aralığındaki küme ya da kümeler atanır. Sonraki aşamada kalan mağazalara aynı şekilde kümeleme yapılarak yola devam edilir. DBSCAN algoritmasının adımları aşağıdaki Şekil 6.2.’de verilmiştir.



Şekil 6.2. DBSCAN aşama işleyiş şeması

Ele aldığımız dört alt problemde de belirli bir aşamaya geldiğimizde 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 mağazaların kümeleri belirlendikten sonra oluşturulan her bir kümedeki mağazalara ana depoda eklenerek dalsınır algoritması ile her bir kümenin rotaları (tur uzunlukları) belirlenir. Sonrasında her bir kümenin tur uzunlukları toplanarak toplam uzunluk bulunmuştur.

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.

6.2.2. Genetik Algoritma Destekli DBSCAN Algoritması

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 yada kümeler atanmaz ve tüm kalan mağazalar ana depo ile birlikte gezgin satıcı problemi gibi düşünülerek genetik algoritma yardımıyla rotalanır. Genetik algoritmanın aşamaları Tablo 6.4.'de belirtilmiştir.

Tablo 6.4. Genetik algoritmasının aşamaları

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 (20 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 kriterlerimiz iterasyon sayısına göre belirlenmiş ve iterasyon sayımız 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 koşturulmuş ve yapılan 20 farklı veri ile ortalama 17 dakikalık bir sürede her bir koşum gerçekleşmiştir. GA sonucunda bulunan rotada ana depodan sonra gelen mağazadan başlanarak kümelerin palet kapasiteleri kamyon kapasitesine (40 palete) yakınsayacak şekilde kümelemeler yapılır. Bunun sonucunda düşük kapasiteli kümelerin oluşmadığı buna istinaden kullanılacak araç sayısında ve araçların kat edeceği mesafede azalmalar sağlandığı gözlemlenmiştir.

Uygulama aşamasında veri setimizdeki mağazaların kuşbakışı uzaklıkları yerine gerçek uzaklıkları hesaba katılmış ve buna göre kümelemeler ve rotalamalar belirlenmiştir.

Bu çalışmada sezgisel bir algoritma olan GA ya başvurmamızın nedeni denemelerde kalan mağaza sayılarının en iyileyen algoritmalar için yüksek bir sayıda seyir etmesidir. GA desteği ile tüm mağazaların kümeleri belirlendikten sonra oluşturulan her bir kümedeki mağazalara ana depoda eklenerek dalsınır algoritması ile her bir kümenin rotaları (tur uzunlukları) belirlenir. Sonrasında her bir kümenin tur uzunlukları toplanarak toplam uzunluk bulunmuştur.

6.3. Alt Problemler ve Çözümleri

6.3.1. Kuru yoğun gün problemi ve çözümleri

Taze yoğun gün alt probleminde mağaza taleplerinin belirtildiği 20 farklı veri seti vardır. Her veri setinde toplam 78 adet mağazaya dağıtım yapılmakta olup, mağazaların talep ettikleri palet sayıları tam sayı olarak 6-14 aralığında değişmekte ve toplamları 724 palet olmaktadır. Bu veri setlerinin varyansları ise 5,5038 ile 6,2831 aralığında değişmektedir.

DBSCAN algoritması çalıştırılmadan önce DBSCAN algoritmasının parametrelerinin baz değerleri belirlenir.

MinPoints'in baz değerini belirleme: En az talebi olan mağaza 6 palet, en fazla talebi olan mağaza 14 palet ve kamyon kapasiteleri 40 palet olan bu alt problemde atanabilecek kümelerin en az 3 mağazalı (örnek olarak $14+14+12=40$), en çok 6 mağazalı (örnek olarak $6+6+6+6+6+6=36$) olabileceği görülmektedir. Bu durumda atanabilecek kümelerinin 3-6 aralığında mağaza sayısına sahip kümeler olduğu ortaya çıkmaktadır.

Veri setimize uygulanacak DBSCAN kümeleme algoritması ile oluşacak kümelerin mağaza sayılarının 3 ve 3'den büyük olmasını sağlamak için MinPoints baz değeri 3 olarak belirlenir.

Eps'nin baz değerini belirleme: MinPoints baz değeri 3 olarak belirledikten sonra Eps baz değerini belirlemek için MinPts baz değerinde sabit tutularak Eps değeri 1'den başlanarak her denemede Eps değerini bir arttırmak suretiyle denemeler yapılır. Bu denemeler Eps değerinin 78 mağazayı tek kümede toplayacağı denemeye kadar devam ettirilir. Her bir deneme sonucunda oluşan 3-6 aralığında mağaza sayısına sahip küme sayıları tespit edilir. Sonrasında denemeler arasından 3-6 aralığında mağaza sayısına sahip küme sayısı en fazla olan denemenin Eps değeri Eps baz değeri olarak kabul edilir.

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

MinPts ve Eps'nin baz değerleri belirlendikten sonra, ilk aşamada 78 mağaza ele alınarak DBSCAN algoritması çalıştırılır. Eğer algoritmanın parametrelerinin baz değerleri ile 32-40 aralığında paletle sahip küme yada kümeler oluşmuş ise bu kümeler direkt atanır ve kalan mağazalara DBSCAN algoritması ile tekrar kümeleme yapılması için ikinci aşamaya geçilir. Eğer bu baz değerlerle yapılan kümelemede 32-40 aralığında paletle sahip bir küme oluşmamış ise MinPts baz değerinde sabit tutularak Eps baz değerinin bir eksiği alınır ve deneme tekrarlanır. Bu deneme sonucunda da 32-40 aralığında paletle sahip küme yada kümeler oluşmamış ise MinPts yine baz değerinde sabit tutularak Eps baz değerinin bir fazlası alınır ve deneme yine tekrarlanır. Bu şekilde her deneme sonucunda atanabilir (32-40 aralığında paletle sahip) bir küme bulununcaya kadar bu denemeler tekrarlanır. Sonuç olarak her aşamanın ilk denemesinde MinPts baz değerinde sabit tutularak Eps baz değeri olan 18 değerinden başlanır ve bunu 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, ... Eps değerlerinin olduğu denemeler atanabilir küme bulunana kadar sırasıyla takip eder. Denemeler sırasında ilk karşılaşılan 32-40 palet aralığındaki küme yada kümeler atanır ve sonraki aşamada kalan mağazalar ile aynı şekilde kümeleme yapılarak yola devam edilir. 20 adet veri seti üzerinde yapılan kümelemeler sonucunda belli bir aşamaya geldiğimizde aşamadaki tüm denemeler yapılmasına rağmen atanabilir bir kümeyle ulaşılamadığı görülmüştür. Bu durumda tüm yapılan denemeler içerisinde oluşan en yüksek palet kapasitesine sahip küme yada kümeler atanarak yola devam edilmiş ve tüm mağazaların kümeleri belirlenene kadar bu aşamalar tekrarlanmıştır.

GA destekli DBSCAN algoritması DBSCAN algoritmasının belli bir aşamaya gelince tüm Eps denemelerine rağmen atanabilir (32-40 palet kapasiteye sahip) küme oluşturamadığı durumda devreye girer ve kümelenemeden geride kalan mağazalara ana depo da eklenerek gezgin satıcı problemi uygulanır. Bu problemde önceki bölümlerde belirtilen GA parametreleri ile GA uygulanır ve rota belirlenir. Belirlenen rotada ana depodan sonra gelen mağazadan başlanarak kümelerin palet kapasiteleri kamyon kapasitesine (40 paletle) yakınsayacak şekilde kümelemeler yapılır.

DBSCAN ve GA destekli DBSCAN algoritmalarının uygulanması sonucunda elde edilen sonuçlar Tablo 6.5.'de verilmiştir. Bu sonuçlar kullanılarak ANOVA testi yapılmış ve sonuçlar karşılaştırılmıştır.

Tablo 6.5. Kuru yoğun gün için GA destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması

Deneme	GA Destekli DBSCAN	Küme Sayısı	DBSCAN	Küme Sayısı
	Toplam Mesafe (km)		Toplam Mesafe (km)	
1	1732,0	20	1892,2	21
2	1781,8	20	1960,3	22
3	1860,5	21	1930,2	22
4	1862,2	21	1948,9	22
5	1807,9	20	1969,8	22
6	1762,7	20	1963,9	22
7	1865,9	21	1926,4	22
8	1785,2	20	1980,3	22
9	1788,4	20	1880,9	21
10	1850,3	21	1937,2	22
11	1707,8	20	1875,6	22
12	1845,3	21	1940,2	22
13	1742,7	20	1900,3	21
14	1752,9	20	1920,4	22
15	1882,6	21	1963,6	22
16	1855,4	21	1945,6	22
17	1804,3	20	1976,3	22
18	1796,2	20	1875,2	21
19	1887,2	21	1956,3	22
20	1754,1	20	1973,2	22

ANOVA testinde öncelikli olarak varyansların eşit olup olmadığı incelendiğinde Tablo 6.6.'daki sig. değeri 0,05 den küçük olduğu için grup varyanslarının eşit olmadığı sonucuna varılmaktadır. Bundan dolayı Tablo 6.7.'deki F testinin sonuçları anlamlı olmayacaktır. Bu nedenle Tablo 6.9.'daki Welch ve Brown-Forsythe testinin sonuçları anlamlı olacaktır. Bu sonuçlara baktığımızda sig. değeri 0,05'den küçük olduğu için iki metot ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir farkın olduğu sonucuna varabiliriz. Tablo 6.8.'deki ortalama değerlere baktığımızda genetik algoritma destekli kümeleme algoritması ile ortalama yol uzunluğu 1806.270 km olur iken DBSCAN kümeleme algoritması ile elde edilen ortalama 1935.840 km olmuştur. Bu sonuçlar ışığında GA destekli DBSCAN algoritmasının DBSCAN algoritmasına göre daha iyi sonuç sağladığı tespit edilmiştir. Ayrıca GA destekli DBSCAN algoritmasında düşük kapasiteli kümeler oluşmadığı ve kullanılacak araç sayısında azalmalar olduğu gözlemlenmiştir.

Tablo 6.6. Kuru yoğun gün için varyansların homojenliği testi

Levene Statistic	Df1	Df2	Sig.
5,796	1	38	,021

Tablo 6.7. Kuru yoğun gün için ANOVA testi sonucu

	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	167883,849	1	167883,849	80,811	,000
Within Groups	78944,050	38	2077,475		
Total	246827,899	39			

Tablo 6.8. Kuru yoğun gün için sonuçlar

	N	Mean	Std. Deviation	Std. Error
1	20	1806,270	54,3228	12,1470
2	20	1935,840	34,6984	7,7588
Total	40	1871,055	79,5545	12,5787
Model	Fixed Effects		45,5793	7,2067
	Random Effects			64,7850

Tablo 6.9. Kuru yoğun gün için welch ve brown-forsythe testi

	Statistica	df1	df2	Sig.
Welch	80,811	1	32,291	,000
Brown-Forsythe	80,811	1	32,291	,000

6.3.2. Taze yoğun gün problemi ve çözümleri

Taze yoğun gün alt probleminde mağaza taleplerinin belirtildiği 20 farklı veri seti vardır. Her veri setinde toplam 78 adet mağazaya dağıtım yapılmakta olup, mağazaların talep ettikleri palet sayıları tam sayı olarak 3-9 aralığında değişmekte ve toplamları 438 palet olmaktadır. Bu veri setlerinin varyansları ise 4,032 ile 4,993 aralığında değişmektedir.

DBSCAN algoritması çalıştırılmadan önce DBSCAN algoritmasının parametrelerinin baz değerleri belirlenir.

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 bu alt 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+3=39) olabileceği

görülmektedir. Bu durumda atanabilecek kümelerinin 4-13 aralığında mağaza sayısına sahip kümeler olduğu ortaya çıkmaktadır.

Veri setimize uygulanacak DBSCAN kümeleme algoritması ile oluşacak kümelerin mağaza sayılarının 4 ve 4'den büyük olmasını sağlamak için MinPoints baz değeri 4 olarak belirlenir.

Eps'nin baz değerini belirleme: MinPoints baz değeri 4 olarak belirledikten sonra Eps baz değerini belirlemek için MinPts baz değerinde sabit tutularak Eps değeri 1'den başlanarak her denemede Eps değerini bir arttırmak suretiyle denemeler yapılır. Bu denemeler Eps değerinin 78 mağazayı tek kümede toplayacağı denemeye kadar devam ettirilir. Her bir deneme sonucunda oluşan 4-13 aralığında mağaza sayısına sahip küme sayıları tespit edilir. Sonrasında denemeler arasından 4-13 aralığında mağaza sayısına sahip küme sayısı en fazla olan denemenin Eps değeri Eps baz değeri olarak kabul edilir.

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

MinPts ve Eps'nin baz değerleri belirlendikten sonra, ilk aşamada 78 mağaza ele alınarak DBSCAN algoritması çalıştırılır. Eğer algoritmanın parametrelerinin baz değerleri ile 32-40 aralığında palet sahibi küme yada kümeler oluşmuş ise bu kümeler direkt atanır ve kalan mağazalara DBSCAN algoritması ile tekrar kümeleme yapılması için ikinci aşamaya geçilir. Eğer bu baz değerlerle yapılan kümelemede 32-40 aralığında palet sahibi bir küme oluşmamış ise MinPts baz değerinde sabit tutularak Eps baz değerinin bir eksiği alınır ve deneme tekrarlanır. Bu deneme sonucunda da 32-40 aralığında palet sahibi küme yada kümeler oluşmamış ise MinPts yine baz değerinde sabit tutularak Eps baz değerinin bir fazlası alınır ve deneme yine tekrarlanır. Bu şekilde her deneme sonucunda atanabilir (32-40 aralığında palet sahibi) bir küme bulununcaya kadar bu denemeler tekrarlanır. Sonuç olarak her aşamanın ilk denemesinde MinPts baz değerinde sabit tutularak Eps baz değeri olan 18 değerinden başlanır ve bunu 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, ... Eps değerlerinin olduğu

denemeler atanabilir küme bulunana kadar sırasıyla takip eder. Denemeler sırasında ilk karşılaşılan 32-40 palet aralığındaki küme yada kümeler atanır ve sonraki aşamada kalan mağazalar ile aynı şekilde kümeleme yapılarak yola devam edilir. 20 adet veri seti üzerinde yapılan kümelemeler sonucunda belli bir aşamaya geldiğimizde aşamadaki 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 edilmiş ve tüm mağazaların kümeleri belirlenene kadar bu aşamalar tekrarlanmıştır.

GA destekli DBSCAN algoritması DBSCAN algoritmasının belli bir aşamaya gelince tüm Eps denemelerine rağmen atanabilir (32-40 palet kapasiteye sahip) küme oluşturamadığı durumda devreye girer ve kümelenmeden geride kalan mağazalara ana depo da eklenerek gezgin satıcı problemi uygulanır. Bu problemde önceki bölümlerde belirtilen GA parametreleri ile GA uygulanır ve rota belirlenir. Belirlenen rotada ana depodan sonra gelen mağazadan başlanarak kümelerin palet kapasiteleri kamyon kapasitesine (40 palete) yakınsayacak şekilde kümelemeler yapılır.

DBSCAN ve GA destekli DBSCAN algoritmalarının uygulanması sonucunda elde edilen Tablo 6.10.'da verilmiştir. Bu sonuçlar kullanılarak ANOVA testi yapılmış ve sonuçlar karşılaştırılmıştır.

Tablo 6.10. Taze yoğun gün için GA destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması

Deneme	GA Destekli DBSCAN	Küme Sayısı	DBSCAN	Küme Sayısı
	Toplam Mesafe (km)		Toplam 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 Tablo 6.11.'de sig. değeri 0,05 büyük olduğu için grup varyanslarının eşit olduğu sonucuna varılmaktadır. Bundan dolayı da Tablo 6.12.'deki F testinin sonuçları anlamlı olabilecektir. ANOVA tablosundaki sig. değeri 0,05'den küçük olduğu için iki metot ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir farkın olduğu sonucuna varabiliriz. Tablo 6.13.'deki ortalama değerlere baktığımızda genetik algoritma destekli kümeleme algoritması ile ortalama yol uzunluğu 1188.835 km olur iken DBSCAN kümeleme algoritması ile elde edilen ortalama 1323.355 km olmuştur. Bu sonuçlar ışığında GA destekli DBSCAN algoritmasının DBSCAN algoritmasına göre daha iyi sonuç sağladığı tespit edilmiştir. Ayrıca GA destekli DBSCAN algoritmasında düşük kapasiteli kümeler oluşmadığı ve kullanılacak araç sayısında azalmalar olduğu gözlemlenmiştir.

Tablo 6.11. Taze yoğun gün için varyansların homojenliği testi

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

Tablo 6.12. Taze yoğun gün için ANOVA testi sonucu

	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	180956,304	1	180956,304	111,622	,000
Within Groups	61694,055	38	1621,159		
Total	242560,359	39			

Tablo 6.13. Taze yoğun gün için sonuçlar

	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	Fixed Effects		40,2636	6,3662
	Random Effects			67,2600

6.3.3. Kuru sakin gün problemi ve çözümleri

Taze yoğun gün alt probleminde mağaza taleplerinin belirtildiği 20 farklı veri seti vardır. Her veri setinde toplam 66 adet mağazaya dağıtım yapılmakta olup, mağazaların talep ettikleri palet sayıları tam sayı olarak 8-14 aralığında değişmekte ve toplamları 652 palet olmaktadır. Bu veri setlerinin varyansları ise 15,636 ile 17,843 aralığında değişmektedir.

DBSCAN algoritması çalıştırılmadan önce DBSCAN algoritmasının parametrelerinin baz değerleri belirlenir.

MinPoints'in baz değerini belirleme: En az talebi olan mağaza 8 palet, en fazla talebi olan mağaza 14 palet ve kamyon kapasiteleri 40 palet olan bu alt problemde atanabilecek kümelerin en az 3 mağazalı (örnek olarak 14+14+12=40), en çok 5 mağazalı (örnek olarak 8+8+8+8+8=40) olabileceği görülmektedir. Bu durumda atanabilecek kümelerinin 3-5 aralığında mağaza sayısına sahip kümeler olduğu ortaya çıkmaktadır.

Veri setimize uygulanacak DBSCAN kümeleme algoritması ile oluşacak kümelerin mağaza sayılarının 3 ve 3'den büyük olmasını sağlamak için MinPoints baz değeri 3 olarak belirlenir.

Eps'nin baz değerini belirleme: MinPoints baz değeri 3 olarak belirledikten sonra Eps baz değerini belirlemek için MinPts baz değerinde sabit tutularak Eps değeri 1'den başlanarak her denemede Eps değerini bir arttırmak suretiyle denemeler yapılır. Bu denemeler Eps değerinin 66 mağazayı tek kümede toplayacağı denemeye kadar devam ettirilir. Her bir deneme sonucunda oluşan 3-5 aralığında mağaza sayısına sahip küme sayıları tespit edilir. Sonrasında denemeler arasından 3-5 aralığında mağaza sayısına sahip küme sayısı en fazla olan denemenin Eps değeri Eps baz değeri olarak kabul edilir.

Veri setimizle yapılan denemeler sonucunda Eps'nin baz değeri 17 olarak tespit edilmiştir.

MinPts ve Eps'nin baz değerleri belirlendikten sonra, ilk aşamada 66 mağaza ele alınarak DBSCAN algoritması çalıştırılır. Eğer algoritmanın parametrelerinin baz değerleri ile 32-40 aralığında palet sahip küme yada kümeler oluşmuş ise bu kümeler direkt atanır ve kalan mağazalara DBSCAN algoritması ile tekrar kümeleme yapılması için ikinci aşamaya geçilir. Eğer bu baz değerlerle yapılan kümelemede 32-40 aralığında palet sahip bir küme oluşmamış ise MinPts baz değerinde sabit tutularak Eps baz değerinin bir eksiği alınır ve deneme tekrarlanır. Bu deneme sonucunda da 32-40 aralığında palet sahip küme yada kümeler oluşmamış ise MinPts yine baz değerinde sabit tutularak Eps baz değerinin bir fazlası alınır ve deneme yine tekrarlanır. Bu şekilde her deneme sonucunda atanabilir (32-40 aralığında palet sahip) bir küme bulununcaya kadar bu denemeler tekrarlanır. Sonuç olarak her aşamanın ilk denemesinde MinPts baz değerinde sabit tutularak Eps baz değeri olan 17 değerinden başlanır ve bunu 16, 18, 15, 19, 14, 20, 13, 21, 12, 22, 11, 23, 10, 24, 9, 25, 8, 26, 7, 27, 6, 28, 5, 29, 4, 30, 3, 31, 2, 32, 1, 33, 1, 34, 35, 36, ... Eps değerlerinin olduğu denemeler atanabilir küme bulunana kadar sırasıyla takip eder. Denemeler sırasında ilk karşılaşılan 32-40 palet aralığındaki küme yada kümeler atanır ve sonraki aşamada kalan mağazalar ile aynı şekilde kümeleme yapılarak yola devam edilir. 20 adet veri seti üzerinde yapılan kümelemeler sonucunda belli bir aşamaya geldiğimizde aşamadaki tüm denemeler yapılmasına rağmen atanabilir bir kümeyle ulaşılamadığı görülmüştür. Bu durumda tüm yapılan denemeler içerisinde oluşan en yüksek palet

kapasitesine sahip küme yada kümeler atanarak yola devam edilmiş ve tüm mağazaların kümeleri belirlenene kadar bu aşamalar tekrarlanmıştır.

GA destekli DBSCAN algoritması DBSCAN algoritmasının belli bir aşamaya gelince tüm Eps denemelerine rağmen atanabilir (32-40 palet kapasiteye sahip) küme oluşturamadığı durumda devreye girer ve kümelenmeden geride kalan mağazalara ana depo da eklenerek gezgin satıcı problemi uygulanır. Bu problemde önceki bölümlerde belirtilen GA parametreleri ile GA uygulanır ve rota belirlenir. Belirlenen rotada ana depodan sonra gelen mağazadan başlanarak kümelerin palet kapasiteleri kamyon kapasitesine (40 palete) yakınsayacak şekilde kümelemeler yapılır.

DBSCAN ve GA destekli DBSCAN algoritmalarının uygulanması sonucunda elde edilen Tablo 6.14.'de verilmiştir. Bu sonuçlar kullanılarak ANOVA testi yapılmış ve sonuçlar karşılaştırılmıştır.

Tablo 6.14. Kuru sakın gün için GA Destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması

Deneme	GA Destekli DBSCAN	Küme Sayısı	DBSCAN	Küme Sayısı
	Toplam Mesafe (km)		Toplam Mesafe (km)	
1	1595,2	18	1678,5	20
2	1679,3	19	1772,8	20
3	1623,9	18	1687,8	19
4	1609,7	18	1676,1	19
5	1603,2	18	1756,7	20
6	1582,5	18	1685,0	19
7	1610,2	18	1699,5	20
8	1675,6	19	1730,4	20
9	1627,3	18	1710,0	20
10	1575,4	18	1680,7	19
11	1593,9	18	1656,2	19
12	1604,7	18	1752,7	20
13	1655,2	19	1725,8	20
14	1599,0	18	1701,3	19
15	1680,3	19	1797,3	20
16	1635,0	18	1715,8	19
17	1610,4	18	1725,6	20
18	1602,7	18	1695,2	19
19	1695,2	19	1785,2	20
20	1625,2	18	1780,3	20

ANOVA testinde öncelikli olarak varyansların eşit olup olmadığı incelendiğinde Tablo 6.15.'de sig. değeri 0,05 büyük olduğu için grup varyanslarının eşit olduğu sonucuna varılmaktadır. Bundan dolayı da Tablo 6.16.'daki F testinin sonuçları

anlamli olabilecektir. ANOVA tablosundaki sig. deęeri 0,05'den k¼¼k olduęu iin iki metot ile elde edilen sonular arasında istatistiksel olarak anlamli bir farkın olduęu sonucuna varabiliriz. Tablo 6.17.'deki ortalama deęerlere baktıęımızda genetik algoritma destekli k¼meleme algoritması ile ortalama yol uzunluęu 1624.195 km olur iken DBSCAN k¼meleme algoritması ile elde edilen ortalama 1720.645 km olmuştur. Bu sonular ıřıęında GA destekli DBSCAN algoritmasının DBSCAN algoritmasına g¼re daha iyi sonu saęladıęı tespit edilmiştir. Ayrıca GA destekli DBSCAN algoritmasında d¼ř¼k kapasiteli k¼meler oluřmadıęı ve kullanılacak ara sayısında azalmalar olduęu g¼zlemlenmiştir.

Tablo 6.15. Kuru sakın g¼n iin varyansların homojenlięi testi

Levene Statistic	Df1	Df2	Sig.
,858	1	38	,360

Tablo 6.16. Kuru sakın g¼n iin ANOVA testi sonucu

T	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	93026,025	1	93026,025	63,568	,000
Within Groups	55609,299	38	1463,403		
Total	148635,324	39			

Tablo 6.17. Kuru sakın g¼n iin sonular

	N	Mean	Std. Deviation	Std. Error
1	20	1624,195	35,0557	7,8387
2	20	1720,645	41,2056	9,2139
Total	40	1672,420	61,7346	9,7611
Model	Fixed Effects		38,2544	6,0486
	Random Effects			48,2250

6.3.4. Taze sakın g¼n problemi ve ¼z¼mleri

Taze yoęun g¼n alt probleminde maęaza taleplerinin belirtildięi 20 farklı veri seti vardır. Her veri setinde toplam 66 adet maęazaya daęıtım yapılmakta olup, maęazaların talep ettikleri palet sayıları tam sayı olarak 5-9 aralıęında deęiřmekte ve toplamları 402 palet olmaktadır. Bu veri setlerinin varyansları ise 6,2358 ile 6,8591 aralıęında deęiřmektedir.

DBSCAN algoritması çalıştırılmadan önce DBSCAN algoritmasının parametrelerinin baz değerleri belirlenir.

MinPoints'in baz değerini belirleme: En az talebi olan mağaza 5 palet, en fazla talebi olan mağaza 9 palet ve kamyon kapasiteleri 40 palet olan bu alt problemde atanabilecek kümelerin en az 4 mağazalı (örnek olarak $9+9+9+9=36$), en çok 8 mağazalı (örnek olarak $5+5+5+5+5+5+5+5=40$) olabileceği görülmektedir. Bu durumda atanabilecek kümelerinin 4-8 aralığında mağaza sayısına sahip kümeler olduğu ortaya çıkmaktadır.

Veri setimize uygulanacak DBSCAN kümeleme algoritması ile oluşacak kümelerin mağaza sayılarının 4 ve 4'den büyük olmasını sağlamak için MinPoints baz değeri 4 olarak belirlenir.

Eps'nin baz değerini belirleme: MinPoints baz değeri 4 olarak belirlendikten sonra Eps baz değerini belirlemek için MinPts baz değerinde sabit tutularak Eps değeri 1'den başlanarak her denemede Eps değerini bir arttırmak suretiyle denemeler yapılır. Bu denemeler Eps değerinin 66 mağazayı tek kümede toplayacağı denemeye kadar devam ettirilir. Her bir deneme sonucunda oluşan 4-8 aralığında mağaza sayısına sahip küme sayıları tespit edilir. Sonrasında denemeler arasından 4-8 aralığında mağaza sayısına sahip küme sayısı en fazla olan denemenin Eps değeri Eps baz değeri olarak kabul edilir.

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

MinPts ve Eps'nin baz değerleri belirlendikten sonra, ilk aşamada 66 mağaza ele alınarak DBSCAN algoritması çalıştırılır. Eğer algoritmanın parametrelerinin baz değerleri ile 32-40 aralığında palet sahibi küme yada kümeler oluşmuş ise bu kümeler direkt atanır ve kalan mağazalara DBSCAN algoritması ile tekrar kümeleme yapılması için ikinci aşamaya geçilir. Eğer bu baz değerlerle yapılan kümelemede 32-40 aralığında palet sahibi bir küme oluşmamış ise MinPts baz değerinde sabit tutularak Eps baz değerinin bir eksiği alınır ve deneme tekrarlanır. Bu deneme sonucunda da

32-40 aralığında palete sahip küme yada kümeler oluşmamış ise MinPts yine baz değerinde sabit tutularak Eps baz değerinin bir fazlası alınır ve deneme yine tekrarlanır. Bu şekilde her deneme sonucunda atanabilir (32-40 aralığında palete sahip) bir küme bulununcaya kadar bu denemeler tekrarlanır. Sonuç olarak her aşamanın ilk denemesinde MinPts baz değerinde sabit tutularak Eps baz değeri olan 18 değerinden başlanır ve bunu 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, ... Eps değerlerinin olduğu denemeler atanabilir küme bulunana kadar sırasıyla takip eder. Denemeler sırasında ilk karşılaşılan 32-40 palet aralığındaki küme yada kümeler atanır ve sonraki aşamada kalan mağazalar ile aynı şekilde kümeleme yapılarak yola devam edilir. 20 adet veri seti üzerinde yapılan kümelemeler sonucunda belli bir aşamaya geldiğimizde aşamadaki 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 edilmiş ve tüm mağazaların kümeleri belirlenene kadar bu aşamalar tekrarlanmıştır.

GA destekli DBSCAN algoritması DBSCAN algoritmasının belli bir aşamaya gelince tüm Eps denemelerine rağmen atanabilir (32-40 palet kapasiteye sahip) küme oluşturamadığı durumda devreye girer ve kümelenmeden geride kalan mağazalara ana depo da eklenerek gezgin satıcı problemi uygulanır. Bu problemde önceki bölümlerde belirtilen GA parametreleri ile GA uygulanır ve rota belirlenir. Belirlenen rotada ana depodan sonra gelen mağazadan başlanarak kümelerin palet kapasiteleri kamyon kapasitesine (40 palete) yakınsayacak şekilde kümelemeler yapılır.

DBSCAN ve GA destekli DBSCAN algoritmalarının uygulanması sonucunda elde edilen Tablo 6.18'de verilmiştir. Bu sonuçlar kullanılarak ANOVA testi yapılmış ve sonuçlar karşılaştırılmıştır.

Tablo 6.18. Taze sakin gün için GA Destekli DBSCAN ve DBSCAN algoritmalarının karşılaştırması

Deneme	GA Destekli DBSCAN	Küme Sayısı	DBSCAN	Küme Sayısı
	Toplam Mesafe (km)		Toplam Mesafe (km)	
1	1023,1	11	1186,2	13
2	1109,0	11	1298,6	13
3	1040,1	11	1195,0	13
4	1036,5	11	1315,9	14
5	1081,0	12	1298,2	14
6	1023,5	11	1245,0	13
7	1091,4	12	1305,7	14
8	1075,7	11	1254,3	13
9	1055,3	11	1317,3	13
10	1055,8	11	1345,5	14
11	1067,4	11	1275,4	14
12	1035,8	11	1155,4	13
13	1110,6	12	1285,3	14
14	1067,9	11	1197,2	13
15	1056,3	12	1300,4	14
16	1075,8	11	1315,3	14
17	1019,0	11	1234,0	13
18	1034,2	11	1247,6	13
19	1125,4	11	1226,4	13
20	1045,0	11	1267,7	13

ANOVA testinde öncelikli olarak varyansların eşit olup olmadığı incelendiğinde Tablo 6.19.'daki sig. değeri 0,05 den küçük olduğu için grup varyanslarının eşit olmadığı sonucuna varılmaktadır. Bundan dolayı Tablo 6.20.'deki F testinin sonuçları anlamlı olmayacaktır. Bu nedenle Tablo 6.22.'deki Welch ve Brown-Forsythe testinin sonuçları anlamlı olacaktır. Bu sonuçlara baktığımızda sig. değeri 0,05'den küçük olduğu için iki metot ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir farkın olduğu sonucuna varabiliriz. Tablo 6.21.'deki ortalama değerlere baktığımızda genetik algoritma destekli kümeleme algoritması ile ortalama yol uzunluğu 1061.440km olur iken DBSCAN kümeleme algoritması ile elde edilen ortalama 1263.320 km olmuştur. Bu sonuçlar ışığında GA destekli DBSCAN algoritmasının DBSCAN algoritmasına göre daha iyi sonuç sağladığı tespit edilmiştir. Ayrıca GA destekli DBSCAN algoritmasında düşük kapasiteli kümeler oluşmadığı ve kullanılacak araç sayısında azalmalar olduğu gözlemlenmiştir.

Tablo 6.19. Taze sakin gün için varyansların homojenliği testi

Levene Statistic	Df1	Df2	Sig.
6,102	1	38	,018

Tablo 6.20. Taze sakin gün için ANOVA testi sonucu

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	407555,344	1	407555,344	224,369	,000
Within Groups	69025,160	38	1816,452		
Total	476580,504	39			

Tablo 6.21. Taze sakin gün için sonuçlar

	N	Mean	Std. Deviation	Std. Error
1	20	1061,440	30,8563	6,8997
2	20	1263,320	51,7764	11,5775
Total	40	1162,380	110,5442	17,4786
Model	Fixed Effects		42,6198	6,7388
	Random Effects			100,9400

Tablo 6.22. Taze sakin gün için welch ve brown-forsythe testi

	Statistica	df1	df2	Sig.
Welch	80,811	1	32,291	,000
Brown-Forsythe	80,811	1	32,291	,000

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Günümüzde şirketler arasındaki rekabet, çok detaylı stratejik kararlardan etkilenebilecek bir durumdur. Bunun nedeni bilgi ve teknolojinin çok daha hızlı yayılabilmesi ve şirketlerin kendi bünyelerindeki işlemlere hızlı bir şekilde entegre edebilmesidir. Özellikle dağıtım sektörünü ilgilendiren ve lojistiğin önemli bir bileşeni olan araç rotalama, sözkonusu rekabetin öne çıktığı bir alandır. Şirketlerin, müşterilere hızlı ve kaliteli hizmet vererek müşteri memnuniyetini arttırmanın yanısıra, şirketin kendi iç verimliliğini de sağlaması ve her iki açıdan da en iyi olacak kararı vermesi gerekmektedir. Gerçek hayatta sıklıkla karşılaşılan ve dağıtım yapılacak müşterilerin belirli taleplere sahip olduğu Kapasite Kısıtlı ARP problemi literatürde sıklıkla incelenen ve çözümü için model geliştirilen bir problemdir. Posta teslimatı, endüstriyel atık toplama, okul servis aracı rotalama ve çizelgeleme gibi belirli durumları yansıtan Kapasite Kısıtlı Araç Rotalama ve Kümeleme Probleminin Kümeleme kısmı için literatürde çeşitli temellere dayanan kümeleme yöntemlerinin kullanıldığı görülmüştür fakat bu kümeleme yöntemlerinin çoğunluğunun tek başlarına kullanıldığı tespit edilmiştir. Bu çalışmada bu kümeleme yöntemlerinin kendisinden farklı mantığa sahip algoritmalarla birlikte kullanılması halinde ne gibi sonuçlar oluşturacağı merak edilmiş ve yoğunluğa dayalı bir kümeleme yöntemi olan DBSCAN'ı tek başına ve GA ile destekleyerek ayrı ayrı kümeleme yaptırılmıştır. Yapılan koşumlar sonucunda elde edilen kümelerdeki mağazalara ana depo da eklenerek en iyileyen bir algoritma olan dalsınır algoritmasıyla mağazalara dağıtımın rotası belirlenmiştir. Elde edilen sonuçlar toplam mesafe açısından karşılaştırılmış ve GA destekli DBSCAN'ın DBSCAN'a göre ortalama; kuru yoğun gün için %6,68, taze yoğun gün için %10,06, kuru sakin gün için %5,59, taze sakin gün için %15,88 iyileştirme sağladığı tespit edilmiştir. Bu iyileşen sonuçlara ulaşmamızı sağlayan faktör GA'nın DBSCAN ile uyumlu bir algoritma olmasının yanısıra GA'nın farklı bir mantıkla çalışıp işleyişte sağladığı çeşitlilikten ileri gelmektedir. DBSCAN algoritmasının belli bir aşamadan sonra düşük kapasiteli yüklemelerinin önüne geçilmiş ve toplam mesafenin yanısıra kullanılan

kamyon sayısında ve şoför sayısında azalmalar sağlamıştır. Ayrıca bu sonuçlar ışığında mağazaların taleplerinin düşük olduğu durumlarda (taze ürün dağıtımı) yani oluşan kümelerdeki mağaza sayısının daha fazla olacağı durumlarda önerdiğimiz çözüm yönteminin daha iyi oranlarda iyileştirme sağladığını söyleyebiliriz. Buna benzer orta ve büyük ölçekli perakende zincirlerinin dağıtım problemlerinde bu yöntem ile maliyetin daha düşük olduğu çözümler bulunabilir. Literatürde bu şekilde kümeleme yönteminin kullanıldığı bir çalışmaya rastlanmaması bu çalışmanın motivasyonlarından biridir.

DBSCAN kümeleme algoritmasının başlangıç değerlerini oluşturan Eps ve MinPts değerleri için kümeleme sırasında Eps sabit tutularak onun yerine MinPts’de değişiklik yapılabilir. Bu şekilde farklı bir kombinasyon kullanılarak DBSCAN kümeleme algoritmasının performansı ayrıca hesaplanabilir. Bu değişikliğe benzer şekilde algoritmanın işleyişinde uygun değişiklikler ile rekabet gücü daha yüksek bir model ortaya konulabilir.

KAYNAKLAR

- [1] LAPORTE, G., NOBERT, Y. VE TAILLEFER, S. , ‘Solving a family of multi depot vehicle routing and location-routing problems’, *Transportation Science*, Cilt 22, Sayı 3, Ağustos, s. 161-172, 1987.
- [2] CRAINIC, T.G. VE LAPORTE, G. , ‘Planning Models For Freight Transportation’, *European Journal Of Operational Research*, Cilt 97, Sayı 3, Mart, s. 409-438, 1997.
- [3] SAVELSBERGH, M.W.P. VE SOL, M. , ‘The General Pick-Up And Delivery Problem’, *Transportation Science*, Cilt 29, Sayı 1, Şubat, s. 17-29, 1995.
- [4] BOWERMAN, R., HALL, B. VE CALMAI, P. , ‘A Multiobjective Optimization Approach To Urban School Bus Routing: Formulation And Solution Method’, *Transportation Research*, Cilt 29, Sayı 2, s. 107-123, 1995.
- [5] TOTH, P. VE VIGO, D. , *The Vehicle Routing Problem*, Philadelphia: Society for Industrial and Applied Mathematics, 2002.
- [6] GENDREAU, M., LAPORTE, G. VE SÉGUIN, R. , ‘Stochastic vehicle routing’, *European Journal of Operational Research*, Cilt 88, Sayı 1, s. 3- 12, 1996.
- [7] LIN, S-W., LEE, Z-J., YING, K-C. VE LEE, C-Y. , ‘Applying hybrid meta-heuristic for capacitated vehicle routing problem’, *Expert Systems with Applications*, Cilt 36, Sayı 2, Bölüm 1, Mart, s.1505 1512, 2009.
- [8] CHRISTOFIDES N., MINGOZZI A. VE TOTH P. , ‘Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations’, *Mathematical Programming*, North-Holland Publishing Company, Cilt 20, Sayı 1, s. 255-282, 1981.
- [9] JACOBS-BLECHA, C. VE GOETSCHALCKX, M. , ‘The Vehicle Routing Problem with Backhauls: Properties and Solution Algorithms’, *Technical Report MHRC-TR-88-13*, Georgia Institute of Technology, Atlanta, Georgia, 1992.
- [10] BIANCHESSI, N. VE RIGHINI, G. , ‘Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery’, *Computers & Operations Research*, Cilt 34, Sayı 2, Şubat, s. 578-594, 2007.

- [11] JIN, M., LIU, K. VE EKSIOGLU, B. , ‘A column generation approach for the split delivery vehicle routing problem’, *Operations Research Letters*, Cilt 36, Sayı 2, Mart, s. 265-270, 2008.
- [12] HO, W., HO, G. T. S., JI, P. VE LAU, H. C. W. , ‘A hybrid genetic algorithm for the multi-depot vehicle routing problem’, *Engineering Applications of Artificial Intelligence*, Cilt 21, Sayı 4, Haziran, s. 548-557, 2008.
- [13] HEMMELMAYR, V. C., DOERNER, K. F. VE HARTL R. F. , ‘A variable neighborhood search heuristic for periodic routing problems’, *European Journal of Operational Research*, Cilt 195, Sayı 3, Haziran, s. 791-802, 2007.
- [14] BADEAU, P., GUERTIN, F., GENDREAU, M., POTVIN J. AND TAILLARD, E.,: A Parallel Tabu Search Heuristic for The Vehicle Routing Problem With Time Windows, *Transportation Research Part C: Emerging Technologies*, Cilt 5, Sayı 2, Nisan, s. 109-122, 1997.
- [15] FISHER, M. L., JAIKUMAR, R. ,, A Generalized Assignment Heuristic For Vehicle Routing, *Networks*, Cilt 11, Sayı 2, Ağustos, s. 109-124, 1981.
- [16] LAWLER, E.L., LENSTRA, J.K., RINOOY KAN, A.H.G., AND SHMOYS, D.B., , *The Traveling Salesman Problem*, New York: John Wiley and Sons, 1985.
- [17] CLARKE, G. AND WRIGHT, J. , *Scheduling Of Vehicles From A Central Depot To A Number Of Delivery Points*, *Operation Research*, Cilt 12, Sayı 4, Temmuz, s. 568-581, 1964.
- [18] CHRISTOFIDES, N. ,, *Vehicle Routing In The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, E.L. Lawlar, J. K. Lenstra, A. H. G. Rinnooy Kan ve D. B. Shmoys, , Wiley, Chichester, s. 431-448, 1985.
- [19] GLOVER, F., VE LAGUNA, M. , *Tabu Search, Modern Heuristic Techniques For Combinatorial Problems*, Oxford: Blackwell Scientific Publications, 1993.
- [20] HOLLAND, J.H. , *Adaption In Natural And Artificial Systems*, Ann Arbor: University Of Michigan Press, 1975.
- [21] KIRKPATRICK, S., GELATT, C.D. AND VECCHI, M.P. , ‘Optimization By Simulated Annealing’, *Science*, New Series, Cilt 220, Sayı 4583, Mayıs, s. 671-680, 1983.
- [22] TATLIDİL, H. , *Uygulamalı Çok Değişkenli İstatistiksel Analiz*, Ankara: Ziraat Matbaası, 2002.
- [23] CAMBELL, M. , *Multivariate Statistics: Cluster Analysis*, [Online], <http://trochim.human.cornell.edu/tutorial/flynn/cluster.html> Erişim Tarihi: 12.11.2011.

- [24] LEAN, P. RICARDO VE POWERS, L. THOMAS , ‘A taxonomy of countries based on inventive activity’, *International Marketing Review*, Cilt 14, Sayı 6, s. 445-460, 1997.
- [25] MALHOTRA, K. NARESH , *Marketing Research: An Applied Orientation*, Second Edition, New Jersey: Prentice Hall Inc., s. 670-672, 1996.
- [26] BRYAN F.J. MANLY , *Multivariate Statistical Methods*, Second Edition, Londra: Chapman & Hall, s. 280, 1994.
- [27] JOHANNES GRABMEIER VE ANDREAS RUDOLPH , ‘Techniques of Cluster Algorithms in Data Mining’, *Data Mining and Knowledge Discovery*, Cilt 6, Sayı 4, Kasım, s. 303-360, 2002.
- [28] KHATTREE RAVINDA VE NAIK N. DAYANAND , *Multivariate Data Reduction and Discrimination*, First Edition, North Caroline: Cary, s. 366, 2002.
- [29] NCSS , *User Guide*, Kaysville: NCSS Inc, 2001.
- [30] GANTI VENKATES, GEHRKE JOHANNES VE RAMAKRISHNAN RAGHU , ‘Mining Very Large Databases’, *IEEE Computer*, Cilt 32, Sayı 8, Ağustos, s. 41, 1999.
- [31] NANCY P. LIN, CHUNG-I CHANG VE CHAO-LUNG PAN , ‘An Adaptable Deflect and Conquer Clustering Algorithm’, *Proceedings of the 6th WSEAS International Conference on Applied Computer Science*, Hangzhou, China, s. 155-159, 2007.
- [32] WILLIAMS GRAHAM , *Math3346 Data Mining Clusters*, [Online] http://datamining.anu.edu.au/student/math3346_2005/050809-maths3346-clusters-2x2.pdf Erişim Tarihi: 08.01.2012.
- [33] WEI WANG , ‘Data Mining Concept, Algorithms and Applications’, *Comp 290-090 Research Seminar*, University of North Carolina at Chapel Hill, 2006.
- [34] HAN, J., KAMBER, M. , *Data Mining Concepts and Techniques*, San Francisco: Morgan Kaufmann Publishers Inc, 2001.
- [35] KANTARDZIC, MEHMED , *Data Mining: Concepts, Models, Methods and Algorithms*, USA: A John Wiley and Sons, Inc., Publication, IEEE Press, 2003.
- [36] SANKAR, K. PAL VE PABITRA MITRA , *Pattern Recognition Algorithms for Data Mining*, Washington: A CRC Press Company, 2004.

- [37] BERKHIN, PAVEL , ‘Survey of Clustering Data Mining Techniques’, Technical Report, Accrue Software Inc., San Jose, California, USA, 2002.
- [38] ALEX A. FREITAS , Data Mining and Knowledge Discovery with Evolutionary Algorithms, New York: Springer-Verlag Berlin Heidelberg, 2002.
- [39] A. JAIN VE R. DUBES , Algorithms for Clustering Data, New Jersey: Prentice-Hall, Englewood Cliffs, 1988.
- [40] KAUFMAN, LEONARD VE PETER J. ROUSSEEUW , Finding Groups in Data: An Introduction to Cluster Analysis, USA: John Wiley and Sons, 1990.
- [41] PANG-NING TAN, MICHAEL STEINBACH VE VIPIN KUMAR , Introduction To Data Mining, Boston, MA: Addison-Wesley is an Imprint of Pearson Education Inc, 2005.
- [42] TREUR, R. , Spatial Clustering Methods In Data Mining, [Online], <http://www.cs.uu.nl/docs/vakken/gdm/ch8.ppt> Erişim Tarihi: 25.05.2012.
- [43] BİLGİN T.T, ÇAMURCU Y. , ‘DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması’, Politeknik Dergisi, Cilt 8, Sayı 2, s. 139-145, 2005.
- [44] NAGPAL P.B., MANN P.A. , ‘Comparative Study of Density based Clustering Algorithms’, International Journal of Computer Applications, Cilt 27, Sayı 11, s. 44-47, 2011.
- [45] ESTER M., KRIEGEL, H.-P., SADER, J. VE XU, X. , ‘A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise’, International Conference of Knowledge Discovery and Data Mining (KDD’96), Portland, USA, s. 226-231, 1996.
- [46] ADRIANO MOREIRA, MARIBEL Y. SANTOS VE SOFIA CAMEIRO , ‘Density-Based Clustering Algorithms-DBSCAN and SNN’, University of Minho, Portugal, 2005.
- [47] RENATA IVANCSY, ATTILA BABOS VE CSABA LEGANY , ‘Analysis and Extensions of Popular Clustering Algorithms’, Proc. of the \$6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest, Hungary, s. 394-395, 2005.
- [48] ZHIWEN YU VE HAU-SAN WONG , ‘GCA: A Real-Time Grid-Based Clustering Algorithm For Large Data Set’, 18th International Conference on Pattern Recognition (ICPR’06), Hong Kong, China, s. 740-743, 2006.
- [49] MICHALEWICZ, Z. , Genetic Algorithms + Data Structures = Evolution Programs, Third Edition, London: Springer, 1998.

- [50] REEVES, C. R. AND ROWE J. E. , Genetic Algorithms-Principles and Perspectives A Guide to GA Theory, Boston: Kluwer Academic Publishers, 2003.
- [51] ROTHLAUF, F. , Representations for Genetic and Evolutionary Algorithms, Berlin: Springer, 2002.
- [52] GEN, M. AND CHENG, R. , Genetic Algorithms and Engineering Design, New York: John Wiley and Sons, 1997.
- [53] SMITH, V. J. R., OSMAN, I. H., REEVES, C. R. AND SMITH, G. D. Modern Heuristic Search Methods, New York: John Wiley and Sons, 1996.

EKLER

Tablo A.1 : Kuru yoğun gün problem veri setleri-1

Mağaza No	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
1	8	7	13	13	12	11	8	9	14	13
2	12	6	6	7	10	9	13	8	12	8
3	14	9	12	10	9	13	10	14	10	13
4	12	7	7	10	12	9	7	9	7	13
5	14	9	9	7	9	6	11	11	12	11
6	6	11	10	11	11	13	7	7	10	12
7	12	8	11	10	7	6	6	10	11	14
8	14	11	7	11	11	11	8	7	7	10
9	12	11	9	7	12	11	7	8	7	8
10	8	6	13	7	7	10	11	11	6	13
11	8	9	7	6	11	11	13	6	10	9
12	12	7	9	6	10	9	6	8	10	8
13	8	11	9	10	9	13	9	6	14	12
14	8	13	6	7	12	9	11	9	13	13
15	12	7	10	13	9	13	7	7	12	6
16	8	10	7	6	8	6	7	8	9	9
17	8	7	14	9	9	11	10	9	11	6
18	12	12	7	11	9	11	8	9	7	11
19	8	12	8	7	6	10	10	12	11	6
20	8	11	10	11	9	12	12	7	6	11
21	14	8	12	6	7	13	12	11	11	8
22	8	7	8	11	12	12	9	13	7	9
23	8	7	8	7	7	6	12	8	6	7
24	6	10	7	7	8	8	6	7	10	14
25	8	14	11	10	12	9	10	10	11	8
26	8	6	13	11	7	7	10	8	7	12
27	6	14	7	7	8	6	10	8	9	8
28	8	9	6	9	10	12	7	8	13	8
29	8	9	9	13	14	7	8	12	8	9
30	6	7	14	8	8	8	10	11	12	7
31	8	9	8	12	12	14	7	10	7	9
32	8	8	9	8	11	13	7	10	14	8
33	6	8	7	8	9	8	11	12	12	12
34	8	12	8	12	9	9	11	7	11	12
35	8	6	7	11	10	13	9	11	12	8
36	6	9	10	12	6	10	10	11	9	9
37	8	14	10	9	11	10	13	9	8	8
38	8	7	6	8	7	10	9	10	12	11
39	6	8	13	12	10	8	11	13	10	9
40	8	7	11	10	9	12	11	8	8	9
41	8	13	6	8	9	11	8	7	6	8
42	6	7	9	6	10	8	14	6	13	11
43	8	11	12	13	8	7	11	6	13	7

44	8	8	8	13	12	7	11	8	11	8
45	6	8	7	11	7	7	7	6	11	11
46	8	10	13	11	11	8	6	6	6	12
47	12	13	7	14	13	8	13	10	10	12
48	12	10	10	10	14	8	10	11	8	10
49	8	10	13	8	8	7	12	7	7	8
50	14	9	13	7	9	8	8	9	9	7
51	14	8	9	9	7	10	9	9	6	9
52	8	10	11	6	8	11	8	6	10	6
53	8	11	7	10	9	14	12	6	6	9
54	14	6	10	6	9	9	11	14	12	7
55	8	14	13	12	14	7	8	6	9	10
56	8	12	13	14	7	6	14	9	6	9
57	12	9	13	7	9	7	13	7	7	6
58	8	13	6	6	14	7	7	6	9	14
59	12	10	6	9	8	13	13	6	7	6
60	12	6	11	7	6	9	6	8	9	8
61	8	7	9	9	6	12	9	13	7	12
62	12	14	11	14	6	8	6	11	9	8
63	14	6	9	9	6	7	11	11	9	7
64	8	10	7	7	9	8	7	14	8	6
65	8	7	8	6	7	6	7	14	8	7
66	8	8	11	8	14	7	6	14	7	12
67	8	14	6	11	6	14	12	11	10	8
68	14	13	6	10	13	7	14	8	11	8
69	8	7	12	11	6	6	14	12	6	6
70	12	6	9	14	6	7	7	10	7	11
71	6	8	6	9	6	6	6	9	9	8
72	8	13	14	8	7	12	7	14	8	7
73	12	10	6	12	9	6	6	14	12	10
74	6	10	10	6	6	9	9	10	9	7
75	8	7	7	14	6	14	9	6	7	8
76	14	6	12	6	12	6	6	6	14	6
77	6	9	7	7	14	10	6	9	6	10
78	12	8	9	6	14	8	7	13	6	14

Tablo A.2 : Kuru yoğun gün problem veri setleri-2

Mağaza No	Set 11	Set 12	Set 13	Set 14	Set 15	Set 16	Set 17	Set 18	Set 19	Set 20
1	13	14	7	9	8	14	8	13	10	13
2	11	9	9	12	12	10	8	13	7	6
3	6	11	9	10	7	9	10	11	9	12
4	13	7	11	8	6	7	13	12	7	7
5	9	10	6	12	10	10	9	8	11	9
6	9	7	7	7	8	7	8	11	13	10
7	10	8	10	6	13	9	12	8	10	11
8	12	11	10	8	13	7	6	13	10	7
9	9	6	11	13	11	9	7	9	8	9
10	13	8	7	13	12	8	8	7	10	13
11	11	6	12	11	12	12	9	12	9	7
12	10	9	12	12	13	7	6	13	7	9
13	7	7	13	12	8	9	12	6	10	8
14	11	8	9	13	12	11	11	9	7	6
15	7	9	12	8	7	8	8	6	7	10
16	8	9	12	12	10	7	12	13	12	7
17	8	9	12	7	6	11	7	11	10	14

18	9	12	6	10	8	14	11	6	7	7
19	10	7	8	7	11	7	7	12	10	8
20	8	11	8	8	9	12	6	8	10	10
21	9	13	7	11	9	11	7	9	10	8
22	7	8	10	9	6	9	13	7	13	8
23	11	7	7	6	8	9	8	12	6	8
24	12	10	12	8	8	10	8	8	11	7
25	10	8	14	8	7	13	10	12	11	11
26	7	8	8	7	8	11	9	8	8	11
27	8	8	6	8	8	8	10	8	12	7
28	13	12	11	8	6	11	11	9	12	6
29	8	11	11	7	13	11	12	7	6	9
30	10	10	8	13	7	10	10	9	7	13
31	10	10	10	7	7	12	8	8	8	8
32	7	12	11	7	9	11	8	12	11	9
33	12	7	7	9	11	11	6	12	8	7
34	9	11	9	11	8	7	7	8	9	8
35	7	11	8	8	10	7	11	7	8	7
36	12	9	7	10	7	13	9	9	11	10
37	13	10	11	7	12	9	12	11	6	10
38	9	13	7	12	13	7	7	9	11	6
39	7	7	7	13	14	8	8	9	9	13
40	6	11	9	11	10	7	12	8	6	11
41	8	8	8	10	11	10	9	8	11	6
42	6	7	11	10	6	12	13	11	7	9
43	9	6	7	7	10	8	13	7	6	12
44	13	8	12	10	10	9	13	8	8	8
45	8	6	9	14	8	8	9	11	6	7
46	8	6	13	8	11	6	12	12	9	13
47	7	10	6	6	7	8	6	12	7	7
48	7	11	9	7	9	11	7	10	9	10
49	8	6	8	9	7	11	9	8	9	13
50	14	8	6	7	10	10	9	7	11	14
51	8	12	7	7	11	7	10	9	8	9
52	10	8	12	11	11	9	11	6	13	11
53	8	13	10	8	8	9	6	9	12	7
54	7	13	6	8	6	13	11	14	13	10
55	13	11	7	14	9	8	6	10	14	13
56	9	9	10	9	6	6	6	9	14	13
57	14	7	6	6	12	7	8	6	12	13
58	7	7	12	7	7	6	11	7	11	6
59	11	6	7	8	9	8	14	14	10	6
60	8	6	7	14	12	7	7	8	9	11
61	7	6	14	9	9	6	11	11	14	9
62	14	7	14	10	10	14	9	6	9	11
63	8	11	12	9	12	11	14	8	7	9
64	7	6	8	6	6	6	6	6	6	7
65	7	8	8	6	6	6	9	12	6	8
66	12	14	10	9	8	6	9	6	6	11
67	6	14	14	10	14	6	8	6	6	6
68	7	8	11	10	6	14	6	7	6	6
69	6	12	12	7	12	11	11	6	6	12
70	7	10	14	7	7	14	7	6	7	10
71	6	8	9	13	13	9	14	8	8	6
72	6	8	6	14	9	10	10	9	7	14
73	12	14	6	7	7	14	10	14	14	9
74	14	8	8	13	14	14	6	12	14	13

75	7	6	7	10	7	9	6	12	12	7
76	10	14	7	6	6	6	6	6	10	12
77	14	14	6	6	7	6	14	6	6	7
78	10	14	14	14	14	6	14	14	14	9

Tablo A.3 : Taze yoğun gün problem veri setleri-1

Mağaza No	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
1	5	3	4	5	3	3	4	3	3	3
2	7	4	5	5	4	3	3	3	3	8
3	9	5	5	5	3	3	4	3	3	4
4	7	5	5	6	4	8	7	3	7	4
5	9	5	5	5	5	6	5	3	4	5
6	3	5	6	6	6	9	7	4	5	8
7	7	5	7	3	7	6	4	4	7	9
8	9	5	7	4	8	9	7	5	8	4
9	7	9	7	7	9	9	3	8	7	3
10	5	9	8	8	3	4	5	9	6	3
11	5	9	8	9	4	9	6	6	3	4
12	7	9	9	9	5	5	3	8	3	5
13	5	8	9	9	5	8	9	6	4	6
14	5	8	4	8	5	3	5	9	5	7
15	7	8	5	6	3	7	3	7	6	8
16	5	6	6	7	4	4	7	8	3	9
17	5	6	3	5	3	6	5	9	3	3
18	7	6	4	4	3	4	3	9	3	5
19	5	3	5	3	3	6	4	9	3	4
20	5	3	7	5	3	3	4	7	4	3
21	9	3	8	6	4	6	3	9	4	3
22	5	5	9	7	5	3	4	3	5	9
23	5	5	9	8	6	6	4	9	6	7
24	3	6	9	5	7	8	4	7	3	4
25	5	7	8	6	8	9	9	3	3	3
26	5	8	7	7	9	8	9	8	7	5
27	3	9	6	4	3	6	4	8	3	8
28	5	9	5	5	3	9	7	7	3	9
29	5	6	4	6	3	7	3	5	4	9
30	3	7	5	3	4	7	3	5	3	6
31	5	5	6	3	4	7	8	5	7	9
32	5	4	7	3	3	3	7	5	8	8
33	3	4	7	4	4	8	4	5	6	4
34	5	4	5	3	5	9	4	7	6	6
35	5	3	4	7	6	3	9	5	5	3
36	3	4	3	5	6	3	4	3	9	9
37	5	5	3	3	7	4	3	3	8	8
38	5	5	4	3	7	4	9	3	4	3
39	3	5	6	3	6	5	4	4	4	9
40	5	5	4	3	5	6	5	7	5	9
41	5	6	6	3	4	7	8	3	6	8
42	3	6	6	5	3	8	5	6	6	3
43	5	6	7	4	4	7	6	6	7	7
44	5	6	7	5	4	7	7	8	9	8
45	3	7	3	6	4	7	7	7	9	5
46	5	7	3	7	3	8	8	6	6	3
47	7	8	3	7	3	8	5	3	6	3
48	7	9	3	8	4	8	5	3	8	7

49	5	9	9	5	8	7	4	7	9	3
50	9	8	3	6	9	8	8	9	9	9
51	9	7	7	4	9	9	9	3	6	7
52	5	8	6	6	8	3	8	8	4	6
53	5	7	5	7	8	9	3	4	6	9
54	9	8	4	3	9	5	3	4	9	3
55	5	4	3	4	9	3	4	7	3	6
56	5	5	3	5	9	4	4	6	7	9
57	7	3	3	7	8	3	5	7	8	6
58	5	4	3	8	4	7	7	7	9	7
59	7	5	5	8	5	3	8	6	7	5
60	7	6	6	3	8	3	6	8	9	8
61	5	7	3	3	8	4	9	4	9	3
62	7	9	3	3	9	4	6	4	9	3
63	9	3	9	3	9	7	8	4	4	4
64	5	3	4	4	3	3	7	6	3	4
65	5	4	3	5	3	3	9	6	5	3
66	5	3	9	6	9	3	9	5	6	5
67	5	3	3	8	4	3	4	3	9	5
68	9	9	9	9	4	7	3	3	9	3
69	5	5	5	9	4	6	7	5	5	6
70	7	3	5	9	9	5	4	5	7	3
71	3	4	9	9	5	3	3	4	3	4
72	5	3	9	9	6	3	7	3	8	9
73	7	3	9	9	6	3	7	3	4	4
74	3	3	8	9	9	3	4	3	6	3
75	5	4	5	8	8	3	5	8	4	6
76	9	5	4	4	9	4	8	7	4	4
77	3	3	4	4	7	4	4	9	4	6
78	7	5	4	3	8	5	9	4	3	5

Tablo A.4 : Taze yoğun gün problem veri setleri-2

Mağaza No	Set 11	Set 12	Set 13	Set 14	Set 15	Set 16	Set 17	Set 18	Set 19	Set 20
1	3	3	7	3	8	3	8	3	4	5
2	4	3	9	3	3	4	8	4	7	8
3	5	3	9	4	7	6	3	5	7	3
4	3	4	3	5	6	7	3	3	9	7
5	9	4	6	6	3	8	4	4	4	4
6	9	3	7	3	4	7	8	4	5	6
7	3	8	3	6	4	9	5	6	4	4
8	4	5	4	8	3	7	6	5	3	6
9	5	6	5	3	4	9	7	9	8	3
10	6	8	7	7	3	8	8	7	3	6
11	7	6	3	4	5	3	6	4	4	4
12	8	9	3	9	4	6	9	3	7	3
13	7	7	3	3	8	9	3	4	3	5
14	9	8	9	4	6	4	3	5	7	8
15	7	9	4	7	7	7	8	6	7	9
16	8	9	3	6	8	8	4	3	3	9
17	8	9	4	5	3	3	5	3	4	6
18	9	3	6	4	4	3	4	6	7	9
19	3	7	8	4	3	7	3	4	5	8
20	8	3	8	3	3	4	6	5	4	4
21	9	4	7	3	9	7	7	9	4	6
22	7	8	3	9	6	8	3	8	5	3

23	3	7	7	6	8	3	7	5	6	9
24	3	5	3	8	8	3	8	4	5	8
25	3	6	4	8	7	4	4	3	4	3
26	4	8	8	7	8	4	9	4	8	9
27	4	8	6	8	8	8	4	5	4	9
28	5	6	3	8	6	5	5	4	3	8
29	6	7	3	7	4	6	5	3	6	3
30	3	8	8	4	7	3	5	5	7	9
31	3	9	3	7	7	3	4	4	8	4
32	4	3	4	7	9	4	8	3	4	3
33	5	7	3	9	4	5	6	5	4	4
34	6	3	4	5	5	6	7	4	3	6
35	7	4	3	8	3	7	4	9	8	8
36	6	9	4	9	7	6	9	8	4	8
37	7	5	3	5	3	7	4	3	6	7
38	8	6	4	4	3	3	7	9	3	3
39	9	4	3	6	4	8	8	9	4	7
40	3	4	4	3	5	7	4	8	6	3
41	4	5	3	4	6	3	4	8	4	4
42	6	7	4	4	7	3	3	3	7	8
43	9	6	3	5	8	8	4	4	6	6
44	3	8	4	5	8	3	5	8	8	3
45	5	6	3	6	8	8	6	5	6	3
46	6	5	6	3	4	6	3	5	9	8
47	7	6	9	7	3	3	6	4	7	3
48	7	7	8	7	9	3	7	3	9	4
49	8	4	5	3	7	4	9	8	9	6
50	9	8	4	3	3	5	9	7	4	7
51	3	4	6	6	3	6	3	9	8	3
52	9	5	7	7	4	3	3	6	4	6
53	8	5	6	7	5	9	4	9	3	3
54	7	6	9	9	6	3	8	7	4	4
55	9	4	9	6	9	3	8	3	5	3
56	3	3	4	9	6	6	5	9	7	5
57	8	7	5	6	3	3	5	6	5	5
58	7	7	7	7	7	4	6	7	4	4
59	3	3	8	9	9	3	7	6	4	6
60	3	4	5	6	3	9	8	8	9	7
61	7	5	4	3	4	9	4	9	9	8
62	3	6	9	4	4	9	9	8	9	7
63	3	7	9	9	5	7	9	8	5	3
64	3	3	9	5	4	9	7	6	4	3
65	3	3	7	5	3	3	7	9	4	6
66	4	4	9	3	9	4	3	3	9	4
67	5	5	4	3	4	9	3	4	3	7
68	3	3	6	7	3	6	3	8	4	5
69	3	4	7	8	3	4	4	3	3	5
70	7	4	6	7	4	3	7	4	5	5
71	7	8	9	8	5	7	5	7	7	6
72	6	9	9	3	9	8	5	7	3	4
73	9	4	5	9	7	3	5	4	9	3
74	5	3	7	3	6	8	6	7	7	9
75	7	3	8	3	7	7	4	8	5	9
76	3	7	5	4	9	4	3	4	7	7
77	3	8	7	3	9	6	9	4	9	8
78	3	4	5	4	8	8	3	5	7	4

Tablo A.5 : Kuru sakın gün problem veri setleri-1

Mağaza No	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
1	8	14	9	14	8	10	11	14	10	8
2	12	9	11	11	8	9	9	10	12	11
3	14	9	12	11	9	8	9	8	9	14
4	12	9	9	10	11	11	8	13	13	8
5	14	8	11	11	13	8	10	9	11	8
6	0	0	0	0	0	0	0	0	0	0
7	12	8	13	9	14	12	12	11	11	8
8	14	14	9	14	10	8	9	13	9	14
9	12	12	10	8	8	8	12	8	8	9
10	8	9	10	9	13	8	8	9	10	14
11	8	13	9	11	9	9	10	10	9	8
12	12	12	11	13	8	10	10	8	9	12
13	8	9	10	9	12	9	10	8	9	8
14	8	14	11	14	13	12	9	8	13	8
15	12	11	9	9	8	14	8	12	9	9
16	8	12	11	9	9	8	10	11	12	9
17	8	14	9	9	8	8	9	10	9	9
18	12	8	9	9	11	11	9	10	9	8
19	8	11	8	14	14	11	11	12	9	12
20	8	9	9	9	11	8	11	9	8	12
21	14	8	8	8	8	10	9	11	8	8
22	8	11	8	8	9	11	10	11	9	9
23	8	10	9	9	9	9	13	9	8	8
24	0	0	0	0	0	0	0	0	0	0
25	8	13	8	9	11	10	13	9	11	12
26	8	14	11	10	12	8	10	10	11	9
27	0	0	0	0	0	0	0	0	0	0
28	8	9	8	9	11	8	11	8	8	8
29	8	8	14	10	8	14	8	13	11	8
30	0	0	0	0	0	0	0	0	0	0
31	8	11	10	9	13	10	14	10	13	8
32	8	8	10	12	9	9	9	9	13	13
33	0	0	0	0	0	0	0	0	0	0
34	8	10	12	9	11	9	11	8	9	9
35	8	8	12	8	9	8	10	10	12	11
36	0	0	0	0	0	0	0	0	0	0
37	8	12	11	11	9	11	11	11	8	9
38	8	9	12	11	9	9	12	11	9	8
39	0	0	0	0	0	0	0	0	0	0
40	8	9	8	9	13	14	8	9	11	8
41	8	9	9	9	9	9	8	10	8	14
42	0	0	0	0	0	0	0	0	0	0
43	8	9	8	9	10	9	13	9	14	14
44	8	9	8	14	9	12	9	11	9	13
45	0	0	0	0	0	0	0	0	0	0
46	8	12	8	8	11	13	8	9	8	12
47	12	11	9	8	14	8	8	14	10	11
48	12	12	8	8	8	12	12	14	8	10
49	8	8	8	8	8	8	8	13	8	10
50	14	12	9	8	8	8	8	8	11	12
51	14	8	9	8	8	14	8	8	8	9
52	8	14	8	8	8	14	8	12	8	8
53	8	8	9	8	8	12	11	11	8	8
54	14	8	9	8	8	9	8	13	12	8

55	8	9	9	14	8	13	14	8	12	10
56	8	14	8	13	8	8	8	8	11	13
57	12	8	8	8	8	8	8	13	14	8
58	8	8	8	8	13	8	14	8	14	8
59	12	9	8	8	8	8	8	14	8	8
60	12	9	8	8	13	8	14	8	8	8
61	8	8	14	8	14	13	13	8	11	8
62	12	8	8	11	13	8	8	8	8	8
63	14	13	8	14	8	8	11	8	8	14
64	8	10	8	8	8	14	13	8	8	10
65	8	8	14	8	8	8	8	8	8	11
66	8	8	8	13	8	8	8	8	8	8
67	8	8	8	8	8	8	12	8	11	8
68	14	8	14	8	11	8	8	8	8	11
69	8	8	14	8	14	8	14	8	12	9
70	12	9	14	8	8	8	8	8	14	11
71	0	0	0	0	0	0	0	0	0	0
72	8	8	8	14	8	13	8	8	8	9
73	12	8	14	10	8	11	8	8	10	12
74	0	0	0	0	0	0	0	0	0	0
75	8	9	14	13	12	9	8	11	13	10
76	14	8	14	10	10	11	8	14	8	14
77	0	0	0	0	0	0	0	0	0	0
78	12	9	10	13	9	14	8	8	8	8

Tablo A.6 : Kuru sakın gün problem veri setleri-2

Mağaza No	Set 11	Set 12	Set 13	Set 14	Set 15	Set 16	Set 17	Set 18	Set 19	Set 20
1	10	8	13	9	12	8	8	8	8	9
2	10	8	13	14	14	13	8	11	14	12
3	11	13	11	9	9	9	8	14	10	9
4	9	13	12	8	8	9	8	11	8	11
5	12	11	12	12	9	12	11	8	9	13
6	0	0	0	0	0	0	0	0	0	0
7	9	13	9	11	9	11	9	9	9	11
8	9	8	10	9	8	8	8	9	14	11
9	11	8	9	8	14	8	8	14	10	10
10	8	10	12	8	8	10	9	9	11	9
11	10	8	14	14	9	13	12	11	11	9
12	9	14	8	9	8	11	10	8	8	9
13	8	8	8	8	14	8	8	8	8	13
14	8	12	11	8	8	11	9	14	14	8
15	8	11	11	9	13	11	10	9	8	8
16	10	12	8	13	9	8	8	8	12	9
17	8	10	10	9	14	8	11	9	9	8
18	9	12	11	14	9	11	9	8	9	8
19	12	8	9	9	11	11	9	14	12	9
20	8	11	9	11	8	8	11	9	9	9
21	8	14	8	14	10	8	11	10	10	13
22	8	8	9	10	9	13	9	9	12	9
23	13	10	11	9	12	8	9	8	8	13
24	0	0	0	0	0	0	0	0	0	0
25	8	9	11	8	8	10	9	11	9	8
26	9	12	9	9	10	9	13	9	9	11
27	0	0	0	0	0	0	0	0	0	0
28	8	8	10	11	10	9	8	11	8	14

29	9	11	10	14	12	13	11	9	9	9
30	0	0	0	0	0	0	0	0	0	0
31	11	9	10	9	9	10	11	9	12	12
32	9	11	8	14	9	13	12	9	9	11
33	0	0	0	0	0	0	0	0	0	0
34	11	9	12	8	10	8	14	11	14	9
35	8	10	8	8	9	11	9	9	12	10
36	0	0	0	0	0	0	0	0	0	0
37	9	10	12	11	10	8	13	14	14	13
38	14	8	8	8	11	13	11	9	9	9
39	0	0	0	0	0	0	0	0	0	0
40	11	8	8	8	8	8	10	11	10	9
41	8	13	11	8	9	11	10	8	12	13
42	0	0	0	0	0	0	0	0	0	0
43	12	9	8	13	12	8	9	12	8	10
44	13	11	8	9	13	12	11	11	9	9
45	0	0	0	0	0	0	0	0	0	0
46	8	8	8	11	11	8	13	11	12	9
47	12	8	10	10	8	13	9	10	10	9
48	8	8	10	10	10	8	9	12	8	8
49	8	8	8	12	11	8	9	11	14	12
50	12	12	12	9	9	8	11	11	8	8
51	11	8	8	11	14	11	8	9	9	8
52	12	8	14	11	8	8	10	9	8	9
53	8	9	12	9	9	11	9	13	8	9
54	11	8	13	10	11	8	8	9	8	8
55	8	8	8	13	11	11	8	14	8	8
56	8	8	8	9	8	11	8	8	8	8
57	8	8	8	8	8	11	8	8	8	8
58	13	8	14	11	8	8	11	8	9	8
59	13	8	8	8	8	13	8	8	13	8
60	8	8	8	14	8	8	10	8	8	8
61	13	11	13	8	8	8	11	8	8	8
62	8	8	8	8	8	8	11	8	9	11
63	11	12	14	11	13	10	13	13	12	12
64	8	8	8	8	8	8	11	8	8	8
65	8	10	8	8	14	8	9	9	11	8
66	13	8	8	8	8	8	14	8	8	8
67	11	11	8	8	8	13	10	8	10	14
68	8	14	8	8	14	8	8	8	10	9
69	9	11	14	8	14	8	8	8	11	11
70	8	13	13	14	8	13	8	13	8	12
71	0	0	0	0	0	0	0	0	0	0
72	11	9	8	9	9	8	8	8	8	13
73	14	11	8	11	9	11	9	8	12	8
74	0	0	0	0	0	0	0	0	0	0
75	8	11	8	8	8	14	11	11	9	13
76	14	11	8	8	8	11	14	12	8	14
77	0	0	0	0	0	0	0	0	0	0
78	12	11	8	8	8	11	14	12	14	8

Tablo A.7 : Taze sakin gün problem veri setleri-1

Mağaza No	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
1	5	8	6	6	8	5	8	5	6	6
2	7	7	5	7	8	9	5	6	6	9

3	9	8	6	5	9	5	9	6	9	5
4	7	6	5	6	9	5	6	5	5	5
5	9	5	6	5	9	6	6	6	7	9
6	0	0	0	0	0	0	0	0	0	0
7	7	8	9	6	5	8	8	5	5	7
8	9	7	7	6	5	9	5	5	5	9
9	7	8	6	6	8	8	5	8	8	6
10	5	7	5	7	5	5	7	5	6	5
11	5	8	6	5	5	8	7	5	5	5
12	7	6	5	6	5	8	6	7	5	5
13	5	5	6	5	9	8	7	5	5	9
14	5	5	5	7	8	5	5	7	5	6
15	7	6	6	8	6	7	7	7	5	7
16	5	5	5	8	5	5	5	6	7	5
17	5	6	6	5	5	5	6	5	9	8
18	7	7	5	5	5	5	9	5	9	5
19	5	9	5	5	9	6	6	5	9	5
20	5	5	6	5	9	7	8	5	6	6
21	9	5	6	6	5	5	7	5	5	6
22	5	6	5	5	5	5	9	5	5	9
23	5	5	6	6	9	5	9	5	6	5
24	0	0	0	0	0	0	0	0	0	0
25	5	9	5	9	6	7	5	5	9	5
26	5	5	5	9	6	6	7	5	5	6
27	0	0	0	0	0	0	0	0	0	0
28	5	5	9	9	9	9	6	5	6	5
29	5	6	5	8	8	5	5	5	6	6
30	0	0	0	0	0	0	0	0	0	0
31	5	7	5	7	8	9	5	6	6	5
32	5	8	6	5	9	5	5	6	9	5
33	0	0	0	0	0	0	0	0	0	0
34	5	9	9	5	9	7	8	6	6	5
35	5	9	6	6	5	9	7	6	5	6
36	0	0	0	0	0	0	0	0	0	0
37	5	9	5	6	5	7	5	7	6	8
38	5	9	6	7	6	6	7	8	5	9
39	0	0	0	0	0	0	0	0	0	0
40	5	5	5	7	5	5	6	5	5	9
41	5	5	5	8	6	6	6	9	6	7
42	0	0	0	0	0	0	0	0	0	0
43	5	5	6	5	5	6	5	8	5	8
44	5	6	5	5	5	9	7	7	5	9
45	0	0	0	0	0	0	0	0	0	0
46	5	9	7	5	5	6	5	8	7	9
47	7	9	8	8	5	6	7	9	6	5
48	7	5	8	5	5	6	5	6	5	5
49	5	5	9	5	5	6	5	8	5	5
50	9	5	9	5	5	5	5	6	5	6
51	9	5	6	8	5	5	5	9	5	5
52	5	5	5	9	5	5	5	7	6	8
53	5	5	6	7	5	6	7	8	9	5
54	9	5	9	5	5	7	5	6	5	5
55	5	6	6	5	5	5	8	9	9	5
56	5	5	9	5	5	6	5	5	8	5
57	7	5	7	5	5	5	5	7	5	5
58	5	5	7	6	5	5	5	5	5	5
59	7	5	5	7	5	5	5	7	5	5

ÖZGEÇMİŞ

Tolga Şen, 13.06.1987 de İstanbul' da doğdu. İlk, orta ve lise eğitimini İstanbul'da tamamladı. 2006 yılında başladığı İstanbul Ticaret Üniversitesi Endüstri Mühendisliği bölümünü 2011 yılında bitirdi. 2011 yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Bölümüne girdi. 2011 – 2014 yılları arasında Marmara Matbaacılık Sanayi ve Tic. Ltd. Şti.nde mühendis olarak çalıştı. Bu süre içerisinde şirketin yeni ürün projeleri yanı sıra verimlilik projeleri ve toplam kalite yönetimi projelerinde aktif rol aldı. Şu anda Marmara Matbaacılık San. ve Tic. Ltd. şirketinde işletme müdürü olarak görev yapmaktadır.