



**OTOMATİK SATIŞ SİSTEMLERİ İÇİN BÜTÜNLEŞİK
YÖNETİM SİSTEMİ**

Gizem GÖÇEN



T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

OTOMATİK SATIŞ SİSTEMLERİ İÇİN BÜTÜNLEŞİK YÖNETİM SİSTEMİ

Gizem GÖÇEN
0000-0003-4095-723X

Doç. Dr. Aslı AKSOY
(Danışman)

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2022
Her Hakkı Saklıdır

ÖZET

Yüksek Lisans Tezi

OTOMATİK SATIŞ SİSTEMLERİ İÇİN BÜTÜNLEŞİK YÖNETİM SİSTEMİ

Gizem GÖÇEN

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Aslı AKSOY

Otomatik satış makineleri, klasik bir satış noktası için gereken kira, enerji personel gibi maliyetleri azaltması, 7/24 hizmet vermesi ve kolay ulaşılabilir olması sebebiyle birçok ülkede günlük hayatta sıkça kullanılan bir alışveriş aracı olmuştur. Artan ürün, personel ve enerji maliyetleri, otomatik satış makineleri yöneticilerini zorlamakta, kurumların karlılığını düşürmektedir. Bu nedenle otomatik satış makineleri yöneticileri maliyetleri düşürecek yeni yönetim stratejileri arayışına girmişlerdir.

Bu tez çalışmasının amacı otomatik satış makineleri sistem yöneticilerine yardımcı olmak adına müşteri hizmet düzeyi ve ürün ulaşılabilirliğini arttırabilmek için envanter yönetim sistemi oluşturulması ile operasyonel yönetim ve kayıp satışları azaltacak, personel ve operasyon etkinliğini arttırırken maliyetleri azaltacak dağıtım planı oluşturmayı içeren bir yönetim sistemi geliştirmektir. Bu amaç doğrultusunda tez çalışması kapsamında otomatik satış makineleri ağının yönetimi için makinelere ürün dağıtımı yapan depoda bulunan ürünler sınıflandırılarak ortak yönetim stratejisi önerilmiştir. Dağıtım personelinin günlük makine ziyaretini planlayan araç rotalama algoritması oluşturulmuştur.

Anahtar Kelimeler: Envanter yönetimi, operasyonel yönetim, araç rotalama, otomatik satış sistemleri

2022, vii + 138 sayfa.

ABSTRACT

MSc Thesis

AN INTEGRATED MANAGEMENT SYSTEM FOR VENDING SYSTEMS

Gizem GÖÇEN

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Assoc. Prof. Dr. Aslı AKSOY

Vending machines have become a frequently used shopping tool in daily life in many countries, as they minimize the costs such as personnel, energy and rent required for a classical point of sale, provide 24/7 service and are easily accessible. Increasing product, personnel and energy costs force vending machine managers and reduce the profitability of enterprises. For this reason, vending machine managers have sought new management strategies to reduce costs.

The aim of this study is to develop a management system that will assist system administrators in inventory management and distribution planning in order to reduce operational costs and lost sales, increase employee and operation efficiency, customer service level and product availability for vending machine system administrators. For this purpose, within the scope of the thesis, aintegrated management strategy has been proposed by classifying the products in the warehouse that distributes products to the machines for the management of the vending machine network. A vehicle routing algorithm has been developed that plans the daily machine visit of the distribution operator.

Key words: Inventory management, operational management, vehicle routing, vending systems

2022, vii +138 pages.

TEŞEKKÜR

Tez çalışmamın her aşamasında değerli katkı ve eleştirileriyle yol gösteren, büyük bir anlayış ve sabırla çalışmamı tamamlamama teşvik eden danışmanım Sayın Doç. Dr. Aslı AKSOY'a, her zaman yanımda olan ve beni her koşulda destekleyen aileme, tez çalışmas kapsamında yapılan araştırmalarda yardımcı olan TÜBİTAK 1507 Kobi Ar-Ge Başlangıç Destek Programı tarafından desteklenen "Vendora IOT Otomat Projesi" başlıklı projenin yürütücüsü Sayın Mehmet KAVİ'ye teşekkürlerimi sunarım.

Gizem GÖÇEN

.../.../.....



İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ	v
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ	vii
1. GİRİŞ	1
2. KAYNAK ÖZETLERİ	4
2.1. Otomatik Satış Makineleri	4
2.2. Envanter Sınıflandırma ve Yönetimi	5
2.3. Araç Rotalama Problemi	9
2.3.1 Araç Rotalama Problemi Çeşitleri	9
2.3.2. Araç Rotalama Problemi Çözüm Yöntemleri	12
3. MATERYAL VE YÖNTEM	19
3.1. Envanter Sınıflandırma ve Yönetimi	20
3.2. Araç Rotalama.....	26
4. BULGULAR VE TARTIŞMA	34
4.1. Sistem Analizi	34
4.2. Envanter Sınıflandırma ve Yönetimi	36
4.3. Araç Rotalama Probleminin Çözümü	45
5. SONUÇ	62
KAYNAKLAR	64
EKLER.....	69
EK 1 k-ortalamalar algoritması kullanılarak elde edilmiş üç kümeye ait bilgiler	70
EK 2 Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler ..	72
EK 3 Hibrit kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler.....	74
EK 4 k-ortalamalar algoritması kullanılarak elde edilmiş iki kümeye ait bilgiler	76
EK 5 Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler..	78
EK 6 Hibrit kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler	80
EK 7 Uzaklık matrisi.....	82
EK 8: İki farklı rota uzunluğunun gösterilmesi	83
EK 9: Sinyal matrisi	84
EK 10 Algoritmanın Visual Basic dilinde yazılan kodu	85
EK 11 Mesafe matrisinin oluşturulması için yazılan kod	94
EK 12 Geliştirilen algoritma için Visual Basic dilinde yazılan kod	99
EK 13 Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod	113
EK 14 Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod.....	124
ÖZGEÇMİŞ	138

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

Açıklama

α	Sıcaklık düşürme kuralı
β	Rasgele bir sayı
σ_D	Birim zamanda gerçekleşen talebin standart sapması
z	Hizmet düzeyi

Kısaltmalar

Açıklama

AB	Arıza Bilgisi
AHP	Analitik Hiyerarşi Prosesi
ARP	Araç Rotalama Problemi
ÇDÖDSTARP	Çok Depolu Önce Dağıt Sonra Topla Araç Rotalama Problemi
ÇKEK	Çok Kriterli Envanter Kontrolü
EOQ	Economic Order Quantity (Ekonomik Sipariş Miktarı)
ES	Emniyet Stoğu
EZTDARP	Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi
GTARP	Geri Toplamalı Araç Rotalama Problemi
IoT	Internet of Things (Nesnelerin İnterneti)
KİU	Küme İçi Uzaklık
KKARP	Kapasite Kısıtlı Araç Rotalama Problemi
MKARP	Mesafe Kısıtlı Araç Rotalama Problemi
MPL	Mathematical Programming Language
ÖDSTARP	Önce Dağıt Sonra Topla Araç Rotalama Problemi
PARP	Periyodik Araç Rotalama Problemi
ZARP	Ayrık Yükleme ve Zaman Pencere Araç Rotalama Problemi

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Tez kapsamında yapılan çalışmalar	19
Şekil 3.2. Envanter kümeleme ve yönetim süreci	21
Şekil 3.3. Optimal küme sayısının seçilmesi	25
Şekil 4. 1. Otomatik satış makinesi yönetim sistemi.....	35
Şekil 4.2. Hiyerarşik kümeleme yönteminin R kodu	37
Şekil 4.3. Hiyerarşik kümeleme sonucunun dendogramda gösterilmesi	38
Şekil 4.4. k-ortalamlar yönteminin R kodu	39
Şekil 4.5. Hibrit kümeleme yönteminin R kodu	41
Şekil 4.6. Elbow yönteminin uygulanması	43
Şekil 4.7. Elbow yöntemine göre küme sayısının belirlenmesi	43
Şekil 4.8. Silhouette indeksinin uygulanması	44
Şekil 4.9. Silhouette indeksine göre küme sayısının belirlenmesi	44
Şekil 4.10. Temel modelin MPL kodu	46
Şekil 4.11. Temel modelin MPL sonucu.....	47
Şekil 4.12. Sinyal durumunun olduğu matematiksel model kodu	49
Şekil 4.13. Sinyal durumunun eklendiği modelin sonucu.....	50
Şekil 4.14. Makine ziyareti planlama algoritmasının kullanıcı ara yüzü	53
Şekil 4.15. Haritadan mesafe matrisi elde etme	54
Şekil 4.16. Rota sonucunun paylaşıldığı ekran	55

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 4.1. Hiyerarşik kümeleme yönteminden elde edilen üç kümeye ait bilgiler	38
Çizelge 4.2. Hiyerarşik kümeleme yönteminden elde edilen iki kümeye ait bilgiler	39
Çizelge 4.3. k-ortalamlar yönteminden elde edilen üç kümeye ait bilgiler.....	40
Çizelge 4.4. k-ortalamlar yönteminden elde edilen iki kümeye ait bilgiler	40
Çizelge 4.5. Hibrit kümeleme yönteminden elde edilen üç kümeye ait bilgiler	41
Çizelge 4.6. Hibrit kümeleme yönteminden elde edilen iki kümeye ait bilgiler.....	42
Çizelge 4.7. Temel model için birim sayısına göre araç rotaları	48
Çizelge 4.8. Sinyal durumunun olduğu modelde birim sayısına göre araç rotaları	51
Çizelge 4.9. İki model sonucunun karşılaştırılması	52
Çizelge 4.10. Araç rotalama algoritmasına göre 30 birim için araç rotaları	55
Çizelge 4.11. Birim sayılarına göre her bir aracın rotası	56
Çizelge 4.12. Arıza bilgisinin öncelikli olduğu durumda 30 birim için elde edilen rota	58
Çizelge 4.13. Öncelikli durum için birim sayılarına göre araç rotaları.....	58
Çizelge 4.14. Matematiksel model ve algoritma sonuçlarının karşılaştırılması	59
Çizelge 4.15. En uzak birimden başlanarak oluşturulan algoritma sonuçları	60

1. GİRİŞ

Otomatik satış makineleri hazır gıda, içecek, gazete, sigara, bilet sağlamak amacıyla halka açık yerlerde bulunmaktadırlar. Son zamanlarda geliştirilen otomatik satış makineleri halka hizmet sunabilmek için bir kontrol sistemine ihtiyaç duymaktadır (Murena vd., 2020).

Son yıllarda otomatik satış makinelerinin popülerliği artmıştır. Otomatik satış makineleri ürünlere hızlı ve kolay ulaşım imkânı sağlamaktadır. Otomatik satış makinesi endüstrisinin ekonomiye katkıları tartışmasız olsa da otomatik satış makinelerinin lojistik operasyonlarının verimliliği söz konusunda zorluklar yaşanmaktadır. Otomatik satış makinelerinin lojistik optimizasyonu karmaşıktır. Bir otomatik satış makinesinde ürünleri yeniden doldurma noktası, ürünlerin eşik değeri ve envanter yenileme için araç rotaları otomatik satış makinelerini yönetimi ve işletimindeki temel zorluklardır (Grzybowska vd., 2020).

Tez çalışması kapsamında otomatik satış makinelerinin yönetimini gerçekleştiren bir firma ele alınmıştır. Firmayla yapılan görüşmeler sonucu otomatik satış makinelerinin yönetimiyle ilgili aşağıdaki sorunlar tespit edilmiştir:

- Otomatik satış makinesi ağı tamamen yönetici deneyimine ve uzmanlığına bağlı olarak yönetilmekte, yönetim sırasında herhangi bir analitik yöntem ve/veya araç kullanılmamaktadır.
- Farklı bölgelerde yer alan otomatik satış makinelerine ürün beslemesini gerçekleştiren personel kendi deneyimine / inisiyatifine bağlı günlük ziyaret planını oluşturmaktadır. Makinelerden herhangi bir satış bilgisi gelmediğinden, dağıtım personeli önce aracı otomatik satış makinesinin bulunduğu binanın otoparkına park etmekte, otomatik satış makinesine yürümekte, otomatik satış makinesini inceleyerek biten ve/veya azalan ürünleri belirlemekte, araca dönüp yükleme yapılacak ürünleri hazırlamakta ve tekrar otomatik satış makinesine dönüp ürün yerleştirme işlemini gerçekleştirmektedir. Bazı sistem yöneticileri ise dağıtıcı personel dışında, belirli bölgeler için (örneğin üniversite kampüsü) bölge çalışanları istihdam etmektedir. Bu kişiler gün boyunca çalıştıkları bölgedeki

otomatik satış makinelerini kontrol ederek, ürün azalması/bitmesi durumunda dağıtıcı personele haber vermekte, dağıtım personeli ilgili makineye geldiğinde ürün dolumu gerçekleştirilmektedir. Bazı otomatik satış makinelerinde ise makine yanına kapalı, kilitli stok dolapları yapılmakta, bölge sorumlusu makinelerde ürün azalması durumunda stoktan makineyi doldurmakta, dağıtıcı personel ise stok alanını doldurmaktadır. Böyle bir sistemin hem çalışan maliyeti fazladır hem de kontrol süreci daha zordur.

- Otomatik satış makinelerinin birçoğu herhangi bir çevrimiçi veri değişim sistemine sahip olmadığından makinelerde arıza bilgisi alınamamaktadır. Makinelerde arıza oluştuğunda, ya arıza sonrası ilk ziyarette fark edilmekte, dağıtım personelinin giderebileceği bir sorunsu orada onarım yapılmakta, personelin gideremeyeceği bir sorunsu teknik ekibe haber verilmektedir. Makine tekrar devreye alınana kadar satış yapılamamakta, müşterilere hizmet sunulamamaktadır. Bazı lokasyonlarda ise müşteriler ve/veya işyeri çalışanları/sahipleri arıza durumu ortaya çıktığında otomatik satış makinelerinin sistem yöneticisine haber verebilmektedir. Ancak bu nadir yaşanan bir durumdur.
- Dağıtım personeli gün başında ziyaret edeceği otomatik satış makinelerini kendisi belirlemekte, dağıtım rotasını da kendisi oluşturmaktadır. Bu operasyon için sektörde kullanılan analitik bir yöntem bulunmamaktadır.
- Ürün talebi belirsiz olduğundan makinelerin ziyaret periyodu deneyime bağlı belirlenmektedir. Deneme yanılma yöntemiyle karar verilmekte, ziyaret periyodunun yanlış belirlenmesi operasyonel maliyetlerin artmasını, karlılığın düşmesine, müşteri memnuniyet seviyesinin düşmesine ve ürün erişilebilirliğinin azalmasına neden olmaktadır.
- Depo yönetiminde birçok sorun yaşanmaktadır. Otomatik satış makinelerine yerleştirilen ürünler düşük hacimli ürünler olduğundan, depoda yerleşim ya da yer bulma problemi yaşanmamakta, ancak ürün çeşitliliği fazla olduğundan çeşit bazında ürün yönetimi etkin bir şekilde gerçekleştirilmemektedir. Bir otomatik satış makinesinde yaklaşık 48-60 adet ürün gözü bulunmakta, farklı lokasyonlardaki, müşteri profiline göre makinelere farklı ürünler yerleştirilmektedir. Küçük bir otomatik satış makinesi ağında bile 100 farklı çeşitte ürün olabilmektedir. Bu ürünler ürün tipine göre yönetilmektedir; örneğin

bisküvi grubu, ikolata grubu, tuzlu grubu Ürün sınıflandırması yapılırken herhangi bir analitik yöntem kullanılmamakta, geçmişten gelen deneyime dayanarak, örneğın bisküvi grubu ürünlerin hepsi aynı stratejiyle yönetilmektedir. Ürünlerin stok yenileme miktarları, emniyet stok miktarları gibi kavramlar bulunmamaktadır.

Yukarıda bahsedilen problemler neticesinde bu tez alışmasında otomatik satış makinelerinin dağıtım sistemi için bir araç rotası ve otomatik satış makinelerine ürün beslemesi yapan depodaki ürünler için bir envanter yönetim stratejisi oluşturulmuştur.

Otomatik satış makinelerinin günlük makine ziyaretlerini planlayan dinamik bir rota oluşturulmuştur. Bu sayede makinelerdeki ürünlerin stok durumlarına göre günlük ziyaret rotaları oluşturulmuş, herhangi bir lokasyonda yer alan makineden arıza bilgisi gelmesi durumunda aracın öncelikli olarak o makineye uğraması sağlanmıştır. Rota, otomatik satış makinelerindeki ürünlerin stok bilgilerinin dışında ürünlerin kullanım süreleri ile de ilişkilendirilerek, kullanım süresi dolan ürünler makinelerden toplanabilecektir. Bu özellik sayesinde mevcutta olmayan birçok otomatik satış makinesine sağlıklı ürün yerleştirilebilecektir.

Bu alışma kapsamında depodaki ürünlerin envanter yönetimini gerçekleştirmek için depoda yer alan ürünlerin sınıflandırılmasını sağlayan ok kriterli ürün sınıflama sistemi oluşturulmuştur. Böylece ürünler sınıflandırılıp farklı stok yönetim stratejileri ile yönetilebilecektir. Stok yönetiminde önemli bir konu benzer ürünleri gruplayarak bu ürünler için ortak yönetim stratejisi geliştirmektir. ABC analizi ürün sınıflandırması için sıklıkla kullanılan bir yöntemdir. Genellikle ürünlerin ekonomik değeri veya satış miktarları baz alınarak sınıflandırma yapılmaktadır. Ancak tek bir kritere bağılı olarak yapılan bu sınıflandırma kritik ürünlerin belirlenmesinde faydalı olmamaktadır. O nedenle ok kriterli sınıflandırma yöntemleri son yıllarda hem literatürde hem de endüstride ilgi eken bir konu olmaktadır. Bu alışmada da benzer ürünler k-ortalamlar, hibrit ve hiyerarşik kümeleme yöntemleriyle gruplara ayrılmış ve her bir ürün grubu için ortak bir envanter yönetim stratejisi önerilmiştir.

2. KAYNAK ÖZETLERİ

Bu bölümde tez kapsamında incelenmiş olan otomatik satış makineleri ve yönetim sistemleri, envanter sınıflandırma, araç rotalama problemi hakkında tez çalışması kapsamında incelenen kaynak özetlerine yer verilmiştir.

2. 1. Otomatik Satış Makineleri

Otomatik satış makineleri birçok ülkede günlük hayatın parçası olmuştur. Otomatik satış makinelerinin doğa dostu alışveriş araçları olduğu da düşünülmektedir; çünkü otomatik makineleri sayesinde tüketiciler ihtiyaçlarına uzun mesafe kat etmeden ulaşabilmektedir (Park ve Yoon, 2012).

Otomatik satış birimleri, doğası gereği rasyonel veya başka bir şekilde karar vermezler. Geleneksel olarak, bilgi aktarmazlar. Yeni teknolojiler ile bilginin paylaşılmasına olanak sağlamaktadırlar (Ketzenberg vd., 2012).

Park ve Yoon (2012), otomatik satış sistemlerinin yönetimi için iki aşamalı bir çözüm geliştirmişlerdir. Çözüm, bütünleşik optimizasyon matematiksel modeli üzerine geliştirilmiştir. İlk aşamada, satış makineleri için makine gözlerine tahsisat ve yeniden doldurma aralıkları belirlenmiştir. İkinci aşamada, satış makinelerini yenilemek için araç rotaları belirlenmiştir.

Park ve Yoo (2012), otomatik satış sistemlerinin yönetimi için ayrıştırma yaklaşımı temelinde bir sezgisel önermişlerdir. Sistemdeki her bir akıllı otomatik satış makinesi için ürün saklama bölmeleri sayısını ve yenileme eşiğini belirlemek için bir tam sayılı doğrusal matematiksel model oluşturmuş ve aynı dağıtım gününü paylaşan akıllı otomatik satış makinelerinin stok yenilemelerini yapmak için Clarke ve Wright'ın kazanım algoritmasını kullanarak araç rotalarını belirlemişlerdir.

2.2. Envanter Sınıflandırma ve Yönetimi

Stoklar, genel anlamda, kullanılmak için depoda hazır konumda bulunan hammadde, yarı mamul, bileşen ve mamuller olarak tanımlanmaktadır. Dolayısıyla stoklar, üretilen ürüne dolaylı ve dolaysız eklenen tüm fiziki olarak bulunan ürünler ve üretilen ürünün kendisini kapsamaktadır (Elder ve Tsoukalas, 2006).

Bir perakende işletmesi içerisinde stok yönetimi yapılmasının faydaları aşağıda belirtildiği gibi sıralanabilir (Ustaahmetoğlu, 2013):

- Stoklara ayrılan bütçe perakendecinin gereksinimlerine göre oluşturulduğundan, etkin şekilde bir finans yönetimi yapılır.
- Perakendecilerin tedarik ve satış harcamaları azalır.
- Perakendecinin satış hedeflerini kolaylıkla ve gerçek olacak şekilde oluşturmaya imkan verir.
- Maliyet hesabı için gerekli veriler çabuk ve doğru olacak şekilde elde edilebilir.
- Stok yönetimi, ürün özelliğinin kaybedilmesine (insan ya da çevre kaynaklı) karşı gerek duyulan uygulama, planlama ve denetim faaliyetlerinin aktif bir biçimde yönetilmesine olanak sağlar.

Envanter kalemlerinin sayısı büyük olduğunda, yönetim (örneğin, tedarik, taşıma ve depolama) zorlaşır. Bu nedenle, etkin bir yönetim sağlamak için, süreci basitleştirmek adına ürünlerin gruplandırılması çoğu zaman gereklidir. Aynı gruptaki öğeler benzer özelliklere sahipse, yönetim etkin hale gelir. Bu yüzden yalnızca kullanım değeri kriterine dayanan geleneksel ABC analizinin belki etkili ancak mutlaka verimli olmadığı açıktır. Bir öğenin tüm kritikliği yalnızca Çok Kriterli Envanter Kontrolü (ÇKEK) yaklaşımlarıyla elde edilebilir (Lolli vd., 2014).

Lolli ve vd. (2014), yaptıkları çalışmada k-ortalamalar algoritmasına ve AHP'ye (Analitik Hiyerarşi Prosesi) dayanan iki yeni hibrit ÇKEK yöntemi önermiştir: AHP-K ve AHP-K-veto. AHP-K, AHP'den elde edilen global önceliklere k-ortalamalar algoritmasını uygular. AHP-K, öğeleri benzerliklerine göre gruplayan ve kompakt ve iyi ayrılmış kümeler üreten k-ortalamalar algoritmasını kullanarak sınıflar oluşturur. Böylece benzer ürünlerin envanter yönetimini kolaylaştırır. AHP-K yöntemindeki eksikliklerden biri

olan tek bir kriterdeki gizli skorlamaları önlemek için AHP-K-Veto yöntemini geliştirmişlerdir.

Ishizaka vd. (2018), ABC envanter sınıflandırma problemini ele alıp Veri Zarflama Analizi'nin (Data Envelopment Analysis-DEA) bir uzantısı olan DEASort yöntemini önermişlerdir. Gerçekçi olmayan sınıflandırmadan kaçınmak için karar vericilerin uzmanlığı sürece dahil edilmiş, her sınıf için tipik ögelere örnekler verilmiş ve AHP ile kriterlerin ağırlıkları belirlenmiştir.

Tez çalışmasında envanter gruplama için hiyerarşik kümeleme, k- ortalamalar algoritması ve hibrit (hiyerarşik ve k-ortalamalar) kümeleme yöntemleri incelenmiştir:

- Hiyerarşik kümeleme: Noor ve Shuib (2015), envanter dağıtım problemi için hiyerarşik kümeleme yönetimini kullanmışlardır.

Hiyerarşik kümeleme yöntemleri, başlangıçta tüm verilerin tek bir küme olarak kabul edilmesi ve daha sonra bu kümenin kademeli bir şekilde alt kümelere ayrılması veya başlangıçta her biri ayrı birer küme olarak ele alınan verilerin kademeli olarak küme şeklinde birleştirilmesi mantığına dayanır. Hiyerarşik kümeleme yöntemleri birleştirici hiyerarşik kümeleme ve ayrıştırıcı hiyerarşik kümeleme yöntemleri adı altında iki gruba ayrılmaktadır. Birleştirici hiyerarşik kümeleme yönteminde başlangıçta her veri ayrı bir küme olarak ele alınır ve daha sonra yinelemeli bir şekilde bütün verileri kapsayan bir küme oluşturulana kadar her bir veri veya veri kümesinin kendisine en yakın olan veri veya veri kümesiyle birleştirilir. Ayrıştırıcı hiyerarşik kümeleme yönteminde ise, başlangıçta tüm veriler bütün bir küme olarak ele alınır, daha sonra tüm veriler için birbirinden bağımsız ayrı birer küme elde edinceye kadar, her bir veri ya da veri kümesi kendisinden en uzak olan veri ya da veri kümesinden çıkarılıp, yeni bir küme oluşturacak biçimde bölünür (Yeşilbudak vd., 2010).

Hiyerarşik kümelemede birleştirilmek üzere birbirine en çok benzeyen kümeleri belirlemek için kümeler arasında uzaklık hesaplanır. İki küme arasındaki uzaklığı hesaplamak için tek bağlantı, tam bağlantı, ortalama bağlantı, merkez bağlantı, komşu birleştirme yöntemi, Ward yöntemi ve ayarlı tam bağlantı yöntemlerinden yararlanılmaktadır (Birant, 2019). Bu yöntemler aşağıda açıklanmaktadır:

- Tek bağlantı: k_1 ve k_2 kümesi arasındaki uzaklık (U), bu iki kümenin birbirine en yakın bulunan iki elemanı (x_1 ve x_2) arasındaki uzaklıktır. Bu uzaklık Eşitlik (2.1) yardımıyla hesaplanmaktadır.

$$U(k_1, k_2) = \min_{x_1 \in k_1, x_2 \in k_2} U(x_1, x_2) \quad (2.1)$$

- Tam bağlantı: k_1 ve k_2 kümesi arasındaki uzaklık (U), bu iki kümenin birbirine en uzak olan iki elemanı (x_1 ve x_2) arasındaki uzaklıktır. Bu uzaklık Eşitlik (2.2) ile hesaplanmaktadır.

$$U(k_1, k_2) = \max_{x_1 \in k_1, x_2 \in k_2} U(x_1, x_2) \quad (2.2)$$

- Ortalama bağlantı: k_1 kümesindeki veriler ile k_2 kümesindeki verilerin arasındaki tüm uzaklıklar hesaplanır ve bu uzaklıkların ortalaması iki küme arasındaki uzaklık (U) olarak kabul edilir. Bu uzaklık Eşitlik (2.3) ile hesaplanmaktadır.

$$U(k_1, k_2) = \frac{1}{|k_1|} \frac{1}{|k_2|} \sum_{x_1 \in k_1} \sum_{x_2 \in k_2} U(x_1, x_2) \quad (2.3)$$

- Merkez bağlantı: Birinci küme (k_1) ve ikinci küme (k_2) merkezleri (p elemanlı ortalama vektör) arasındaki uzaklık hesaplanır. Bu hesaplama Eşitlik (2.4) yardımıyla yapılmaktadır.

$$U(k_1, k_2) = U\left(\left(\frac{1}{k_1} \sum_{x \in k_1} \vec{x}\right), \left(\frac{1}{k_2} \sum_{x \in k_2} \vec{x}\right)\right) \quad (2.4)$$

- Komşu birleştirme yöntemi: Bu yöntemde diğer yöntemlerden farklı olarak veriler arasındaki tüm ikili uzaklıklar hesaplanır. Uzaklıkları minimum olan iki veri seçilip uzaklık matrisinden silinir. Silinen verileri temsil eden yeni ana verinin diğer verilere uzaklıkları hesaplanır ve uzaklık matrisine bu değer eklenir. İki eleman kalıncaya kadar bu işlemlere tekrarlı bir şekilde devam edilir.
- Ward yöntemi: Bir kümenin merkezinde yer alan verinin, kümede yer alan diğer verilerden ortalama uzaklığını dikkate almaktadır. Toplam küme içi varyansı minimize etmeyi hedeflemektedir. Bu amaçla, küme içi kareli sapmalardan yararlanarak hata kareler toplamını hesaplamaktadır. Bu yöntem için Eşitlik (2.5) kullanılmaktadır.

$$TU_{k_1 \cup k_2} = \sum_{x \in k_1 \cup k_2} U(x, \mu_{k_1 \cup k_2})^2 \quad (2.5)$$

- Ayarlı tam bağlantı: k_1 ve k_2 kümelerinin birbirine en uzak olan iki verisinin (x_1 ve x_2) arasındaki uzaklıktan, kümelerin küme içi uzaklık (KİU) değerlerinden büyük olanının çıkartılması ile elde edilir. Bu uzaklık Eşitlik (2.6) ile hesaplanmaktadır.

$$U(k_1, k_2) = \max_{x_1 \in k_1, x_2 \in k_2} U(x_1, x_2) - \max_{i \in \{1,2\}} KIU(k_i) \quad (2.6)$$

Tez çalışmasında hiyerarşik kümeleme yöntemlerinden birleştirici hiyerarşik yöntem kullanılmıştır. İki küme arasındaki uzaklığın hesaplanmasında ortalama bağlantı yöntemi kullanılmıştır.

- k-ortalamlar algoritması: Az boyutlu ve büyük veri setlerinin olduğu birçok uygulamada kullanılan etkili bir algoritmadır. Kümeleme algoritmaları arasında en çok kullanılan klasik yöntemlerden biri olan k-ortalamlar algoritması Mac Queen tarafından geliştirilmiştir. k-ortalamlar, kümelemenin gözetimsiz öğrenme fonksiyonunu yerine getirmenin yanı sıra oldukça fazla kullanılmaktadır (Yücelen ve Baykal, 2021).

k-ortalamlar algoritmasında başlangıçta verilen k küme sayısı kadar veri setinden rastgele küme merkezi seçilir ve veriler küme merkezlerine olan uzaklıklarına göre kendisine yakın olan kümelere atanırlar. Bu işlem küme merkezlerinde değişiklik olmayıncaya kadar tekrarlanır (Çınaroğlu ve Bulut, 2018).

- Hibrit kümeleme: Kümeleme yöntemleri bazı avantaj ve dezavantajlara sahiptir. Bu nedenle çeşitli hibrit ve bulanık kümeleme yöntemleri geliştirilmiştir. Hibrit yöntemlerde, farklı kümeleme yöntemlerinin üstün oldukları taraflarının kullanılıp eksik oldukları taraflarının bertaraf edilebileceği en az iki kümeleme yönteminin birlikte kullanılmasıdır (Chen vd., 2010).

2.3. Araç Rotalama Problemi

Araç rotalama problemi (ARP) literatürde ilk defa 1959 yılında Dantzig ve Ramser tarafından ele alınmıştır (Redi vd., 2021). Araç rotalama problemi genel olarak, depodan yola çıkan aracın tüm müşterilere en kısa zamanda ve en az maliyetle hizmet götürmesine olanak sağlayan en uygun rotasının oluşturulmasıdır. Bir araç rotalama probleminin çözümünün oluşturulması için başlangıçta optimize edilecek olan problemin amaç fonksiyonu, müşterilerin talepleri, müşterilerin birbirleriyle ve depo arasındaki mesafeleri ve kullanılacak araç sayısı ile bu araçların kapasiteleri bilinmelidir. Klasik araç rotalama probleminde, belirli bir kapasiteye sahip olan aynı tür araçlar, merkez bir depodan hareket ederek talepleri önceden bilinen müşterileri ziyaret etmekte ve en sonunda depoya geri dönmektedir. Bu problemde, her müşteriye yalnızca bir araç ziyaret etmektedir. Her aracın kendine ait tek bir rotası vardır. Rotada bulunan müşterilerin talepleri toplamı araç kapasitesini aşamaz. Araçlar başlangıçta depodan çıkıp rotanın sonunda depoya geri dönmek zorundadırlar (Yücenur ve Demirel, 2011).

2.3.1 Araç Rotalama Problemi Çeşitleri

Araç rotalama problemi çeşitleri aşağıda belirtilmiştir (Keskindür vd., 2015):

- **Kapasite Kısıtlı Araç Rotalama Problemi (KKARP):** Kapasite kısıtlı araç rotalama problemi (KKARP), ARP'nin standart versiyonudur. Coğrafi koordinatları ve talepleri önceden bilinen müşterilere hizmet vermek için araçların kapasitesini aşmayacak şekilde en az maliyetle araç rotalarının belirlenmesini sağlamaktadır. Rota boyunca her müşteri bir araç tarafından yalnızca bir kere ziyaret edilmelidir. Kullanılan araçlar özdeş ve aynı kapasiteye sahiptir. Bazı durumlarda, her araç rotası için maksimum seyahat süresini sınırlayan bir süre üst sınırı atanır.

KKARP'yi tanımlamak için bir G çizgesi kullanılır. $V = \{v_0, v_1, \dots, v_{n+1}\}$ nokta kümesi ve $A = \{(v_i, v_j) | v_i, v_j \in V, v_i \neq v_j\}$ yol kümesi olmak üzere $G = (V, A)$ bir toplam çizge olsun. v_0 ve v_{n+1} noktaları, K tane homojen aracın bulunduğu depoyu, kalan köşeler ise müşterileri ifade etmektedir. Negatif olmayan maliyet

$c_{v_i v_j}$, her bir (v_i, v_j) yolu ile ilişkilidir ve v_i noktasından v_j noktasına seyahat mesafesini verir. Her müşterinin dağıtım talebi q_i ' dir. KKARP, aşağıdaki gibi bir dizi en düşük maliyetli araç rotasının belirlenmesinden oluşur:

- Her araç rota depodan hareket ederek başlar. Rotanın sonunda depoya geri döner.
- Rotada bulunan her müşteri bir araç tarafından yalnızca bir kez ziyaret edilir.
- Herhangi bir araca atanan müşterilerin toplam talebi, araç kapasitesini Q aşmamalıdır.
- Süre kısıtlı KKARP için, her rotanın süresi bir üst sınır D 'yi geçmemelidir. Böyle bir durumda her v_i ($i > 0$) müşterisi belirli bir servis süresi s_i gerektirir. Süre, bu rotadaki tüm müşterilere hizmet vermek için toplam seyahat süresine ve toplam hizmet süresine eşittir. Burada v_i ve v_j arasındaki seyahat süresi bunların arasındaki $c_{v_i v_j}$ Öklid uzaklığına eşittir (Chen vd., 2010).
- Mesafe Kısıtlı Araç Rotalama Problemi (MKARP): Mesafe kısıtlı araç rotalama probleminde, araçların rota boyunca katedeceği mesafe sınırlandırılmaktadır. Bu durum araçların belirlenen seyahat süresinden daha uzun süre çalışmasını önlemek ve meydana gelen olumsuz bir durumdan dolayı gecikme yaşanmaması amacıyla kullanılmalıdır (Oropeza vd., 2012).
- Zaman Pencere Araç Rotalama Problemi: Zaman pencere araç rotalama probleminde her müşteriyi ziyaret etmek için gerekli bir zaman aralığı kısıtı vardır. Bu problem çeşidinde rotası oluşturulan araç, her bir müşteriye belli bir zaman periyodunun içinde uğramak zorundadır (Düzakın ve Demircioğlu, 2009).
- Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi (EZTDARP): Eş zamanlı topla dağıt araç rotalama problemine araç, müşterilerin talepleri doğrultusunda ürün dağıtımını yaparken eş zamanlı olarak müşterilerden ürün toplama operasyonunu da yapmaktadır (Cömert vd., 2019).

EZTDARP’de depodan çıkış yapan araç, rotası üzerinde yer alan müşterilere ürün dağıtırken aynı anda da toplar. Araç rotası sırasında kapasite kontrolü yapılarak müşterilerin tüm taleplerinin karşılanması ve müşterilerden ürün toplama operasyonunun gerçekleştirilmesi sağlanır. Her müşteriye bir araç yalnızca bir kez ziyaret edebilmektedir. Aynı aracın toplama ve dağıtım operasyonlarını eş zamanlı yapmasından dolayı EZTDARP, toplama ve dağıtım faaliyetlerin ayrı zamanlarda yapıldığı diğer araç rotalama problemlerine oranla daha maliyetli ve daha verimlidir (Zacharias vd., 2009).

- **Önce Dağıt Sonra Topla Araç Rotalama Problemi (ÖDSTARP):** Önce dağıt sonra topla araç problemlerinde, dağıtım araçları depodan çıkarak önce müşterilerin talep ettiği ürünlerin tamamı dağıtırlar. Daha sonra müşterilerden depoya gönderilecek ürünlerin toplanması operasyonunu gerçekleştirirler. Bu nedenle araçlar müşterilere birden çok kere uğrar (Küçüköğlu ve Öztürk, 2010).

Çoğu araştırmacı müşterileri mal alan müşteriler ve mal gönderen müşteriler olmak üzere iki gruba ayrılabilceği varsayımında bulunur. Ayrıca araçlar ancak tüm yüklerini teslim ettikten sonra malları alabilirler (Nagy ve Salhi, 2005).

Bu problem türünde rotalar oluşturulurken, araç kapasitesinin aşılması için depoda araca yüklenecek ürün miktarı ve müşterilerden toplanacak ürün miktarlarının toplamının bilinmesi gerekmektedir. Oluşturulan rotada hep dağıtım hem toplama işlemi yapılacak müşteri varsa toplama işlemi yapılacak müşteriye dağıtım işlemi bittikten sonra uğranmalıdır (Keskintürk vd., 2015).

- **Ayrık Yükleme ve Zaman Pencere Araç Rotalama Problemi (ZARP):** Zaman pencere Araç rotalama probleminin bir çeşidi olarak incelenmektedir. Bu problem hakkında yapılmış çok sayıda çalışma olmamasına rağmen son zamanlarda araç rotalama problemi için ayrık zamanlara göre karar verme ile ilgili birkaç çalışma mevcuttur. Bu problem türünde araçlar müşterilere belirlenmiş zaman aralıklarında dağıtım faaliyeti yapmaktadır. Burada belirlenmiş zaman aralıklarının sayısı arttıkça teslimat sayıları artmakta bu durum da zaman kısıtı üzerindeki baskının artmasına yol açmaktadır (Keskintürk vd., 2015).

- Geri Toplamalı Araç Rotalama Problemi (GTARP): Müşterilerin ellerindeki ürünleri geri gönderdiği veya talepte bulunduğu araç rotalama problemi türüdür. Bu yüzden toplama dağıtım araç rotalama problemlerinde müşterilerin geri gönderecekleri ürünlerin araç kapasitesini aşmaması gerekmektedir. Burada toplama işlemine başlayabilmek için önce dağıtım işlemlerinin tamamlanması gerekmektedir. Bunun sebebi araçlara yüklemenin arkadan yapılmasıdır. Çünkü dağıtım birimlerine hizmet verildiği sırada araçlardaki ürünlerin yerleşiminin düzenlenmesi ekonomik ve verimli olmayacaktır. Araçların müşterilere dağıtacağı ve toplayacağı ürün miktarları sabittir ve önceden bilinmektedir (Şeker, 2007).
- Periyodik Araç Rotalama Problemi (PARP) : Periyodik araç rotalama probleminde ilk olarak dönemsel plan yapılır. Müşteriler bu planlanan dönemde birden fazla kez hizmet görebilmektedirler. Müşterilere yapılacak servis sayısı müşterilerin talep miktarlarına, stok alanlarına göre değişmektedir. Eğer bir müşterinin talep miktarı çok fazla ise az miktarda talebi olan müşteriye göre ya da stoklama alanı küçük ise büyük olan müşteriye göre daha fazla ziyaret edilecektir. Bu problem sınıfı bakkaliye, içecek endüstrisi, atık toplama gibi alanlarda ortaya çıkmaktadır (Dursun, 2009).

2.3.2. Araç Rotalama Problemi Çözüm Yöntemleri

Araç Rotalama Problemlerinin çözümü için literatürde birçok yöntem geliştirilmiştir. Bu yöntemleri kesin, klasik sezgiseller ve meta sezgiseller olmak üzere üç ana gruba ayırmak mümkündür. Kesin çözüm yöntemleri ile optimal çözüm sağlanırken, klasik ve meta sezgisel yöntemler ile optimuma yakın çözümler kısa sürede elde edilebilmektedir (Keskintürk vd., 2015).

2.3.2.1. Araç Rotalama Problemi İçin Önerilen Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri, optimum sonuca götüren yöntemler olup literatürde yaygın olarak kullanılan yöntemler aşağıda verilmiştir (Keskintürk vd., 2015):

- Kesme Düzlemi: Bu yöntem dal sınır algoritmasına alternatif bir yöntem olması amacıyla oluşturulmuştur. Tamsayılı doğrusal programlama problemi için sınırların tam sayı olması gerekmektedir. Eğer sınırlar tam sayı değilse ilk olarak sınırların tam sayı olması sağlanır. Bunun için sınırların değiştirilmesi durumu ortaya çıkabilir. Daha sonra problem için en iyi çözüm tablosu bulunur. En iyi çözümün tam sayı olduğu noktada problemin çözümüne ulaşılmış demektir. (Alparslan, 2015).

Ruiz vd. (2020), açık araç rotalama problemi için kesme düzlemi ve karışık tam sayılı programlamadan oluşan hibrit bir yöntem önermişlerdir.

- Dal ve Sınır Algoritması: Dal ve sınır algoritması NP-zor optimizasyon problemlerine kesin çözümler üretmek için yaygın olarak kullanılan bir yöntemdir. Bu yöntem, alt problemler olarak adlandırılan kısmi çözümleri bir ağaç yapısında depolayarak, söz konusu problemin tüm olası çözümlerini örtük olarak sıralar. Ağaçtaki keşfedilmemiş düğümler, çözüm alanını yinelemeli olarak çözülebilen (dallanma) daha küçük bölgelere bölerek çocukları oluşturur ve kurallar, arama alanının kanıtlanabilir şekilde alt optimal olan (sınır) bölgelerinin üstünde kullanılır. Ağacın tamamı keşfedildikten sonra, aramada bulunan en iyi çözüm geri gelir. Algoritmanın performansı üzerinde önemli etkileri olabilecek üç bileşen vardır. Bu bileşenler çözüm stratejisi (ağaçtaki alt problemlerin keşfedilme sırası), dallanma stratejisi (ağaçta yeni alt problemler üretmek için çözüm alanının nasıl bölündüğü) ve budama kurallarıdır (keşfetmeyi engelleyen kurallar) (Morrison vd., 2016).
- Dal ve Kesme Algoritması: Tamsayılı programlama problemleri için kullanılan etkili bir yöntem olup kesme düzlemi algoritması ve dal-sınır yöntemlerinin bir karmasıdır. Bu yöntemin başlangıcı tamsayılı programlama modelinin doğrusal programlama ile yapılmış olan çözümüdür. Yöntem bu yönüyle diğer tamsayılı programlama algoritmalarıyla (Dal-sınır, Kesme düzlemi) benzerdir. Bir tam

sayılı programlama modelini sadece kesme düzlemi yöntemi ile verimli bir şekilde çözmek mümkün olmadığı için diğer optimum çözümleri bulabilmek için dallandırma yapmak gerekli olmaktadır. Kesme düzlemi algoritmasının uygulanmasıyla dal-sınır yaklaşımı hızlandırılabilir. Dallandırma yapılmadan kesme ekleyebilmekle beraber ağacın tüm düğümlerinin çözüm aşamasında da kesmeler kullanılabilir (Mitchel, 1998).

- **Dinamik Programlama:** Dinamik programlama, birden çok karar verme operasyonunun optimizasyonunu sağlayan matematiksel işlemler bütünüdür. Genel olarak dinamik programlama, problemin veya problemin bir bölümünün parçalara ayrılması ve bu parçaların çözülerek elde edilen sonuçların saklanması prensibine dayanmaktadır. Bu çözümler, ihtiyaç halinde, tekrar çözülmek yerine canlandırılarak problemin genel çözümüne dahil edilir. Böylece en son çözüme ulaşılmaktadır (Yılmaz, 2008).

2.3.2.2. Araç Rotalama Problemi İçin Önerilen Klasik Sezgisel Çözüm Yöntemleri

Araç rotalama problemleri çözüm yöntemleri genel olarak tur kurucu sezgiseller, tur geliştirici sezgiseller ve iki aşamalı metodlar olmak üzere üç ana sınıfa ayrılmaktadır (Eryavuz ve Gencer, 2001).

- **Tur Kurucu Sezgiseller:** Çözüme başlangıçta mümkün olmayan atamalar yapılır, uygun çözüme ulaşmak için her seferinde iki düğüm arasına bir dal ekleyerek devam ederler. Dal ekleme işlemi yapılırken araç kapasitesini aşıp aşılmadığı kontrol edilir. Eklenecek dal seçilirken bazı maliyet tasarrufları göz önüne alınır. Tur kurucu sezgiseller arasında Clarke ve Wright' ın, Dantzig ve Ramser'den esinlenerek ürettikleri tasarruf algoritması en çok kullanılmaktadır. Tasarruf algoritması üzerinden çeşitli algoritmalar üretilmiştir (Eryavuz ve Gencer, 2001). Bu algoritmalara ek olarak en yakın komşu (nearest neighbour) ve yerleştirme (insertion) yöntemleri de bu tur kurucu sezgisel yöntemlere dahildir (Emel ve Taşkın, 2005).

- **Tur Geliştirici Sezgiseller:** Algoritma başlangıcı olarak bir mümkün çözümü alırlar ve o çözümü geliştirerek ilerlerler. Olası dal kombinasyonları, her bir iterasyonda değiştirilir ve bu değişimin mümkün olan çözümü elde edip etmediği, maliyette bir değişim yaratıp yaratmadığı kontrol edilir. Tur geliştirme sezgiselleri Seyyar Satıcı Problemi (SSP) sezgiseline dayanır (Eryavuz ve Gencer, 2001).
- **İki Aşamalı Metotlar:** İki aşamalı metodun ilk aşamasında araç kapasitesini geçmeyecek şekilde düğümler rotaya dahil edilir. İkinci aşamasında ise seyyar satıcı problemi için kullanılan sezgisel yöntemlerden yararlanarak araçların rotaları oluşturulur. İki aşamalı metodlara örnek olarak önce grupta-sonra rotalama türündeki araç rotalama algoritmaları verilebilir (Eryavuz ve Gencer, 2001).

2.3.2.3. Araç Rotalama Problemi İçin Önerilen Meta Sezgisel Çözüm Yöntemleri

Literatürde araç rotalama problem çözümünde sıklıkla tercih edilen metasezgisel çözüm yöntemleri aşağıda belirtilmiştir (Keskintürk vd., 2015):

- **Tabu Arama:** Tabu arama algoritmasında bir başlangıç çözümü üretildikten sonra bu çözümü belirlenmiş değişim yöntemleri kullanılarak uygun çözümü elde edilmeye çalışılır (Chelouah ve Siarry, 2000). Tabu arama algoritmasında kullanılacak başlıca değişim teknikleri olarak eleman eklemek ya da çıkarmak, değişkenin değerini değiştirme, iki çözüm arasında değerleri birbiri ile değiştirmek gösterilebilir. Tabu arama algoritmasının esnek yapısı sayesinde yeni çözümler türetmek ve uygun çözümün elde edilmesi daha etkin olmaktadır. Bu esneklik sayesinde değişik problem türlerine uygulanabilmektedir. Tabu arama algoritmasının adımları aşağıdaki belirtilmiştir (Gürbüz, 2015):
1. Başlangıç çözümünü belirle. Başlangıç çözümünü mevcut çözüm ve en iyi çözüm olarak hafızaya kaydet.
 2. Belirlenmiş değiştirme fonksiyonu ile geçiş yapılabilecek aday komşu çözümleri bul.

- a. Tabu olmayan veya tabu olsa dahi tabu yıkma kriterlerini sağlayan bir komşu çözümü seç.
- b. Tabu olarak, mevcut çözümden yeni çözüme geçişi belirle.
- c. Yeni çözüm o süreye kadarki en iyi çözüm ise yeni çözümü uygun çözüm olarak belirle.

3. Durdurma kriteri sağlanana kadar adım 2'yi tekrar et.

- Genetik Algoritma: Genetik algoritmada, çözümü oluşturan her bir parçaya gen denilmektedir. Genetik algoritma, temel olarak, aynı genlerin değişik operatörler yoluyla çeşitli dizilişlerini sağlayarak değişik bireyler elde etmektedir. Elde edilen bu bireylerden ise en iyi olanı seçmektedir (Gözüdeli ve Akcayol, 2007).

Genetik algoritmanın adımları aşağıda belirtilmiştir (Alparslan, 2015):

Adım 1: Başlangıç popülasyonu oluşturulur.

Adım 2: Popülasyondaki her bir bireyin uygunluk değeri hesaplanır.

Adım 3: Popülasyondaki bireyler, çarpazlama yapılmak amacıyla birey seçim yöntemine göre seçilir.

Adım 4: Seçilen bireyler, çarpazlama olasılığına uygun olarak çarpazlanır ve yeni bireyler oluşturulur.

Adım 5: Yeni bireylere mutasyon olasılığına göre mutasyon uygulanır .

Adım 6: Oluşturulan bu yeni bireyler popülasyonun mevcut bireyleridir.

Genetik algoritmalar çözüm arama uzayının karmaşık ve büyük olduğu problemlerde kullanılmaya uygun bir yöntemdir. Problemin uygun çözümünü bulmayı garanti etmezler ancak kabul edilebilir iyi çözümleri makul bir süre zarfında bulabilirler (Alparslan, 2015).

- Tavlama Benzetimi: Tavlama benzetimi algoritması, mantığını metallerin tavlama sürecinden esinlenerek almış belirli bir maliyet fonksiyonunun küresel en iyiye yaklaşmak için tasarlanmış sezgisel bir optimizasyon yöntemidir. Tavlama benzetiminin temel prensibi katıların ısıtılması ve ısıtmanın ardından yavaşça

soğutulmasına dayanmaktadır. Isıtılan katıların sıcaklığı azaldığında, katının iç parçacıkları her sıcaklıkta bir denge durumuna ulaşmaktadır. Tavlama benzetimi algoritması, ısı yükseldikçe, en iyi yerel optimuma ulaşmak için komşu bölgeye gidecektir. Yavaşça soğumaya başladığında ise en iyi yerel optimum noktasında durmaya çalışacaktır (Metin, 2021).

- Karınca Kolonisi: Gerçek karıncaların yiyecek arama sürecinden yola çıkarak geliştirilen karınca kolonisi algoritması ilk olarak 1992 yılında yayımlanan bir doktora tezinde gezgin satıcı problemi üzerinde uygulanarak literatüre kazandırılmıştır. Karınca kolonisi algoritmasında karıncalar, başlangıçta rasgele yollar üzerinden yiyecek aramaya başlar ve bir turu tamamladıktan sonra yuvalarına dönmektedirler. Tur sırasında karıncalar izledikleri yollar üzerine feromon salgılamaktadırlar. İlk iterasyon bittikten sonra ikinci iterasyonda karıncaların bir kısmı daha fazla feromon bulunan yolları kullanacaklardır. Daha kısa yollarda daha fazla feromon bulunmaktadır. Her iterasyon bittiğinde yollarda bulunan feromon salgısı miktarları kullanıcının belirlediği bir oranda azaltılarak güncellenmektedir. Bu sayede çok kötü çözümler engellendiği gibi, feromon miktarının daha fazla olduğu rotalar da en iyi çözüm olarak kabul edilmemektedir. Böylece algoritmanın yerel optimumlara takılmasına engel olunmaktadır. Global feromon güncellemesi sırasında ise en iyi çözümü sağlayan rota ek feromon eklemesi yapılarak yapay karıncalar için daha avantajlı olmaktadır.(Kuzu vd., 2014).
- Yapay Arı Kolonisi: Karaboğa (2005), arı kolonilerinin besin arama sürecinden yola çıkarak yapay arı kolonisi algoritmasını geliştirmiştir. Yapay arı kolonisi algoritmasında, yemek yerleri olası çözüm noktalarını, nektar miktarları ise amaç fonksiyonuna karşılık gelen çözüm değerlerini göstermektedir. Algoritmada modellenen kâşif arı, gözcü arı ve işçi arı olmak üzere üç tür arı davranışı olmakla birlikte, her biri algoritma ayrı bir aşamasında çalışır.

Algoritma ilk olarak kâşif arıların yiyecek yer aramasıyla başlar. Keşif arıları yiyecek kaynaklarını bulduktan sonra işçi arılara dönüşürler. Algoritmadaki görevli arıların sayısı ile toplam yiyecek kaynağının sayısı birbirine eşittir. Gözcü arıların sayısı ile işçi arıların sayısı aynıdır

Belirlenen besin noktalarından elde edilen nektarların kalitesini belirlemek için problemin amaç fonksiyonu kullanılır. Bir minimizasyon problemi için hesaplanan en düşük maliyet değeri genel minimum olarak girilirken, bu çözüme ait yemek yeri de koloninin ortak hafızasına genel çözüm olarak girilir (Yılmaz vd., 2020).

Kaşif arılar başlangıçta rassal olarak yiyecek aramaktadırlar. Yiyecek kaynağını tespit eden kaşif arı artık görev sahibi bir arı haline gelmiş ve kovana nektar götürmeye başlamıştır. Görevli arı kovana nektarını taşıırken dans alanında bulunan gözücü arılara dans ederek yem taşıdığı kaynaklar ilgili bilgiyi aktarır. Kaşif arının keşfettiği kaynaktaki nektar tükendiğinde kaşif arı gözücü arıya dönüşecektir (Küçüksille ve Tokmak, 2011).

Yılmaz vd. (2020), daha önce birçok çalışmada ele alınmış olan Hanoi şehri su dağıtım şebekesi, Alperovits ve Shamir şebekesi ve bir yenileme projesi olan New York şehri su dağıtım şebekesi üzerinde yapay arı kolonisi yöntemini kullanarak maliyet optimizasyonu yapmışlardır.

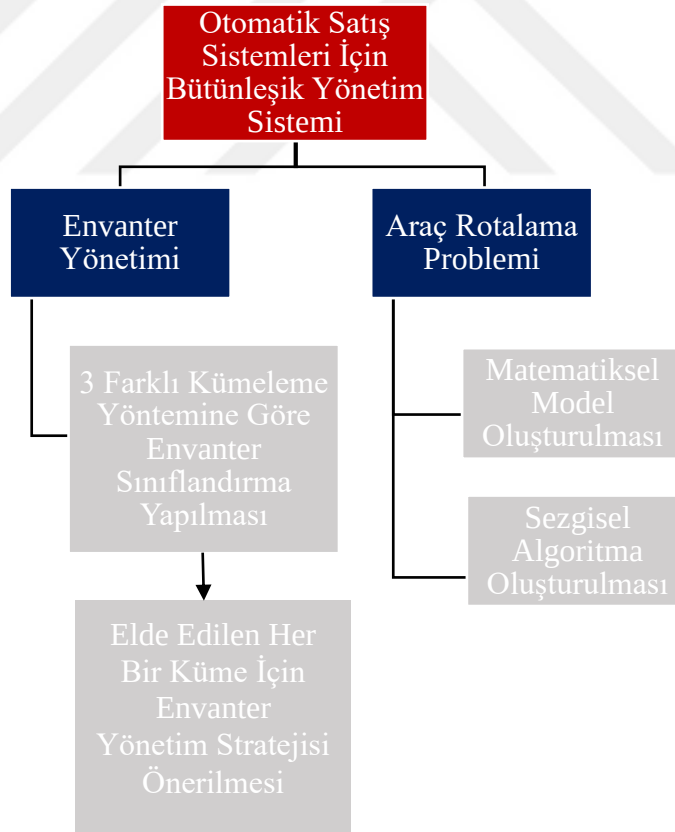
- Parçacık Sürü Optimizasyonu: Parçacık sürü optimizasyonu, genetik algoritmalar gibi hesaplama yöntemleriyle birçok benzerliğe sahiptir. Yöntem rastgele çözümler barındıran bir popülasyonla başlatılır ve nesilleri güncelleyerek en uygun çözümü bulmaya çalışır. Parçacık sürü optimizasyonunda parçacık olarak adlandırılan olası muhtemel çözümler, o andaki optimum parçacığı izleyerek problem uzayında dolaşırlar. Parçacık sürü optimizasyonunun türev bilgisine ihtiyaç duymaması onun klasik optimizasyon tekniklerinden en önemli farklıdır. Bu özellik sayesinde birçok problemin çözümü için gerekli olan karmaşık işlem yükünün azaltılmış olmaktadır. Parametre sayısının düşük olması parçacık sürü optimizasyonunu uygulamayı kolaylaştırmaktadır. (Çevik ve Koçer, 2013).

Demirtaş ve Özdemir (2017), dinamik araç rotalama problemi için parçacık sürü optimizasyonuna dayalı bir çözüm önermişlerdir. Önerilen yöntemi dağıtım yapan bir firmada uygulamışlardır.

3. MATERYAL VE YÖNTEM

Tez çalışması kapsamında otomatik satış makinelerinin yönetimini sağlayan bir firmanın yönetim sisteminin iyileştirilmesi amaçlanmıştır. Bu sebeple etkili bir envanter yönetimi yapılabilmesi için firmanın deposundaki envanter kalemleri için ABC sınıflandırmasına dayalı envanter kümeleme yapılmıştır. Her bir küme için ekonomik sipariş miktarları ve emniyet stoğu miktarları önerilmiştir.

Tez çalışmasının bir diğer aşamasında ise otomatik satış makinelerine hizmet veren personelin günlük makine ziyaretini planlayan araç rotası oluşturulmuştur. Bunun için ilk olarak matematiksel model oluşturulmuştur. Oluşturulan matematiksel model ele alınan problemi çözmede yeterli olmadığından araç rotama problemi için en yakın komşu algoritmasına dayanan sezgisel algoritma oluşturulmuştur. Yukarıda açıklanan çalışmalar Şekil 3.1’deki akış diyagramında özetlenmiştir.



Şekil 3.1. Tez kapsamında yapılan çalışmalar

3.1. Envanter Sınıflandırma ve Yönetimi

Nihai tüketiciye satılmak için perakendecinin elinde tuttuğu mallara kısaca stok denir. Perakende işletmeleri amaçlarına ulaşabilmek için uygun miktarda ürünü elinde bulundurup tüketici ihtiyaçlarına cevap vermektedirler. Gereğinden daha fazla miktarda stok bulundurmak, ürün çeşitliğini ve kârlılığını etkilediği gibi az miktarda stok bulundurmak da satış ve müşteri kaybı yaratacaktır (Ustaahmetoğlu, 2013).

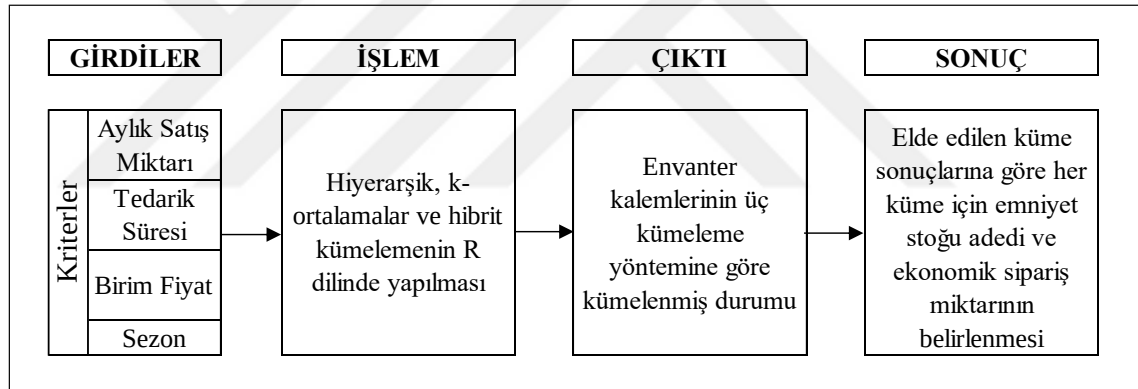
Envanter yönetimi, günümüzde artan rekabetle birlikte işletmeler için değer kazanan kavramlardan biri haline gelmiştir. İşletmeler dengeli bir envanter yönetimiyle maliyetlerini azaltırken müşteri ihtiyaçlarına da zamanında cevap verebilmektedir. İşletmelerin dengeli bir envanter yönetimi gerçekleştirebilmeleri için stokta yer alan ürünleri sınıflandırmaları gerekmektedir. ABC analizi bu sınıflandırma için kullanılan yöntemlerden biridir. ABC analizi ürünlerin parasal değerine göre sınıflandırma yapmaktadır (Dursun ve Gürgen, 2020). Stok kontrolünde ABC yöntemi, stok kalemlerinin önem derecelerine göre sınıflandırılmasından oluşur. Bu yöntemde stokları meydana getiren ürünler parasal değerlerine göre tüm ürünler içindeki oranlarına göre A grubu, B grubu ve C grubu stoklar olmak üzere üç ana gruba ayrılır (Özdemir ve Özveri, 2004).

Envanter birimlerinin etkin bir şekilde kontrol edilmesi ve bunlara uygun sipariş politikalarının belirlenmesi için en yaygın üretim ve stok kontrol tekniklerinden biri olan çok kriterli ABC sınıflandırması kullanılmaktadır. Bu sınıflandırmada, ürünlerin parasal değerlerinin yanı sıra farklı kriterler de dikkate alınmakta ve envanter birimleri, önceliklerine göre farklı sipariş politikaları ile üç sınıfa ayrılmaktadır (Hadi-Vencheh ve Mohamadghasemi, 2011).

Ferraira vd. (2018), envanter yönetimi için envanter birimlerini üç sınıfa ayırırken kritiklik, talep, birim fiyat, teslim zamanı ve potansiyel tedarikçi sayısı kriterlerini kullanmışlardır.

Özdemir ve Özveri (2004), yaptıkları çalışmada envanter birimlerine ABC sınıflandırması uygularken fiyat, talep, teslim zamanı, ikame olanakları ve kritiklik kriterlerini göz önüne almışlardır.

Bu tez çalışmasında otomatik satış makinelerine ürün beslemesi yapan bir firma ele alınmıştır. Tez çalışması kapsamında ürün gruplarını kümelere ayırmak için uzman görüşlerine dayanılarak ve literatürde de sıklıkla kullanılan aylık satış miktarı, tedarik süresi, birim fiyat, sezon (yaz/kış) kriterleri ele alınmıştır. Kriter ağırlıkları eşit olarak kabul edilmiştir. Firmanın deposundaki envanterlerini etkili bir şekilde yönetebilmesi için aylık satış miktarı, tedarik süresi, birim fiyat ve sezon kriterlerine göre birbirine benzer ürünler kümelenecek iki ve üç adet ürün grubu oluşturulmuştur. Birbirine benzer ürün grupları için ortak envanter yönetim stratejileri önerilmiştir. Ürünler gruplandırılırken tezin kaynak özetleri kısmında da anlatıldığı gibi literatürde sıkça kullanılan k-ortalamlar algoritması, hiyerarşik kümeleme ve hibrit kümeleme (k-ortalamlar ve hiyerarşik kümelemenin bir arada olduğu durum) yöntemleri kullanılmıştır. Bahsedilen envanter kümeleme ve yönetim süreci Şekil 3.2’de gösterilmiştir.



Şekil 3.2. Envanter kümeleme ve yönetim süreci

Firmadan 59 adet ürünün aylık satış miktarı, tedarik süresi, birim fiyat ve sezon kriterlerine ait bilgiler elde edilmiştir. Kriter bilgileri elde edilen ürünlere hiyerarşik, k-ortalamlar ve hibrit kümeleme yöntemleri uygulanmıştır. Bu yöntemler iki ve üç küme elde etmek amacıyla uygulanmıştır. Kümeleme sonucunda elde edilen ürün gruplarının envanter yönetimi için emniyet stoğu adetleri ve ekonomik sipariş miktarları belirlenmiştir.

Kümeleme işleminde kullanılan yöntemler aşağıda açıklanmaktadır:

- k-ortalamlar algoritması: k-ortalamlar, uzun yıllardır sıkça kullanılan bölümleyici yöntemlerden biridir. Nesne sınıflandırma, görüntü bölümlenme, veri madenciliği, makine öğrenmesi gibi bilişim uygulamaları yanında iktisat, müşteri yönetimi, pazarlama, biyoinformatik ve mühendislik araştırmaları gibi birçok branşta en çok kullanılan yöntemleri arasındadır. k-ortalamlar aşağıdaki eşitlik (3.1) ile gösterilen amaç fonksiyonunu minimize etmeyi amaçlayan bölümleyici bir kümeleme algoritmasıdır:

$$J_{Ko}(\mathbf{X}; \mathbf{V}) = \sum_{i=1}^n \sum_{j=1}^k D_{ij}^2 \quad (3.1)$$

Eşitlik (3.1)'de k küme sayısını ve n nesne sayısını ifade etmektedir. D_{ij}^2 nesneler ile küme merkezleri arasındaki uzaklık ölçüsüdür. Kümeleme analizlerinde uzaklık yöntemi olarak genellikle Öklid uzaklıkları kullanılmakta ve $\|x_{ij} - v_j\|^2$, $1 \leq j \leq k$ olarak hesaplanmaktadır. Burada x_{ij} , j . kümedeki i . nesneyi; v_j ise j . küme merkezini ifade etmektedir. k-ortalamlar algoritmasının adımları aşağıda belirtilmiştir (Cebeci vd., 2015):

- 1) $\mathbf{X}$ veri setinden rastgele k adet küme merkezi seçilir.
- 2) Veri noktaları ile küme merkezleri arasındaki uzaklıklar hesaplanır.
- 3) Veri noktaları, merkezleri kendilerine en yakın olan kümelerle atanır.
- 4) Eşitlik (3.2) kullanılarak küme merkezleri güncellenir. Eşitlik (3.2)'de x_{ij} , j . kümedeki i . nesneyi; n_j ise j . kümedeki nesne sayısını göstermektedir.

$$v_j = \sum_{i=1}^{n_j} x_{ij} / n_j ; 1 \leq j \leq k \quad (3.2)$$

- 5) Verilerde küme değişimi olmadığında ya da birbirini izleyen iki iterasyonda hata karelerindeki artış belirlenmiş bir yaklaşma değerine eşit veya küçükse kümeleme işlemi sonlandırılır, değilse 2. adıma geçilerek işlem adımları tekrarlanır.

Tez çalışmasında algoritmanın ikinci adımı olan veri noktaları ile küme merkezleri arasındaki uzaklığı hesaplamak için Eşitlik (3.3)'de verilen Öklid uzaklığı yöntemi kullanılmıştır.

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (3.3)$$

- Hiyerarşik kümeleme: Hiyerarşik kümeleme yöntemleri uygulanırken küme sayısı önceden bilinmemektedir. Belirlenen kriterlere göre birbirine en çok benzeyen nesneler aynı kümenin içinde yer alır. Kümeleme işlemi, veri kümesi içindeki son eleman kendisine en çok benzeyen kümeye atanınca son bulur. Hiyerarşik kümeleme yönteminde birbirine benzeyen elemanların birleştirilme mantığının anlaşılabilmesi için dendogramdan faydalanılmaktadır. Kümeleme sürecinin başlangıcında her eleman bir küme olarak (ağacın dalları) kabul edilmektedir. Kümeleme sürecinin sonunda ise tüm elemanlar tek bir kümede birleştirilir (ağacın gövdesi). Hiyerarşik kümeleme yönteminin temel adımları aşağıda belirtilmiştir (Yılmaz ve Temurlenk, 2010):

1. Başlangıç olarak n tane küme alınır.
2. En yakın olan iki küme birleştirilir.
3. Küme sayısı bir azaltılarak uzaklıklar matrisi güncellenir.
4. 2 ve 3 numaralı adımlar $n-1$ kez tekrarlanır.

Tez çalışmasında hiyerarşik kümeleme yönteminde iki küme arasındaki uzaklığı hesaplamak için Eşitlik (3.4) kullanılmıştır. Birinci küme (k_1) ile ikinci küme (k_2) elemanları arasındaki bütün uzaklıklar hesaplanır ve bu uzaklıkların ortalaması iki küme arasında uzaklık (U) olarak kabul edilir (Birant, 2019):

$$U(k_1, k_2) = \frac{1}{|k_1|} \frac{1}{|k_2|} \sum_{x_1 \in k_1} \sum_{x_2 \in k_2} U(x_1, x_2) \quad (3.4)$$

- Hibrit kümeleme: Tez çalışmasında hiyerarşik algoritmanın kümeleme sonucunu k-ortalamar algoritmasının başlangıç küme merkezleri olarak alan hibrit algoritma uygulanmıştır. Bunun için ilk olarak ürünler yukarıda adımları açıklanan hiyerarşik yöntemle kümelenecek ve hiyerarşik kümeleme yönteminden elde edilen kümelerin merkezleri bulunmuştur. Daha sonra hiyerarşik kümelemekten elde edilen küme merkezleri kullanılarak k-ortalamar algoritması uygulanmıştır.

K-ortalamlar kümeleme yönteminde ilk aşama olarak k (küme sayısı) değerinin belirlenmesi gerekmektedir. Bu değer doğru olarak belirlenemediğinde yanlış sonuçlar elde edilebilir. Literatür incelendiğinde bu değeri belirleyebilmek için farklı yöntemler kullanıldığı görülmektedir. Elbow yöntemi ve Silhouette indeksi k değerini hesaplamak için literatürde sıkça karşılaşılan yöntemler arasındadır (Özari ve Demirkale, 2020). Tez çalışmasında envanter kalemleri kümeleme işlemi yapılırken veri setine uygun optimal küme sayısını belirleyebilmek için Elbow yöntemi ve Silhouette indeksi kullanılmıştır. Çalışmada küme sayısı sektör uzmanlarının da görüşü alınarak iki ve üç olarak belirlenmiştir.

Elbow yöntemi, eski bir yöntem olup en iyi küme sayısının belirlenmesi için başvuru en yaygın yöntemlerden biridir. Kümelemenin temel mantığı, birbirine benzeyen (birbirine yakın olan) elemanların aynı kümede; birbirine benzemeyen (birbirinden uzak olan) elemanların ise ayrı kümelerde bulunmasıdır. Elbow yöntemindeki bu mantık, kümeler içi uzaklıkların kareleri toplamı olarak işlemektedir. Kümeleme işlemi için k (küme sayısı) değerleri ayrı ayrı ele alınarak, her bir k sayısı için küme merkezine olan uzaklıklarının karelerinin toplamı hesaplanır.

Şekil 3.3'te küme sayısı ile küme içi uzaklıkların karelerinin toplamı arasındaki ilişkiyi gösteren grafik verilmiştir. Şekil 3.3'teki düşey eksen her bir noktanın küme merkezine olan uzaklığının karesinin toplamını göstermektedir. İyi kurgulanmış bir modelde küme içi uzaklıkların karelerinin toplamının daha az olması beklenmektedir. Şekil 3.3'de $k=3$ e kadar küme içi uzaklık karelerinin toplamı önemli ölçüde bir azalma göstermektedir ve $k=3$ den sonra ise kümeler içi uzaklıkların kareleri toplamının düşüş hızı azalmaktadır. Şekil 3.3'den $k=3$ değerinin en iyi küme sayısı olduğu anlaşılmaktadır. Çünkü daha fazla sayıda küme, küme içi uzaklıkların kareleri toplamını önemli ölçüde azaltmadığı gibi modelin yorumlanabilirliğini azaltmaktadır.



Şekil 3.3. Optimal küme sayısının seçilmesi

Elbow yöntemi kullanılırken veri setinin iyi anlaşılması gerekmektedir. Algoritma kullanılırken kümeleme aralıkları kullanıcı tarafından verileceğinden, Elbow yöntemi küme sayısını belirlemede yardımcı olabilecek etkili bir yöntem olarak kabul edilebilir (Üstünel, 2018).

Silhouette indeks değerinin hesaplanması için uygulanan yöntemde; ilk olarak herhangi k değeri için kümelere ayrılma işlemi yapıldıktan sonra veri setinde bulunan her veri için Silhouette değeri hesaplanır. Hesaplanan tüm Silhouette değerlerin ortalaması k sayısı için Silhouette indeks değerini vermektedir. Herhangi i verisi için Silhouette indeksi Eşitlik (3.5) ile hesaplanır:

$$Sil(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3.5)$$

Eşitlik (3.5)'de $a(i)$, i . verinin küme içindeki tüm verilere olan uzaklıklarının ortalamasını, $b(i)$ ise i . verinin diğer kümelerdeki tüm verilere olan uzaklıklarının ortalamalarının minimum değeridir (Özari ve Demirkale, 2020).

Silhouette indeks değerine göre kümeleme işleminin başarısı farklı kümelere ait verilerin birbirinden ne kadar uzak ve aynı kümeye ait verilerin birbirine ne kadar yakın olduğuna bakılarak ölçülür. Elde edilen sonuç $[-1, 1]$ aralığında olur. -1 kümeleme başarısının minimum ve +1 kümeleme başarısının maksimum olduğunu gösterir (Şenol ve Karacan, 2020).

Tez çalışmasında k-ortalamlar algoritması, hiyerarşik kümeleme ve hibrit kümeleme olmak üzere üç kümeleme yöntemine göre ürünler gruplandırıldıktan sonra her bir ürün grubuna ait envanter yönetim stratejisini belirlemek için ekonomik sipariş miktarı (Economic Order Quantity-EOQ) ve emniyet stoğu belirlenmiştir. Ekonomik sipariş miktarı belirlenirken eşitlik (3.6) kullanılmıştır (Ustaahmetoğlu, 2013):

$$EOQ = \sqrt{\frac{2C_0D}{C_1U}} \quad (3.6)$$

Eşitlik (3.6)'daki; EOQ ekonomik sipariş miktarını, C_0 sipariş maliyetini, C_1 elde bulundurma maliyetini, D talebi (satışı), U ise her birimin maliyetini ifade etmektedir.

Emniyet stoğunu (ES) hesaplamak için eşitlik (3.7) kullanılmıştır. Burada z istenilen hizmet yüzdesine karşılık normal dağılım tablosundan alınan değeri, σ_d birim zamanda gerçekleşen talebin standart sapmasını, L ise ortalama tedarik süresini (birim zamanlı) göstermektedir. Tedarik süresi sabit, talep ise değişkendir (Parıldar ve Akyürek, 2021).

$$ES = z * \sigma_d * \sqrt{L} \quad (3.7)$$

3.2. Araç Rotalama

Abdelmaguid (2004), belirli bir planlama ufku üzerinden bir takım müşteriler için araç rotalama, envanter tutma ve geri sipariş kararlarının alınacağı bir karışık tam sayılı matematiksel model önermiştir. Bu matematiksel modele alt sınır sağlamak için Lagrange gevşetmesine ve Bender ayrışmasına dayalı iki yaklaşım sunmuştur.

Chen vd. (2010), kapasite kısıtlı araç rotalama problemini çözmek için çoklu operatör optimizasyonuna dayalı değişken komşu iniş sezgiselli bir hibrit sezgisel metot önermişlerdir.

Tez çalışmasında otomatik satış makinesi operatörlerinin araç rotalarını belirlemek amacıyla ilk olarak temel araç rotalama probleminden yola çıkarak klasik bir kapasite kısıtlı araç rotalama problemi için matematiksel model oluşturulmuştur. Oluşturulan

matematiksel model aşağıda açıklanmıştır. Modelde i ve j indisleri birimleri v indisi aracı ifade etmektedir.

Parametreler:

n = Toplam birim sayısı

V = Araç sayısı

q_i = i biriminin talebi ($i=0,1,\dots,n$)

c_{ij} = i birimi ile j birimi arasındaki uzaklık ($i,j=0,1,\dots,n$)

d_{ij} = i biriminden j birimine gidiş süresi ($i,j=0,1,\dots,n$)

Q_v = v aracının çalışabileceği günlük mesai süresi

Değişkenler:

$X_{ijv} = \begin{cases} 1, & v \text{ aracı } i \text{ biriminden } j \text{ birimine gidiyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

u_i = Yardımcı değişken

Amaç fonksiyonu:

$$\text{Min } \sum_{i=0}^n \sum_{j=0, j \neq i}^n \sum_{v=1}^V c_{ij} X_{ijv} \quad (3.8)$$

$$\sum_{j=1}^n x_{0jv} = 1 \quad (v = 1, 2, \dots, V) \quad (3.9)$$

$$\sum_{i=1}^n x_{i0v} = 1 \quad (v = 1, 2, \dots, V) \quad (3.10)$$

$$\sum_{i \neq k}^n X_{ikv} - \sum_{j \neq k}^n X_{kjv} = 0 \quad (k = 1, 2, \dots, n; v = 1, 2, \dots, V) \quad (3.11)$$

$$\sum_{i=0, i \neq j}^n \sum_{v=1}^V x_{ijv} = 1 \quad (j = 1, 2, \dots, n) \quad (3.12)$$

$$\sum_{j=0, j \neq i}^n \sum_{v=1}^V x_{ijv} = 1 \quad (i = 1, 2, \dots, n) \quad (3.13)$$

$$u_i - u_j + n x_{ijv} \leq n - 1 \quad \left(\begin{matrix} i = 1, 2, \dots, n; j = 1, 2, \dots, n; v = 1, 2, \dots, V; \\ i \neq j \end{matrix} \right) \quad (3.14)$$

$$u_j \geq 0 \quad (j = 1, 2, \dots, n) \quad (3.15)$$

$$\sum_{i \neq j}^n \sum_{j \neq i}^n X_{ijv} * q_i + \sum_{i \neq j}^n \sum_{j \neq i}^n X_{ijv} * d_{ij} \leq Q_v \quad (v = 1, 2, \dots, V) \quad (3.16)$$

Amaç fonksiyonu araçlar tarafından kat edilecek toplam mesafeyi minimize etmeye çalışmaktadır. Kısıt (3.9) ve kısıt (3.10) sırasıyla her araç için depodan çıkma ve depoya dönmeyi sağlamaktadır. Kısıt (3.11) bir düğüme giren aracın o düğümden çıkmasını sağlamaktadır. Eğer araç k müşterisine gittiyse, k müşterisinden de mutlaka bir j müşterisine gitmelidir. (3.12) ve (3.13) numaralı kısıtlar deponun dışında bulunan her müşterinin bir araçla yalnızca bir defa ziyaret gerçekleşmesi gerektiğini sağlamaktadırlar. Kısıt (3.14) ve kısıt (3.15) alt tur oluşumunu engellemektedir. Kısıt (3.16) kapasite kısıtı olup ele alınan problem için rotanın bir vardiya süresini (450 dakika) aşmamasını sağlamaktadır.

Kapasite kısıtlı araç rotalama problemi için kullanılan matematiksel model oluşturulduktan sonra tez çalışması kapsamında ele alınan gerçek probleme uygun olması açısından yeni bir model geliştirilmiştir. Tez kapsamında ele alınan problemde günlük makine ziyaretlerini yapacak olan aracın para miktarı, ürün miktarı, arıza bilgisi ve sağlıklı ürün sinyallerinin geldiği makineleri ziyaret etmesi amaçlanmaktadır. Böylece araç sinyal olmayan makinelere gereksiz yere uğramayacak ve maliyet tasarrufu sağlanacaktır. Bu amaç doğrultusunda oluşturulan model aşağıdaki gibidir:

Parametreler:

n = Toplam birim sayısı

V = Araç sayısı

q_i = i biriminin talebi ($i=0,1,\dots,n$)

c_{ij} = i birimi ile j birimi arasındaki uzaklık ($i,j=0,1,\dots,n$)

d_{ij} = i biriminden j birimine gidiş süresi ($i,j=0,1,\dots,n$)

Q_v = v aracının çalışabileceği günlük mesai süresi

$s_i = \begin{cases} 1, & i \text{ biriminde herhangi bir sinyal varsa} \\ 0, & \text{diğer durumda} \end{cases}$

Değişkenler:

$$X_{ijv} = \begin{cases} 1, & v \text{ aracı } i \text{ biriminden } j \text{ birimine gidiyorsa} \\ 0, & \text{diğer durumda} \end{cases}$$

u_i = Yardımcı değişken

Amaç fonksiyonu:

$$\text{Min } \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{v=1}^V c_{ij} X_{ijv} \quad (3.17)$$

$$\sum_{\substack{j=1 \\ s_j=1}}^n x_{0jv} = 1 \quad (v = 1, 2, \dots, V) \quad (3.18)$$

$$\sum_{\substack{i=1 \\ s_i=1}}^n x_{i0v} = 1 \quad (v = 1, 2, \dots, V) \quad (3.19)$$

$$\sum_{\substack{i=1 \\ i \neq k}}^n X_{ikv} - \sum_{\substack{j=1 \\ j \neq k}}^n X_{k j v} = 0 \quad (k = 1, 2, \dots, n; v = 1, 2, \dots, V; s_i = 1) \quad (3.20)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n \sum_{v=1}^V x_{ijv} = 1 \quad (j = 1, 2, \dots, n; s_j = 1) \quad (3.21)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^n \sum_{v=1}^V x_{ijv} = 1 \quad (i = 1, 2, \dots, n; s_i = 1) \quad (3.22)$$

$$u_i - u_j + n x_{ijv} \leq n - 1 \quad \left(\begin{matrix} i = 1, 2, \dots, n; j = 1, 2, \dots, n; v = 1, 2, \dots, V; \\ i \neq j; s_i = 1; s_j = 1 \end{matrix} \right) \quad (3.23)$$

$$u_j \geq 0 \quad (j = 1, 2, \dots, n; s_j = 1) \quad (3.24)$$

$$\sum_{\substack{i=0 \\ i \neq j \\ s_i=1}}^n \sum_{\substack{j=0 \\ j \neq i \\ s_j=1}}^n X_{ijv} * q_i + \sum_{\substack{i=0 \\ i \neq j}}^n \sum_{\substack{j=0 \\ j \neq i}}^n X_{ijv} * d_{ij} \leq Q_v \quad (v = 1, 2, \dots, V) \quad (3.25)$$

Denklem (3.17) ile gösterilen amaç fonksiyonu araçlar tarafından kat edilecek toplam mesafeyi minimize etmeye çalışmaktadır. Kısıt (3.18) her aracın depodan hareket etmesini, kısıt (3.19) her aracın depoya dönmesini sağlamaktadır. Kısıt (3.20) sinyal gelen bir düğüme giren aracın o düğümden çıkarak sinyal gelen başka bir düğüme gitmesini sağlamaktadır. Eğer araç k düğüme gittiyse, k düğümden de mutlaka bir j düğüme gitmelidir. (3.21) ve (3.22) numaralı kısıtlar deponun dışında bulunan her düğümün bir araçla yalnızca bir defa ziyaret gerçekleşmesi gerektiğini sağlamaktadırlar. Kısıt (3.23) ve kısıt (3.24) alt tur oluşumunu engellemektedir. Kısıt (3.16) kapasite kısıtı olup ele alınan problem için her bir aracın rotasının bir vardiya süresini (450 dakika) aşmamasını sağlamaktadır.

Yeni oluşturulan modelin amaç fonksiyonu ve kısıtları kapasite kısıtlı araç rotalama problemi için kullanılan modelle benzerdir. Yeni oluşturulan model ile yapılmak istenen aracın tüm birimlere gitmesi yerine sadece sinyal gelen makineleri ziyaret etmesini sağlamaktır.

Problem çözümü esnasında, araç rotalama problemlerinin literatürde bulunan çözüm yöntemleri gözden geçirilip incelenen problemde araç rotalarının optimum şekilde oluşturabilmek için probleme uygun matematiksel model hazırlanmıştır. Problemin karmaşıklığı sebebiyle kurulan matematiksel modelde mevcut problem için optimum sonuca ulaşamamıştır. Bu nedenele tez kapsamında ele alınan probleme en uygun çözümü bulabilmek için en yakın komşu algoritmasına dayalı sezgisel araç rotalama algoritması oluşturulmuştur.

En yakın komşu algoritması rastgele seçilen bir şehirden başlar. Seçilen şehrin uzaklık matrisindeki satırı taranarak en yakın komşusu seçilir ve rotaya eklenir. Tüm şehirler rotaya eklenene kadar en yakın komşu ekleme işlemi tekrarlanır. Algoritma en son rotaya eklenen şehirden depoya olan mesafenin de maliyet değerine eklenmesiyle son bulur. Yöntem tek bir rota üzerinden hareket edilen seri ve aynı anda birden fazla rota üzerinden hareket edilen paralel olmak üzere iki gruba ayrılır. Karmaşıklık düzeyi $Q(n^2)$ olan yöntemde rotaya eklenen son şehirlerin toplam maliyet üzerindeki etkileri çok yüksek olabilmektedir (Babayiğit ve Yıldız, 2019).

Peya vd. (2019), yaptıkları çalışmada kapasite kısıtlı araç rotalama için farklı algoritmaları kullanarak denemeler yapmışlardır. Kullandıkları algoritmadan biri de en yakın komşu algoritmasıdır. En yakın komşu algoritması için bir uygulamada depoya en yakın birimden, diğer bir uygulamada ise depoya en uzak birimden başlanarak rota oluşturulmuştur.

Gün içinde makinelere ziyaret gerçekleştirecek olan aracın sadece sinyal gelen makinelere uğramasını sağlayan araç rotalama algoritmasının adımları aşağıdaki gibidir. Burada Süre $[k]$, k aracının harcadığı toplam süreyi; Süre $[m]$, m . makineye gidiş süresini; servis süresi $[m]$, m . makinede harcanan servis süresini; mesai süresi $[k]$, k aracı için tanımlanmış mesai süresini; sinyal $[m]$, m . makinedeki para miktarı, ürün miktarı, arıza bilgisi ya da sağlıklı ürün sinyallerinin bulunmasını ifade etmektedir.

Depodan başlayan yeni bir rota oluştur r_k , $k:=0$, $i:=0$

1. Eğer $k > \text{Araç Sayısı}$

O Halde, 7.adıma git

2. Atanmamış ve $\text{Sinyal}[m]=1$ olan müşteriler için i . noda en yakın müşteriyi seç

3. **Eğer** $\text{Süre}[k] + \text{Süre}[m] + \text{Servis Süresi}[m] + \text{Depoya Dönüş Süresi}[m] < \text{Mesai Süresi}[k]$ ise,

O Halde, i . müşteriyi r_k rotasına ekle, atanmamış müşteri kümesinden çıkar ve $\text{Süre}[k] = \text{Süre}[k] + \text{Süre}[m] + \text{Servis Süresi}[m]$ yap

Diğer durumda, 5.adıma git

4. **Eğer** Atanmamış ve $\text{Sinyal}[m]=1$ müşteri kümesi eleman sayısı > 0 ise,

O Halde, 2. Adıma git

Diğer durumda, 5. Adıma git

5. $\text{Süre}[k] = \text{Süre}[k] + \text{Depoya Dönüş Süresi}[m]$ yap

6. $k=k+1$ yap, 1. adıma git.

7. Bitir

Tez kapsamında çalışma yapılan firma için makinelerdeki arıza bilgisi sinyali önem teşkil etmektedir. Arızası bulunan makine hizmet verememektedir. Dolayısıyla arızasına erken müdahale edilmeyen makine uzun süre hizmet veremeyecek ve firma için zarara yol açacaktır. Bu nedenele arıza bilgisi sinyali, para miktarı, ürün miktarı ve sağlıklı ürün sinyallerine göre önceliklidir.

Arıza bilgisi sinyali, otomatik satış makinesi yönetimini gerçekleştiren işletmeler için daha kritik olduğundan başlangıçta oluşturulun sezgisel algoritma arıza bilgisi sinyali önceliklendirilerek geliştirilmiştir. Arıza bilgisi sinyalini öncelikli olduğu durum için oluşturulan algoritma adımları aşağıdaki gibidir:

Depodan başlayan yeni bir rota oluştur r_k , $k:=0$, $i:=0$

1. Eğer $k > \text{Araç Sayısı}$

O Halde, 10.adıma git

2. Atanmamış ve $AB[m]=1$ olan müşteriler için i . noda en yakın müşteriyi seç

3. **Eğer** $Süre[k] + Süre[m] + Servis Süresi[m] + Depoya Dönüş Süresi[m] < Mesai Süresi[k]$ ise,

O Halde, i . müşteriyi r_k rotasına ekle, atanmamış müşteri kümesinden çıkar ve $Süre[k] = Süre[k] + Süre[m] + Servis Süresi[m]$ yap

Diğer durumda, 8.adıma git

4. **Eğer** Atanmamış ve $AB[m]=1$ müşteri kümesi eleman sayısı > 0 ise,

O Halde, 2. Adıma git

Diğer durumda, 5. Adıma git

5. Atanmamış ve $Sinyal[m]=1$ olan müşteriler için i . noda en yakın müşteriyi seç

6. **Eğer** $Süre[k] + Süre[m] + Servis Süresi[m] + Depoya Dönüş Süresi[m] < Mesai Süresi[k]$ ise,

O Halde, i . müşteriyi r_k rotasına ekle, atanmamış müşteri kümesinden çıkar ve $Süre[k] = Süre[k] + Süre[m] + Servis Süresi[m]$ yap

Diğer durumda, 8.adıma git

7. **Eğer** Atanmamış ve $Sinyal [m]=1$ müşteri kümesi eleman sayısı > 0 ise,

O Halde, 5. Adıma git

Diğer durumda, 8. Adıma git

8. $Süre[k] = Süre[k] + Depoya Dönüş Süresi[m]$ yap

9. $k=k+1$ yap, 1. adıma git.

10. Bitir

Burada $AB [m]$, m . makinedeki arıza bilgisi sinyalinini; $Süre [k]$, k aracının harcadığı toplam süreyi; $Süre [m]$, m . makineye gidiş süresini; servis süresi $[m]$, m . makinede

harcanan servis süresini; mesai süresi $[k]$, k aracı için tanımlanmış mesai süresini; sinyal $[m]$, m . makinedeki arıza bilgisi dışında bulunan sinyalleri ifade etmektedir.



4. BULGULAR VE TARTIŞMA

Tez çalışmasında otomatik satış makineleri için makinelere ürün beslemesi yapan depodaki envanter yönetimini sağlamak amacıyla çok kriterli envanter sınıflandırma yapılmıştır. Aylık satış miktarı, tedarik süresi, birim fiyat ve sezon bilgilerine sahip olan envanter kalemlerinin kriter olarak birbirine benzer olanları kümelenerek her bir grup için ayrı bir envanter yönetim stratejisi oluşturulmuştur. Böylece ürünlerin tek tek envanter kontrolünün yapılması yerine benzer ürünlerden oluşan küme için ortak bir envanter kontrolünün yapılması sağlanmıştır. Envanter kalemleri için kümeleme işlemi yapılırken k-ortalamlar algoritması, hiyerarşik kümeleme ve hibrit kümeleme yöntemi kullanılmıştır. Belirlenen kümelere ait ekonomik sipariş miktarı ve emniyet stoğu adeti hesaplanmıştır. Her bir yönteme ait sonuçlar değerlendirilmiştir.

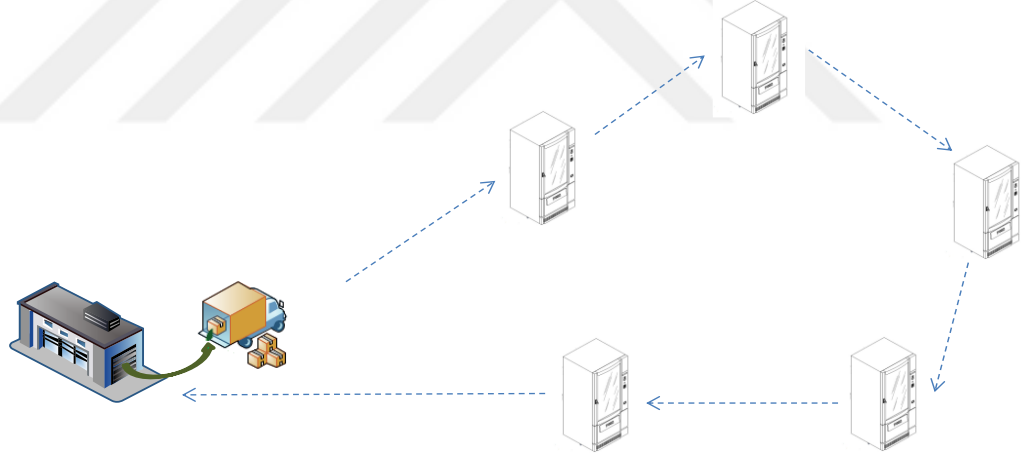
Tez çalışmanın diğer bir aşamasında ise otomatik satış makinelerine ürün yükleme ve boşaltma yapan, arıza durumuyla ilgilenen ve kasadaki parayı toplayan personelin günlük makine ziyaretini planlamak için araç rotalama problemi üzerine çalışılmıştır. Para miktarı, ürün miktarı, arıza bilgisi, sağlıklı ürün (son tüketim tarihini tamamlamış ürünler) kriterlerine ait sinyallerin bulunduğu bir sistemde en uygun araç rotası oluşturulmaya çalışılmıştır.

Bu bölümde ilk olarak tez kapsamında ele alından sistemin analizi ardından da 3. bölümde bahsedilen çalışmaların uygulamalarına ve sonuçlarına yer verilmiştir. Elde edilen çıktılar yorumlanmıştır.

4.1. Sistem Analizi

Otomatik satış makinesi yöneticileri farklı lokasyonlarda yer alan en az 100-200 otomatik satış makinesi ağını ortak bir (veya birkaç) depodan yönetmekle sorumludurlar. Bir otomatik satış makinesi ağında öncelikle farklı lokasyonlara (fabrikalar, okullar, iş ofisleri, duraklar, tüketicinin yoğun olduğu bölgeler...) makineler yerleştirilir. Otomatik satış makinesi yöneticileri bu ağda yer alan makinelere ürün dağıtımı ve biriken nakit paranın toplamasını organize etmekle sorumludurlar. Ortak bir depodan makinelere ürün dağıtımı yapacak araçlara ürünler yüklenmekte, günlük ya da periyodik makine

ziyaretleriyle makinelere ürün yüklemesi gerçekleştirilmektedir. Makinelere ürün yüklemesi yapıldıktan sonra bir sonraki ürün yükleme zamanına kadar makineler ziyaret edilmemekte (herhangi bir acil durum, ya da arıza durumu olmaması durumunda), bu süre zarfında makinelerde gerçekleştirilen satış bilgisi ve/veya ürün stok bilgisi alınmamaktadır. Makinede biriken paraların toplanması için sektörde iki farklı uygulama yer almaktadır: bazı yöneticiler (otomatik satış makinesi ağındaki makine sayısı ≤ 100 adet) nakit para toplama işlemini kendileri gerçekleştirmektedir. Bunun nedenini dağıtım işlemini gerçekleştiren personele duyulan düşük güvenle açıklamaktadırlar. İkinci grup yönetici ise ağda yer alan otomatik satış makinesi sayısının çok olması nedeniyle nakit toplama işlemini dağıtım gerçekleştiren personele bırakmaktadır. Bu durumda herhangi bir otomatik satış makinesine giden dağıtım personeli hem eksilen ve/veya biten ürünlerin makineye yerleştirilmesi hem de nakit paranın toplanmasından sorumlu olmaktadır. Sistemin temel işleyişi Şekil 4.1’de gösterilmektedir.



Şekil 4. 1. Otomatik satış makinesi yönetim sistemi

Böyle bir otomatik satış makinesi ağında sistem yöneticisinin temelde iki görev ve sorumluluğu bulunmaktadır.

1. Otomatik satış makinesi yönetimi: Yönetici makinelerde yer alan ürün çeşitliliği ve miktarlarını belirlemeli, hangi makinede hangi tip ürünün satılacağı deneyimine sahip olmalı, makinelere ziyaret sıklığını belirlemeli, makinelerden

nakit para toplama sürecini organize etmeli, makinelerde herhangi bir arıza oluşması durumunda teknik ekibi en kısa süre içinde yönlendirmesini yapmalı, makinelerin bakım onarım faaliyetlerini organize etmelidir.

2. Depo ve dağıtım yönetimi: sistem yöneticisi makinelerin satış rakamlarından elde ettiği verilere dayanarak, ürünlerin stok planını yapmalı, stok yenileme sistemini planlamalı, dağıtım personeli için dağıtım çizelgesi oluşturmalıdır.

Her bir otomatik satış makinesi ziyaretinde, makinelerdeki ürün stokları gözden geçirilmekte ve makineler tam kapasite yüklenmektedir. Talebin belirsiz olması nedeniyle bir sonraki ziyaret zamanının ne zaman yapılacağı kişisel deneyime dayanarak belirlenmekte, herhangi bir veri toplanmamaktadır. Eğer bir sonraki ziyaret kısa bir zaman sonrasında planlanırsa, ihtiyaç olmamasına rağmen pek çok ziyaretin gerçekleşmesine neden olurken, ziyaretin uzun zaman sonra planlanması kayıp satışa neden olmaktadır.

Tez çalışması kapsamında etkili bir otomatik satış makinesi ağı yönetimi için bütünlüklük bir yönetim sistemi oluşturulmuştur. Yönetim sisteminin oluşturulmasında depo ve dağıtım yönetimi ayrı ayrı ele alınmıştır. Bu bölümde ilk olarak otomatik satış makinelerine ürün beslemesi yapan depo için oluşturulan envanter yönetimi anlatılmıştır. Ardından otomatik satış makinelerinin arıza tamirini, ürün beslemesini, kasasındaki paranın toplanmasını ve sağlıklı ürün ömürleri kontrollerini yapan araçlar için oluşturulan rotanın belirlenmesi anlatılmıştır.

4.2. Envanter Sınıflandırma ve Yönetimi

Tez çalışması kapsamında otomatik satış makinelerine ürün tedarikini sağlayan depo için çok kriterli ABC sınıflandırmasına dayalı envanter sınıflandırma yapılmıştır. Kriterler sektör uzmanlarının paylaştığı bilgiler ve literatür araştırmaları birlikte harmanlanarak belirlenmiştir. Sınıflandırma için R paket programı kullanılmıştır.

Çalışmada otomatik satış makinelerinin yönetimini gerçekleştiren bir firmanın 59 adet ürünü ele alınmıştır. Firmadan ürünlere ait aylık satış miktarı, tedarik süresi, birim fiyat ve sezon bilgileri gibi envanter yönetimi için önem arz eden bilgiler elde edilmiştir. Bu bilgiler toplandıktan sonra ürünler hiyerarşik, k-ortalamlar ve hibrit kümeleme olmak üzere üç farklı kümeleme yöntemine göre iki ve üç kümeye ayrılmıştır. Elde edilen her küme için ekonomik sipariş miktarı, minimum, maksimum ve ortalama emniyet stoğu adetleri hesaplanmıştır. R programında kodlanan kümeleme yöntemleri aşağıda açıklanmıştır.

```
library(readxl)
data<- read_excel("C:/Users/HP/Desktop/RData/data.xlsx")
data<- as.data.frame(data)
data <- data.frame(data, row.names = 1)
#View(data)
summary(data)

#Plotting
boxplot(data)

#Hierarchical clustering
#Scaling
data<-as.data.frame(scale(data[,1:4]))
summary(data)

#uzaklık ölçümü
dist_mat <- dist(data, method = 'euclidean')
hclust_avg <- hclust(dist_mat, method = 'average')
plot(hclust_avg)

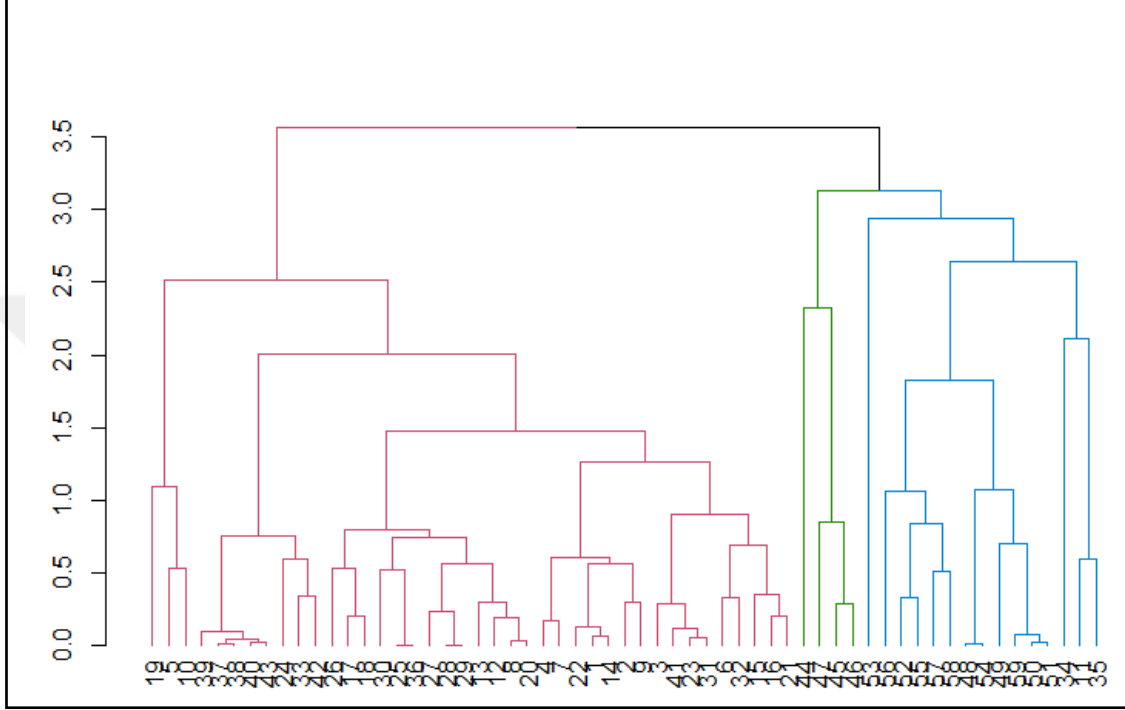
cut_avg <- cutree(hclust_avg, k = 3)
plot(hclust_avg)
rect.hclust(hclust_avg , k = 3, border = 2:6)
abline(h = 3, col = 'red')
print(cut_avg)

#Dendrogramda kümeleri renklendirerek görme
suppressPackageStartupMessages(library(dendextend))
avg_dend_obj <- as.dendrogram(hclust_avg)
avg_col_dend <- color_branches(avg_dend_obj, k = 3)
plot(avg_col_dend)
```

Şekil 4.2. Hiyerarşik kümeleme yönteminin R kodu

Hiyerarşik kümeleme yönteminin uygulaması: Şekil 4.2’de kodu gösterilen hiyerarşik kümelemenin ilk adımında 66 satır (59 ürün) ve beş sütundan (dört kriter) oluşan veri seti Excel dosyasından okutularak programa aktarılmıştır. Veri setindeki kriterlere ait değerlerin ortalama ve standart sapmaları gibi bilgileri görebilmek için “summary” komutu kullanılmıştır. Veri setinde “scale” komutu kullanılarak ölçeklendirme yapıp hiyerarşik kümeleme uygulanmaya uygun hale getirilmiştir. Bu yöntemde veriler arasındaki uzaklığı ölçmek için Öklid uzaklığı kullanılmıştır. Hiyerarşik kümeleme uygulandıktan sonra veriler dendogramda gösterilmiştir.

Hiyerarşik kümeleme yöntemine göre elde edilen üç kümeye ait dendogram Şekil 4.3’de gösterilmiştir. Eleman sayısı en yüksek olan küme birinci kümedir. Kümelere ait ürünler ve ürün bilgileri ayrıntılı olarak EK-2’de verilmiştir.



Şekil 4.3. Hiyerarşik kümeleme sonucunun dendogramda gösterilmesi

Hiyerarşik kümeleme uygulandıktan sonra elde edilen kümelere ait bilgiler Çizelge 4.1’de gösterilmiştir. Burada birinci küme için ekonomik sipariş miktarı 314, ikinci küme için 1949, üçüncü küme için 130 olarak hesaplanmıştır.

Çizelge 4.1. Hiyerarşik kümeleme yönteminden elde edilen üç kümeye ait bilgiler

	Satış Miktarı (Adet)				EOQ	Emniyet Stoğu Miktarı		
	Ortalama	Minimum	Maksimum	Standart Sapma		Ortalama	Minimum	Maksimum
1.KÜME	1965,111111	210	3782	1031,957289	313,3397807	116,3832121	15,49708743	408,9835245
2.KÜME	4319,875	2846,5	6450,75	1330,397432	1948,161526	92,24134708	70,6129543	141,8258098
3.KÜME	921,5	33,5	2870	864,665976	129,4217911	46,26879133	3,724027138	114,9217503

İki küme elde etmek için veri setine hiyerarşik kümeleme yöntemi uygulandıktan sonra hesaplanan satış miktarının ortalama, minimum, maksimum değeri ile standart sapması, ekonomik sipariş miktarı, emniyet stoğunun ortalama, minimum, maksimum değeri Çizelge 4.2’de gösterilmiştir.

Çizelge 4.2. Hiyerarşik kümeleme yönteminden elde edilen iki kümeye ait bilgiler

	Satış Miktarı (Adet)				EOQ	Emniyet Stoğu Miktarı		
	Ortalama	Minimum	Maksimum	Standart Sapma		Ortalama	Minimum	Maksimum
1.KÜME	1933,86	191	3880	1020,745075	291,703	144,424	14,7592	404,331
2.KÜME	1692,18	30	6444	1753,304149	198,116	131,463	29,8376	415,806

Çizelge 4.2’de de görülebileceği gibi hiyerarşik kümeleme yöntemine göre birinci küme için önerilen ekonomik sipariş miktarı 292 adet adet, ikinci küme için 199 adet olarak hesaplanmıştır.

k-ortalamlar yönteminin uygulamasına ilişkin R kodu Şekil 4.4’de gösterilmiştir. Gösterilen kod iki kümeye ayırma işlemi için kullanılmıştır. Veri seti üç kümeye ayrılmak istendiğinde koddaki “kmeans.df<-(df[,1:4],2)” satırındaki “2” ifadesi “3” ile değiştirilmiştir.

```
library(readxl)
data<- read_excel("C:/Users/HP/Desktop/RData/data.xlsx")
data<- as.data.frame(data)
data <- data.frame(data, row.names = 1)
#view(data)
summary(data)

#k-means ile kümeleme işlemi
set.seed(15)
kmeans.df<-kmeans(df[,1:4], 2)
kmeans.df
df<- as.data.frame(df)
```

Şekil 4.4. k-ortalamlar yönteminin R kodu

k-ortalamlar yöntemi kullanılarak üç kümeye ayrılmış ürünler EK-1’de verilmiştir. Bu yöntemin uygulanması sonucunda elde edilen kümelere ait ortalama satış miktarı, minimum satış miktarı, maksimum satış miktarı, satış miktarındaki standart sapma, ekonomik sipariş miktarı, ortalama emniyet stoğu miktarı, minimum emniyet stoğu miktarı ve maksimum emniyet stoğu miktarı bilgileri Çizelge 4.3’ de paylaşıldığı gibidir.

Çizelge 4.3. k-ortalamlar yönteminden elde edilen üç kümeye ait bilgiler

	Satış Miktarı (Adet)				EOQ	Emniyet Stoğu Miktarı		
	Ortalama	Minimum	Maksimum	Standart Sapma		Ortalama	Minimum	Maksimum
1.KÜME	1523,17308	210	2871,75	784,704369	313,3397807	110,863	15,4971	408,98352
2.KÜME	1425,51563	33,5	4267	1331,304355	129,4217911	55,6056	3,72403	141,82581
3.KÜME	3417,47727	2179	6450,75	1058,503873	1476,143624	126,194	60,6611	202,01607

k-ortalamlar algoritması uygulanarak elde edilen iki kümedeki ürünler ve ürünlerin detaylı bilgileri EK-4'tedir. k-ortalamlar algoritmasından elde edilen iki kümeye ait satış miktarlarının ortalama, minimum, maksimum adetleri ile satış değerlerinin standart sapması, ekonomik sipariş miktarı ve emniyet stoğunun ortalama, minimum, maksimum miktarlar Çizelge 4.4'de gösterilmiştir. Bu yöntemle göre birinci ürün kümesi için minimum ekonomik sipariş miktarının 329, ikinci küme için 326 olması önerilmiştir.

Çizelge 4.4. k-ortalamlar yönteminden elde edilen iki kümeye ait bilgiler

	Satış Miktarı (Adet)				EOQ	Emniyet Stoğu Miktarı		
	Ortalama	Minimum	Maksimum	Standart Sapma		Ortalama	Minimum	Maksimum
1.KÜME	1933,86	191	3880	1020,75	328,634	175,085	21,9239	627,514
2.KÜME	1515,47	13	6444	1736,63	325,064	131,166	41,5098	218,748

Hibrit kümeleme yöntemi k-ortalamlar ve hiyerarşik kümelemenin bir arada kullanılmasından oluşmaktadır. Bu yüzden Şekil 4.5'te gösterildiği gibi veri setine ilk olarak k-ortalamlar algoritması ve ardından hiyerarşik kümeleme yöntemi uygulanmıştır. Daha sonra hiyerarşik kümelemekten elde edilen küme merkezleri bulunmuştur. Hiyerarşik kümelemekten elde edilen küme merkezleri k-ortalamlar algoritmasında kullanılarak hibrit bir kümeleme yöntemi gerçekleştirilmiştir.

```

library(readxl)
data<- read_excel("C:/Users/HP/Desktop/RData/data.xlsx")
data<- as.data.frame(data)
data <- data.frame(data, row.names = 1)
#View(data)
summary(data)

#k-means ile kümeleme işlemi
set.seed(15)
kmeans.df<-kmeans(df[,1:4], 2)#2 kumeye boluyoruz.
kmeans.df
df<- as.data.frame(df)

#Hierarchical clustering
#hiyerarşik kümelemenin uygulanması ve k-kümenin dendogramda gösterilmesi:
hc <- hclust(dist(df[,1:4]), method="ave")
plot(hc, hang = -1, labels=df$Aylık.Satış.Miktarı,cex=.4)
rect.hclust(hc, k=2)
groups <- cutree(hc, k=2)
print(groups)

#hiyerarşik kümelemede bulunan kümelerin merkezlerinin bulunması:
df<-scale(df[,1:4])
clus.centers <- aggregate(df, list(groups), mean)
clus.centers

# ilk sütunu atma:
clus.centers <- clus.centers[, -1]
clus.centers

#hiyerarşik kümelemeden elde edilen küme merkezleri kullanarak k-ortalamlar algoritmasının kullanılması:
km.res2 <- eclust(df, "kmeans", k = clus.centers, graph = FALSE)
fviz_silhouette(km.res2)
# sonucu görüntüleme
km.res2

```

Şekil 4.5. Hibrit kümeleme yönteminin R kodu

R programlama dilinde hibrit kümeleme yöntemi kullanılarak elde edilen üç kümeye ait detaylı bilgi EK-3’de verilmiştir. Elde edilen kümelerden yola çıkılarak hesaplanan bilgiler Çizelge 4.5’te gösterilmiştir.

Çizelge 4.5. Hibrit kümeleme yönteminden elde edilen üç kümeye ait bilgiler

	Satış Miktarı (Adet)				EOQ	Emniyet Stoğu Miktarı		
	Ortalama	Minimum	Maksimum	Standart Sapma		Ortalama	Minimum	Maksimum
1.KÜME	1967,527778	210	3782	1031,929609	313,3397807	131,9689205	15,49708743	631,8491896
2.KÜME	759,125	33,5	1883,75	683,5187337	129,4217911	45,4586535	3,724027138	114,9217503
3.KÜME	4029,9	2846,5	6450,75	1323,747659	1021,585221	84,99116672	55,99044528	141,8258098

Çalışmada hibrit kümeleme yöntemiyle veri seti için iki kümeye ayırma işlemi de gerçekleştirilmiştir. Elde edilen iki küme için hesaplanan satış miktarının ortalama, minimum, maksimum, standart sapma değerleri, ekonomik sipariş miktarı, emniyet stopunun ortalama, minimum ve maksimum değerleri Çizelge 4.6’da belirtildiği gibidir.

Çizelge 4.6. Hibrit kümeleme yönteminden elde edilen iki kümeye ait bilgiler

	Satış Miktarı (Adet)				EOQ	Emniyet Stoğu Miktarı		
	Ortalama	Minimum	Maksimum	Standart Sapma		Ortalama	Minimum	Maksimum
1.KÜME	1933,86	191	3880	1020,745075	18,8481	1193,62	328,634	2553,625
2.KÜME	1515,47	13	6444	1736,632233	325,064	131,166	41,5098	218,7485

Çizelge 4.6’ ya göre birinci küme için önerilen ekonomik sipariş miktarı 19 adet, ikinci küme için 326 adet olarak hesaplanmıştır.

Kümeleme yöntemlerinin kümeleme mantığının birbirinden farklı olması, küme merkezi seçim yöntemlerinin birbirinden farklı olmasından dolayı küme sayısı aynıyken bile farklı yöntemler için farklı ekonomik sipariş miktarları bulunmuştur.

Envanter birimlerinin sayısı fazla olduğunda tedarikçisinin, depolanmasının yönetimi güçleşmektedir. Envanter birimlerini etkin bir şekilde yönetebilmek adına ürünlerin gruplandırılması yönetim sürecini kolaylaştıracaktır. Böylece her bir envanter kalemi için ayrı ayrı envanter yönetim stratejisi belirlemek yerine benzer kriterlere sahip envanter kalemleri için ortak envanter yönetim stratejisi uygulanmış olacaktır.

Tez kapsamında yapılan çalışmada 59 envanter birimi üç farklı kümeleme yöntemiyle gruplara ayrılmıştır. Böylece 59 ürün için ayrı ayrı envanter yönetim stratejisi belirlemek yerine her bir küme için ortak bir envanter stratejisi önerilmiştir. Her küme için ekonomik sipariş miktarı, emniyet stoğunun minimum, ortalama ve maksimum miktarları belirlenmiştir. Bu sayede küme için önerilen ortak strateji kümede bulunan tüm envanter kalemleri için uygulanacaktır.

Kümeleme yapılırken küme sayısının belirlenebilmesi için literatürde sıkça kullanılan Elbow yöntemi ve Silhouette indeksi kullanılmıştır.

Elbow yönteminin R dilinde kodlanmış hali Şekil 4.6’da gösterilmiştir. Yöntemi uygulamak için R programının “factoextra” paketi kullanılmıştır.

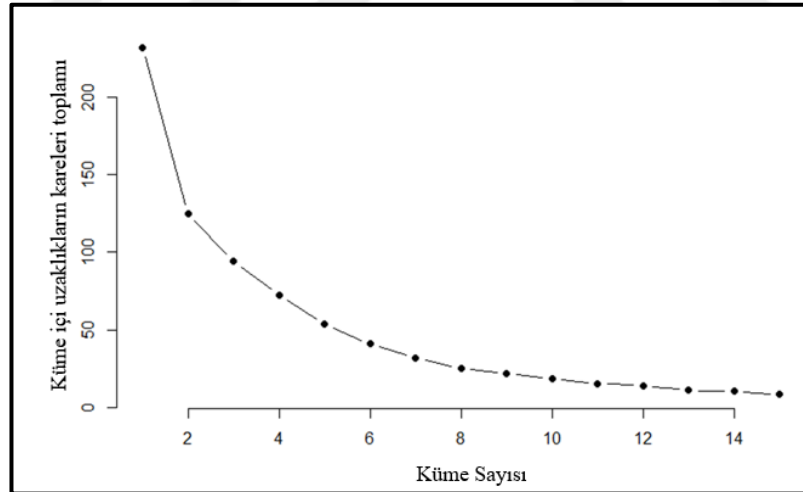
```

library(factoextra)
#Optimal sınıf sayısını belirlemek için Elbow yöntemi
set.seed(15)
#k=2 den k=15 e kadar wss nin hesaplanması ve grafik haline getirilmesi
k.max <- 15
df<-scale(data[,1:4])
wss <- sapply(1:k.max,
              function(k){kmeans(df, k, nstart=50,iter.max = 15 )$tot.withinss})
wss
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters k",
     ylab="Total within-clusters sum of squares")

```

Şekil 4.6. Elbow yönteminin uygulanması

Şekil 4.7’de Elbow yöntemi uygulandıktan sonra küme sayısı ile kümeler içi kareler toplamı arasındaki ilişkiyi gösteren grafik yer almaktadır. Başarılı bir kümeleme uygulamasında kümeler içi kareler toplamının düşük olması beklenmektedir. Şekil 4.1’e göre $k=2$ ve $k=6$ noktalarına kadar kümeler içi kareler toplamında önemli bir düşüş görülmekte, bu noktalardan sonra ise düşüş miktarı azalmaktadır. $k=2$ ve $k=6$ noktalarından sonra grafikteki düşüş miktarı azaldığı ve yatay seyretmeye başladığından bu veri setini iki kümeye ayırmanın uygun olduğu belirlenmiştir.



Şekil 4.7. Elbow yöntemine göre küme sayısının belirlenmesi

Elbow yöntemi dirsek noktasının belirlenmesindeki öznellikten dolayı her zaman doğru sonuç vermez. Daha düz bir eğrinin olduğu, k değerini belirlemenin belirsiz olduğunda optimal k sayısını belirlemek için Silhouette indeksi gibi farklı yöntemler denenebilir (Gove, 2022).

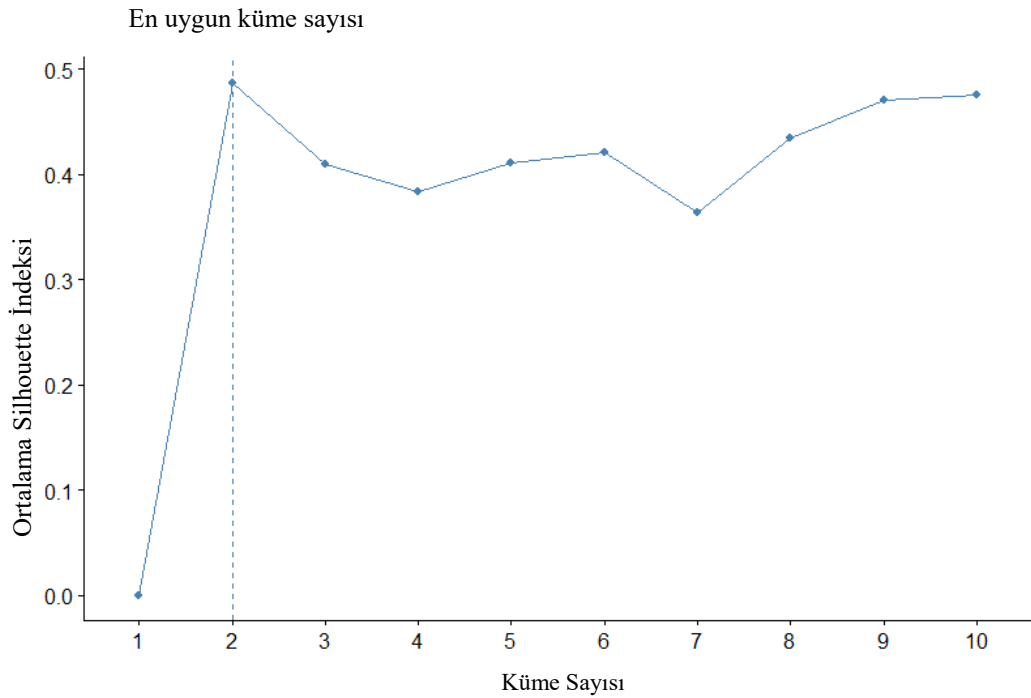
Şekil 4.7 incelendiğinde dirsek noktasının belirlenmesinde belirsizlik yaşanmaktadır. Bu sebeple optimal küme sayını belirlemek için literatürde de sıklıkla kullanılan Silhouette indeksi kullanılmıştır.

Silhouette indeksinin R dilinde kodlanmış hali Şekil 4.8’de gösterildiği gibidir.

```
# Silhouette yöntemi  
fviz_nbclust(df, kmeans, method = "silhouette")+  
labs(subtitle = "Silhouette method")
```

Şekil 4.8. Silhouette indeksinin uygulanması

Uygulama sonucunda Silhouette indeksi 0,49 olarak bulunmuştur. Bu değer [-1,+1] aralığında bulunduğu için kümeleme işlemi başarılı olarak kabul edilebilir. Silhouette indeksine göre elde edilen grafik Şekil 4.9’da verilmiştir. Program çıktısına göre optimal küme sayısı iki olarak elde edilmiştir.



Şekil 4.9. Silhouette indeksine göre küme sayısının belirlenmesi

Sektör uzmanlarıyla yapılan görüşme sonucunda üzerinde çalışan envanter kalemlerini üç gruba ayırmanın daha uygun olacağı belirtilmiştir. Ancak tez kapsamında envanter kalemlerinin hem iki kümeye hem de üç kümeye ayrılmış haline yer verilmiştir.

4.3. Araç Rotalama Probleminin Çözümü

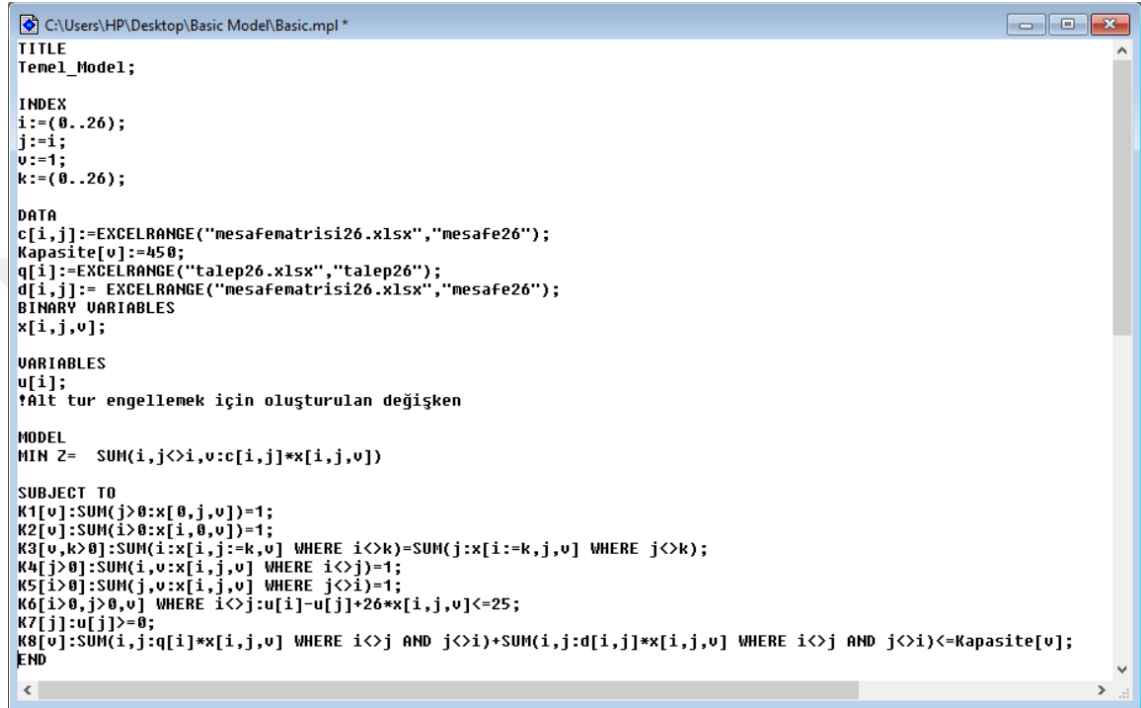
Murena ve ark. (2020), otomatik satış makinelerinin daha etkili çalışabilmesi için geliştirdikleri kontrol sisteminde makinelerin alarm sistemi, ürün dağıtımı, soğutma ve ödeme sistemi fonksiyonlarını takip ve kontrol etmişlerdir.

Tez çalışması kapsamında ele alınan otomatik satış makinelerinin yönetim sisteminde mevcut durumda araç rotalarının belirlenmesi için herhangi bir yöntem kullanılmamaktadır. Sistem dahilinde bulunan dağıtım personeli gün içindeki yapacağı ziyaret planını kendi tecrübelerine göre belirlemektedir. Personel kendi deneyimine dayanarak en çok satış yapan makineleri önceliklendirerek bir ziyaret planı oluşturmaktadır. Bu durumda makine arızası, kasadaki para miktarının dolu olduğu durum, ürünlerin tükendiği ve sağlıklı ürünlerin raf ömrünün tükendiği durumlar kısa sürede fark edilememektedir. Bu olumsuz durumun yaratacağı kayıpları önlemek adına makinelerdeki para miktarının dolduğu, makinede arıza meydana geldiği, makinedeki ürün/ürünlerin bittiği, sağlıklı ürünlerin raf ömrünün tükendiği durumda sinyal ileten bir sistem olduğunda araç rotasının nasıl olması gerektiğine karar veren bir karar sisteminin oluşturulması amaçlanmıştır. Bu amaç doğrultusunda literatürde var olan otomatik satış makineleri yönetimi ve araç rotalama problemleri incelenmiştir. Tez kapsamında ele alınan sisteme uygun araç rotasının belirlenebilmesi için ilk olarak kapasite kısıtlı araç rotalama problemine uygun bir matematiksel model oluşturulmuştur. Ardından model, problem ihtiyacına uygun olarak araçların sadece sinyal gelen birimlere uğramasını sağlayacak şekilde geliştirilmiştir. Matematiksel model problem çözümü için yetersiz kaldığından sistem ihtiyacına uygun olarak üçüncü bölümde algoritma adımları verilen araç rotalama algoritması oluşturulmuştur.

Matematiksel modeli uygulama aşamasında gerekli olan otomatik satış makineleri arasındaki uzaklık için Google Haritalar kullanılmıştır. Harita üzerinden istenen düğümler işaretlenerek düğümler arası mesafe matrisi oluşturulmuştur. 30 düğüm için oluşturulan uzaklık matrisi EK-7’de gösterilmiştir. Matriste yer alan değerler kilometre cinsindedir.

Bölüm 3.2’de anlatılan matematiksel modeller Mathematical Programming Language (MPL)’in Gurobi çözücüsü yardımıyla 2,5 Ghz ve 8 GB belleğe sahip bir bilgisayarda çözdürülmüştür.

Kapasite kısıtlı araç rotalama problemi için oluşturulan temel modelin 26 birim için MPL’de kodlanmış hali Şekil 4.10’da gösterilmiştir.



```
C:\Users\HP\Desktop\Basic Model\Basic.mpl *
TITLE
Temel_Model;

INDEX
i:=(0..26);
j:=i;
v:=1;
k:=(0..26);

DATA
c[i,j]:=EXCELRange("mesafematrisi26.xlsx","mesafe26");
Kapasite[v]:=450;
q[i]:=EXCELRange("talep26.xlsx","talep26");
d[i,j]:= EXCELRange("mesafematrisi26.xlsx","mesafe26");
BINARY VARIABLES
x[i,j,v];

VARIABLES
u[i];
!Alt tur engellemek için oluşturulan değişken

MODEL
MIN Z= SUM(i,j<>i,v:c[i,j]*x[i,j,v])

SUBJECT TO
K1[v]:SUM(j>0:x[0,j,v])=1;
K2[v]:SUM(i>0:x[i,0,v])=1;
K3[u,k>0]:SUM(i:x[i,j:=k,v] WHERE i<>k)=SUM(j:x[i:=k,j,v] WHERE j<>k);
K4[j>0]:SUM(i,v:x[i,j,v] WHERE i<>j)=1;
K5[i>0]:SUM(j,v:x[i,j,v] WHERE j<>i)=1;
K6[i>0,j>0,v] WHERE i<j:u[i]-u[j]+26*x[i,j,v]<=25;
K7[j]:u[j]>=0;
K8[v]:SUM(i,j:q[i]*x[i,j,v] WHERE i<j AND j<i)+SUM(i,j:d[i,j]*x[i,j,v] WHERE i<j AND j<i)<=Kapasite[v];
END
```

Şekil 4.10. Temel modelin MPL kodu

Oluşturulan model 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet talep noktası için test edilmiştir. Model 26 birime kadar sonuç verirken 26 birim sonrası için sonuç verememiştir. Modelin kodlanmasıyla 26 birim için MPL’den elde edilen sonuç ekranı Şekil 4.11’de gösterilmiştir. Sonuç ekranına göre aracın kat ettiği toplam mesafe 48,8854 kilometredir. Araçlar tüm birimlere uğramaktadırlar.

View File: Basic.sol

Optimal solution found

MIN Z = 48.8854

DECISION VARIABLES

VARIABLE x[i,j,v] :

i	j	v	Activity	Reduced Cost
0	25	1	1.0000	0.0000
1	17	1	1.0000	0.0000
2	16	1	1.0000	0.0000
3	11	1	1.0000	0.0000
4	10	1	1.0000	0.0000
5	21	1	1.0000	0.0000
6	2	1	1.0000	0.0000
7	14	1	1.0000	0.0000
8	22	1	1.0000	0.0000
9	13	1	1.0000	0.0000
10	12	1	1.0000	0.0000
11	24	1	1.0000	0.0000
12	3	1	1.0000	0.0000
13	0	1	1.0000	0.0000
14	5	1	1.0000	0.0000
15	20	1	1.0000	0.0000
16	9	1	1.0000	0.0000
17	18	1	1.0000	0.0000
18	4	1	1.0000	0.0000
19	1	1	1.0000	0.0000
20	6	1	1.0000	0.0000
21	15	1	1.0000	0.0000
22	7	1	1.0000	0.0000
23	8	1	1.0000	0.0000
24	26	1	1.0000	0.0000
25	19	1	1.0000	0.0000
26	23	1	1.0000	0.0000

Şekil 4.11. Temel modelin MPL sonucu

Oluşturulan model 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet talep noktası için test edilmiştir. Bu talep noktalarının tümünü kapsayan rotalar Çizelge 4.7’de gösterilmiştir.

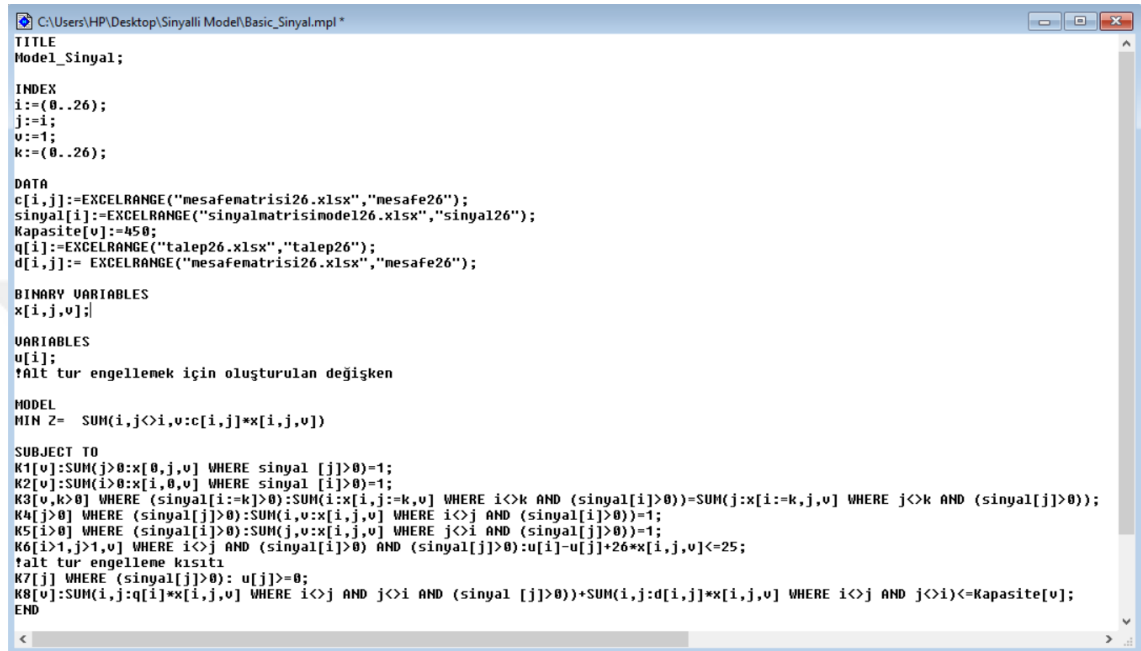
Çizelge 4.7. Temel model için birim sayısına göre araç rotaları

Müşteri Sayısı	Aracın Rotası	Toplam Mesafe (km)
8	0-3-8-6-2-7-5-4-1-0	52,3870
10	0-3-8-6-2-9-7-5-10-4-1-0	54,223
12	0-1-4-10-5-7-9-2-6-8-11-12-3-0	54,452
14	0-1-4-10-12-3-11-8-5-14-7-6-2-9-13-0	45,211
16	0-1-4-10-12-3-11-15-5-14-7-8-6-2-16-9-13-0	45,6004
18	0-1-17-18-4-10-12-3-11-15-5-14-7-8-6-2-16-9-13-0	45,5764
20	0-19-1-17-18-4-10-12-3-11-8-7-14-5-15-20-6-2-16-9-13-0	44,7014
22	0-19-1-17-18-4-10-12-3-11-8-22-7-14-5-21-15-20-6-2-16-9-13-0	47,3024
24	0-13-9-16-2-6-20-15-21-5-14-7-22-8-23-24-11-3-12-10-4-18-17-1-19-0	48,1544
26	0-25-19-1-17-18-4-10-12-3-11-24-26-23-8-22-7-14-5-21-15-20-6-2-16-9-13-0	48,8854

Çizelge 4.7’ de gösterilen rotalarda “0” ile belirtilen birim depoyu ifade etmektedir. Depo dışında aracın uğradığı diğer birimler birden başlanarak numaralandırılmıştır.

Çizelge 4.7 incelendiğinde 12 birimin ziyaret edildiği durumda aracın katettiği toplam mesafe 54,452 kilometre iken 14 birimin ziyaret edildiği durumda aracın katettiği toplam mesafenin 45,211 kilometre olduğu görülmektedir. Burada birim sayısı artarken gidilen mesafenin azalması rota değişikliğinden dolayı kullanılan yolların değişmesinden kaynaklanmaktadır. Söz konusu durum EK-8’de açıklanmaktadır.

Oluşturulan matematiksel model ele alınan problem için 26 talep noktasına kadar optimum çözümleri verebilmiştir. Ancak ele alınan problemde ziyaretleri gerçekleştirecek olan aracın sadece sinyal olan makinelere uğraması istenmektedir. Bu amaç doğrultusunda oluşturulan matematiksel model MPL’de kodlanmıştır. MPL koduna ait ekran görüntüsü Şekil 4.12’de sunulmuştur.



```

CAUsers\HP\Desktop\Sinyalli Model\Basic_Sinyal.mpl
TITLE
Model_Sinyal;

INDEX
i:=(0..26);
j:=i;
v:=1;
k:=(0..26);

DATA
c[i,j]:=EXCELRange("mesafenetrisi26.xlsx","mesafe26");
sinyal[i]:=EXCELRange("sinyalnetrisi26.xlsx","sinyal26");
Kapasite[v]:=450;
q[i]:=EXCELRange("talep26.xlsx","talep26");
d[i,j]:= EXCELRange("mesafenetrisi26.xlsx","mesafe26");

BINARY VARIABLES
x[i,j,v];

VARIABLES
u[i];
!alt tur engellemek için oluşturulan değişken

MODEL
MIN Z= SUM(i,j<i,v:c[i,j]*x[i,j,v])

SUBJECT TO
K1[v]:SUM(j>0:x[i,j,v] WHERE sinyal [j]>0)=1;
K2[v]:SUM(i>0:x[i,0,v] WHERE sinyal [i]>0)=1;
K3[v,k>0] WHERE (sinyal[i:=k]>0):SUM(i:=k,v] WHERE i>k AND (sinyal[i]>0))=SUM(j:=k,v] WHERE j<k AND (sinyal[j]>0));
K4[j>0] WHERE (sinyal[j]>0):SUM(i,v:x[i,j,v] WHERE i>j AND (sinyal[i]>0))=1;
K5[i>0] WHERE (sinyal[i]>0):SUM(j,v:x[i,j,v] WHERE j<i AND (sinyal[j]>0))=1;
K6[i>1,j>1,v] WHERE i>j AND (sinyal[i]>0) AND (sinyal[j]>0):u[i]-u[j]+26*x[i,j,v]<=25;
!alt tur engelleme kısıtı
K7[j] WHERE (sinyal[j]>0): u[j]>=0;
K8[v]:SUM(i,j:=q[i]*x[i,j,v] WHERE i>j AND j<i AND (sinyal [j]>0))+SUM(i,j:=d[i,j]*x[i,j,v] WHERE i>j AND j<i)<=Kapasite[v];
END

```

Şekil 4.12. Sinyal durumunun olduğu matematiksel model kodu

Şekil 4.12’de anlatılan modelde kullanılan uzaklık matrisi ve sinyal bilgisi verisi Excel’den okutulmuştur. Oluşturulan model her aracın sadece sinyal geldiği birimlere uğramasını sağlamaktadır.

Sinyal durumunun eklendiği model 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet birim için test edilmiştir. Model yirmi altı birime kadar optimum sonucu verebilmiştir. Yirmi altı birim için MPL’den elde edilen sonucun ekran görüntüsü Şekil 4.13’deki gibidir. MPL sonucuna göre 26 birim için gidilecek mesafe en az 23,8254 kilometredir. Şekil 4.13’deki sonuca göre araçların 1, 4, 6, 7, 10, 11, 12, 15, 17, 19 ve 24 numaralı birimlere uğramamasının sebebi o düğümlerde o anda herhangi bir sinyal bulunmamasıdır.

View File: Basic_Sinyal.sol

Optimal solution found

MIN Z = 23.8254

DECISION VARIABLES

VARIABLE x[i,j,v] :

i	j	v	Activity	Reduced Cost
0	13	1	1.0000	0.0000
2	20	1	1.0000	0.0000
3	25	1	1.0000	0.0000
5	21	1	1.0000	0.0000
8	23	1	1.0000	0.0000
9	16	1	1.0000	0.0000
13	9	1	1.0000	0.0000
14	5	1	1.0000	0.0000
16	2	1	1.0000	0.0000
18	3	1	1.0000	0.0000
20	22	1	1.0000	0.0000
21	18	1	1.0000	0.0000
22	8	1	1.0000	0.0000
23	14	1	1.0000	0.0000
25	0	1	1.0000	0.0000

Şekil 4.13. Sinyal durumunun eklendiği modelin sonucu

Sinyal durumunun eklendiği model 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet birim için test edilmiştir. Birim sayısına göre elde edilen rotalar ve bu rotaların uzunluğu Çizelge 4.8’de gösterilmiştir. Modele sinyal durumu dahil edilerek aracın ziyarete ihtiyacı olmayan birimlere uğraması engellenmiştir. Çizelge 4.8’de sekiz birim için oluşturulan rota incelendiğinde aracın herhangi bir sinyal gelmeyen 1, 4, 6 ve 7 numaralı birimlere uğramadığı görülmektedir. Oluşturulan temel modelde araç bu birimlere de uğramakta dolayısıyla iş gücü verimsiz kullanılmaktaydı.

Çizelge 4.8. Sinyal durumunun olduğu modelde birim sayısına göre araç rotaları

Müşteri Sayısı	Araçın Rotası	Toplam Mesafe (km)
8	0-3-5-2-8-0	28,6900
10	0-3-5-9-2-8-0	30,0200
12	0-3-5-9-2-8-0	30,0200
14	0-13-2-9-8-5-14-3-0	18,729
16	0-3-14-5-8-9-2-16-13-0	18,7164
18	0-18-3-14-5-8-9-2-16-13-0	20,6564
20	0-18-3-14-5-8-20-2-16-9-13-0	21,3164
22	0-3-18-21-5-14-8-22-20-2-16-9-13-0	21,9344
24	0-13-9-16-2-20-22-8-21-5-14-23-3-18-0	22,1474
26	0-13-9-16-2-20-22-8-23-14-5-21-18-3-25-0	23,8254

Matematiksel modeller MPL’de kodlanıp farklı birim sayıları için sonuçlar elde edildikten sonra iki modelin sonuçları karşılaştırılmıştır. Elde edilen karşılaştırma tablosu Çizelge 4.9’da gösterilmiştir. Çizelge 4.9 incelendiğinde sinyal durumu eklenerek geliştirilen modelin temel modelden ortalama %51,82 daha iyi sonuç verdiği görülmektedir. Temel modelde araç sistemde bulunan her birime en kısa mesafeyi kat edecek şekilde uğramaktadır. Sinyal durumu eklenerek geliştirilen model sadece sinyal bilgisinin geldiği birimlere uğradığından hizmet almaya ihtiyacı olmayan makinelere uğramayacak dolayısıyla araçların gittiği mesafeler kısalarak maliyet azaltılacaktır.

Çizelge 4.9. İki model sonucunun karşılaştırılması

Müşteri Sayısı	Temel Model Sonucu (km)	Geliştirilen Model Sonucu (km)	İyileşme Oranı
8	52,3870	28,6900	45,23%
10	54,2230	30,0200	44,64%
12	54,4520	30,0200	44,87%
14	45,2110	18,7290	58,57%
16	45,6004	18,7164	58,96%
18	45,5764	20,6564	54,68%
20	44,7014	21,3164	52,31%
22	47,3024	21,9344	53,63%
24	48,1544	22,1474	54,01%
26	48,8854	23,8254	51,26%

Sinyal durumu eklenerek geliştirilen matematiksel model 26 birim için çözüm üretebilmiştir. Ele alınan 30 birimlik problem için yetersiz kalmıştır. Bu sebepten dolayı tez kapsamında ele alınan problemi çözebilmek için en yakın komşu algoritmasına dayalı sezgisel bir araç rotalama algoritması oluşturulmuştur. Oluşturulan algoritma Visual Studio 2019 programında Visual Basic programlama dili ile kodlanmıştır. Visual Basic dilinde kodlanmış algoritma EK-10’da gösterilmiştir. Algoritma için tez çalışmasında kullanılan veri setinin genel özellikleri aşağıda belirtilmiştir:

- Depo ile birlikte toplamda otuz bir adet birim bulunmaktadır.
- Firmanın dağıtımda kullanacağı iki adet araç bulunmaktadır.
- Günlük mesai süresi dört yüz elli dakika olarak kabul edilmiştir.

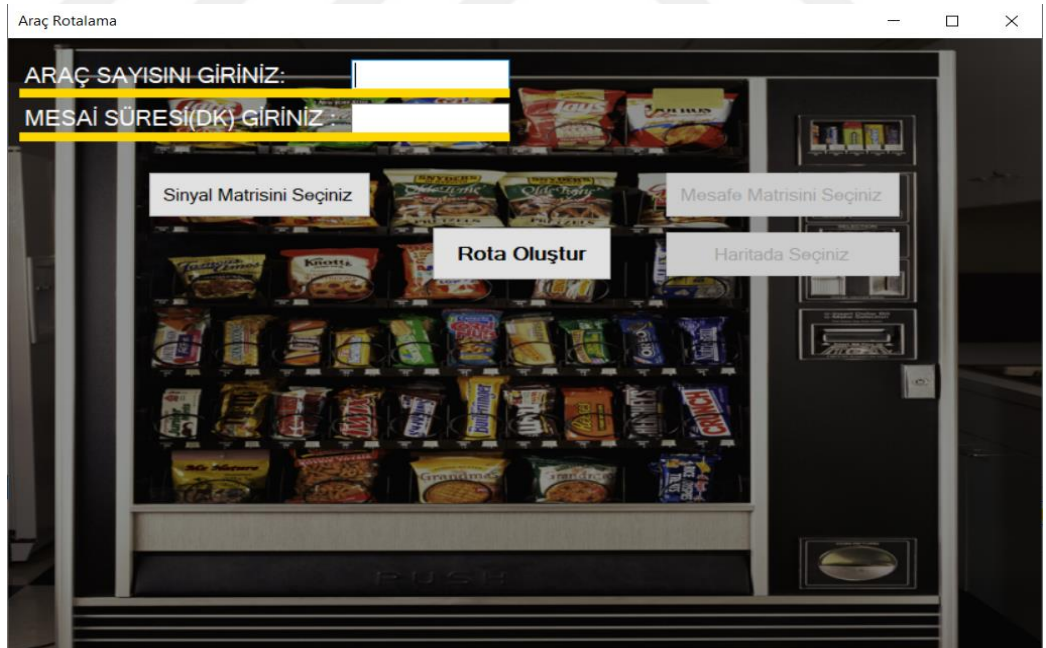
Tez çalışmasında, bulut tabanlı sistemde makinelere takılacak akıllı cihazlar ile Nesnelerin İnterneti (Internet of Things-IoT) teknolojisi kullanılarak otomatik satış makinelerinden karar sistemine anlık veri sağlanmaktadır. Bu veriler makinede oluşan arıza bilgisini, kasadaki para miktarının doluluğunu, bitmiş ürün durumunu ve son tüketim tarihini tamamlamış sağlıklı ürünlerle ilgili bilgileri içermektedir. Sayılan kriterlere ait bilgiler Excel’de sinyal matrisinde tutulmaktadır.

Çalışmada kullanılan sinyal matrisi EK-9’de gösterilmiştir. Bu matriste birimler sütununda “0” ile belirtilen birim depodur. Depo dışındaki birimler 1’den 30’a kadar

numaralandırılmıştır. Para miktarının otomatik satış makinesinin kasasını tam olarak doldurduğu durumda o makineye ait para miktarı bilgisi “1” değerini almaktadır. Eğer para miktarıyla ilgili bir sorun yoksa “0” değerini almaktadır. Diğer kriterler (bitmiş ürün, arıza bilgisi, son tüketim tarihini tamamlamış sağlıklı ürün) için de aynı süreç işlemektedir. Kriterlere ait sinyaller geldikçe matriste ilgili bölüm “1” değerini almaktadır.

Oluşturulan algoritma gün içinde ziyaret yapacak araçların, belirlenmiş mesai süresini aşmadan, sadece sinyal (para miktarı, ürün miktarı, arıza bilgisi, sağlıklı ürün) gelen makinelere uğramasını sağlamaktadır. Başlangıçta depodan çıkan araç depoya en yakın ve sinyal gelen makineye uğrar. Burada görevini tamamlayan araç bu makineye en yakın ve sinyal gelen makineye gider. Mesai süresinin aşılması için algoritma bir taraftan aracın yolculuk süresini ve birimlerdeki hizmet süresini toplar. Mesai süresi veya sinyal gelen birim sayısı tükenene kadar bu döngü devam eder.

Algoritmanın uygulanması için Visual Studio’da Şekil 4.14’de gösterilen ara yüz tasarlanmıştır.

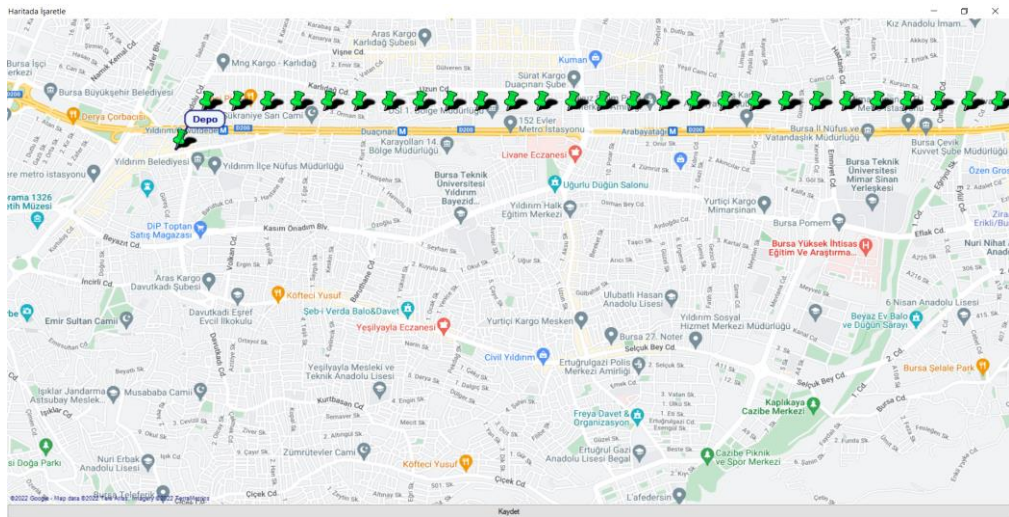


Şekil 4.14. Makine ziyareti planlama algoritmasının kullanıcı ara yüzü

Kullanıcı manuel olarak sistemde uygun olan araç sayısını, günlük çalışma süresini girmektedir. Tez çalışması için bir vardiya süresi 450 dakika olarak alınmıştır. Kullanıcı

araç sayısı ve günlük çalışma süresini girdikten sonra sinyal matrisi ve mesafe matrisinin bulunduğu Excel dosyalarını sırasıyla seçip algoritmayı çalıştırır. Kullanıcı algoritmanın çalışması için gerekli bilgileri girdikten sonra “Rota Oluştur” butonuna basar. Algoritma sonucunda oluşturulan rota yine aynı ara yüzde görülür.

Mesafe matrisinin yüklenmesine alternatif olarak arayüze “Haritada seçiniz” butonu eklenmiştir. Sisteme yeni eklenen, çıkarılan veya konumu değişen birimler olması durumunda kullanıcı mesafe matrisini bu butona bastıktan sonra oluşturabilir. Mesafe matrisinin oluşturulması için arayüzde GMap.Net kütüphanesinden yararlanılarak birimler ve depo haritadan seçilerek mesafe matrisi arka planda oluşturulmaktadır. Böylece, kullanıcının depolar ve müşteriler arası mesafeyi tek tek hesaplama zahmetinin azaltılması amaçlanmıştır. Ek olarak, müşteri eklenip çıkarıldığında veya müşterilerin konumu değiştiğinde de kullanıcının mesafe matrisini tekrar hesaplayıp yeni bir Excel dosyası oluşturmaya gerek kalmayacaktır. Arayüzden “Haritada seçiniz” butonuna tıkladığında her bir müşteri ve depo için Şekil 4.15’de belirtilen haritada işaret belirmektedir. Bu işaretleri haritada ilgili yere sürükleyip bırakarak konumları belirtilmektedir. Tüm birimler ve depo konumlarından emin olunca “Kaydet” butonuna basıldığında her ikili konumun mesafesi alınıp arka planda mesafe matrisi hesaplanmaktadır. Mesafe matrisinin oluşturulması amacıyla yazılan kod EK-11’dadır.



Şekil 4.15. Haritadan mesafe matrisi elde etme

Algoritma çalıştırıldıktan sonra elde edilen ara yüz ekranı Şekil 4.16’da gösterilmiştir. Sonuç ekranında her bir aracın uğraması gereken makineler sırasıyla gösterilmiştir.

Bunun yanında sonuç ekranında iki birim arasındaki mesafe, o birimin kaçınıcı sırada ziyaret edildiği ve aracın gittiği mesafe bilgileri de yer almaktadır.

Araç Rotalama

ARAÇ SAYISINI GİRİNİZ: 2

MESAI SÜRESİ(DK) GİRİNİZ: 450

Sinyal Matrisini Seçiniz

Mesafe Matrisini Seçiniz

Rota Oluştur

Haritada Seçiniz

ARAÇLARIN ROTALARI:

Araç No	Sıra	Müşteri	Uzaklık	Gidilen Mesafe	Anıza Tipi
1	1	Depo	0	0	Depodan Cıkıs
1	2	Müşteri_13	0,814	0,814	X
1	3	Müşteri_16	0,94	1,754	X
1	4	Müşteri_2	0,0574	1,8114	X
1	5	Müşteri_27	0,176	1,9874	X
1	6	Müşteri_20	1,19	3,1774	X
1	7	Müşteri_30	2,45	5,6274	X
1	8	Müşteri_9	0,109	5,7364	X
1	9	Müşteri_22	9,26	14,9964	X
1	10	Müşteri_8	1,28	16,2764	X

Şekil 4.16. Rota sonucunun paylaşıldığı ekran

Yukarıda bahsedilen algoritmanın Visual Basic dilinde kodlanması sonucu ele alınan problem için elde edilen çözüm Çizelge 4.10'da gösterilmiştir. Oluşturulan rota iki araç için bir vardiya süresi içinde geçerlidir.

Çizelge 4.10. Araç rotalama algoritmasına göre 30 birim için araç rotaları

Müşteri Sayısı	1. Aracın Rotası	1. Aracın Gittiği Mesafe (km)	2. Aracın Rotası	2. Aracın Gittiği Mesafe (km)	Toplam Gidilen Mesafe (km)
30	0-13-16-2-27-20-30-9-22-8-23-14-5-21-0	22,1904	0-25-3-18-29-0	11,32	33,5104

Çizelge 4.10'daki algoritma sonucuna göre birinci araç 13. birimden başlayıp 13 birime uğrayıp depoya dönmektedir. İkinci araç ise depodan başlayıp dört birime uğrayıp tekrar depoya dönmektedir. Algoritmanın çalışma mantığında ilk araç dolmadan ikinci araç hizmete başlamamaktadır. Bu durum ikinci aracın hizmet vermesine gereksinim olmadığı durumda avantaj sağlamaktadır. Sadece bir araçla hizmet verilebilecek durumda ikinci araç hizmet vermeyecek ve böylece maliyet azalacaktır.

Algoritma her iki araç için de hizmet sürelerini hesaplamaktadır, İki araç içinde hizmet süresi 450 dakikayı geçmemektedir. Algoritma oluşturulurken arıza olan birimdeki hizmet süresi yaklaşık olarak 30 dakika, diğer sinyal bilgilerinin olduğu birimler için 10 dakika olarak alınmıştır. Aracın bir kilometrelik mesafeyi bir dakikada gittiği varsayılmaktadır. Burada 30 birimin her birine uğranmamasının sebebi bazı birimlerden herhangi bir sinyal gelmemesidir.

Oluşturulan algoritma 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet birim için test edilmiştir. Birim sayılarına göre her araç için elde edilen rotalar Çizelge 4.11'de gösterilmiştir. 26 birime kadar günlük ziyaret için bir aracın kullanılması yeterli olacaktır. Birinci aracın harcadığı toplam süre mesai süresini aşmadığı için ikinci aracın hizmete açılmasına gerek kalmayacaktır. Böylece iş gücünden tasarruf elde edilmiş olacaktır. Tablodaki verilerde 10 birimlik veri seti ile 12 birimlik veri setinin sonuçları arasında bir fark olmamasının sebebi 10., 11. ve 12. makinelerden herhangi bir sinyal gelmemesinden dolayı araçların o birimlere uğramamasıdır.

Çizelge 4.11'de müşteri sayılarına göre rotaları gösterilen araç, depodan çıkıp depoya en yakın olan ve herhangi bir sinyali olan müşteriye uğrar. Bir mesai saati içinde en az bir sinyali olan makinelere uğrayıp rotanın sonunda yine depoya döner.

Çizelge 4.11. Birim sayılarına göre her bir aracın rotası

Müşteri Sayısı	Aracın Rotası	Toplam Mesafe (km)
8	0-3-8-5-2-0	29,71
10	0-3-8-5-2-0	31,04
12	0-3-8-5-2-0	31,04

Çizelge 4.11. Birim sayılarına göre her bir aracın rotası (devam)

Müşteri Sayısı	Aracın Rotası	Toplam Mesafe (km)
14	0-13-2-9-8-14-5-3-0	18,779
16	0-13-16-2-9-8-14-5-3-0	18,7664
18	0-13-16-2-9-8-14-5-3-18-0	20,7064
20	0-13-16-2-20-9-8-14-5-3-18-0	22,9664
22	0-13-16-2-20-9-22-8-14-5-21-3-18-0	23,8144
24	0-13-16-2-20-9-22-8-23-14-5-21-3-18-0	24,0554
26	0-13-16-2-20-9-22-8-23-14-5-21-3-18-25-0	26,7954

Makinelerdeki arıza durumunun geç fark edilmesi ve çözüm bulunamaması otomatik satış makinelerinin yönetim sistemi için maliyet açısından fazlaca kayıp yaratmaktadır. Dolayısıyla bir otomatik satış makinesi ağı yöneticisi için arıza bilgisi sinyali diğer sinyallere (ürün miktarı, para miktarı, sağlıklı ürün) göre daha önemlidir. Arıza sinyalinin geldiği makineye diğer makinelerden daha erken ziyaret edilip arızasının giderilebilmesi için, oluşturulan sezgisel algoritmaya arıza bilgisi sinyalinin öncelikli olması durumu dahil edilerek geliştirilmiştir. Geliştirilen algoritma gün içinde ziyarete çıkacak araçlar için ilk olarak arıza bilgisi sinyali gelen makinelerle uğramasını sağlamaktadır. Arızası bulunan tüm makinelere ziyaretini tamamlayan araç son ziyaret ettiği makineye en yakın bulunan ve para miktarı, ürün miktarı, sağlıklı ürün raf ömrü sinyallerinden en az birinin bulunduğu makineye gidecektir. Bahsedilen süreç bir vardiya süresi içinde ve tüm araçlar için geçerlidir. Arıza bilgisinin hiç olmadığı durumda araçlar diğer sinyallerin geldiği makinelere en yakın komşu algoritması mantığıyla uğrayacaklardır.

Algoritmaya göre bir birimde arıza bilgisi sinyalinin yanı sıra diğer sinyaller de var ise araç o birime uğradığında tüm sinyaller için gerekli hizmeti verecektir. Örneğin arıza bilgisi sinyali gelen bir makinede aynı zamanda ürün miktarının bittiğine dair bir sinyal varsa görevli o makineye gittiğinde hem arızayı giderecek hem de bitmiş ürünlerin doldurma işlemini gerçekleştirecektir. Görevlinin bu makinedeki hizmet süresi arızayı giderme süresi ve ürün doldurma işlemini yaptığı sürenin toplamı olacaktır.

Arıza bilgisi sinyali önceliklendirilerek geliştirilen algoritmanın kodu EK-12

'de verilmiştir.

Geliştirilen algoritmanın tez kapsamında ele alınan problem için uygulanması sonucu elde edilen rota Çizelge 4.12’de gösterilmiştir. Problemden toplamda 30 birim bulunmaktadır. Bu 30 birimin 17 tanesinden sinyal gelmektedir. Çizelge 4.12’ye göre birinci araç bir vardiya süresi boyunca depodan başlayıp 12 birime uğrayabilmektedir. Bu 12 birim dışında kalan ve sinyal gelen diğer birimlere ikinci araç uğramaktadır. İki aracın toplamda kat ettikleri mesafe 35,0634 kilometredir.

Çizelge 4.12. Arıza bilgisinin öncelikli olduğu durumda 30 birim için elde edilen rota

Müşteri Sayısı	1. Aracın Rotası	1. Aracın Gittiği Mesafe (km)	2. Aracın Rotası	2. Aracın Gittiği Mesafe (km)	Toplam Mesafe (km)
30	0-13-16-2-27-20-30-9-5-21-29-14-23-0	21,4734	0-25-3-8-22-18-0	13,59	35,0634

Arıza bilgisi sinyalinin diğer sinyallere göre önceliklendirilmesiyle oluşturulan algoritma 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet birimden oluşan veri setlerinde test edilmiştir. Her bir veri seti için elde edilen rotalar ve araçların bu rotalar boyunca kat ettikleri yol uzunlukları Çizelge 4.13’te gösterilmiştir.

Çizelge 4.13. Öncelikli durum için birim sayılarına göre araç rotaları

Müşteri Sayısı	Aracın Rotası	Toplam Mesafe (km)
8	0-5-2-8-3-0	28,71
10	0-5-9-2-8-3-0	30,04
12	0-5-9-2-8-3-0	30,04
14	0-13-2-9-5-14-8-3-0	19,139
16	0-13-16-2-9-5-14-8-3-0	19,1264
18	0-13-16-2-9-5-14-8-3-18-0	21,0664
20	0-13-16-2-20-9-5-14-8-3-18-0	23,3264
22	0-13-16-2-20-9-5-21-14-8-22-3-18-0	26,3414
24	0-13-16-2-20-9-5-21-14-23-8-22-3-18-0	26,5824
26	0-13-16-2-20-9-5-21-14-23-8-22-3-18-25-0	29,3224

Oluşturulan algoritmanın sonucunun değerlendirilmesi amacıyla matematiksel modelde kullanılan test verileri algoritma için de kullanılmıştır. Matematiksel model sistemde en fazla 26 adet otomatik satış birimin olduğu duruma kadar bir rota oluşturabilmektedir. Bu yüzden algoritma sonucuyla matematiksel model sonucunu karşılaştırabilmek adına 8, 10, 12, 14, 16, 18, 20, 22, 24 ve 26 adet birimden oluşan veri setleri kullanılmıştır. Bu birimler için matematiksel model ve algorithmadan elde edilen sonuçlar Çizelge 4.14’de gösterilmiştir.

Çizelge 4.14 incelendiğinde 14, 16, 18, 20, 22, 24 ve 26 birimlik veri setleri için matematiksel modelin ulaşılabilen en iyi sonucu verdiği görülmektedir. Oluşturulan algoritma hizmet aracının öncelikle arıza bilgisinin sinyalinin olduğu ardından da diğer sinyallerin olduğu makinelerle gitmesini sağlamaktadır. Tez kapsamında ele alınan problem için önemli olan konu hizmet veren aracın önce arıza bilgisi veren ardından da diğer sinyallerin geldiği makinelere ziyaret gerçekleştirilmesidir.

Çizelge 4.14. Matematiksel model ve algoritma sonuçlarının karşılaştırılması

	Matematiksel Model	Öncelik Olmayan Algoritma	Arızanın Öncelikli Olduğu Algoritma
Birim Sayısı	Toplam Mesafe (km)	Toplam Mesafe (km)	Toplam Mesafe (km)
8	28,6900	29,7100	28,7100
10	30,0200	31,0400	30,0400
12	30,0200	31,0400	30,0400
14	18,7290	18,7790	19,1390
16	18,7164	18,7664	19,1264
18	20,6564	20,7064	21,0664
20	21,3164	22,9664	23,3264
22	21,9344	23,8144	26,3414
24	22,1474	24,0554	26,5824
26	23,8254	26,7954	29,3224

Oluşturulan algoritma en yakın komşu algoritması temeline dayanmaktadır. Depodan rotaya başlayan araç ilk olarak depoya en yakın birime uğrar. Ardından sırasıyla bulunduğu birime en yakın birimlere uğrayarak depoya döner. Burada aracın uğradığı son birimin depoya olan uzaklığı rotanın uzunluğunu etkilemektedir. En son ziyaret edilen birim depoya yakınsa rota uzunluğu kısa olurken en son ziyaret edilen birimin depoya

uzak olması durumunda rota uzunluğu büyük olmaktadır. Bu durumu iyileştirmek amacıyla en yakın komşu algoritmasının depoya en uzak birimden başlamasına yönelik bir uygulama yapılmıştır. Böylece depodan çıkan araç ilk olarak depoya en uzak birime gidecek ardından sırasıyla bulunduğu birime en yakın birime gidip tüm ziyaretlerini tamamladıktan sonra depoya dönecektir. Bu amaç doğrultusunda sinyal durumunda öncelik olmayan durum için ve arıza bilgisinin öncelikli olduğu durum için oluşturulan algoritma Visual Basic dilinde kodlanmıştır. Sinyal durumunun öncelikli olmadığı algoritmaya ait kod EK-13'te, arıza bilgisinin öncelikli olduğu algoritmaya ait kod EK-14'te verilmiştir.

En uzak birimden rotaya başlayan algoritmayı test etmek için tez çalışmasında kullanılan veri setleri kullanılmıştır. Algoritma sonuçları ilk uygulanan en yakın komşu algoritması sonuçlarıyla karşılaştırılmıştır. Her iki algoritmaya ait veri setlerinden elde edilen rota mesafeleri Çizelge 4.15'de gösterilmiştir.

Çizelge 4.15. En uzak birimden başlanarak oluşturulan algoritma sonuçları

Birim Sayısı	Sezgisel Algoritma (Rotaya en yakın birimden başlanan durum)		Sezgisel Algoritma (Rotaya en uzak birimden başlanan durum)	
	Öncelik Olmayan	AB Öncelikli	Öncelik Olmayan	AB Öncelikli
8	29,71	28,71	29,18	29,71
10	31,04	30,04	30,78	31,31
12	31,04	30,04	30,78	31,31

Çizelge 4.15'de sekiz, on ve on iki birimlik veri setleri için elde edilen rota mesafeleri kilometre cinsinden verilmiştir. Rotaya en uzak birimden başlanmasını sağlayan algoritma 12 birime kadar olan veri setleri için sonucu iyileştirirken 12 birimden daha fazla birime sahip veri setleri için sonucu kötüleştirmiştir. Bu sebepten dolayı yeni geliştirilen algoritma için denemelere 12 birimlik veri setinden sonrası için devam edilmemiştir.

Park ve Yoon (2012), yaptıkları çalışmada otomatik satış makinelerinin yönetimi için iki aşamalı bir yöntem önermişlerdir. Önerdikleri yöntemin ilk aşamasında her bir akıllı otomatik satış makinesi için ürün saklama bölmelerine ürün yerleştirme ve tahsis etme aralıklarını belirlemek için doğrusal olmayan bir matematiksel model kullanmışlardır.

Yöntemin ikinci aşamasında ise otomatik satış makinelerine ürün beslemesi yapan araçların rotasını belirlemek için taşıma ve yok satma maliyetini en azlayan kazanım algoritması temelli bir algoritma kullanmışlardır. Tez çalışmasında otomatik satış makinesindeki ürün saklama bölmelerindeki ürün miktarlarını bitmesi durumu makine üzerinde bulunan akıllı cihazlar sayesinde elde edilmektedir. Dolayısıyla makinelere yüklenecek ürün miktarını belirlemeyle ilgili herhangi bir analitik yöntem kullanılmasına gerek duyulmamaktadır. Yapılan çalışmada otomatik satış makinelerine ürün tedarikini sağlayan depodaki ürünlerin yönetimini kolaylaştırmak adına çok kriterli envanter sınıflandırma yapılmış ve elde edilen sınıflar için envanter yönetim stratejisi önerilmiştir. Otomatik satış makinelerine hizmet veren araçlar için ise en yakın komşu algoritmasına dayalı sezgisel bir yöntem geliştirilmiştir.

Otomatik satış makinelerinin yönetimiyle ilgili Park ve Yoo (2012) yapmış oldukları çalışmada iki aşamalı sezgisel bir yöntem geliştirmişlerdir. Otomatik satış makinelerindeki ürün saklama bölmelerini sayısını ve bu bölmeler için ürün yenileme eşiğini belirlemek için tam sayılı doğrusal matematiksel model kullanmışlardır. Sistemdeki tüm otomatik satış makinelerinin ürün saklama bölmelerinin sayısı ve ürün yenileme eşiği belirlendikten sonra bu ürün yenilemesini yapan ve aynı dağıtım gününü paylaşan araçlar için Clarke ve Wright'ın kazanım algoritmasını kullanmışlardır. Tez çalışmasında araçlar için günlük ziyaret planı oluşturulmuştur. Her bir aracın günlük ziyaret rotasını oluşturabilmek için en yakın komşu algoritmasına dayalı bir sezgisel algoritma geliştirilmiştir. Envanter yönetimi için otomatik satış makinelerinin deposu ele alınmış olup çok kriterli envanter sınıflandırma yapılarak her bir sınıf için ortak envanter yönetim stratejisi belirlenmiştir. Böylece otomatik satış makinelerinde en fazla satılan ürünlerin yönetimi kolaylaşmış ve yok satma maliyetleri en küçüklenmeye çalışılmıştır.

5. SONUÇ

Otomatik satış makineleri kolay ulaşılabilir olmaları ve geniş ürün çeşitliliğine sahip olmaları nedeniyle tüketiciler açısından son zamanlarda oldukça ilgi görür hale gelmiştir. Tüketicinin ihtiyacına uzun mesafeler kat etmeden ulaşması otomatik satış makinelerinin sağladığı en önemli avantajlardan biridir.

Otomatik satış makinelerine olan talep artarken otomatik satış makinelerinin yönetiminin daha etkili hale getirilmesi konusu önem kazanmıştır. Bir otomatik satış makinesi ağının yönetim sürecinde iki faktör söz konusudur. Bunlardan biri otomatik satış makinesine ürün beslemesi yapan deponun yönetimi, diğeri ise otomatik satış makinelerine ürün dağıtımı ve makine bakımı yapan personelin araç rotasının oluşturulmasıdır. Bu süreçlerin etkili yönetilmesi otomatik satış makinelerinin değerini daha da arttıracaktır.

Tez çalışması kapsamında otomatik satış makineleri ağını yöneten bir firma ele alınmıştır. Firmada yapılan gözlemler sonucunda depodaki envanter takibi ve dağıtım için kullanılan araçların rotasının oluşturulması için herhangi bir sistematik yaklaşımın olmadığı görülmüştür.

Depodaki envanter yönetimini gerçekleştirmek adına envanter kalemlerinde çok kriterli ABC sınıflandırmasına dayalı bir yöntem geliştirilmiştir. Aylık satış miktarı, tedarik süresi, birim fiyatı ve sezon kriterleri bulunan ürün gruplarının envanter yönetimini sağlamak için benzer ürünler gruplandırılmıştır. Bu gruplama işlemi için k-ortalamalar yöntemi, hiyerarşik kümeleme ve hibrit kümeleme yöntemi kullanılarak envanter kalemleri iki ve üç gruba ayrılmıştır. Her bir ürün grubu için ekonomik sipariş miktarları ile emniyet stoğunun minimum, maksimum ve ortalama değerleri hesaplanmıştır envanter yönetim stratejisi önerilmiştir. Böylece elli dokuz envanter kalemi için ayrı ayrı envanter yönetimi stratejisi belirlemek yerine iki ve üç küme için ortak envanter yönetimi stratejisi belirlenmiş olacaktır. Her bir küme için belirlenen strateji küme içinde bulunan envanter birimleri için uygulanabilecektir.

Otomatik satış makinelerine ürün dağıtımı ve bakım faaliyetlerini gerçekleştiren personelin araç rotasını belirlemek için bir matematiksel model oluşturulmuştur. Oluşturulan model en fazla yirmi altı birime sahip sistem için optimal çözüm vermiştir. Tez kapsamında ele alınan sistemde otuz birim bulunmaktadır. Dolayısıyla oluşturulan

model ele alınan sistemdeki birimler için bir rota oluşturamamıştır. Ayrıca kullanıcının oluşturulan matematiksel modeli çözdürebilmesi için program satın alması gerekmektedir. Bu durum da ek makliyetle sebep olmaktadır. Bu sebeplerden dolayı kullanıcıya daha kolay ulaşabileceği bir araç sunulmuştur. Makinelere arıza bilgisi, kasa doluluğu (para miktarı), bitmiş ürün ve son tüketim tarihini tamamlamış sağlıklı ürün sinyalleri geldiği durumda araç rotasının nasıl olması gerektiğine karar veren bir algoritma oluşturulmuştur. Oluşturulan algoritma Visual Studio 2019 programında Visual Basic dilinde kodlanmıştır. Algoritma sayesinde araç her birime uğramak yerine sadece sinyallerin geldiği makinelere uğrayacaktır. Böylece gereksiz ziyaretler önlenerek maliyet minimizasyonu sağlanırken hizmet almaya ihtiyacı olan birimlerin ihtiyacına daha kısa sürede cevap verilecektir.

Otomatik satış makinesi yöneticisi için makinelerde arıza olması durumu önem arz etmektedir. Çünkü arızası bulunan makine hizmete açık olmadığı için kayıp satışa neden olmaktadır. Bu durum göz önüne alınarak algoritma geliştirilerek aracın önce arıza sinyali gelen makinelere, ardından diğer sinyallerin geldiği makinelere uğraması sağlanmıştır. Böylece kayıp satıştan kaynaklanan zarar azaltılmış olacaktır.

Algoritma sayesinde aracın sistemde bulunan her birime uğraması yerine sadece sinyal gelen birimlere uğraması sağlanacaktır. Bu durumda hem ziyarete ihtiyacı olan makinenin ihtiyacı en kısa zamanda çözülmüş hem de gereksiz ziyaretlerin önüne geçilmiş olacaktır.

Tez kapsamında yapılan envanter yönetimi ve araç rotalama çalışmalarıyla otomatik satış makineleri yönetimi, yöneticinin insiyatifinden arındırılarak analitik yöntemlerle yönetilebilir hale getirilmiştir.

KAYNAKLAR

- Babayiğit, B., Yıldız, K. (2019). Kapasite kısıtlı araç rotalama problemi için melez bir algoritma. *SETSCI Conference Proceedings, Ankara*, 4 (1), 508-513. http://www.set-science.com/manage/uploads/ISAS2019-ENS_0042/SETSCI_ISAS2019-ENS_0042_0096.pdf
- Birant, D. (2019). Farklı Bağlantı Yöntemleri İle Hiyerarşik Kümeleme Topluluğu, *Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi*, 7(1), 154-164. <https://dergipark.org.tr/tr/download/article-file/664873>
- Cebeci, Z., Yıldız, F., & Kayaalp, G.T. (2015). K-Ortalamlar Kümelemesinde Optimum K Değeri Seçilmesi. Ü. Özen vd. (Ed.), *Bildiriler Kitabı* içinde (s. 231-242). Erzurum: Orka Ofset Matbaacılık. https://www.researchgate.net/profile/Zeynel_Cebeci/publication/282733012_K-Ortalamlar_Kumelemesinde_Optimum_K_Degeri_Secilmesi/links/561a744708aea8036722b405.pdf
- Chen, B., Rhodes, C., Kline, C., & Irvin, L. (2010). Protein Sequence Motif Information Generated by Fuzzy-Hybrid Hierarchical K-means Clustering Algorithm, *International Conference on Bioinformatics & Computational Biology (BIOCOMP'10)*'de sunulmuştur. Nevada-USA. https://www.researchgate.net/publication/221051803_Protein_Sequence_Motif_Information_Generated_by_Fuzzy-Hybrid_Hierarchical_K-means_Clustering_Algorithm
- Chen, Ping., Huang, Hou-kuan., & Dong, Xing-Ye. (2010). Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, 37(2), 1620-1627. <https://www.sciencedirect.com/science/article/abs/pii/S0957417409006071?via%3Dihub>
- Cömert, S. E., Yazgan, H. R., & Görgülü, N. (2019). Eş Zamanlı Topla Dağıt Araç Rotalama Problemi için İki Aşamalı Bir Çözüm Yöntemi Önerisi. *International Journal of Advances in Engineering and Pure Sciences*, 2, 107-125. DOI:[10.7240/jeps.461528](https://doi.org/10.7240/jeps.461528)
- Chelouah, R., ve Siarry, P. (2000). Tabu Search applied to global optimization. *European Journal of Operational Research* , 23(2)256-270. DOI:[10.1016/S0377-2217\(99\)00255-6](https://doi.org/10.1016/S0377-2217(99)00255-6)
- Çevik, K.K. ve Koçer, H. E. (2013). Parçacık Sürü Optimizasyonu ile Yapay Sinir Ağları Eğitimine Dayalı Bir Esnek Hesaplama Uygulaması, *Süleyman Demirel Üniversitesi Fen bilimleri Enstitüsü Dergisi*, 17(2), 39-45. <https://dergipark.org.tr/tr/download/article-file/193819>
- Çınaroğlu, S., & Bulut, H. (2018). K-ortalamlar ve parçacık sürü optimizasyonu tabanlı kümeleme algoritmaları için yeni ilklendirme yaklaşımları, *Mühendislik Mimarlık Fakültesi Dergisi*, 33(2), 413-423. <https://dergipark.org.tr/en/download/article-file/499878>

- Demirtaş, Y. E., & Özdemir, E. (2017). Dinamik Araç Rotalama Problemleri İçin Yeni Bir Çözüm Önerisi, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 22(3), 807-823. <https://dergipark.org.tr/pub/sduibfd/issue/52994/704161>
- Dursun, P. (2009). Zaman Pancereli Araç Rotalama Probleminin Genetik Algoritma ile Modellenmesi (Tez No. 11527) [Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi]. <http://hdl.handle.net/11527/5720>
- Dursun, E. & Gürgen, E. (2020). Konteyner terminal stok yönetiminde abc analizi ve bulanık sınıflandırma. *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 17(46), 563-583. <https://dergipark.org.tr/en/download/article-file/1019390>
- Düzakın, E., & Demircioğlu, M. (2009). Araç Rotalama Problemleri ve Çözüm Yöntemleri, *Çukurova Üniversitesi İİBF Dergisi*, 13(1), 68-87. <https://dergipark.org.tr/tr/download/article-file/46729>
- Emel, G. G., & Taşkın, Ç. (2005). Araç rotalama problemlerinin iki aşamalı çözümünde genetik algoritma kullanımı, *Gazi Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi*, 7(1), 1-17. <https://dergipark.org.tr/tr/download/article-file/287621>
- Elder, R., & Tsoukalas, J. (2006). Investing in Inventories. *Bank of England. Quarterly Bulletin*, 2. https://www.researchgate.net/publication/228284909_Investing_in_Inventories
- Eryavuz, M., & Gencer C. (2001). Araç Rotalama Problemine Ait Bir Uygulama, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi*, 6(1), 139-155. <https://dergipark.org.tr/tr/download/article-file/195138>
- Ferraira, L. M. D. F., Maganha, I., Magalhaes, V.S.M., & Almeida, M. (2018). A multicriteria decision framework for the management of maintenance spares- a case study. *IFAC (International Federation of Automatic Control) Papers Online*, 11(2018), 531-537. <https://doi.org/10.1016/j.ifacol.2018.08.373>
- Gove, R. (2022, 20 Şubat). Using the elbow method to determine the optimal number of clusters for k-means clustering. <https://bl.ocks.org/rpgove/0060ff3b656618e9136b>
- Gözüdeli, Y. ve Akcayol, M. A. (2007). Genetik algoritma ile web sayfası düzeninin gerçek zamanlı optimizasyonu, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 22(2), 431-49. <https://dergipark.org.tr/en/download/article-file/75607>
- Gürbüz, Ö. (2015). *Tabu Arama Algoritmasının Kuyruk Problemine Uygulanması* (Tez No. 1842) [Yüksek Lisans Tezi, Hacettepe Üniversitesi]. <http://www.openaccess.hacettepe.edu.tr:8080/xmlui/bitstream/handle/11655/2103/39cfc127-e234-4c06-aa04-46b456a87336.pdf?sequence=1>
- Grzybowska, A., Kerfeld, B., Gretton, C., & Waller, S.T. (2020). A simulation-optimisation genetic algorithm approach to product allocation in vending machine systems. *Expert Systems with Applications*, 145 (2020), 110-113. <https://doi.org/10.1016/j.eswa.2019.113110>

- Hadi-Vencheh, A., & Mohamadghasemi, A. (2011). A fuzzy AHP-DEA approach for multiple criteria ABC inventory classification. *Expert Systems with Applications*, 38 (4), 3346-3352. <https://doi.org/10.1016/j.eswa.2010.08.119>
- Ishizaka, A., Lolli, F., Balugani, E., Cavallieri, R., & Gamberini, R. (2018). DEA-Sort: Assigning Items With Data Envelopment Analysis in ABC Classes. *International Journal of Production Economics*, 199: 7-15. DOI: [10.1016/j.ijpe.2018.02.007](https://doi.org/10.1016/j.ijpe.2018.02.007)
- Karaboğa, D. (2005). An Idea Based On Honey Bee Swarm For Numerical Optimization. Technical Report-TR06, Erciyes University Engineering Faculty Computer Engineering Department, Kayseri. https://www.researchgate.net/publication/255638348_An_Idea_Based_on_Honey_Bee_Swarm_for_Numerical_Optimization_Technical_Report_-_TR06
- Keskintürk, T., Topuk, N., & Özyeşil, O. (2015). Araç rotalama problemleri ile çözüm yöntemlerinin sınıflandırılması ve bir uygulama. *İşletme Bilimi Dergisi*, 3(2), 77-107. <https://dergipark.org.tr/tr/download/article-file/213329>
- Küçükoglu, İ. (2010). *Zaman Kısıtlı Araç Rotalama Problemi ve Hizmet Sisteminde Bir Uygulama* [Yayınlanmamış yüksek lisans tezi]. <https://acikerisim.uludag.edu.tr/bitstream/11452/2192/1/340046.pdf>
- Küçüksille, E. U., & Tokmak, M. (2011). Yapay arı kolonisi algoritması kullanarak otomatik ders çizelgeleme, *Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü Dergisi*, 15(3), 203-210. <https://dergipark.org.tr/tr/pub/sdufenbed/issue/20796/222135>
- Kuzu, S., Önay, O., Şen, U., Tunçer, M., Yıldırım, B. F., & Keskintürk, T. (2014). Gezgin satıcı problemlerinin metasezgiseller ile çözümü, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 43(1), 1-27. <https://dergipark.org.tr/tr/pub/iuisletme/issue/9254/115779>
- Lolli, F., Ishizaka, A., & Gamberini R. (2014). New AHP-based approaches for multi-criteria inventory classification, *Int. J. Production Economics*, 156(2014), 62-74. <https://www.sciencedirect.com/science/article/abs/pii/S0925527314001789>
- Metin, S. (2021). Benzetimli tavlama algoritması ile eksik veri tamamlama, *Fırat Üniversitesi Müh. Bil. Dergisi*, 33(1), 295-301. <https://dergipark.org.tr/tr/download/article-file/1372653>
- Mitchell, J. E. (1998). Branch-And-Cut Algorithms for Integer Programming, Mathematical Sciences, *Rensselaer Polytechnic Institute, USA*. http://eaton.math.rpi.edu/faculty/Mitchell/papers/bc_hao.pdf
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J., & Sewell, E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19, 79-102. <https://doi.org/10.1016/j.disopt.2016.01.005>
- Murena, E., Sibanda, V., Sibanda, S., & Mporu, K. (2020). Design of a Control System for a Vending Machine, *Procedia CIRP*, 91, 758-763. DOI: [10.1016/j.procir.2020.04.136](https://doi.org/10.1016/j.procir.2020.04.136)

- Nagy, G. ve Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162, 126–141. DOI:[10.1016/j.ejor.2002.11.003](https://doi.org/10.1016/j.ejor.2002.11.003)
- Noor, N. M. ve Shuib, A. (2015). Mult-Depot Instances for Inventory Routing Problem Using Clustering Techniques, *Journal of Industrial and Intelligent Information*, 3(2), 97-101. <http://www.jiii.org/uploadfile/2014/1112/20141112112618639.pdf>
- Oropeza, A., Cruz-Chávez, M., Cruz-Rosales M., Bernal, P., & Abarca, J. (2012). Unsupervised Clustering Method for the Capacited Vehicle Routing Problem, *Ninth Electronics, Robotics and Automotive Mechanics konferansında yayınlanmıştır*. Mexico. <https://0211m73tc-y-https-ieeeexplore-ieee-org.proxy.uludag.deep-knowledge.net/stamp/stamp.jsp?tp=&arnumber=6524580>
- Özari, Ç., & Demirkale, Ö. (2020). K-ortalamlar kümeleme yöntemi ile temel makroekonomik ve finansal göstergeler ile değerlendirilmesi: kırılmalı beşli ülkelerinin Örneği. *Finans Ekonomi ve Sosyal Araştırmalar Dergisi*, 5, 2602 – 2486. <https://doi.org/10.29106/fesa.649176>
- Parıldar, O., & Akyürek, Ç.E. (2021). Hastanelerde emniyet stoğu seviyesinin ve yeniden sipariş noktasının olasılıklı stok modeli ile belirlenmesi: bir kamu hastanesi örneği. *Sosyoekonomi*, 29(47), 229-284. <https://doi.org/10.17233/sosyoekonomi.2021.01.11>
- Park, Y., & Yoo, J. (2012). A heuristic for the inventory management of smart vending machine systems. *Journal of Industrial Engineering and Management*, 5 (2): 354-363. DOI:[10.3926/jiem.587](https://doi.org/10.3926/jiem.587)
- Park, Y., Yoon, S. (2012). The operation of vending machine systems with stock-out-based, one-stage item substitution. *International Journal of Industrial Engineering*, 19 (11), 412-427. https://www.researchgate.net/publication/287956905_The_operation_of_vending_machine_systems_with_stock-out-based_one-stage_item_substitution
- Peya, Z. J., Akhand, M. A. H., & Rahman, M. M. H. (2019). Distance based sweep nearest algorithm to solve capacitated vehicle routing problem. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 10 (10), 259-264. https://thesai.org/Downloads/Volume10No10/Paper_36-Distance_based_Sweep_Nearest_Algorithm.pdf
- Ruiz, E., Garcia-Calvillo, I. & Nucamendi-Guillen, S. (2020). Open vehicle routing problem with split deliveries: mathematical formulations and cutting-plane method. *Operational Research*, <https://doi.org/10.1007/s12351-020-00580-8>
- Şeker, Ş. (2007). *Araç rotalama problemleri ve zaman pencereli stokastik araç rotalama problemine genetik algoritma yaklaşımı* [Yayınlanmamış yüksek lisans tezi]. Yıldız Teknik Üniversitesi. <http://dspace.yildiz.edu.tr/xmlui/bitstream/handle/1/6126/0030533.pdf?sequence=1>
- Şenol, A., & Karacan, H. (2020). K-boyutlu ağaç ve uyarlanabilir yarıçap (KD-AR Stream) tabanlı gerçek zamanlı akan veri kümeleme. *Gazi Üniversitesi*

Ustaahmetoğlu, E. (2013). Stok yönetimi ve miktar kararları. Z. Erdoğan (Ed.), *Perakendecilikte ürün yönetimi içinde* (s.142-163). Eskişehir: Anadolu Üniversitesi Yayınları.

https://www.researchgate.net/publication/278406028_Ustaahmetoglu_E_Stok_Yonetimi_ve_Miktar_Kararlari_Editor_Zafer_Erdogan_Perakende_Urun_Yoneti_mi_Anadolu_Universitesi_Yayinlari_2013_Eskisehir

Üstünel, M. (2018). *K - ortalamalar algoritmasına dayalı kümeleme analizi sistemi ve perakendecilik sektöründe uygulaması* (Tez no. 10202073) [Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi].
https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=EiQHWxCbRV_RGJk_fOTqTZw&no=F6VrY5l_r0RQ7JcGc5hYuA

Yeşilbudak, M., Kahraman, H. T., & Karacan, H. (2011). *Veri madenciliğinde nesne yönelimli birleştirici hiyerarşik kümeleme modeli*, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 26(1), 27-39.
<https://dergipark.org.tr/en/pub/gazimmfd/issue/6687/88712>

Yılmaz, Ö., & Temurlenk, M. S. (2010). Türkiye'deki İstatistik Bölgelerin Kişi Başına Düşen Gelir Açısından Hiyerarşik ve Hiyerarşik Olmayan Kümeleme Analizi İle Değerlendirilmesi: 1965-2001. *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 19(2), 75-92.
<https://dergipark.org.tr/en/pub/atauniiibd/issue/2688/35321>

Yılmaz, Ş. (2008). *Çok depolu araç rotalama probleminin karınca kolonisi optimizasyonu ile modellenmesi ve bir çözüm önerisi* (Tez No. 0033747) [Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi].
<http://dspace.yildiz.edu.tr/xmlui/handle/1/6131>

Yılmaz, V., Büyükyıldız, M., & Baykan, Ö. K. (2020). Yapay arı kolonisi algoritması kullanılarak su dağıtım şebekelerinin optimizasyonu, *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 9(1), 377-392.
<https://dergipark.org.tr/tr/download/article-file/950293>

Yücelen, A. M., & Baykal, A. (2021). K-ortalamalar kümeleme yöntemi için çift k başlatma algoritması. *Avrupa Bilim ve Teknoloji Dergisi*, 23, 280-287.
<https://doi.org/10.31590/ejosat.866830>

Yücenur, G. N., & Demirel, N. Ç. (2011). A hybrid algorithm with genetic algorithm and ant colony optimization for solving multi-depot vehicle routing problems. *Journal of Engineering and Natural Sciences*, 29, 340-350.
<https://www.ytusigmadergisi.com/pdfs/173.pdf>

EKLER

EK 1 k-ortalamlar algoritması kullanılarak elde edilmiş üç kümeye ait bilgiler

EK 2 Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler

EK 3 Hibrit kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler

EK 4 k-ortalamlar algoritması kullanılarak elde edilmiş iki kümeye ait bilgiler

EK 5 Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler

EK 6 Hibrit kümeleme yöntemi kullanılarak elde edilmiş 2 kümeye ait bilgiler

EK 7 Uzaklık matrisi

EK 8 Sinyal matrisi

EK 9 Algoritmanın Visual Basic dilinde yazılan kodu

EK 10 Mesafe matrisinin oluşturulması için yazılan kod

EK 11 Geliştirilen algoritma için Visual Basic dilinde yazılan kod

EK 12 Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

EK 13 Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

EK 1 k-ortalamalar algoritması kullanılarak elde edilmiş üç kümeye ait bilgiler

Küme1				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ BURÇAK KLASİK BİSKÜVİ 131 GR*KL24 2511000	1882	2	1,75	1
ETİ CİN TEK LOKMALIK BÜSKÜVİ-114G*18 3031700	1922	2	2	1
ETİ BURÇAK SÜTLÜ KREMALİ YULAFLI BÜSKÜVİ	2824	2	1,5	1
ETİ TUTKU KAK. KREM.MOZAİK100 GR-2233500	2613	2	2	1
ETİ BENİMOLOK. 80GR*24 KL-3532900	1480	2	2,25	1
ETİ SULTANİ BÜSKÜVİ 2411500	2373	2	2	1
BİSCOLATA STARZ SÜTLÜ 50G*24 AD 5721	781	2	1,25	1
ETİ NEGRO BÜSKÜVİ 4031800	1509	2	2	1
ETİ BUR.SÜTLÜ ÇİK. 114GR*12-1635000	884	2	2,4	1
BİSCOLATA STARZ SAND.BIS.ÇİK 50G*24 AD 5720	1019	2	1,25	1
BİS.NİRVANA ROLLS FIN.4KT 12AD 28G 9703	442	2	1,25	1
ETİ CANGA NUGA BAR 45GR*16 TV-1672500	1969	2	1,75	1
ETİ GONG 40*18 DP 2246700	1365	2	1,25	1
ETİ MAXIMUS YER FIS.BZS 40G*24 TV-2173500	1991	2	1,25	1
BOOMBASTIC FIND.KEK BAR 6KT12 AD 9684	297	2	1,5	1
ETİ AHENK BOL SÜTLÜ ÇİK.35GR*12 1087800	276	2	2,75	1
ETİ GONG BALLI HARDALLI 40*18 DP 2248200	730	2	1,25	1
ETİ HOŞBEŞ FIN. KR. GOFRET 40 GR*24 KL1752200	1714	2	1,25	1
KARAM GURME 66 GR*12*12 KL-3754100	2104	2	1,75	1
ETİ BİDOLU 36GR*15	2410	2	1,5	1
ETİ BİDOLU YERFİS. KAK. 36GR*1--24 KL-1855100	737	2	1,5	1
BİSCOLATA VENİ ÇİK.96191A 50GR*12	1062	2	1,5	1
BİSCOLATA VENİ FIND..96131A 50GR*12	1069	2	1,5	1
ETİ CRAX PEYNİRLİ ÇUBUK KRAKER 50 GR-2041400	191	2	1	1
ETİ SUSAMLI ÇUBUK KRAKER 70 GR* 14 KL 3540600	2336	2	1,5	1
ETİ BROWNİ INTENSE KAK.50 GR*16 TV-1863700	2532	2	1,5	1

EK 1 (devam) k-ortalamalar algoritması kullanılarak elde edilmiş üç kümeye ait bilgiler

Küme2				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ÜLKER ÇİZİ VİÇ 90 GR	1803	3	2,25	1
ÜLKER KRAKER ÇİZİ PEYNİRLİ	433	3	1,5	1
SIRMA LIMON AROMALI 0.2 VİTAMİNLİ, MADEN SUYU	4180	3	1,5	0
SIRMA ELMA C-PLUS VİTAMİNLİ MADEN SUYU	3786	3	1,5	0
SIRMA MANDALINA C-PLUS VİTAMİNLİ MADEN SUYU	2815	3	1,5	0
SIRMA SCHORLE ELMA	30	3	1	0
SIRMA 0.5 LT 12 Lİ PET SU	1048	3	1,5	0
SIRMA MAN. C-PLUS VİTAMİNLİ MADEN SUYU ÇEVİR AÇ	35	3	1,5	0
SIRMA SCHORLE LİMON	71	3	1,5	0
COCO COLA 200 ML	1247	3	2	0
BLACK BRUIN ENERJİ İÇECEĞİ	2923	3	2,75	0
JUSS SOĞUK ÇAY ŞEFTALİ 330 ML	1695	3	2	0
JUSS SOĞUK ÇAY LİMON 330 ML	36	3	2	0
OHBE GAZOZ ŞİŞE 250ML	1047	3	2,25	0
MAN KAFA 250ML	1023	3	2,5	0
JUSS ELMA-ŞEFTALİ SUYU 1/5 27 ADET	151	3	1,5	0

Küme3				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ KREMALI BÜSKÜVİ BZS 77 GR*24 KL-1730100	2032	2	1	1
ETİ WANTED M.B.GEV.KRM 22*24 KL1687100	3289	2	0,5	1
ETİ CRAX BAH.ÇUBUK KRAKER 50 GR--5742300	2489	2	1	1
ETİ CRAX ÇUBUK KRAKER 45 GR-3540100	3880	2	0,75	1
ETİ POP KEK ÇİKOLATALI60 GR5837100	3246	2	1	1
ETİ POP KEK MUZLU 60 GR*24 KL 4837200	3263	2	1	1
ETİ TOP KEK MEYVELİBSZ 40 GR*24 TV 1361200	3098	2	1	1
ETİ TOP KEK KAKAOLU BSZ 35 GR*24 TV 1361300	3214	2	1	1
LUPPO BAR KEK24 AD 30 GR 5959	3414	2	0,75	1
LUPPO ÇİK.BAR KEK24 AD 30 GR 5869	3182	2	1	1
SIRMA MULTIPACK MADEN SUYU 24 AD	6444	3	1	0

EK 2 Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler

Küme1				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ AHENK BOL SÜTLÜ ÇİK.35GR*12 1087800	276	2	2,75	1
ETİ BENİMOLOK. 80GR*24 KL-3532900	1480	2	2,25	1
ETİ BUR.SÜTLÜ ÇİK. 114GR*12-1635000	884	2	2,4	1
ETİ TOP KEK MEYVELİBSZ 40 GR*24 TV 1361200	3098	2	1	1
ETİ POP KEK ÇİKOLATALI60 GR5837100	3246	2	1	1
ETİ POP KEK MUZLU 60 GR*24 KL 4837200	3263	2	1	1
ETİ TOP KEK KAKAOLU BSZ 35 GR*24 TV 1361300	3214	2	1	1
LUPPO ÇİK.BAR KEK24 AD 30 GR 5869	3182	2	1	1
ETİ WANTED M.B.GEV.KRM 22*24 KL1687100	3289	2	0,5	1
ETİ CRAX ÇUBUK KRAKER 45 GR-3540100	3880	2	0,75	1
LUPPO BAR KEK24 AD 30 GR 5959	3414	2	0,75	1
BOOMBASTIC FIND.KEK BAR 6KT12 AD 9684	297	2	1,5	1
ETİ CRAX PEYNİRLİ ÇUBUK KRAKER 50 GR-2041400	191	2	1	1
ETİ BİDOLU YERFİS. KAK. 36GR*1--24 KL-1855100	737	2	1,5	1
BİSCOLATA VENİ ÇİK.96191A 50GR*12	1062	2	1,5	1
BİSCOLATA VENİ FIND..96131A 50GR*12	1069	2	1,5	1
BİS.NİRVANA ROLLS FİN.4KT 12AD 28G 9703	442	2	1,25	1
BİSCOLATA STARZ SAND.BİS.ÇİK 50G*24 AD 5720	1019	2	1,25	1
BİSCOLATA STARZ SÜTLÜ 50G*24 AD 5721	781	2	1,25	1
ETİ GONG BALLI HARDALLI 40*18 DP 2248200	730	2	1,25	1
ETİ TUTKU KAK. KREM.MOZAİK100 GR-2233500	2613	2	2	1
ETİ SULTANİ BÜSKÜVİ 2411500	2373	2	2	1
KARAM GURME 66 GR*12*12 KL-3754100	2104	2	1,75	1
ETİ BURÇAK KLASİK BİSKÜVİ 131 GR*KL24 2511000	1882	2	1,75	1
ETİ CANGA NUGA BAR 45GR*16 TV-1672500	1969	2	1,75	1
ETİ CİN TEK LOKMALIK BÜSKÜVİ-114G*18 3031700	1922	2	2	1
ETİ NEGRO BÜSKÜVİ 4031800	1509	2	2	1
ETİ BURÇAK SÜTLÜ KREMALİ YULAFLI BÜSKÜVİ	2824	2	1,5	1
ETİ BROWNİ INTENSE KAK.50 GR*16 TV-1863700	2532	2	1,5	1
ETİ BİDOLU 36GR*15	2410	2	1,5	1
ETİ SUSAMLI ÇUBUK KRAKER 70 GR* 14 KL 3540600	2336	2	1,5	1
ETİ KREMALİ BÜSKÜVİ BZS 77 GR*24 KL-1730100	2032	2	1	1
ETİ CRAX BAH.ÇUBUK KRAKER 50 GR--5742300	2489	2	1	1
ETİ GONG 40*18 DP 2246700	1365	2	1,25	1
ETİ MAXİMUS YER FİS.BZS 40G*24 TV-2173500	1991	2	1,25	1
ETİ HOŞBEŞ FİN. KR. GOFRET 40 GR*24 KL1752200	1714	2	1,25	1

EK 2 (devam) Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler

Küme2				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
SIRMA MULTIPACK MADEN SUYU 24 AD	6444	3	1	0
SIRMA MANDALINA C-PLUS VİTAMİNLİ MADEN SUYU	2815	3	1,5	0
SIRMA LIMON AROMALI 0.2 VİTAMİNLİ, MADEN SUYU	4180	3	1,5	0
SIRMA ELMA C-PLUS VİTAMİNLİ MADEN SUYU	3786	3	1,5	0

Küme3				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
BLACK BRUIN ENERJİ İÇECEĞİ	2923	3	2,75	0
JUSS SOĞUK ÇAY LİMON 330 ML	36	3	2	0
COCO COLA 200 ML	1247	3	2	0
JUSS SOĞUK ÇAY ŞEFTALİ 330 ML	1695	3	2	0
OHBE GAZOZ ŞİŞE 250ML	1047	3	2,25	0
MAN KAFA 250ML	1023	3	2,5	0
SIRMA SCHORLE ELMA	30	3	1	0
SIRMA 0.5 LT 12 Lİ PET SU	1048	3	1,5	0
JUSS ELMA-ŞEFTALİ SUYU 1/5 27 ADET	151	3	1,5	0
SIRMA MAN. C-PLUS VİTAMİNLİ MADEN SUYU ÇEVİR AÇ	35	3	1,5	0
SIRMA SCHORLE LİMON	71	3	1,5	0
ÜLKER ÇİZİ VİÇ 90 GR	1803	3	2,25	1
ÜLKER KRAKER ÇİZİ PEYNİRLİ	433	3	1,5	1

EK 3 Hibrit kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler

Küme1

İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ BURÇAK KLASİK BİSKÜVİ 131 GR*KL24 2511000	1882	2	1,75	1
ETİ CİN TEK LOKMALIK BÜSKÜVİ-114G*18 3031700	1922	2	2	1
ETİ BURÇAK SÜTLÜ KREMALI YULAFLI BÜSKÜVİ	2824	2	1,5	1
ETİ TUTKU KAK. KREM.MOZAİK100 GR-2233500	2613	2	2	1
ETİ BENİMOLOK. 80GR*24 KL-3532900	1480	2	2,25	1
ETİ KREMALI BÜSKÜVİ BZS 77 GR*24 KL-1730100	2032	2	1	1
ETİ SULTANİ BÜSKÜVİ 2411500	2373	2	2	1
BİSCOLATA STARZ SÜTLÜ 50G*24 AD 5721	781	2	1,25	1
ETİ NEGRO BÜSKÜVİ 4031800	1509	2	2	1
ETİ BUR.SÜTLÜ ÇİK. 114GR*12-1635000	884	2	2,4	1
BİSCOLATA STARZ SAND.BIS.ÇİK 50G*24 AD 5720	1019	2	1,25	1
BİS.NİRVANA ROLLS FIN.4KT 12AD 28G 9703	442	2	1,25	1
ETİ CANGA NUGA BAR 45GR*16 TV-1672500	1969	2	1,75	1
ETİ GONG 40*18 DP 2246700	1365	2	1,25	1
ETİ MAXIMUS YER FIS.BZS 40G*24 TV-2173500	1991	2	1,25	1
BOOMBASTIC FIND.KEK BAR 6KT12 AD 9684	297	2	1,5	1
ETİ AHENK BOL SÜTLÜ ÇİK.35GR*12 1087800	276	2	2,75	1
ETİ GONG BALLI HARDALLI 40*18 DP 2248200	730	2	1,25	1
ETİ HOŞBEŞ FIN. KR. GOFRET 40 GR*24 KL1752200	1714	2	1,25	1
KARAM GURME 66 GR*12*12 KL-3754100	2104	2	1,75	1
ETİ BİDOLU 36GR*15	2410	2	1,5	1
ETİ WANTED M.B.GEV.KRM 22*24 KL1687100	3289	2	0,5	1
ETİ BİDOLU YERFIS. KAK. 36GR*1--24 KL-1855100	737	2	1,5	1
BİSCOLATA VENİ ÇİK.96191A 50GR*12	1062	2	1,5	1
BİSCOLATA VENİ FIND..96131A 50GR*12	1069	2	1,5	1
ETİ CRAX PEYNİRLİ ÇUBUK KRAKER 50 GR-2041400	191	2	1	1
ETİ SUSAMLI ÇUBUK KRAKER 70 GR* 14 KL 3540600	2336	2	1,5	1
ETİ CRAX BAH.ÇUBUK KRAKER 50 GR--5742300	2489	2	1	1
ETİ CRAX ÇUBUK KRAKER 45 GR-3540100	3880	2	0,75	1
ETİ POP KEK ÇİKOLATALI60 GR5837100	3246	2	1	1
ETİ POP KEK MUZLU 60 GR*24 KL 4837200	3263	2	1	1
ETİ TOP KEK MEYVELİBSZ 40 GR*24 TV 1361200	3098	2	1	1
ETİ TOP KEK KAKAOLU BSZ 35 GR*24 TV 1361300	3214	2	1	1
ETİ BROWNİ INTENSE KAK.50 GR*16 TV-1863700	2532	2	1,5	1
LUPPO BAR KEK24 AD 30 GR 5959	3414	2	0,75	1
LUPPO ÇİK.BAR KEK24 AD 30 GR 5869	3182	2	1	1

EK 3 (devam) Hibrit kümeleme yöntemi kullanılarak elde edilmiş üç kümeye ait bilgiler

Küme2				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ÜLKER ÇİZİ VİÇ 90 GR	1803	3	2,25	1
ÜLKER KRAKER ÇİZİ PEYNİRLİ	433	3	1,5	1
SIRMA SCHORLE ELMA	30	3	1	0
SIRMA 0.5 LT 12 Lİ PET SU	1048	3	1,5	0
SIRMA MAN. C-PLUS VİTAMİNLİ MADEN SUYU ÇEVİR AÇ	35	3	1,5	0
SIRMA SCHORLE LİMON	71	3	1,5	0
COCO COLA 200 ML	1247	3	2	0
JUSS SOĞUK ÇAY ŞEFTALİ 330 ML	1695	3	2	0
JUSS SOĞUK ÇAY LİMON 330 ML	36	3	2	0
OHBE GAZOZ ŞİŞE 250ML	1047	3	2,25	0
MAN KAFA 250ML	1023	3	2,5	0
JUSS ELMA-ŞEFTALİ SUYU 1/5 27 ADET	151	3	1,5	0

Küme3				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
SIRMA MULTIPACK MADEN SUYU 24 AD	6444	3	1	0
SIRMA LIMON AROMALI 0.2 VİTAMİNLİ, MADEN SUYU	4180	3	1,5	0
SIRMA ELMA C-PLUS VİTAMİNLİ MADEN SUYU	3786	3	1,5	0
SIRMA MANDALINA C-PLUS VİTAMİNLİ MADEN SUYU	2815	3	1,5	0
BLACK BRUIN ENERJİ İÇECEĞİ	2923	3	2,75	0

EK 4 k-ortalamlar algoritması kullanılarak elde edilmiş iki kümeye ait bilgiler**Küme-1**

İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ BURÇAK KLASİK BİSKÜVİ 131 GR*KL24 2511000	1882	2	1,75	1
ETİ CİN TEK LOKMALIK BÜSKÜVİ-114G*18 3031700	1922	2	2	1
ETİ BURÇAK SÜTLÜ KREMALI YULAFLI BÜSKÜVİ	2824	2	1,5	1
ETİ TUTKU KAK. KREM.MOZAİK100 GR-2233500	2613	2	2	1
ETİ BENİMOLOK. 80GR*24 KL-3532900	1480	2	2,25	1
ETİ KREMALI BÜSKÜVİ BZS 77 GR*24 KL-1730100	2032	2	1	1
ETİ SULTANİ BÜSKÜVİ 2411500	2373	2	2	1
BİSCOLATA STARZ SÜTLÜ 50G*24 AD 5721	781	2	1,25	1
ETİ NEGRO BÜSKÜVİ 4031800	1509	2	2	1
ETİ BUR.SÜTLÜ ÇİK. 114GR*12-1635000	884	2	2,4	1
BİSCOLATA STARZ SAND.BIS.ÇİK 50G*24 AD 5720	1019	2	1,25	1
BİS.NİRVANA ROLLS FIN.4KT 12AD 28G 9703	442	2	1,25	1
ETİ CANGA NUGA BAR 45GR*16 TV-1672500	1969	2	1,75	1
ETİ GONG 40*18 DP 2246700	1365	2	1,25	1
ETİ MAXIMUS YER FIS.BZS 40G*24 TV-2173500	1991	2	1,25	1
SÜTLÜ FINDIKLIN ÇİK12*12*35G 401099	12	2	1,5	1
BOOMBASTIC FIND.KEK BAR 6KT12 AD 9684	297	2	1,5	1
ETİ AHENK BOL SÜTLÜ ÇİK.35GR*12 1087800	276	2	2,75	1
ETİ GONG BALLI HARDALLI 40*18 DP 2248200	730	2	1,25	1
ETİ HOŞBEŞ FIN. KR. GOFRET 40 GR*24 KL1752200	1714	2	1,25	1
KARAM GURME 66 GR*12*12 KL-3754100	2104	2	1,75	1
ETİ BİDOLU 36GR*15	2410	2	1,5	1
ETİ WANTED M.B.GEV.KRM 22*24 KL1687100	3289	2	0,5	1
ETİ WANTED BUMBA 40GR*24 KL 1687200	15	2	1,25	1
KARAM %45 K.80GR*6*12 KL 1980500	3	2	1,75	1
ETİ BİDOLU YERFIS. KAK. 36GR*1--24 KL-1855100	737	2	1,5	1
BİSCOLATA VENİ ÇİK.96191A 50GR*12	1062	2	1,5	1
BİSCOLATA VENİ FIND..96131A 50GR*12	1069	2	1,5	1
ETİ CRAX PEYNİRLİ ÇUBUK KRAKER 50 GR-2041400	191	2	1	1
ETİ SUSAMLI ÇUBUK KRAKER 70 GR* 14 KL 3540600	2336	2	1,5	1
ETİ CRAX BAH.ÇUBUK KRAKER 50 GR--5742300	2489	2	1	1
ETİ CRAX ÇUBUK KRAKER 45 GR-3540100	3880	2	0,75	1
ETİ BROWNİ INTENSE 35 GR*P24 D	13	2	1,25	1
ETİ POP KEK ÇİKOLATALI60 GR5837100	3246	2	1	1
ETİ POP KEK MUZLU 60 GR*24 KL 4837200	3263	2	1	1
ETİ TOP KEK MEYVELİBSZ 40 GR*24 TV 1361200	3098	2	1	1
ETİ TOP KEK KAKAOLU BSZ 35 GR*24 TV 1361300	3214	2	1	1
ETİ BROWNİ INTENSE KAK.50 GR*16 TV-1863700	2532	2	1,5	1
LUPPO BAR KEK24 AD 30 GR 5959	3414	2	0,75	1
LUPPO ÇİK.BAR KEK24 AD 30 GR 5869	3182	2	1	1

EK 4 (devam) k-ortalamlar algoritması kullanılarak elde edilmiş iki kümeye ait bilgiler

Küme-2				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
TORKU DELİ DOLU KAKAO KREM.24*110 G 401974	13	3	1,25	1
ÜLKER ÇİZİ VİÇ 90 GR	1803	3	2,25	1
ÜLKER KRAKER ÇİZİ PEYNİRLİ	433	3	1,5	1
SIRMA MULTIPACK MADEN SUYU 24 AD	6444	3	1	0
SIRMA LIMON AROMALI 0.2 VİTAMİNLİ, MADEN SUYU	4180	3	1,5	0
SIRMA ELMA C-PLUS VİTAMİNLİ MADEN SUYU	3786	3	1,5	0
SIRMA MANDALINA C-PLUS VİTAMİNLİ MADEN SUYU	2815	3	1,5	0
SIRMA SCHORLE ELMA	30	3	1	0
SIRMA 0.5 LT 12 Lİ PET SU	1048	3	1,5	0
SIRMA MAN. C-PLUS VİTAMİNLİ MADEN SUYU ÇEVİR AÇ	35	3	1,5	0
SIRMA SCHORLE LİMON	71	3	1,5	0
COCO COLA 200 ML	1247	3	2	0
BLACK BRUIN ENERJİ İÇECEĞİ	2923	3	2,75	0
SIRMA SADE SODA	14	3	1	0
JUSS SOĞUK ÇAY ŞEFTALİ 330 ML	1695	3	2	0
JUSS SOĞUK ÇAY LİMON 330 ML	36	3	2	0
OHBE GAZOZ ŞİŞE 250ML	1047	3	2,25	0
MAN KAFA 250ML	1023	3	2,5	0
JUSS ELMA-ŞEFTALİ SUYU 1/5 27 ADET	151	3	1,5	0

EK 5 Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler

Küme-1				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ AHENK BOL SÜTLÜ ÇİK.35GR*12 1087800	276	2	2,75	1
ETİ BENİMOLOK. 80GR*24 KL-3532900	1480	2	2,25	1
ETİ BUR.SÜTLÜ ÇİK. 114GR*12-1635000	884	2	2,4	1
ETİ TOP KEK MEYVELİBSZ 40 GR*24 TV 1361200	3098	2	1	1
ETİ POP KEK ÇİKOLATALI60 GR5837100	3246	2	1	1
ETİ POP KEK MUZLU 60 GR*24 KL 4837200	3263	2	1	1
ETİ TOP KEK KAKAOLU BSZ 35 GR*24 TV 1361300	3214	2	1	1
LUPPO ÇİK.BAR KEK24 AD 30 GR 5869	3182	2	1	1
ETİ WANTED M.B.GEV.KRM 22*24 KL1687100	3289	2	0,5	1
LUPPO BAR KEK24 AD 30 GR 5959	3414	2	0,75	1
KARAM %45 K.80GR*6*12 KL 1980500	3	2	1,75	1
SÜTLÜ FINDIKLIN ÇİK12*12*35G 401099	12	2	1,5	1
BOOMBASTIC FIND.KEK BAR 6KT12 AD 9684	297	2	1,5	1
ETİ CRAX PEYNİRLİ ÇUBUK KRAKER 50 GR-2041400	191	2	1	1
ETİ WANTED BUMBA 40GR*24 KL 1687200	15	2	1,25	1
ETİ BROWNİ INTENSE 35 GR*P24 D	13	2	1,25	1
ETİ BİDOLU YERFİS. KAK. 36GR*1--24 KL-1855100	737	2	1,5	1
BİSCOLATA VENİ ÇİK.96191A 50GR*12	1062	2	1,5	1
BİSCOLATA VENİ FIND..96131A 50GR*12	1069	2	1,5	1
BİS.NİRVANA ROLLS FIN.4KT 12AD 28G 9703	442	2	1,25	1
BİSCOLATA STARZ SAND.BİS.ÇİK 50G*24 AD 5720	1019	2	1,25	1
BİSCOLATA STARZ SÜTLÜ 50G*24 AD 5721	781	2	1,25	1
ETİ GONG BALLI HARDALLI 40*18 DP 2248200	730	2	1,25	1
ETİ TUTKU KAK. KREM.MOZAİK100 GR-2233500	2613	2	2	1
ETİ SULTANİ BÜSKÜVİ 2411500	2373	2	2	1
KARAM GURME 66 GR*12*12 KL-3754100	2104	2	1,75	1
ETİ BURÇAK KLASİK BİSKÜVİ 131 GR*KL24 2511000	1882	2	1,75	1
ETİ CANGA NUGA BAR 45GR*16 TV-1672500	1969	2	1,75	1
ETİ CİN TEK LOKMALIK BÜSKÜVİ-114G*18 3031700	1922	2	2	1
ETİ NEGRO BÜSKÜVİ 4031800	1509	2	2	1
ETİ BURÇAK SÜTLÜ KREMALİ YULAFLI BÜSKÜVİ	2824	2	1,5	1
ETİ BROWNİ INTENSE KAK.50 GR*16 TV-1863700	2532	2	1,5	1
ETİ BİDOLU 36GR*15	2410	2	1,5	1
ETİ SUSAMLI ÇUBUK KRAKER 70 GR* 14 KL 3540600	2336	2	1,5	1
ETİ KREMALİ BÜSKÜVİ BZS 77 GR*24 KL-1730100	2032	2	1	1
ETİ CRAX BAH.ÇUBUK KRAKER 50 GR--5742300	2489	2	1	1
ETİ GONG 40*18 DP 2246700	1365	2	1,25	1
ETİ MAXIMUS YER FİS.BZS 40G*24 TV-2173500	1991	2	1,25	1
ETİ HOŞBEŞ FIN. KR. GOFRET 40 GR*24 KL1752200	1714	2	1,25	1
ETİ CRAX ÇUBUK KRAKER 45 GR-3540100	3880	2	0,75	1

EK 5 (devam) Hiyerarşik kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler

Küme-2				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
SIRMA MULTIPACK MADEN SUYU 24 AD	6444	3	1	0
SIRMA MANDALINA C-PLUS VİTAMİNLİ MADEN SUYU	2815	3	1,5	0
SIRMA LIMON AROMALI 0.2 VİTAMİNLİ, MADEN SUYU	4180	3	1,5	0
SIRMA ELMA C-PLUS VİTAMİNLİ MADEN SUYU	3786	3	1,5	0
BLACK BRUIN ENERJİ İÇECEĞİ	2923	3	2,75	0
JUSS SOĞUK ÇAY LİMON 330 ML	36	3	2	0
COCO COLA 200 ML	1247	3	2	0
JUSS SOĞUK ÇAY ŞEFTALİ 330 ML	1695	3	2	0
OHBE GAZOZ ŞİŞE 250ML	1047	3	2,25	0
MAN KAFA 250ML	1023	3	2,5	0
SIRMA SCHORLE ELMA	30	3	1	0
SIRMA SADE SODA	14	3	1	0
SIRMA 0.5 LT 12 Lİ PET SU	1048	3	1,5	0
JUSS ELMA-ŞEFTALİ SUYU 1/5 27 ADET	151	3	1,5	0
SIRMA MAN. C-PLUS VİTAMİNLİ MADEN SUYU ÇEVİR AÇ	35	3	1,5	0
SIRMA SCHORLE LİMON	71	3	1,5	0
ÜLKER ÇİZİ VİÇ 90 GR	1803	3	2,25	1
TORKU DELİ DOLU KAKAO KREM.24*110 G 401974	13	3	1,25	1
ÜLKER KRAKER ÇİZİ PEYNİRLİ	433	3	1,5	1

EK 6 Hibrit kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler**Küme-1**

İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
ETİ BURÇAK KLASİK BİSKÜVİ 131 GR*KL24 2511000	1882	2	1,75	1
ETİ CİN TEK LOKMALIK BÜSKÜVİ-114G*18 3031700	1922	2	2	1
ETİ BURÇAK SÜTLÜ KREMALI YULAFLI BÜSKÜVİ	2824	2	1,5	1
ETİ TUTKU KAK. KREM.MOZAİK100 GR-2233500	2613	2	2	1
ETİ BENİMOLOK. 80GR*24 KL-3532900	1480	2	2,25	1
ETİ KREMALI BÜSKÜVİ BZS 77 GR*24 KL-1730100	2032	2	1	1
ETİ SULTANİ BÜSKÜVİ 2411500	2373	2	2	1
BİSCOLATA STARZ SÜTLÜ 50G*24 AD 5721	781	2	1,25	1
ETİ NEGRO BÜSKÜVİ 4031800	1509	2	2	1
ETİ BUR.SÜTLÜ ÇİK. 114GR*12-1635000	884	2	2,4	1
BİSCOLATA STARZ SAND.BIS.ÇİK 50G*24 AD 5720	1019	2	1,25	1
BİS.NİRVANA ROLLS FIN.4KT 12AD 28G 9703	442	2	1,25	1
ETİ CANGA NUGA BAR 45GR*16 TV-1672500	1969	2	1,75	1
ETİ GONG 40*18 DP 2246700	1365	2	1,25	1
ETİ MAXIMUS YER FIS.BZS 40G*24 TV-2173500	1991	2	1,25	1
SÜTLÜ FINDIKLIN ÇİK12*12*35G 401099	12	2	1,5	1
BOOMBASTIC FIND.KEK BAR 6KT12 AD 9684	297	2	1,5	1
ETİ AHENK BOL SÜTLÜ ÇİK.35GR*12 1087800	276	2	2,75	1
ETİ GONG BALLI HARDALLI 40*18 DP 2248200	730	2	1,25	1
ETİ HOŞBEŞ FIN. KR. GOFRET 40 GR*24 KL1752200	1714	2	1,25	1
KARAM GURME 66 GR*12*12 KL-3754100	2104	2	1,75	1
ETİ BİDOLU 36GR*15	2410	2	1,5	1
ETİ WANTED M.B.GEV.KRM 22*24 KL1687100	3289	2	0,5	1
ETİ WANTED BUMBA 40GR*24 KL 1687200	15	2	1,25	1
KARAM %45 K.80GR*6*12 KL 1980500	3	2	1,75	1
ETİ BİDOLU YERFIS. KAK. 36GR*1--24 KL-1855100	737	2	1,5	1
BİSCOLATA VENİ ÇİK.96191A 50GR*12	1062	2	1,5	1
BİSCOLATA VENİ FIND..96131A 50GR*12	1069	2	1,5	1
ETİ CRAX PEYNİRLİ ÇUBUK KRAKER 50 GR-2041400	191	2	1	1
ETİ SUSAMLI ÇUBUK KRAKER 70 GR* 14 KL 3540600	2336	2	1,5	1
ETİ CRAX BAH.ÇUBUK KRAKER 50 GR--5742300	2489	2	1	1
ETİ CRAX ÇUBUK KRAKER 45 GR-3540100	3880	2	0,75	1
ETİ BROWNİ INTENSE 35 GR*P24 D	13	2	1,25	1
ETİ POP KEK ÇİKOLATALI60 GR5837100	3246	2	1	1
ETİ POP KEK MUZLU 60 GR*24 KL 4837200	3263	2	1	1
ETİ TOP KEK MEYVELİBSZ 40 GR*24 TV 1361200	3098	2	1	1
ETİ TOP KEK KAKAOLU BSZ 35 GR*24 TV 1361300	3214	2	1	1
ETİ BROWNİ INTENSE KAK.50 GR*16 TV-1863700	2532	2	1,5	1
LUPPO BAR KEK24 AD 30 GR 5959	3414	2	0,75	1
LUPPO ÇİK.BAR KEK24 AD 30 GR 5869	3182	2	1	1

EK 6 (devam) Hibrit kümeleme yöntemi kullanılarak elde edilmiş iki kümeye ait bilgiler

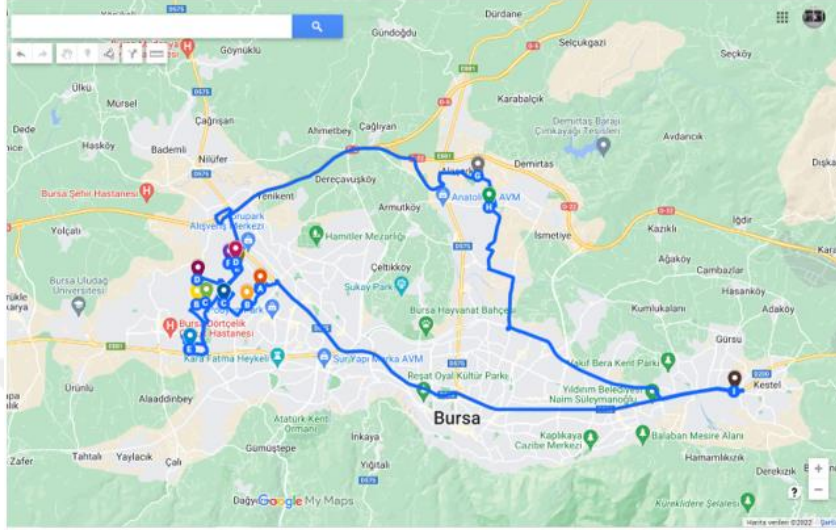
Küme-2				
İsim	Aylık Satış Miktarı	Tedarik Süresi(Hafta)	Br Fiyat	Sezon
TORKU DELİ DOLU KAKAO KREM.24*110 G 401974	13	3	1,25	1
ÜLKER ÇİZİ VİÇ 90 GR	1803	3	2,25	1
ÜLKER KRAKER ÇİZİ PEYNİRLİ	433	3	1,5	1
SIRMA MULTIPACK MADEN SUYU 24 AD	6444	3	1	0
SIRMA LIMON AROMALI 0.2 VİTAMİNLİ, MADEN SUYU	4180	3	1,5	0
SIRMA ELMA C-PLUS VİTAMİNLİ MADEN SUYU	3786	3	1,5	0
SIRMA MANDALINA C-PLUS VİTAMİNLİ MADEN SUYU	2815	3	1,5	0
SIRMA SCHORLE ELMA	30	3	1	0
SIRMA 0.5 LT 12 Lİ PET SU	1048	3	1,5	0
SIRMA MAN. C-PLUS VİTAMİNLİ MADEN SUYU ÇEVİR AÇ	35	3	1,5	0
SIRMA SCHORLE LİMON	71	3	1,5	0
COCO COLA 200 ML	1247	3	2	0
BLACK BRUIN ENERJİ İÇECEĞİ	2923	3	2,75	0
SIRMA SADE SODA	14	3	1	0
JUSS SOĞUK ÇAY ŞEFTALİ 330 ML	1695	3	2	0
JUSS SOĞUK ÇAY LİMON 330 ML	36	3	2	0
OHBE GAZOZ ŞİŞE 250ML	1047	3	2,25	0
MAN KAFA 250ML	1023	3	2,5	0
JUSS ELMA-ŞEFTALİ SUYU 1/5 27 ADET	151	3	1,5	0

EK 7 Uzaklık matrisi

	Depo	Mişen 1	Mişen 2	Mişen 3	Mişen 4	Mişen 5	Mişen 6	Mişen 7	Mişen 8	Mişen 9	Mişen 10	Mişen 11	Mişen 12	Mişen 13	Mişen 14	Mişen 15	Mişen 16	Mişen 17	Mişen 18	Mişen 19	Mişen 20	Mişen 21	Mişen 22	Mişen 23	Mişen 24	Mişen 25	Mişen 26	Mişen 27	Mişen 28	Mişen 29	Mişen 30
Depo	0	1,86	13,77	2,3	2,81	3,91	22,68	4,36	3,8	13,9	2,56	3,01	2,48	0,814	4,05	3,43	13,74	2,31	2,59	3,09	12,86	4,03	4,67	3,4	3,09	2,02	3,11	13,79	1,53	3,85	13,94
Mişen 1	1,86	0	13,27	1,16	1,1	2,26	22,62	2,95	2,72	12,8	0,793	2,08	1,12	12,27	2,36	1,81	12,75	0,589	0,82	2,48	12,02	2,35	3,88	2,04	1,97	2,56	1,84	12,86	0,631	2,15	12,85
Mişen 2	13,77	13,27	0	11,74	12,72	10,76	12,66	9,96	10,14	1,47	12,17	10,87	11,76	1,01	10,68	11,17	0,0574	13,01	12,87	13,67	1,24	10,88	9,1	10,79	10,88	11,99	10,99	0,176	12,59	11,09	1,37
Mişen 3	2,3	1,16	11,74	0	1,72	1,69	21,45	2,08	1,6	11,77	0,945	0,915	0,063	11,05	1,61	1,24	11,7	1,59	1,65	2,63	10,91	1,05	2,73	1,12	0,861	1,93	0,838	11,78	0,854	1,73	11,82
Mişen 4	2,81	1,1	12,72	1,72	0	1,95	22,97	2,76	2,97	12,72	0,886	2,59	1,73	12,08	2,19	1,71	12,81	0,487	0,215	3,47	12,15	2,02	4,25	2,2	2,36	3,52	2,11	12,92	1,61	1,8	12,78
Mişen 5	3,91	2,26	10,76	1,69	1,95	0	21,25	0,817	1,28	10,62	1,52	1,57	1,62	10,13	0,075	0,488	10,71	2,3	2,14	4,29	10,05	0,318	2,47	0,718	1,21	3,4	0,97	10,81	2,37	0,417	10,68
Mişen 6	22,68	22,62	12,66	21,45	22,97	21,25	0	20,43	20,14	13,94	22,24	20,56	21,51	13,86	21,16	21,51	12,73	23,03	23,01	22,21	12,45	21,47	10,86	20,92	20,76	20,66	20,99	12,51	22,13	21,64	13,83
Mişen 7	4,36	2,95	9,96	2,08	2,76	0,817	20,43	0	0,989	9,87	2,22	1,57	2,05	9,62	0,741	1,2	9,93	3,05	2,91	4,65	9,24	1,04	1,79	0,948	0,952	3,47	1,24	1,23	2,89	1,22	9,93
Mişen 8	3,8	2,72	10,14	1,6	2,97	1,28	20,14	0,989	0	10,17	2,13	0,885	1,61	9,67	1,25	1,38	10,11	2,97	2,91	3,93	9,33	1,42	1,28	0,788	0,732	2,63	0,878	10,19	2,47	1,68	10,22
Mişen 9	13,9	12,8	1,47	11,77	12,72	10,62	13,94	9,87	10,17	0	12,09	10,95	11,78	1,26	10,54	11,06	1,41	12,92	12,76	13,89	2,49	10,71	9,26	10,76	10,91	12,26	10,98	1,62	12,63	10,92	0,109
Mişen 10	2,56	0,793	12,17	0,945	0,886	1,52	22,24	2,22	2,13	12,09	0	1,71	0,852	11,29	1,6	1,05	12,13	0,845	0,785	3,09	11,41	1,56	3,39	1,39	1,47	2,78	1,26	12,22	1,07	1,37	12,15
Mişen 11	3,01	2,08	10,87	0,915	2,59	1,57	20,56	1,57	0,885	10,95	1,71	0	0,961	10,04	1,6	1,37	10,83	2,48	3,11	11	10,02	1,89	1,82	0,852	0,376	1,88	0,699	10,91	1,73	1,87	11
Mişen 12	2,48	1,12	11,76	0,963	1,73	1,62	21,51	2,05	1,61	11,78	0,852	0,961	0	10,89	1,66	1,2	11,73	1,54	1,58	2,66	10,94	1,83	2,77	1,12	0,886	2	0,83	11,8	0,849	1,7	11,84
Mişen 13	0,814	12,27	1,01	11,05	12,08	10,13	13,86	9,62	9,67	1,26	11,29	10,04	10,89	0	9,76	10,25	0,94	12,1	11,96	12,88	1,37	9,95	8,27	9,9	10,01	11,22	10,1	1,11	11,72	10,15	1,25
Mişen 14	4,05	2,36	10,68	1,61	2,19	0,075	21,16	0,741	1,25	10,54	1,6	1,6	1,66	9,76	0	0,965	10,63	2,37	2,21	4,31	9,97	0,362	2,41	0,703	1,2	3,4	0,976	10,74	2,41	0,488	10,61
Mişen 15	3,43	1,81	11,17	1,24	1,71	0,488	21,51	1,2	1,38	11,06	1,05	1,37	1,2	10,25	0,565	0	11,12	1,85	1,71	3,82	3,67	0,61	2,63	0,588	0,995	3,06	0,709	11,22	1,89	0,527	11,13
Mişen 16	13,74	12,75	0,0574	11,7	12,81	10,71	12,73	9,93	10,11	1,41	12,13	10,83	11,73	0,94	10,63	11,12	0	12,96	12,83	13,63	1,25	10,83	9,07	10,75	10,85	11,96	10,95	0,218	12,55	11,04	1,32
Mişen 17	2,31	0,589	13,01	1,59	0,487	2,3	23,03	3,05	2,97	12,92	0,845	2,48	1,54	12,1	2,37	1,85	12,96	0	0,272	2,99	12,25	2,27	4,21	2,24	2,29	3,16	2,09	13,06	1,21	2,06	12,98
Mişen 18	2,59	0,82	12,87	1,65	0,215	2,14	23,01	2,91	2,91	12,76	0,785	2,48	1,58	11,96	2,21	1,71	12,83	0,272	0	3,25	12,14	2,07	4,17	2,15	2,26	3,31	2,04	12,93	1,39	1,87	12,82
Mişen 19	3,09	2,48	13,67	2,63	3,47	4,29	22,21	4,65	3,93	13,89	3,09	3,11	2,66	12,88	4,31	3,82	13,63	2,99	3,25	0	12,71	4,49	4,65	3,73	3,34	1,7	3,43	13,7	2,04	4,33	13,94
Mişen 20	12,86	12,02	1,24	10,91	12,15	10,05	12,45	9,24	9,33	2,49	11,41	10,02	10,94	1,37	9,97	3,67	1,25	12,25	12,14	12,71	0	10,2	8,21	10,02	10,06	11,01	10,19	11,74	10,41	2,45	
Mişen 21	4,03	2,35	10,88	1,85	2,02	0,318	21,47	1,04	1,42	10,71	1,56	1,89	1,83	9,95	0,362	0,61	10,83	2,27	2,07	4,49	10,2	0	2,77	1,82	1,49	3,65	1,22	10,94	2,48	0,208	10,77
Mişen 22	4,67	3,88	9,1	2,73	4,25	2,47	18,86	1,79	1,28	9,36	3,39	1,82	2,77	8,27	2,41	2,63	9,07	4,21	4,17	4,65	8,21	2,77	0	2,87	1,93	3,04	2,14	9,13	3,54	2,88	9,31
Mişen 23	3,4	2,04	10,79	1,12	2,2	0,718	20,92	0,948	0,788	10,76	1,39	0,852	1,12	9,9	0,703	0,588	10,75	2,24	2,15	3,73	10,02	1,02	2,07	0	0,493	2,68	0,286	10,83	1,92	1,03	10,81
Mişen 24	3,09	1,97	10,88	0,861	2,36	1,21	20,76	0,952	0,732	10,91	1,47	0,576	0,886	10,01	1,2	0,995	10,85	2,29	2,26	3,34	10,06	1,49	1,93	0,493	0	2,22	0,388	10,93	1,71	1,49	10,96
Mişen 25	2,02	2,56	11,99	1,93	3,52	3,4	20,66	3,47	2,63	12,26	2,78	1,88	2	11,22	3,4	3,06	11,96	3,16	3,31	1,7	11,01	3,65	3,04	2,68	2,22	0	2,43	12	1,95	3,61	12,29
Mişen 26	3,11	1,84	10,99	0,838	2,11	0,97	20,99	1,24	0,878	10,98	1,26	0,699	0,83	10,1	0,976	0,709	10,95	2,09	2,04	3,43	10,19	1,22	2,14	0,286	0,388	2,43	0	11,03	1,67	1,23	11,04
Mişen 27	13,79	12,86	0,176	11,78	12,92	10,81	12,51	1,23	10,19	1,62	12,22	10,91	11,8	1,11	10,74	11,22	0,218	13,06	12,93	13,7	1,19	10,94	9,13	10,83	10,93	12	11,83	0	12,62	11,14	1,49
Mişen 28	1,53	0,631	12,59	0,854	1,61	2,37	22,13	2,89	2,47	12,63	1,07	1,73	0,849	11,72	2,41	1,89	12,55	1,21	1,39	2,04	11,74	2,48	3,54	1,92	1,71	1,95	1,67	12,62	0	2,33	12,69
Mişen 29	3,85	2,15	11,09	1,73	1,8	0,417	21,64	1,22	1,68	10,92	1,37	1,87	1,7	10,15	0,488	0,527	11,04	2,06	1,87	4,33	10,41	0,208	2,88	1,83	1,49	3,61	1,23	11,14	2,33	0	10,98
Mişen 30	13,94	12,85	1,37	11,82	12,78	10,68	13,83	9,93	10,22	0,109	12,15	11	11,84	1,25	10,61	11,13	1,32	12,98	12,82	13,94	2,45	10,77	9,31	10,81	10,96	12,29	11,84	1,49	12,69	10,98	0

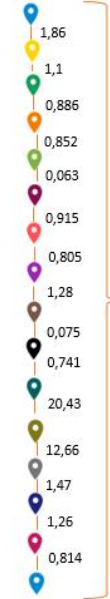
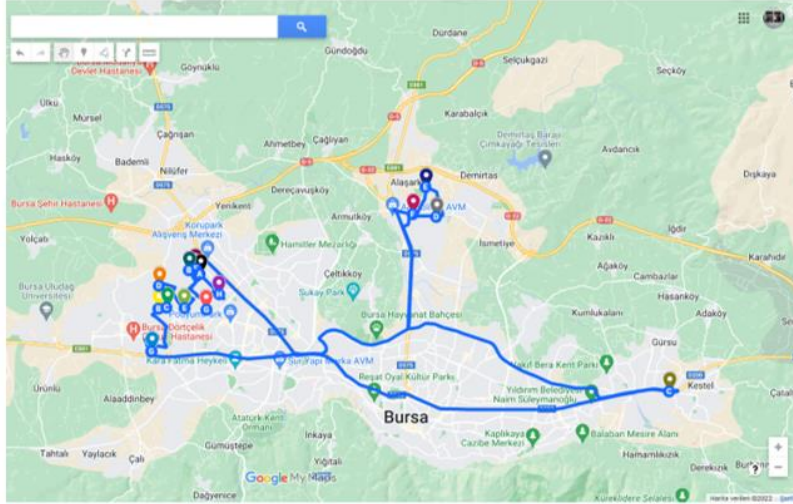
EK 8: İki farklı rota uzunluğunun gösterilmesi

12 BİRİMİN ROTASI



54,452 km

14 BİRİMİN ROTASI



45,211 km

EK 9: Sinyal matrisi

Birimler	PM	ÜM	AB	SÜ
Depo	0	0	0	0
Müşteri_1	0	0	0	0
Müşteri_2	1	1	1	0
Müşteri_3	0	1	0	0
Müşteri_4	0	0	0	0
Müşteri_5	0	0	1	0
Müşteri_6	0	0	0	0
Müşteri_7	0	0	0	0
Müşteri_8	1	1	0	1
Müşteri_9	1	1	1	1
Müşteri_10	0	0	0	0
Müşteri_11	0	0	0	0
Müşteri_12	0	0	0	0
Müşteri_13	0	0	1	0
Müşteri_14	1	1	0	1
Müşteri_15	0	0	0	0
Müşteri_16	0	0	1	0
Müşteri_17	0	0	0	0
Müşteri_18	1	0	0	0
Müşteri_19	0	0	0	0
Müşteri_20	0	0	1	0
Müşteri_21	0	0	1	0
Müşteri_22	1	1	0	0
Müşteri_23	1	0	0	1
Müşteri_24	0	0	0	0
Müşteri_25	1	0	0	0
Müşteri_26	0	0	0	0
Müşteri_27	0	0	1	0
Müşteri_28	0	0	0	0
Müşteri_29	0	0	1	0
Müşteri_30	0	0	1	0

EK 10 Algoritmanın Visual Basic dilinde yazılan kodu

```
Imports System.IO
```

```
Imports OfficeOpenXml
```

```
Public Class Form1
```

```
    Dim sinyalMatrisi As List(Of SignalMatrix) = New List(Of SignalMatrix)
```

```
    Public mesafeMatrisi As List(Of DistanceMatrix) = New List(Of DistanceMatrix)
```

```
    Dim results As List(Of ResultClass)
```

```
    Dim aracSayisi As Int32 = 0
```

```
    Dim mesai As Decimal = 0
```

```
    Dim arizaSuresi As Decimal = 30
```

```
    Dim sinyalSuresi As Decimal = 10
```

```
    Dim katedilenMesafe As Decimal = 0
```

```
    Dim dosyaYoluSignalMatrix As String
```

```
    Dim dosyaYoluDistanceMatrix As String
```

```
    Public haritaMi As Boolean = True
```

```
    Public Class DistanceMatrix
```

```
        Public FromNode As String
```

```
        Public ToNode As String
```

```
        Public Distance As Decimal
```

```
    End Class
```

```
    Public Class SignalMatrix
```

```
        Public Customer As String
```

```
        Public AB As Int32
```

```
        Public PM As Int32
```

```
        Public UM As Int32
```

```
        Public SU As Int32
```

```
        Public isAssigned As Boolean
```

```
    End Class
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
Public Class ResultClass
```

```
    Public Property VehicleNo As Int32
```

```
    Public Property Order As Int32
```

```
    Public Property Customer As String
```

```
    Public Property Distance As Decimal
```

```
    Public Property ServiceTime As Int32
```

```
    Public Property Time As Decimal
```

```
    Public Property Type As String
```

```
    Public Property DistanceTraveled As Decimal
```

```
End Class
```

```
Private Sub btnRun_Click(sender As Object, e As EventArgs) Handles btnRun.Click
```

```
    Try
```

```
        If haritaMi = False Then
```

```
            mesafeMatrisi.Clear()
```

```
            readDistanceXLS(dosyaYoluDistanceMatrix)
```

```
        End If
```

```
    ,
```

```
    pnlSonuc.Visible = True
```

```
    lblSonuc.Visible = True
```

```
    DataGridView1.Visible = True
```

```
    DataGridView1.DataSource = Nothing
```

```
    sinyalarMatrisi.Clear()
```

```
    readSignalXLS(dosyaYoluSignalMatrix)
```

```
    If String.IsNullOrEmpty(txtShiftTime.Text) Then
```

```
        MessageBox.Show("Lütfen mesai süresini giriniz !")
```

```
        Return
```

```
    End If
```

```
    If String.IsNullOrEmpty(txtVehicleNo.Text) Then
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
MessageBox.Show("Lütfen araç sayısını giriniz !")

Return

End If

Try

    aracSayisi = Convert.ToInt32(txtVehicleNo.Text)

    mesai = Convert.ToDecimal(txtShiftTime.Text)

Catch ex As Exception

    MessageBox.Show("Lütfen araç sayısı ve mesai süresine sayı giriniz !")

End Try

Dim aracMesai(aracSayisi) As Decimal

Dim unassigned As List(Of SignalMatrix) = New List(Of SignalMatrix)

Dim toBeAssigned As String = "Depo"

Dim currentNode As String = "Depo"

Dim toBeAssignedDistance As Decimal = 0

results = New List(Of ResultClass)

For i = 0 To aracSayisi - 1

    katedilenMesafe = 0

    Dim result As ResultClass = New ResultClass

    Dim sayac As Int16 = 0

    currentNode = "Depo"

    aracMesai(i) = aracMesai(i)

    unassigned = (From x In sinyalMatrisi Where x.AB = 1 Or x.PM = 1 Or x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()

    If unassigned.Count <> 0 Then

        sayac += 1

        result = New ResultClass()

        result.Customer = "Depo"

        result.Distance = 0
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
result.Order = sayac
result.Time = aracMesai(i)
result.ServiceTime = 0
result.Type = "Depodan Cikis"
result.VehicleNo = i + 1
results.Add(result)
End If
'AB varsa
While unassigned.Count <> 0
    result = New ResultClass()
    unassigned = (From x In sinyalMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()
    toBeAssigned = (From x In mesafeMatrisi Where x.FromNode =
currentNode And unassigned.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderBy(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()
    toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()
    Dim returnDepotTimePrevious As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault()
    Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()
    Dim serviceTime As Decimal = (serviceSignal.AB * arizaSuresi) +
(serviceSignal.PM + serviceSignal.SU + serviceSignal.UM) * sinyalSuresi
    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()
    If aracMesai(i) + toBeAssignedDistance + serviceTime + returnDepotTime
> mesai Then
        Exit While
    End If
End While
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
Else
    sayac += 1
    Dim assigned As SignalMatrix = unassigned.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()
    assigned.isAssigned = True
    aracMesai(i) = aracMesai(i) + serviceTime + toBeAssignedDistance
    katedilenMesafe += toBeAssignedDistance
    currentNode = toBeAssigned
    result.Customer = toBeAssigned
    result.Distance = toBeAssignedDistance
    result.DistanceTraveled = katedilenMesafe
    result.Order = sayac
    result.Time = aracMesai(i)
    result.ServiceTime = serviceTime
    result.Type = "X"
    result.VehicleNo = i + 1
    results.Add(result)
End If
    unassigned = (From x In sinyaliMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()
End While
If currentNode <> "Depo" Then
    sayac += 1
    result = New ResultClass()
    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = currentNode And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()
    aracMesai(i) = aracMesai(i) + returnDepotTime
    katedilenMesafe += returnDepotTime
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
        result.Customer = "Depo"
        result.Distance = returnDepotTime
        result.DistanceTraveled = katedilenMesafe
        result.Order = sayac
        result.Time = aracMesai(i)
        result.ServiceTime = 0
        result.Type = "Depoya Donus"
        result.VehicleNo = i + 1
        results.Add(result)
    End If
Next
Dim ResultBindingSource As New BindingSource
ResultBindingSource.DataSource = results
DataGridView1.DataSource = ResultBindingSource
Catch ex As Exception
End Try
End Sub

Sub readSignalXLS(FilePath As String)
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial
    Dim existingFile As FileInfo = New FileInfo(FilePath)
    Using package As ExcelPackage = New ExcelPackage(existingFile)
        Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)
        Dim colCount As Int32 = worksheet.Dimension.End.Column
        Dim rowCount As Int32 = worksheet.Dimension.End.Row
        For row = 2 To rowCount
            Dim sinyal As SignalMatrix = New SignalMatrix()
            For col = 2 To colCount
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
sinyal.Customer = worksheet.Cells(row, 1).Value?.ToString().Trim()
If col = 2 Then
    sinyal.PM = Convert.ToInt32(worksheet.Cells(row, col).Value)
ElseIf col = 3 Then
    sinyal.UM = Convert.ToInt32(worksheet.Cells(row, col).Value)
ElseIf col = 4 Then
    sinyal.AB = Convert.ToInt32(worksheet.Cells(row, col).Value)
ElseIf col = 5 Then
    sinyal.SU = Convert.ToInt32(worksheet.Cells(row, col).Value)
End If
sinyal.isAssigned = False
Next
sinyalMatrisi.Add(sinyal)
Next
End Using
End Sub

Sub readDistanceXLS(FilePath As String)
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial
    Dim existingFile As FileInfo = New FileInfo(FilePath)
    Using package As ExcelPackage = New ExcelPackage(existingFile)
        Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)
        Dim colCount As Int32 = worksheet.Dimension.End.Column
        Dim rowCount As Int32 = worksheet.Dimension.End.Row
        For row = 2 To rowCount
            For col = 2 To colCount
                Dim mesafe As DistanceMatrix = New DistanceMatrix()
                mesafe.FromNode = worksheet.Cells(row, 1).Value?.ToString().Trim()
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
mesafe.ToNode = worksheet.Cells(1, col).Value?.ToString().Trim()
mesafe.Distance = Convert.ToDecimal(worksheet.Cells(row, col).Value)
mesafeMatrisi.Add(mesafe)

Next
Next
End Using
End Sub

Private Sub btnLoadSignalMatrix_Click(sender As Object, e As EventArgs) Handles
btnLoadSignalMatrix.Click
    Dim file As OpenFileDialog = New OpenFileDialog
    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
    file.Filter = "Excel Dosyası |*.xls| Excel Dosyası|*.xlsx"
    file.FilterIndex = 2
    file.Title = "Excel Dosyası Seçiniz.."
    If file.ShowDialog() = DialogResult.OK Then
        dosyaYoluSignalMatrix = file.FileName
        readSignalXLS(dosyaYoluSignalMatrix)
    End If
    btnHarita.Enabled = True
    btnLoadDistanceMatrix.Enabled = True
End Sub

Private Sub btnLoadDistanceMatrix_Click(sender As Object, e As EventArgs)
Handles btnLoadDistanceMatrix.Click
    Dim file As OpenFileDialog = New OpenFileDialog
    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
    file.Filter = "Excel Dosyası |*.xls| Excel Dosyası|*.xlsx"
    file.FilterIndex = 2
```

EK 10 (devam) Algoritmanın Visual Basic dilinde yazılan kodu

```
file.Title = "Excel Dosyası Seçiniz.."
If file.ShowDialog() = DialogResult.OK Then
    dosyaYoluDistanceMatrix = file.FileName
    readDistanceXLS(dosyaYoluDistanceMatrix)
End If
haritaMi = False
End Sub
Private Sub btnHarita_Click(sender As Object, e As EventArgs) Handles
btnHarita.Click
    mesafeMatrisi.Clear()
    Dim frm As Form2 = New Form2()

    frm.list = (From x In sinyalMatrisi).Select(Function(x) x.Customer).ToList()
    frm.myCaller = Me
    frm.ShowDialog()
End Sub
End Class
```

EK 11 Mesafe matrisinin oluşturulması için yazılan kod

```
Imports GMap.NET
```

```
Imports GMap.NET.MapProviders
```

```
Imports GMap.NET.WindowsForms
```

```
Imports GMap.NET.WindowsForms.Markers
```

```
Public Class Form2
```

```
    Public myCaller As Form1
```

```
    Dim markers As GMapOverlay
```

```
    Public list As List(Of String) = New List(Of String)
```

```
    Dim isMouseDown As Boolean = False
```

```
    Dim sayac As Int32 = 0
```

```
    Dim markNo As Int32 = 0
```

```
    Dim mesafeMatrisi As List(Of Form1.DistanceMatrix) = New List(Of  
Form1.DistanceMatrix)
```

```
    Private Sub btnKaydet_Click(sender As Object, e As EventArgs) Handles  
btnKaydet.Click
```

```
        If markers.Markers.Count() > 0 Then
```

```
            Dim route As GMapRoute = New GMapRoute("getDistance")
```

```
            route.Points.Add(markers.Markers(0).Position)
```

```
            route.Points.Add(markers.Markers(1).Position)
```

```
            Dim distance As Double = 0
```

```
            mesafeMatrisi.Clear()
```

```
            If list.Count > 0 Then
```

EK 11 (devam) Mesafe matrisinin oluşturulması için yazılan kod

```
For i = 0 To list.Count - 1

    For j = 0 To list.Count - 1

        If i <> j Then

            'MessageBox.Show(markers.Markers(i).ToolTipText.ToString() + "-"
+ j.ToString() + "-->" + distance.ToString())

            route.Points.Add(markers.Markers(i).Position)

            route.Points.Add(markers.Markers(j).Position)

            distance = route.Distance

            'MessageBox.Show(i.ToString() + "-" + j.ToString() + "-->" +
distance.ToString())

            route.Clear()

            'MessageBox.Show(Convert.ToDecimal(distance))

        Else

            distance = 0

            'MessageBox.Show(i.ToString() + "-" + j.ToString() + "-->" +
distance.ToString())

        End If

        Dim mesafe As Form1.DistanceMatrix = New Form1.DistanceMatrix()

        mesafe.FromNode = markers.Markers(i).ToolTipText.Trim()

        mesafe.ToNode = markers.Markers(j).ToolTipText.Trim()

        mesafe.Distance = Convert.ToDecimal(distance)

        mesafeMatrisi.Add(mesafe)

    'route = Nothing
```

EK 11 (devam) Mesafe matrisinin oluşturulması için yazılan kod

```
Next
Next
'MessageBox.Show(mesafeMatrisi(0).FromNode + mesafeMatrisi(0).ToNode
+ mesafeMatrisi(0).Distance.ToString())

myCaller.mesafeMatrisi = mesafeMatrisi

myCaller.haritaMi = True

MessageBox.Show("Kayıt başarılı !")

End If

Else

MessageBox.Show("Önce Sinyal Matrisini yüklemelisiniz !")

End If

End Sub
```

Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load

```
mapa.DragButton = MouseButtons.Right

mapa.CanDragMap = True

mapa.MapProvider = GMapProviders.GoogleMap

mapa.ShowCenter = False

mapa.Position = New PointLatLng(40.19266, 29.08403)

mapa.MinZoom = 0

mapa.MaxZoom = 24

mapa.Zoom = 9

mapa.AutoScroll = True
```

EK 11 (devam) Mesafe matrisinin oluşturulması için yazılan kod

```
GMaps.Instance.Mode = AccessMode.ServerAndCache

markers = New GMapOverlay("markers")

Dim X As Double = 29.08403

Dim Y As Double = 40.19266

sayac = 0

For Each item In list

    Dim marker As GMarkerGoogle = New GMarkerGoogle(New PointLatLng(Y,
X), GMarkerGoogleType.green_pushpin)

    marker.ToolTipText = item

    marker.ToolTipMode = MarkerTooltipMode.OnMouseOver

    marker.Tag = sayac

    markers.Markers.Add(marker)

    mapa.Overlays.Add(markers)

    sayac += 1

    'Y += 0.001

    X += 0.002

Next

End Sub

Private Sub mapa_MouseDown(sender As Object, e As MouseEventArgs) Handles
mapa.MouseDown

    isMouseDown = True

    For Each mark In markers.Markers
```

EK 11 (devam) Mesafe matrisinin oluşturulması için yazılan kod

```
If mark.IsMouseOver Then

    markNo = mark.Tag

    isMouseDown = True

End If

Next

End Sub

Private Sub mapa_MouseUp(sender As Object, e As MouseEventArgs) Handles
mapa.MouseUp

    If markers.Markers.Count() > 0 Then

        If isMouseDown Then

            Dim X As Double = mapa.FromLocalToLatLng(e.X, e.Y).Lng

            Dim Y As Double = mapa.FromLocalToLatLng(e.X, e.Y).Lat

            markers.Markers(markNo).Position = New PointLatLng(Y, X)

            'markers.Markers(markNo).ToolTipText      =      "merhaba"      +
markers.Markers(0).Position.ToString()

        End If

    End If

    isMouseDown = False

End Sub

End Class
```

EK 12 Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Imports System.IO
```

```
Imports OfficeOpenXml
```

```
Public Class Form1
```

```
    Dim sinyalMatrisi As List(Of SignalMatrix) = New List(Of SignalMatrix)
```

```
    Public mesafeMatrisi As List(Of DistanceMatrix) = New List(Of DistanceMatrix)
```

```
    Dim results As List(Of ResultClass)
```

```
    Dim aracSayisi As Int32 = 0
```

```
    Dim mesai As Decimal = 0
```

```
    Dim arizaSuresi As Decimal = 30
```

```
    Dim sinyalSuresi As Decimal = 10
```

```
    Dim katedilenMesafe As Decimal = 0
```

```
    Dim dosyaYoluSignalMatrix As String
```

```
    Dim dosyaYoluDistanceMatrix As String
```

```
    Public haritaMi As Boolean = True
```

```
    Public Class DistanceMatrix
```

```
        Public FromNode As String
```

```
        Public ToNode As String
```

```
        Public Distance As Decimal
```

```
    End Class
```

```
    Public Class SignalMatrix
```

```
        Public Customer As String
```

```
        Public AB As Int32
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Public PM As Int32

Public UM As Int32

Public SU As Int32

Public isAssigned As Boolean

End Class

Public Class ResultClass

    Public Property VehicleNo As Int32

    Public Property Order As Int32

    Public Property Customer As String

    Public Property Distance As Decimal

    Public Property ServiceTime As Int32

    Public Property Time As Decimal

    Public Property Type As String

    Public Property DistanceTraveled As Decimal

End Class

Private Sub btnRun_Click(sender As Object, e As EventArgs) Handles btnRun.Click

    Try

        If haritaMi = False Then

            mesafeMatrisi.Clear()

            readDistanceXLS(dosyaYoluDistanceMatrix)

        End If

        pnlSonuc.Visible = True

    End Try

End Sub
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
lblSonuc.Visible = True

DataGridView1.Visible = True

DataGridView1.DataSource = Nothing

sinyalMatrisi.Clear()

readSignalXLS(dosyaYoluSignalMatrix)

If String.IsNullOrEmpty(txtShiftTime.Text) Then
    MessageBox.Show("Lütfen mesai süresini giriniz !")
    Return
End If

If String.IsNullOrEmpty(txtVehicleNo.Text) Then
    MessageBox.Show("Lütfen araç sayısını giriniz !")
    Return
End If

Try
    aracSayisi = Convert.ToInt32(txtVehicleNo.Text)
    mesai = Convert.ToDecimal(txtShiftTime.Text)
Catch ex As Exception
    MessageBox.Show("Lütfen araç sayısı ve mesai süresine sayı giriniz !")
End Try

Dim aracMesai(aracSayisi) As Decimal

Dim unassignedAB As List(Of SignalMatrix) = New List(Of SignalMatrix)
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Dim unassignedNotAB As List(Of SignalMatrix) = New List(Of SignalMatrix)

Dim toBeAssigned As String = "Depo"

Dim currentNode As String = "Depo"

Dim toBeAssignedDistance As Decimal = 0

results = New List(Of ResultClass)

For i = 0 To aracSayisi - 1
    katedilenMesafe = 0

    Dim result As ResultClass = New ResultClass

    Dim sayac As Int16 = 0

    currentNode = "Depo"

    unassignedAB = (From x In sinyalMatrisi Where x.AB = 1 And x.isAssigned
= False).ToList()

    unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or x.SU = 1
Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

    If unassignedAB.Count <> 0 Or unassignedNotAB.Count <> 0 Then

        sayac += 1

        result = New ResultClass()

        result.Customer = "Depo"

        result.Distance = 0

        result.Order = sayac

        result.Time = aracMesai(i)
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
result.ServiceTime = 0

result.Type = "Depodan Cikis"

result.VehicleNo = i + 1

results.Add(result)

End If

'AB varsa
While unassignedAB.Count <> 0
    result = New ResultClass()

    unassignedAB = (From x In sinyalMatrisi Where x.AB = 1 And x.isAssigned
= False).ToList()

    toBeAssigned = (From x In mesafeMatrisi Where x.FromNode =
currentNode And unassignedAB.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderBy(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

    toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

    Dim returnDepotTimePrevious As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault()

    Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Dim serviceTime As Decimal = (serviceSignal.AB * arizaSuresi) +  
(serviceSignal.PM + serviceSignal.SU + serviceSignal.UM) * sinyalSuresi
```

```
Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)  
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)  
x.Distance).FirstOrDefault()
```

```
If aracMesai(i) + toBeAssignedDistance + serviceTime + returnDepotTime  
> mesai Then
```

```
Exit While
```

```
Else
```

```
sayac += 1
```

```
Dim assignedAB As SignalMatrix = unassignedAB.Where(Function(x)  
x.Customer = toBeAssigned).FirstOrDefault()
```

```
assignedAB.isAssigned = True
```

```
aracMesai(i) = aracMesai(i) + serviceTime + toBeAssignedDistance
```

```
katedilenMesafe += toBeAssignedDistance
```

```
currentNode = toBeAssigned
```

```
result.Customer = toBeAssigned
```

```
result.Distance = toBeAssignedDistance
```

```
result.DistanceTraveled = katedilenMesafe
```

```
result.Order = sayac
```

```
result.Time = aracMesai(i)
```

```
result.ServiceTime = serviceTime
```

```
result.Type = "Arıza"
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
result.VehicleNo = i + 1

results.Add(result)

End If

unassignedAB = (From x In sinyalMatrisi Where x.AB = 1 And x.isAssigned
= False).ToList()

End While

unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or x.SU = 1
Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

'//Diğer sinyaller varsa

While (unassignedNotAB.Count <> 0)

result = New ResultClass()

unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or x.SU =
1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

toBeAssigned = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And unassignedNotAB.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderBy(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Dim returnDepotTimePrevious As Decimal =  
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =  
"Depo").Select(Function(x) x.Distance).FirstOrDefault()
```

```
Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)  
x.Customer = toBeAssigned).FirstOrDefault()
```

```
Dim serviceTime As Decimal = (serviceSignal.PM + serviceSignal.SU +  
serviceSignal.UM) * sinyalsuresi + (serviceSignal.AB * arizaSuresi)
```

```
If (aracMesai(i) + toBeAssignedDistance + serviceTime + returnDepotTime  
> mesai) Then
```

```
Exit While
```

```
Else
```

```
sayac += 1
```

```
Dim assignedAB As SignalMatrix =  
unassignedNotAB.Where(Function(x) x.Customer = toBeAssigned).FirstOrDefault()
```

```
assignedAB.isAssigned = True
```

```
aracMesai(i) = aracMesai(i) + toBeAssignedDistance + serviceTime
```

```
katedilenMesafe += toBeAssignedDistance
```

```
currentNode = toBeAssigned
```

```
result.Customer = toBeAssigned
```

```
result.Distance = toBeAssignedDistance
```

```
result.DistanceTraveled = katedilenMesafe
```

```
result.Order = sayac
```

```
result.Time = aracMesai(i)
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
result.ServiceTime = serviceTime

result.Type = "Sinyal"

result.VehicleNo = i + 1

results.Add(result)

End If

unassignedNotAB = sinyaliMatrisi.Where(Function(x) (x.PM = 1 Or x.SU = 1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

End While

If currentNode <> "Depo" Then

    sayac += 1

    result = New ResultClass()

    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode = "Depo").Select(Function(x) x.Distance).FirstOrDefault()

    aracMesai(i) = aracMesai(i) + returnDepotTime

    katedilenMesafe += returnDepotTime

    result.Customer = "Depo"

    result.Distance = returnDepotTime

    result.DistanceTraveled = katedilenMesafe

    result.Order = sayac

    result.Time = aracMesai(i)

    result.ServiceTime = 0
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
        result.Type = "Depoya Donus"

        result.VehicleNo = i + 1

        results.Add(result)

    End If

Next

Dim ResultBindingSource As New BindingSource
ResultBindingSource.DataSource = results
DataGridView1.DataSource = ResultBindingSource

Catch ex As Exception

End Try

End Sub

Sub readSignalXLS(FilePath As String)

    ExcelPackage.LicenseContext = LicenseContext.NonCommercial

    Dim existingFile As FileInfo = New FileInfo(FilePath)

    Using package As ExcelPackage = New ExcelPackage(existingFile)

        Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)

        Dim colCount As Int32 = worksheet.Dimension.End.Column

        Dim rowCount As Int32 = worksheet.Dimension.End.Row

        For row = 2 To rowCount

            Dim sinyal As SignalMatrix = New SignalMatrix()

            For col = 2 To colCount
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
sinyal.Customer = worksheet.Cells(row, 1).Value?.ToString().Trim()  
  
If col = 2 Then  
    sinyal.PM = Convert.ToInt32(worksheet.Cells(row, col).Value)  
  
ElseIf col = 3 Then  
    sinyal.UM = Convert.ToInt32(worksheet.Cells(row, col).Value)  
  
ElseIf col = 4 Then  
    sinyal.AB = Convert.ToInt32(worksheet.Cells(row, col).Value)  
  
ElseIf col = 5 Then  
    sinyal.SU = Convert.ToInt32(worksheet.Cells(row, col).Value)  
  
End If  
  
sinyal.isAssigned = False  
  
Next  
  
sinyalMatrisi.Add(sinyal)  
  
Next  
  
End Using  
  
End Sub  
  
Sub readDistanceXLS(FilePath As String)  
  
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial  
  
  
    Dim existingFile As FileInfo = New FileInfo(FilePath)  
  
    Using package As ExcelPackage = New ExcelPackage(existingFile)
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)

Dim colCount As Int32 = worksheet.Dimension.End.Column

Dim rowCount As Int32 = worksheet.Dimension.End.Row

For row = 2 To rowCount

    For col = 2 To colCount

        Dim mesafe As DistanceMatrix = New DistanceMatrix()

        mesafe.FromNode = worksheet.Cells(row, 1).Value?.ToString().Trim()

        mesafe.ToNode = worksheet.Cells(1, col).Value?.ToString().Trim()

        mesafe.Distance = Convert.ToDecimal(worksheet.Cells(row, col).Value)

        mesafeMatrisi.Add(mesafe)

    Next

Next

End Using

End Sub
```

```
Private Sub btnLoadSignalMatrix_Click(sender As Object, e As EventArgs) Handles
btnLoadSignalMatrix.Click
```

```
    Dim file As OpenFileDialog = New OpenFileDialog

    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)

    file.Filter = "Excel Dosyası |*.xls| Excel Dosyası|*.xlsx"

    file.FilterIndex = 2

    file.Title = "Excel Dosyası Seçiniz.."
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
If file.ShowDialog() = DialogResult.OK Then

    dosyaYoluSignalMatrix = file.FileName

    readSignalXLS(dosyaYoluSignalMatrix)

End If

btnHarita.Enabled = True

btnLoadDistanceMatrix.Enabled = True

End Sub

Private Sub btnLoadDistanceMatrix_Click(sender As Object, e As EventArgs)
Handles btnLoadDistanceMatrix.Click

    Dim file As OpenFileDialog = New OpenFileDialog

    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)

    file.Filter = "Excel Dosyasi |*.xls| Excel Dosyasi|*.xlsx"

    file.FilterIndex = 2

    file.Title = "Excel Dosyası Seçiniz.."

    If file.ShowDialog() = DialogResult.OK Then

        dosyaYoluDistanceMatrix = file.FileName

        readDistanceXLS(dosyaYoluDistanceMatrix)

    End If

    haritaMi = False

End Sub
```

EK 12 (devam) Geliştirilen algoritma için Visual Basic dilinde yazılan kod

```
Private Sub btnHarita_Click(sender As Object, e As EventArgs) Handles  
btnHarita.Click  
  
    mesafeMatrisi.Clear()  
  
    Dim frm As Form2 = New Form2()  
  
    frm.list = (From x In sinyalMatrisi).Select(Function(x) x.Customer).ToList()  
  
    frm.myCaller = Me  
  
    frm.ShowDialog()  
  
End Sub  
  
End Class
```

EK 13 Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
Imports System.IO
```

```
Imports OfficeOpenXml
```

```
Public Class Form1
```

```
    Dim sinyalMatrisi As List(Of SignalMatrix) = New List(Of SignalMatrix)
```

```
    Public mesafeMatrisi As List(Of DistanceMatrix) = New List(Of DistanceMatrix)
```

```
    Dim results As List(Of ResultClass)
```

```
    Dim aracSayisi As Int32 = 0
```

```
    Dim mesai As Decimal = 0
```

```
    Dim arizaSuresi As Decimal = 30
```

```
    Dim sinyalSuresi As Decimal = 10
```

```
    Dim katedilenMesafe As Decimal = 0
```

```
    Dim dosyaYoluSignalMatrix As String
```

```
    Dim dosyaYoluDistanceMatrix As String
```

```
    Public haritaMi As Boolean = True
```

```
    Public Class DistanceMatrix
```

```
        Public FromNode As String
```

```
        Public ToNode As String
```

```
        Public Distance As Decimal
```

```
    End Class
```

```
    Public Class SignalMatrix
```

```
        Public Customer As String
```

```
        Public AB As Int32
```

```
        Public PM As Int32
```

```
        Public UM As Int32
```

```
        Public SU As Int32
```

```
        Public isAssigned As Boolean
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
End Class
```

```
Public Class ResultClass
```

```
    Public Property VehicleNo As Int32
```

```
    Public Property Order As Int32
```

```
    Public Property Customer As String
```

```
    Public Property Distance As Decimal
```

```
    Public Property ServiceTime As Int32
```

```
    Public Property Time As Decimal
```

```
    Public Property Type As String
```

```
    Public Property DistanceTraveled As Decimal
```

```
End Class
```

```
Private Sub btnRun_Click(sender As Object, e As EventArgs) Handles btnRun.Click
```

```
    Try
```

```
        If haritaMi = False Then
```

```
            mesafeMatrisi.Clear()
```

```
            readDistanceXLS(dosyaYoluDistanceMatrix)
```

```
        End If
```

```
        pnlSonuc.Visible = True
```

```
        lblSonuc.Visible = True
```

```
        DataGridView1.Visible = True
```

```
        DataGridView1.DataSource = Nothing
```

```
        sinyalMatrisi.Clear()
```

```
        readSignalXLS(dosyaYoluSignalMatrix)
```

```
        If String.IsNullOrEmpty(txtShiftTime.Text) Then
```

```
            MessageBox.Show("Lütfen mesai süresini giriniz !")
```

```
            Return
```

```
        End If
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
If String.IsNullOrEmpty(txtVehicleNo.Text) Then
    MessageBox.Show("Lütfen araç sayısını giriniz !")
    Return
End
Try
    aracSayisi = Convert.ToInt32(txtVehicleNo.Text)
    mesai = Convert.ToDecimal(txtShiftTime.Text)
Catch ex As Exception
    MessageBox.Show("Lütfen araç sayısı ve mesai süresine sayı giriniz !")
End Try
Dim aracMesai(aracSayisi) As Decimal
Dim unassigned As List(Of SignalMatrix) = New List(Of SignalMatrix)
    Dim toBeAssigned As String = "Depo"
    Dim currentNode As String = "Depo"
    Dim toBeAssignedDistance As Decimal = 0
results = New List(Of ResultClass)
For i = 0 To aracSayisi - 1
    katedilenMesafe = 0
    Dim result As ResultClass = New ResultClass
    Dim sayac As Int16 = 0
    currentNode = "Depo"
    aracMesai(i) = aracMesai(i)
    unassigned = (From x In sinyaliMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()
    If unassigned.Count <> 0 Then
        sayac += 1
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
result = New ResultClass()
result.Customer = "Depo"
result.Distance = 0
result.Order = sayac
result.Time = aracMesai(i)
result.ServiceTime = 0
result.Type = "Depodan Cikis"
result.VehicleNo = i + 1
results.Add(result)
#Region "Once en uzaga git"
For j = 1 To 1
    result = New ResultClass()
    unassigned = (From x In sinyalMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()

    toBeAssigned = (From x In mesafeMatrisi Where x.FromNode =
currentNode And unassigned.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderByDescending(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

    toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode
= currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

    'Dim returnDepotTimePrevious As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault()

    Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()

    Dim serviceTime As Decimal = (serviceSignal.AB * arizaSuresi) +
(serviceSignal.PM + serviceSignal.SU + serviceSignal.UM) * sinyaliSuresi
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()

If aracMesai(i) + toBeAssignedDistance + serviceTime +
returnDepotTime > mesai Then

Exit For

Else

sayac += 1

Dim assigned As SignalMatrix = unassigned.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()

assigned.isAssigned = True

aracMesai(i) = aracMesai(i) + serviceTime + toBeAssignedDistance

katedilenMesafe += toBeAssignedDistance

currentNode = toBeAssigned

result.Customer = toBeAssigned

result.Distance = toBeAssignedDistance

result.DistanceTraveled = katedilenMesafe

result.Order = sayac

result.Time = aracMesai(i)

result.ServiceTime = serviceTime

result.Type = "X"

result.VehicleNo = i + 1

results.Add(result)

End If

unassigned = (From x In sinyalMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()

Next

#End Region
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
End If

While unassigned.Count <> 0

    result = New ResultClass()

    unassigned = (From x In sinyalMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()

    toBeAssigned = (From x In mesafeMatrisi Where x.FromNode =
currentNode And unassigned.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderBy(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

    toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode
= currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

    Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()

    Dim serviceTime As Decimal = (serviceSignal.AB * arizaSuresi) +
(serviceSignal.PM + serviceSignal.SU + serviceSignal.UM) * sinyalSuresi

    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()

    If aracMesai(i) + toBeAssignedDistance + serviceTime + returnDepotTime
> mesai Then

        Exit While

    Else

        sayac += 1

        Dim assigned As SignalMatrix = unassigned.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()

        assigned.isAssigned = True

        aracMesai(i) = aracMesai(i) + serviceTime + toBeAssignedDistance

        katedilenMesafe += toBeAssignedDistance

        currentNode = toBeAssigned
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
result.Customer = toBeAssigned
result.Distance = toBeAssignedDistance
result.DistanceTraveled = katedilenMesafe
result.Order = sayac
result.Time = aracMesai(i)
result.ServiceTime = serviceTime
result.Type = "X"
result.VehicleNo = i + 1
results.Add(result)
End If

unassigned = (From x In sinyalMatrisi Where x.AB = 1 Or x.PM = 1 Or
x.UM = 1 Or x.SU = 1).Where(Function(x) x.isAssigned = False).ToList()
End While

If currentNode <> "Depo" Then
    sayac += 1
    result = New ResultClass()

    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = currentNode And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()

    aracMesai(i) = aracMesai(i) + returnDepotTime
    katedilenMesafe += returnDepotTime
    result.Customer = "Depo"
    result.Distance = returnDepotTime
    result.DistanceTraveled = katedilenMesafe
    result.Order = sayac
    result.Time = aracMesai(i)
    result.ServiceTime = 0
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
        result.Type = "Depoya Donus"
        result.VehicleNo = i + 1
        results.Add(result)
    End If
Next
    Dim ResultBindingSource As New BindingSource
    ResultBindingSource.DataSource = results
    DataGridView1.DataSource = ResultBindingSource
    Catch ex As Exception
    End Try
End Sub
Sub readSignalXLS(FilePath As String)
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial
    Dim existingFile As FileInfo = New FileInfo(FilePath)
    Using package As ExcelPackage = New ExcelPackage(existingFile)
        Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)
        Dim colCount As Int32 = worksheet.Dimension.End.Column
        Dim rowCount As Int32 = worksheet.Dimension.End.Row
        For row = 2 To rowCount
            Dim sinyal As SignalMatrix = New SignalMatrix()
            For col = 2 To colCount
                sinyal.Customer = worksheet.Cells(row, 1).Value?.ToString().Trim()
                If col = 2 Then
                    sinyal.PM = Convert.ToInt32(worksheet.Cells(row, col).Value)
                ElseIf col = 3 Then
                    sinyal.UM = Convert.ToInt32(worksheet.Cells(row, col).Value)
                End If
            Next col
        Next row
    End Using
End Sub
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
ElseIf col = 4 Then
    sinyal.AB = Convert.ToInt32(worksheet.Cells(row, col).Value)
ElseIf col = 5 Then
    sinyal.SU = Convert.ToInt32(worksheet.Cells(row, col).Value)
End If
sinyal.isAssigned = False
Next
sinyalMatrisi.Add(sinyal)
Next
End Using
End Sub
Sub readDistanceXLS(FilePath As String)
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial
    Dim existingFile As FileInfo = New FileInfo(FilePath)
    Using package As ExcelPackage = New ExcelPackage(existingFile)
        Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)
        Dim colCount As Int32 = worksheet.Dimension.End.Column
        Dim rowCount As Int32 = worksheet.Dimension.End.Row
        For row = 2 To rowCount
            For col = 2 To colCount
                Dim mesafe As DistanceMatrix = New DistanceMatrix()

                mesafe.FromNode = worksheet.Cells(row, 1).Value?.ToString().Trim()
                mesafe.ToNode = worksheet.Cells(1, col).Value?.ToString().Trim()
                mesafe.Distance = Convert.ToDecimal(worksheet.Cells(row, col).Value)
                mesafeMatrisi.Add(mesafe)
            
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
Next
Next
End Using
End Sub

Private Sub btnLoadSignalMatrix_Click(sender As Object, e As EventArgs) Handles
btnLoadSignalMatrix.Click
    Dim file As OpenFileDialog = New OpenFileDialog
    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
    file.Filter = "Excel Dosyası |*.xls| Excel Dosyası|*.xlsx"
    file.FilterIndex = 2
    file.Title = "Excel Dosyası Seçiniz.."
    If file.ShowDialog() = DialogResult.OK Then
        dosyaYoluSignalMatrix = file.FileName
        readSignalXLS(dosyaYoluSignalMatrix)
    End If
    btnHarita.Enabled = True
    btnLoadDistanceMatrix.Enabled = True
End Sub

Private Sub btnLoadDistanceMatrix_Click(sender As Object, e As EventArgs)
Handles btnLoadDistanceMatrix.Click
    Dim file As OpenFileDialog = New OpenFileDialog
    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
    file.Filter = "Excel Dosyası |*.xls| Excel Dosyası|*.xlsx"
    file.FilterIndex = 2
    file.Title = "Excel Dosyası Seçiniz.."
```

EK 13 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (sinyalde önceliklendirme olmayan durum için) için Visual Basic dilinde yazılan kod

```
If file.ShowDialog() = DialogResult.OK Then
    dosyaYoluDistanceMatrix = file.FileName
    readDistanceXLS(dosyaYoluDistanceMatrix)
End If

haritaMi = False

End Sub

Private Sub btnHarita_Click(sender As Object, e As EventArgs) Handles
btnHarita.Click
    mesafeMatrisi.Clear()
    Dim frm As Form2 = New Form2()
    frm.list = (From x In sinyalMatrisi).Select(Function(x) x.Customer).ToList()
    frm.myCaller = Me
    frm.ShowDialog()
End Sub

End Class
```

EK 14 Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
Imports System.IO
```

```
Imports OfficeOpenXml
```

```
Public Class Form1
```

```
    Dim sinyalMatrisi As List(Of SignalMatrix) = New List(Of SignalMatrix)
```

```
    Public mesafeMatrisi As List(Of DistanceMatrix) = New List(Of DistanceMatrix)
```

```
    Dim results As List(Of ResultClass)
```

```
    Dim aracSayisi As Int32 = 0
```

```
    Dim mesai As Decimal = 0
```

```
    Dim arizaSuresi As Decimal = 30
```

```
    Dim sinyalSuresi As Decimal = 10
```

```
    Dim katedilenMesafe As Decimal = 0
```

```
    Dim dosyaYoluSignalMatrix As String
```

```
    Dim dosyaYoluDistanceMatrix As String
```

```
    Public haritaMi As Boolean = True
```

```
    Public Class DistanceMatrix
```

```
        Public FromNode As String
```

```
        Public ToNode As String
```

```
        Public Distance As Decimal
```

```
    End Class
```

```
    Public Class SignalMatrix
```

```
        Public Customer As String
```

```
        Public AB As Int32
```

```
        Public PM As Int32
```

```
        Public UM As Int32
```

```
        Public SU As Int32
```

```
        Public isAssigned As Boolean
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
End Class
```

```
Public Class ResultClass
```

```
    Public Property VehicleNo As Int32
```

```
    Public Property Order As Int32
```

```
    Public Property Customer As String
```

```
    Public Property Distance As Decimal
```

```
    Public Property ServiceTime As Int32
```

```
    Public Property Time As Decimal
```

```
    Public Property Type As String
```

```
    Public Property DistanceTraveled As Decimal
```

```
End Class
```

```
Private Sub btnRun_Click(sender As Object, e As EventArgs) Handles btnRun.Click
```

```
    Try
```

```
        If haritaMi = False Then
```

```
            mesafeMatrisi.Clear()
```

```
            readDistanceXLS(dosyaYoluDistanceMatrix)
```

```
        End If
```

```
        pnlSonuc.Visible = True
```

```
        lblSonuc.Visible = True
```

```
        DataGridView1.Visible = True
```

```
        DataGridView1.DataSource = Nothing
```

```
        sinyalMatrisi.Clear()
```

```
        readSignalXLS(dosyaYoluSignalMatrix)
```

```
        If String.IsNullOrEmpty(txtShiftTime.Text) Then
```

```
            MessageBox.Show("Lütfen mesai süresini giriniz !")
```

```
            Return
```

```
        End If
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
If String.IsNullOrEmpty(txtVehicleNo.Text) Then
    MessageBox.Show("Lütfen araç sayısını giriniz !")
    Return
End If
Try
    aracSayisi = Convert.ToInt32(txtVehicleNo.Text)
    mesai = Convert.ToDecimal(txtShiftTime.Text)
Catch ex As Exception
    MessageBox.Show("Lütfen araç sayısı ve mesai süresine sayı giriniz !")
End Try
Dim aracMesai(aracSayisi) As Decimal
Dim unassignedAB As List(Of SignalMatrix) = New List(Of SignalMatrix)
Dim unassignedNotAB As List(Of SignalMatrix) = New List(Of SignalMatrix)
Dim toBeAssigned As String = "Depo"
Dim currentNode As String = "Depo"
Dim toBeAssignedDistance As Decimal = 0
results = New List(Of ResultClass)
For i = 0 To aracSayisi - 1
    katedilenMesafe = 0
    Dim result As ResultClass = New ResultClass
    Dim sayac As Int16 = 0
    currentNode = "Depo"
    unassignedAB = (From x In sinyaliMatrisi Where x.AB = 1 And x.isAssigned = False).ToList()
    unassignedNotAB = sinyaliMatrisi.Where(Function(x) (x.PM = 1 Or x.SU = 1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()
    If unassignedAB.Count <> 0 Or unassignedNotAB.Count <> 0 Then
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
sayac += 1
result = New ResultClass()
result.Customer = "Depo"
result.Distance = 0
result.Order = sayac
result.Time = aracMesai(i)
result.ServiceTime = 0
result.Type = "Depodan Cikis"
result.VehicleNo = i + 1
results.Add(result)
#Region "Once en uzaga git"
If unassignedAB.Count > 0 Then
    For j = 1 To 1
        result = New ResultClass()
        unassignedAB = (From x In sinyaliMatrisi Where x.AB = 1 And
x.isAssigned = False).ToList()
        toBeAssigned = (From x In mesafeMatrisi Where x.FromNode =
currentNode And unassignedAB.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderByDescending(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault() 'en kısa mesafedeki müşteri
geldi
        toBeAssignedDistance = mesafeMatrisi.Where(Function(x)
x.FromNode = currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault() 'mesafeyi alıyo
        'Dim returnDepotTimePrevious As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault() 'mevcuttakinden depoya
dönme süresi 1-3 için 1den dönme mesafesi
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
Dim serviceSignal As SignalMatrix =  
sinyalMatrisi.Where(Function(x) x.Customer = toBeAssigned).FirstOrDefault()  
'atanacak müşteriyi sadece getirmemi sağlıyo
```

```
Dim serviceTime As Decimal = (serviceSignal.AB * arizaSuresi) +  
(serviceSignal.PM + serviceSignal.SU + serviceSignal.UM) * sinyaSuresi
```

```
Dim returnDepotTime As Decimal =  
mesafeMatrisi.Where(Function(x) x.FromNode = toBeAssigned And x.ToNode =  
"Depo").Select(Function(x) x.Distance).FirstOrDefault() '3ten depoya dönme mesafesi
```

```
If aracMesai(i) + toBeAssignedDistance + serviceTime +  
returnDepotTime > mesai Then
```

```
Exit For
```

```
Else
```

```
sayac += 1
```

```
'previousNode = currentNode
```

```
Dim assignedAB As SignalMatrix =  
unassignedAB.Where(Function(x) x.Customer = toBeAssigned).FirstOrDefault()
```

```
assignedAB.isAssigned = True
```

```
aracMesai(i) = aracMesai(i) + serviceTime + toBeAssignedDistance  
'3teki hizmet, 1den 3 e girme mesafesi
```

```
katedilenMesafe += toBeAssignedDistance
```

```
currentNode = toBeAssigned 'mevcut müşteriyi atanana eşitliyo
```

```
result.Customer = toBeAssigned
```

```
result.Distance = toBeAssignedDistance
```

```
result.DistanceTraveled = katedilenMesafe
```

```
result.Order = sayac
```

```
result.Time = aracMesai(i)
```

```
result.ServiceTime = serviceTime
```

```
result.Type = "Arıza"
```

```
result.VehicleNo = i + 1
```

```
results.Add(result)
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
End If

unassignedAB = (From x In sinyalMatrisi Where x.AB = 1 And
x.isAssigned = False).ToList()

Next

Else

For j = 1 To 1

unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or
x.SU = 1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

toBeAssigned = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And unassignedNotAB.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderByDescending(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

toBeAssignedDistance = mesafeMatrisi.Where(Function(x)
x.FromNode = currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

Dim returnDepotTime As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = toBeAssigned And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault()

Dim returnDepotTimePrevious As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault()

Dim serviceSignal As SignalMatrix =
sinyalMatrisi.Where(Function(x) x.Customer = toBeAssigned).FirstOrDefault()

Dim serviceTime As Decimal = (serviceSignal.PM + serviceSignal.SU
+ serviceSignal.UM) * sinyalSuresi + (serviceSignal.AB * arizaSuresi)

If (aracMesai(i) + toBeAssignedDistance + serviceTime +
returnDepotTime > mesai) Then

Exit For

Else

sayac += 1
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
Dim assignedAB As SignalMatrix =  
unassignedNotAB.Where(Function(x) x.Customer = toBeAssigned).FirstOrDefault()  
    assignedAB.isAssigned = True  
    aracMesai(i) = aracMesai(i) + toBeAssignedDistance + serviceTime  
    katedilenMesafe += toBeAssignedDistance  
    currentNode = toBeAssigned  
    result.Customer = toBeAssigned  
    result.Distance = toBeAssignedDistance  
    result.DistanceTraveled = katedilenMesafe  
    result.Order = sayac  
    result.Time = aracMesai(i)  
    result.ServiceTime = serviceTime  
    result.Type = "Sinyal"  
    result.VehicleNo = i + 1  
    results.Add(result)  
End If  
    unassignedNotAB = sinyaliMatrisi.Where(Function(x) (x.PM = 1 Or  
x.SU = 1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()  
Next  
#End Region  
End If  
End If  
'AB varsa  
While unassignedAB.Count <> 0  
    result = New ResultClass()  
    unassignedAB = (From x In sinyaliMatrisi Where x.AB = 1 And  
x.isAssigned = False).ToList()
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
toBeAssigned = (From x In mesafeMatrisi Where x.FromNode =
currentNode And unassignedAB.Select(Function(y
y.Customer).ToList().Contains(x.ToNode)).OrderBy(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()

Dim serviceTime As Decimal = (serviceSignal.AB * arizaSuresi) +
(serviceSignal.PM + serviceSignal.SU + serviceSignal.UM) * sinyalSuresi

Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()

If aracMesai(i) + toBeAssignedDistance + serviceTime + returnDepotTime
> mesai Then

Exit While

Else

sayac += 1

Dim assignedAB As SignalMatrix = unassignedAB.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()

assignedAB.isAssigned = True

aracMesai(i) = aracMesai(i) + serviceTime + toBeAssignedDistance

katedilenMesafe += toBeAssignedDistance

currentNode = toBeAssigned

result.Customer = toBeAssigned

result.Distance = toBeAssignedDistance

result.DistanceTraveled = katedilenMesafe

result.Order = sayac

result.Time = aracMesai(i)
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
result.ServiceTime = serviceTime
result.Type = "Arıza"
result.VehicleNo = i + 1
results.Add(result)
End If

unassignedAB = (From x In sinyalMatrisi Where x.AB = 1 And
x.isAssigned = False).ToList()
End While

unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or x.SU = 1
Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

'//Diğer sinyaller varsa
While (unassignedNotAB.Count <> 0)
    result = New ResultClass()
    unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or x.SU =
1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()

    toBeAssigned = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And unassignedNotAB.Select(Function(y)
y.Customer).ToList().Contains(x.ToNode)).OrderBy(Function(x)
x.Distance).Select(Function(x) x.ToNode).FirstOrDefault()

    toBeAssignedDistance = mesafeMatrisi.Where(Function(x) x.FromNode =
currentNode And x.ToNode = toBeAssigned).Select(Function(x)
x.Distance).FirstOrDefault()

    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = toBeAssigned And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()

    Dim returnDepotTimePrevious As Decimal =
mesafeMatrisi.Where(Function(x) x.FromNode = currentNode And x.ToNode =
"Depo").Select(Function(x) x.Distance).FirstOrDefault()

    Dim serviceSignal As SignalMatrix = sinyalMatrisi.Where(Function(x)
x.Customer = toBeAssigned).FirstOrDefault()
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
Dim serviceTime As Decimal = (serviceSignal.PM + serviceSignal.SU +  
serviceSignal.UM) * sinyalSuresi + (serviceSignal.AB * arizaSuresi)
```

```
If (aracMesai(i) + toBeAssignedDistance + serviceTime +  
returnDepotTime > mesai) Then
```

```
Exit While
```

```
Else
```

```
sayac += 1
```

```
Dim assignedAB As SignalMatrix =  
unassignedNotAB.Where(Function(x) x.Customer = toBeAssigned).FirstOrDefault()
```

```
assignedAB.isAssigned = True
```

```
aracMesai(i) = aracMesai(i) + toBeAssignedDistance + serviceTime
```

```
katedilenMesafe += toBeAssignedDistance
```

```
currentNode = toBeAssigned
```

```
result.Customer = toBeAssigned
```

```
result.Distance = toBeAssignedDistance
```

```
result.DistanceTraveled = katedilenMesafe
```

```
result.Order = sayac
```

```
result.Time = aracMesai(i)
```

```
result.ServiceTime = serviceTime
```

```
result.Type = "Sinyal"
```

```
result.VehicleNo = i + 1
```

```
results.Add(result)
```

```
End If
```

```
unassignedNotAB = sinyalMatrisi.Where(Function(x) (x.PM = 1 Or x.SU =  
1 Or x.UM = 1)).Where(Function(x) x.isAssigned = False).ToList()
```

```
End While
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
If currentNode <> "Depo" Then
    sayac += 1
    result = New ResultClass()
    Dim returnDepotTime As Decimal = mesafeMatrisi.Where(Function(x)
x.FromNode = currentNode And x.ToNode = "Depo").Select(Function(x)
x.Distance).FirstOrDefault()
    aracMesai(i) = aracMesai(i) + returnDepotTime
    katedilenMesafe += returnDepotTime
    result.Customer = "Depo"
    result.Distance = returnDepotTime
    result.DistanceTraveled = katedilenMesafe
    result.Order = sayac
    result.Time = aracMesai(i)
    result.ServiceTime = 0
    result.Type = "Depoya Donus"
    result.VehicleNo = i + 1
    results.Add(result)
End If
Next
Dim ResultBindingSource As New BindingSource
ResultBindingSource.DataSource = results
DataGridView1.DataSource = ResultBindingSource
Catch ex As Exception
End Try
End Sub
Sub readSignalXLS(FilePath As String)
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
Dim existingFile As FileInfo = New FileInfo(FilePath)
Using package As ExcelPackage = New ExcelPackage(existingFile)
    Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)
    Dim colCount As Int32 = worksheet.Dimension.End.Column
    Dim rowCount As Int32 = worksheet.Dimension.End.Row
    For row = 2 To rowCount
        Dim sinyal As SignalMatrix = New SignalMatrix()
        For col = 2 To colCount
            sinyal.Customer = worksheet.Cells(row, 1).Value?.ToString().Trim()
            If col = 2 Then
                sinyal.PM = Convert.ToInt32(worksheet.Cells(row, col).Value)
            ElseIf col = 3 Then
                sinyal.UM = Convert.ToInt32(worksheet.Cells(row, col).Value)
            ElseIf col = 4 Then
                sinyal.AB = Convert.ToInt32(worksheet.Cells(row, col).Value)
            ElseIf col = 5 Then
                sinyal.SU = Convert.ToInt32(worksheet.Cells(row, col).Value)
            End If
            sinyal.isAssigned = False
        Next
        sinyalMatrisi.Add(sinyal)
    Next
End Using
End Sub
Sub readDistanceXLS(FilePath As String)
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
Dim existingFile As FileInfo = New FileInfo(FilePath)
Using package As ExcelPackage = New ExcelPackage(existingFile)
    Dim worksheet As ExcelWorksheet = package.Workbook.Worksheets(0)
    Dim colCount As Int32 = worksheet.Dimension.End.Column
    Dim rowCount As Int32 = worksheet.Dimension.End.Row

    For row = 2 To rowCount
        For col = 2 To colCount
            Dim mesafe As DistanceMatrix = New DistanceMatrix()
            mesafe.FromNode = worksheet.Cells(row, 1).Value?.ToString().Trim()
            mesafe.ToNode = worksheet.Cells(1, col).Value?.ToString().Trim()
            mesafe.Distance = Convert.ToDecimal(worksheet.Cells(row, col).Value)
            mesafeMatrisi.Add(mesafe)
        Next
    Next
End Using
End Sub

Private Sub btnLoadSignalMatrix_Click(sender As Object, e As EventArgs) Handles
btnLoadSignalMatrix.Click
    Dim file As OpenFileDialog = New OpenFileDialog
    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
    file.Filter = "Excel Dosyasi |*.xls| Excel Dosyasi|*.xlsx"
    file.FilterIndex = 2
    file.Title = "Excel Dosyası Seçiniz.."
    If file.ShowDialog() = DialogResult.OK Then
        dosyaYoluSignalMatrix = file.FileName
    End If
End Sub
```

EK 14 (devam) Rotaya en uzak birimden başlanmasını sağlayan algoritma (arıza bilgisi sinyalinin öncelikli olduğu durum için) için Visual Basic dilinde yazılan kod

```
        readSignalXLS(dosyaYoluSignalMatrix)

    End If

    btnHarita.Enabled = True

    btnLoadDistanceMatrix.Enabled = True

End Sub

Private Sub btnLoadDistanceMatrix_Click(sender As Object, e As EventArgs) Handles btnLoadDistanceMatrix.Click

    Dim file As OpenFileDialog = New OpenFileDialog

    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)

    file.Filter = "Excel Dosyasi |*.xls| Excel Dosyasi|*.xlsx"

    file.FilterIndex = 2

    file.Title = "Excel Dosyası Seçiniz.."

    If file.ShowDialog() = DialogResult.OK Then

        dosyaYoluDistanceMatrix = file.FileName

        readDistanceXLS(dosyaYoluDistanceMatrix)

    End If

    haritaMi = False

End Sub

Private Sub btnHarita_Click(sender As Object, e As EventArgs) Handles btnHarita.Click

    mesafeMatrisi.Clear()

    Dim frm As Form2 = New Form2()

    frm.list = (From x In sinyaliMatrisi).Select(Function(x) x.Customer).ToList()

    frm.myCaller = Me

    frm.ShowDialog()

End Sub

End Class
```