

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

**BİR SÜPERMARKET ZİNCİRİNDE ROTALAMA
PROBLEMİNİN METASEZGİSEL ALGORİTMALAR
İLE ÇÖZÜLMESİ**

YÜKSEK LİSANS TEZİ

Serap ERCAN

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

Haziran 2014

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BİR SÜPERMARKET ZİNCİRİNDE ROTALAMA
PROBLEMİNİN METASEZGİSEL ALGORİTMALAR
İLE ÇÖZÜLMESİ**

YÜKSEK LİSANS TEZİ

Serap ERCAN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

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



**Prof. Dr. İ. Hakkı
Cedimoğlu
Jüri Başkanı**



**Doç. Dr. Harun R.
Yazgan
Üye**



**Doç. Dr. Cemil
Öz
Üye**

TEŐEKKÜR

Yüksek lisans eğitimin ve tez çalışmam boyunca yardım ve katkılarıyla beni yönlendiren değerli hocam Doç. Dr. Harun Reşit YAZGAN'a teşekkürü bir borç bilirim. Ayrıca tez çalışmamda yazılım konusunda destek sağlayan Öğr. Gör. M. Akif ŞAHMAN hocama ve her zaman arkamda olan, beni sürekli destekleyen ve bugünlere gelmemde en büyük emeğe sahip olan değerli aileme teşekkürlerimi sunarım.

Son olarak yüksek lisans tezimi Yurt İçi Yüksek Lisans Bursu ile maddi olarak destekleyen TÜBİTAK'a teşekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	viii
TABLolar LİSTESİ	ix
ÖZET	xi
SUMMARY	xii

BÖLÜM.1.

GİRİŞ	1
-------------	---

BÖLÜM.2.

GEZGİN SATICI VE ARAÇ ROTALAMA PROBLEMİ	3
2.1. Gezgin Satıcı Problemi (GSP)	3
2.2. Araç Rotalama Problemi (ARP)	4
2.2.1. Araç rotalama probleminin tanımı	4
2.2.2. Araç rotalama probleminin temel bileşenleri.....	5
2.2.3. Araç rotalama probleminin matematiksel modeli.....	6
2.3. Araç Rotalama Probleminin Türleri.....	8
2.3.1. Kapasite kısıtlı ARP	8
2.3.2. Zaman pencereci ARP	9
2.3.3. Dağıtım ve toplamalı ARP	10
2.3.4. Mesafe kısıtlı ARP	11
2.3.5. Çok depolu ARP.....	12
2.4. Araç Rotalama Problemlerinin Çözüm Yöntemleri.....	13
2.4.1. Kesin yöntemler	14
2.4.1.1. Dal ve sınır yöntemi.....	14

2.4.1.2. Kesme düzlemi yöntemi	16
2.4.1.3. Dal ve kesme yöntemi.....	17
2.4.1.4. Dinamik programlama	17
2.4.2. Klasik sezgisel yöntemler	18
2.4.2.1. Yapısal sezgisel (tur kurucu) algoritmalar	18
2.4.2.2. İyileştirmeli sezgisel (tur geliştirici) algoritmalar.....	21
2.4.2.3. İki aşamalı sezgisel yöntemler	25
2.4.3. Metasezgisel yöntemler.....	30
2.4.3.1. Tavlama benzetimi	31
2.4.3.2. Tabu arama algoritması.....	32
2.4.3.3. Parçacık sürü optimizasyonu	33
2.4.3.4. Karınca kolonisi algoritması	34
2.4.3.5. Genetik algoritma.....	34
2.4.3.6. Yapay arı kolonisi algoritması	35

BÖLÜM.3.

ÇALIŞMADA KULLANILAN ALGORİTMALAR	36
3.1. Literatür Araştırması	36
3.2. Karınca Kolonisi Algoritması	40
3.2.1. Gerçek karıncalar	40
3.2.2. Yapay karıncalar ve karınca kolonisi algoritması	41
3.2.2.1. Geçiş kuralı	42
3.2.2.2. Feromon güncellemesi	43
3.3. Genetik Algoritma.....	46
3.3.1. Genetik algoritmanın tanımı	46
3.3.2. Genetik algoritmanın terimleri.....	46
3.3.3. Genetik algoritmanın parametreleri	48
3.3.3.1. Popülasyon büyüklüğü.....	48
3.3.3.2. Üreme Parametresi.....	48
3.3.3.3. Çaprazlama olasılığı.....	49
3.3.3.4. Mutasyon olasılığı.....	49
3.3.3.5. Kuşak farkı.....	49
3.3.3.6. Seçim stratejisi	49
3.3.3.7. Uygunluk fonksiyonunun ölçeklenmesi	50

3.3.4. Genetik algoritmanın çalışma prensibi	50
3.4. Yapay Arı Kolonisi Algoritması.....	51
3.4.1. Arılar ve sürü zekası	51
3.4.2. Arılarda besin kaynağı bulma	52
3.4.3. Yapay arı kolonisi algoritması.....	53
BÖLÜM.4.	
BİR SÜPERMARKET ZİNCİRİNDE ROTALAMA PROBLEMİNİN METASEZGİSEL ALGORİTMALAR İLE ÇÖZÜLMESİ.....	58
4.1. Problemin Tanımı.....	58
4.2. Üç Farklı Algoritma ile Problemin Çözümü.....	61
4.3. Sonuçların ANOVA Testi ile Yorumlanması.....	74
4.3.1. Taze ürünün yoğun günde dağıtımını alt problemi için ANOVA testi.....	74
4.3.2. Kuru ürünün yoğun günde dağıtımını alt problemi için ANOVA testi.....	76
4.3.3. Taze ürünün sakin günde dağıtımını alt problemi için ANOVA testi.....	78
4.3.4. Kuru ürünün sakin günde dağıtımını alt problemi için ANOVA testi.....	80
4.3.5. Alt problemlerin ANOVA testi sonuçlarının karşılaştırılması ...	82
BÖLÜM.5.	
SONUÇLAR VE ÖNERİLER	84
KAYNAKLAR.....	87
EKLER.....	96
ÖZGEÇMİŞ	108

SİMGELER VE KISALTMALAR LİSTESİ

Q	: Araç kapasitesi
N	: Müşteri veya durak sayısı
q_i	: i müşterisinin talep miktarı
d_{ij}	: i müşterisi ile j müşterisi arasındaki uzaklık
d_j	: j müşterisinin dağıtım talebi miktarı
p_i	: i müşterisinin toplama talebi miktarı
s_i	: Servis süresi
L	: Bir aracın gidebileceği maksimum mesafe
s_{ij}	: i ve j müşteri çifti için tasarruf değeri
d_{ik}	: Sezgisel maliyet
l_{ij}	: Gezgin satıcı problemi turunda i 'den j 'ye gitme maliyeti
E_i	: i durumunda bulunan katının enerjisi
q_0	: Sözde-rastlantısal-orantılı durum geçiş kuralı
$\tau(i, j)$: i ve j noktaları arasındaki feromon miktarı
$\eta(i, j)$: i ve j noktaları arasındaki mesafeyle ters orantılı olan seçilebilirlik parametresi
α	: Feromon miktarının önemini belirleyen sabit
β	: Sezgisel bilginin önemini belirleyen sabit
$j_k(i)$: i noktasından gidilebilecek ve henüz k karıncası tarafından ziyaret edilmemiş noktalar kümesi
$p_k(i, j)$: Yolların seçilme olasılığı
ρ	: Buharlaşma oranı
τ_0	: Bütün kenarlara başlangıçta atanmış olan feromon miktarı
P_c	: Çaprazlama olasılığı
P_m	: Mutasyon olasılığı
SN	: Besin kaynaklarının sayısı

D	: Optimize edilecek parametre sayısı
x_j^{\min}	: j. parametrenin alt sınırı
x_j^{\max}	: j. parametrenin üst sınırı
failure _i	: Başlangıç kaynaklarına ait deneme sayaçları
fitness _i	: Kaynağın kalitesi
ANOVA	: Varyans analizi
ARP	: Araç rotalama problemi
ÇDARP	: Çok depolu araç rotalama problemi
DTARP	: Dağıtım ve toplamalı araç rotalama problemi
GA	: Genetik algoritma
GAP	: Genelleştirilmiş atama problemi
GSP	: Gezgin satıcı problemi
KKARP	: Kapasite kısıtlı araç rotalama problemi
KKA	: Karınca kolonisi algoritması
MKARP	: Mesafe kısıtlı araç rotalama problemi
NP	: Non polinomial
PSO	: Parçacık sürü optimizasyonu
TB	: Tavlama benzetimi
YAKA	: Yapay arı kolonisi algoritması
ZPARP	: Zaman pencereli araç rotalama problemi

ŞEKİLLER LİSTESİ

Şekil 2.1. Kapasite kısıtlı ARP örneği.....	8
Şekil 2.2. Zaman pencereci ARP örneği.....	9
Şekil 2.3. Dağıtım ve toplamalı ARP örneği.....	10
Şekil 2.4. Çok depolu ARP örneği.....	13
Şekil 2.5. Çözüm uzayı ve araştırma ağacındaki gösterimi.....	15
Şekil 2.6. Çözüm uzayının alt kümelerine ayrılması ve araştırma ağacında gösterimi.	15
Şekil 2.7. Çözüm alt uzaylarının değerlendirilmesi ve yeni alt çözümlere ulaşılması.	15
Şekil 2.8. Periyodik transfer uygulamadan önceki rotalar.....	23
Şekil 2.9. Periyodik transfer uygulandıktan sonraki rotalar.....	23
Şekil 2.10. Zincir çaprazlama örneği.....	24
Şekil 2.11. Zincir deęiş-tokuşu örneği.....	24
Şekil 2.12. Zincir yer deęiştirme örneği.....	24
Şekil 3.1. Gerçek karıncaların en kısa yolu bulma adımları.....	41
Şekil 3.2. Karınca kolonisi algoritmasının akış diyagramı.....	45
Şekil 3.3. Genetik algoritmanın akış diyagramı.....	51
Şekil 3.4. Arıların besin kaynağı arama davranışları.....	53
Şekil 3.5. Yapay arı kolonisi algoritmasının akış diyagramı.....	57
Şekil 4.1. Ana depo ve mağazaların haritadaki gösterimi.....	59
Şekil 4.2. KKA ile oluşturulan kümelerin haritadaki gösterimi.....	67
Şekil 4.3. GA ile oluşturulan kümelerin haritadaki gösterimi.....	70
Şekil 4.4. YAKA ile oluşturulan kümelerin haritadaki gösterimi.....	73

TABLolar LİSTESİ

Tablo 3.1. Çalışmada kullanılan algoritmalar ile yapılmış bazı ARP çalışmaları.....	36
Tablo 4.1. Mağazaların numaraları ve çeşitleri	60
Tablo 4.2. Önerilen yöntemin çözüm aşamaları.....	61
Tablo 4.3. Karınca kolonisi algoritması ile elde edilen rota.....	64
Tablo 4.4. Taze ürünün yoğun günde dağıtım alt probleminin 1. haftadaki talepleri	65
Tablo 4.5. KKA sonucunda oluşan kümeler ve uğranacak mağazalar ..	66
Tablo 4.6. KKA sonucunda araçların izleyeceği rotalar ve kat ettikleri mesafeler. ...	68
Tablo 4.7. Genetik algoritma ile elde edilen rota	68
Tablo 4.8. GA sonucunda oluşan kümeler ve uğranacak mağazalar	69
Tablo 4.9. GA sonucunda araçların izleyeceği rotalar ve kat ettikleri mesafeler.....	71
Tablo 4.10. Yapay arı kolonisi ile elde edilen rota	71
Tablo 4.11. YAKA sonucunda oluşan kümeler ve uğranacak mağazalar	72
Tablo 4.12. YAKA sonucunda araçların izleyeceği rotalar ve kat ettikleri mesafeler.	74
Tablo 4.13. Taze ürünün yoğun günde dağıtım alt problemi için KKA, GA ve YAKA sonuçları	75
Tablo 4.14. Taze ürünün yoğun günde dağıtım alt problemi için varyansların homojenliği testi	76
Tablo 4.15. Taze ürünün yoğun günde dağıtım alt problemi için varyans analizi tablosu	76
Tablo 4.16. Taze ürünün yoğun günde dağıtım alt problemi için tanımlayıcı istatistikler	76
Tablo 4.17. Kuru ürünün yoğun günde dağıtım alt problemi için KKA, GA ve YAKA sonuçları	77
Tablo 4.18. Kuru ürünün yoğun günde dağıtım alt problemi için varyansların homojenliği testi	78

Tablo 4.19. Kuru ürünün yoğun günde dağıtımı alt problemi için varyans analizi tablosu	78
Tablo 4.20. Taze ürünün sakın günde dağıtımı alt problemi için KKA, GA ve YAKA sonuçları	78
Tablo 4.21. Taze ürünün sakın günde dağıtımı alt problemi için varyansların homojenliği testi	80
Tablo 4.22. Taze ürünün sakın günde dağıtımı alt problemi için varyans analizi tablosu	80
Tablo 4.23. Taze ürünün sakın günde dağıtımı alt problemi için Welch ve Brown-Forsythe testi.	80
Tablo 4.24. Taze ürünün sakın günde dağıtımı alt problemi için tanımlayıcı istatistikler	80
Tablo 4.25. Kuru ürünün sakın günde dağıtımı alt problemi için KKA, GA ve YAKA sonuçları	81
Tablo 4.26. Kuru ürünün sakın günde dağıtımı alt problemi için varyansların homojenliği testi	82
Tablo 4.27. Kuru ürünün sakın günde dağıtımı alt problemi için varyans analizi tablosu	82
Tablo 4.28. Alt problemlerin ANOVA testi sonuçları..	82

ÖZET

Anahtar kelimeler: Araç Rotalama Problemi, Karınca Kolonisi Algoritması, Genetik Algoritma, Yapay Arı Kolonisi Algoritması

Kapasite kısıtlı araç rotalama problemi (KKARP), toplam kat edilen mesafe minimum olacak şekilde, araçların ortak bir depodan talepleri belirli olan müşterilere servis yapmak için izlemesi gereken rotaların kapasite kısıtı altında oluşturulduğu problemlerdir.

Bu tez çalışmasında, bir süpermarket zincirinin haftalık taleplerinin karşılanması için en uygun rotanın belirlenmesi problemi çözülmüştür. Ele alınan problem NP-zor olduğundan dolayı kesin çözümlü matematiksel modeller yerine metasezgisel algoritmaların kullanılması uygun olacaktır. Bu amaçla karınca kolonisi, genetik ve yapay arı kolonisi algoritmaları uyarlanmış ve elde edilen sonuçlar ANOVA testi yardımıyla karşılaştırılmıştır.

SOLUTION OF ROUTING PROBLEM IN THE SUPERMARKET CHAIN BY THE METAHEURISTIC ALGORITHMS

SUMMARY

Key Words: Vehicle Routing Problem, Ant Colony Algorithm, Genetic Algorithm, Artificial Bee Colony Algorithm

Capacity constraint vehicle routing problem (CCVRP) deals with the minimum distance routes for vehicles that serve customers who have specific demands from a common warehouse under capacity constraints.

In this study, optimal routing problem that meet the weekly demands of a supermarket was solved. The kind of this problem is non-polynomial-hard (NP-hard), so solution of this will be appropriate to employ metaheuristics rather than exact methods. Metaheuristic algorithms such as an ant colony, a genetic and an artificial bee colony algorithms were implemented to solve the problem. An ANOVA test was carried out to investigate effectiveness of the algorithms.

BÖLÜM 1. GİRİŞ

Günümüzdeki zorlu rekabet ortamında işletmelerin rakiplerine göre önde olabilmeleri için ürettikleri ürün ve hizmetleri müşterilerine en hızlı şekilde ulaştırmaları gerekmektedir. İşletmeler bu nedenle tedarik zincirine ve tedarik zincirinin içerisindeki lojistik sistemlere her zamankinden daha çok önem vermek zorundadır.

Lojistik; bir ürünü tedarikçilerinden müşterilere ulaştırabilmek için gerekli olan tüm faaliyetlerdir. Bu faaliyetler tedarik zinciri olarak adlandırılan bir akış içinde gerçekleştirilir. Lojistik maliyetlerinin en aza indirilmesinde araçların rotalanması çok önemlidir.

Araç rotalama problemi (ARP), bir işletmenin konumları belli olan “n” adet müşterisine bir veya birden fazla depodan hizmet verebilmek için araçların optimum rotalarının belirlenmesi problemidir. Lojistik sisteminde yer alan kısıtlara göre birçok ARP çeşidi bulunmaktadır. Bu kısıtların en önemlileri araç kapasitesi, aracın bir defada yol alabileceği maksimum mesafe ve süre kısıtıdır.

Literatürde araç rotalama problemlerinin çözümü için birçok yöntem geliştirilmiştir. Bu yöntemler kesin, klasik sezgisel ve metasezgisel olmak üzere üçe ayrılmaktadır. Araç rotalama problemi NP-zor problem sınıfından olduğu için çok değişkenli problemlerde çözüme kesin yöntemlerle ulaşmak oldukça zor ve hatta bazen imkânsız olmaktadır. Problemin kısıtları arttıkça problem daha da karmaşık hale gelmekte ve optimum sonuca ulaşmak daha da zorlaşmaktadır. Bu nedenle problemin boyutu büyüdükçe kısa sürede iyi kalitede sonuçlar üreten klasik sezgiseller ve çözüm uzayındaki olurlu bölgelerde derin araştırmalar yapabilen metasezgiseller tercih edilmektedir.

Bu çalışmada ilk olarak (ikinci bölümde) araç rotalama problemi irdelenmekte, literatürde yer alan temel araç rotalama problemi türleri ve çözüm yöntemleri açıklanmaktadır.

Üçüncü bölümde metasezgisel yöntemler içinde kısaca açıklanan karınca kolonisi, genetik ve yapay arı kolonisi algoritması hakkında ayrıntılı bilgiler yer almaktadır.

Dördüncü bölümde bir süpermarket zincirinin haftalık taleplerinin karşılanmasında ortaya çıkan araç rotalama problemi geliştirilen karınca kolonisi, genetik ve yapay arı kolonisi algoritmasıyla çözülmüş, sonuçları da ANOVA testi yardımıyla karşılaştırılmıştır.

Son bölümde ise yapılan çalışmalar ve elde edilen sonuçlar özetlenmiştir.

BÖLÜM 2. GEZGİN SATICI VE ARAÇ ROTALAMA PROBLEMİ

2.1. Gezgin Satıcı Problemi (GSP)

Gezgin satıcı problemi (GSP), gezgin satıcının listedeki bir şehirden başlayıp, diğer tüm şehirleri bir kez ziyaret ettikten sonra tekrardan başladığı şehre geri dönmesi için gereken en kısa yolu bulma problemidir.

Gezgin satıcı problemi, araç rotalama probleminin genelleştirilmiş halidir. Araç rotalama problemi, gezgin satıcı problemine birden fazla araç ve çeşitli kısıtların eklenmesi ile elde edilir. Örneğin; izlenecek rotada kullanılacak araçların yükleme kapasiteleri kısıtlı olduğunda GSP kapasite kısıtlı araç rotalama problemine dönüşmektedir.

Literatürde GSP için önerilen kesin çözüm yöntemlerinden en yaygın tercih edilenleri dal ve sınır ile dal ve kesme yöntemleri olduğu görülmektedir. Kesin çözüm yöntemleri belli bir boyuta kadar olan problemler için iyi sonuçlar vermektedir. GSP'de şehir sayısı arttıkça problemin çözüm uzayı faktöriyel olarak artmaktadır. Örneğin 5 şehirli bir GSP'nin çözüm uzayı $5!=120$, 10 şehirli bir GSP'nin çözüm uzayı ise $10!=3628800$ 'dir. Dolayısıyla şehir sayısı arttıkça çözüm karmaşıklığı da artmaktadır. Bu nedenle GSP NP-zor problem sınıfına girmektedir. NP-zor problemlerde, problemin çözüm karmaşıklığı arttıkça optimum sonuca kesin çözüm yöntemleri ile ulaşmak imkansız hale geldiğinden araştırmacılar daha çok sezgisel ve metasezgisel yöntemler kullanmaya yönelmişlerdir. Bu yöntemler arasında kurucu ve geliştirici sezgisel yöntemler, tabu arama, tavlama benzetimi, yapay sinir ağları, karınca kolonisi, genetik ve yapay arı kolonisi algoritmaları yer almaktadır.

2.2. Araç Rotalama Problemi (ARP)

2.2.1. Araç rotalama probleminin tanımı

Araç rotalama problemi (ARP), bir veya bir kaç depoya sahip ve her biri farklı bir yerleşimde bulunan belirli sayıda müşterileri veya şehirleri olan işletmenin, ürün dağıtımını yaparken seyahat mesafesini veya süresini en küçüklemeyi amaçlayan problemdir [1].

Araç Rotalama Problemi (ARP) ilk olarak 1959 yılında Dantzig ve Ramser tarafından literatüre kazandırılmıştır. Yazarlar bu çalışmalarında benzin istasyonlarına benzin dağıtımını problemi üzerinde durmuşlar ve problemin çözümü için ilk matematiksel programlama modelini kurmuşlardır. Daha sonra 1964 yılında Clarke ve Wright probleme sezgisel bir çözüm önermiş ve bu çalışmadan sonra ARP'ye ilgi daha da artarak büyümüştür. ARP şu ana kadar üzerinde en fazla yöntem geliştirilen optimizasyon problemlerinden biridir [2].

Tipik araç rotalama probleminde bir araç filosu için minimum maliyetli rota kümesi tasarlanmaktadır. Her rota bir depodan başlar ve talepleri bilinen müşteri kümesine hizmet sağladıktan sonra yine aynı depoda sonlanır. Her müşteri bir araca atanmalı ve araca atanan müşterilerin toplam talebi o aracın kapasitesini aşmamalıdır. Araç kapasitelerinin tüm müşteri taleplerinin en büyüğünden daha büyük olduğu varsayılarak her bir araca bir veya birden fazla müşterinin atanması sağlanır [3].

Araç rotalama problemlerinin gerçek hayattaki uygulamalarında çeşitli kısıtlarla karşılaşmaktadır. Bu kısıtları aşağıdaki gibi özetleyebiliriz [4]:

1) Araçlarla ilgili kısıtlar

- Araç kapasite kısıtı (ağırlık veya hacim olarak)
- Toplam zaman kısıtı
- Sürücünün çalışma saatleri için yasal sınırlamalar

2) Müşteriler ile ilgili kısıtlar

- Her bir müşterinin bir tür ürün talep etmesi veya belirli çeşitte ürün dağıtılması
- Dağıtımın yapılabilmesi için belirli zaman aralıklarının bulunması

3) Diğer kısıtlar

- Aynı araç ile birden fazla tur yapılması
- Bir turun bir günden uzun olması
- Birden fazla depo olması

Araç rotalama problemlerinde kısıtların yanında bir diğer önemli konuda amacın belirlenmesidir. Bir araç rotalama probleminin amaçlardan bazıları şunlardır [5]:

- Araç veya araçların kullanım zamanını en çok yapmak
- Araç veya araçların kapasite kullanım oranını en çok yapmak
- Yolculuk mesafesini en az yapmak
- Kullanılan araç sayısını en az yapmak

2.2.2. Araç rotalama probleminin temel bileşenleri

Araç rotalama problemlerinin temel bileşenleri; talep yapısı, taşınacak malzemenin cinsi, dağıtım/toplama noktaları ve araç filosudur.

Talep Yapısı: Araç rotalama problemlerinde talep statik veya dinamik olabilir. Statik talep durumunda talep önceden bilinir. Dinamik talep durumunda ise bazı noktalardaki talep bilinmekte bazıları ise araç rotasında devam ederken belirli olmaktadır.

Malzeme Cinsi: Araçlarla taşınan malzemeler çok çeşitli olabilir. Tehlikeli maddeler, gıda maddeleri, çöp toplama bütün bunlar basit paketler olarak adlandırılır ve probleme ilave bir karmaşıklık getirmezler. Diğer taraftan öğrenci servisleri; güvenlik, etkinlik gibi ilave bazı amaçlardan dolayı daha karmaşık bir yapıya

sahiptir. Tehlikeli maddeleri taşıyan araçların rotalarının belirlenmesinde ise coğrafi özellikler büyük önem kazanır.

Dağıtım/Toplama noktaları: Birçok araç rotalama probleminde, dağıtım noktaları müşterilerin buldukları yerler, toplama noktaları ise depolardır.

Depo genellikle aracın rotasına başladığı ve geri döndüğü noktadır. Depo sayısına göre problem, tek depolu ve çok depolu diye adlandırılabilir. Dağıtım noktaları sabit ve önceden biliniyorsa hangi noktalara, hangi araçların hizmet vereceği belirlenmelidir. Diğer durumda dağıtım noktaları potansiyel yerler arasından seçileceği için ilave bir yerleştirme kararı gerekir. Bazı araç rotalama problemlerinde dağıtım ve toplama noktaları aynıdır. Örneğin öğrenci servislerinde okul, gidişte dağıtım noktası, duraklar toplama noktası; öğrenciler evlerine dönerken ise okul depo, duraklar ise dağıtım noktalarıdır.

Filo: Bütün araç rotalama problemlerinde araçların kapasitesinin bilindiği ve çoğunlukla araçların homojen yani aynı kapasitede olduğu varsayılır. Filo heterojen ise filodaki araçların taşıma kapasiteleri farklıdır. Bu durum hangi araç tipinin, hangi rotaya hizmet vereceğinin belirlenmesini, yani ilave bir kararı gerektirir [6].

2.2.3. Araç rotalama probleminin matematiksel modeli

Klasik bir araç rotalama probleminin matematiksel modeli aşağıdaki gibidir.

Parametreler:

Q = araç kapasitesi,

N = müşteri veya durak sayısı,

$q_i = i (i > 0)$ müşterisinin talep miktarı,

d_{ij} = i müşterisi ile j müşterisi arasındaki uzaklık,

Değişkenler:

$$x_{ij} = \begin{cases} 1, & \text{eğer araç } i \text{ müşterisinden } j \text{ müşterisine gidiyorsa} \\ 0, & \text{aksi takdirde} \end{cases}$$

$$i \neq j, \quad i, j \in \{0, \dots, N\} \text{ ve } 0 \text{ ana depo}$$

Amaç Fonksiyonu:

$$\text{En az } Z = \sum_{i=0}^N \sum_{j=0, i \neq j}^N d_{ij} x_{ij} \quad (2.1)$$

Kısıtlar:

$$\sum_{i=1, i \neq j}^N x_{ij} = 1, \quad \forall j, \quad j \in \{1, \dots, N\} \quad (2.2)$$

$$\sum_{j=1, j \neq i}^N x_{ij} = 1, \quad \forall i, \quad i \in \{1, \dots, N\} \quad (2.3)$$

$$\sum_{i=0}^N \sum_{j=0, i \neq j}^N x_{ij} + x_{ji} \leq 1, \quad (2.4)$$

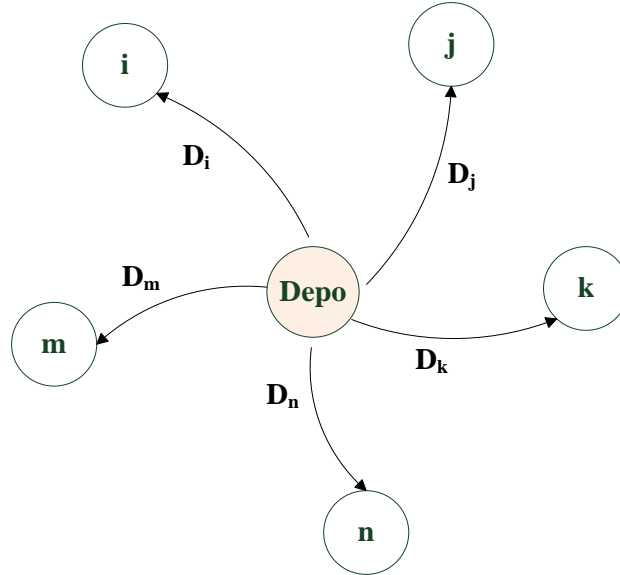
$$\sum_{i=1}^N q_i \sum_{j=0, i \neq j}^N x_{ij} \leq Q, \quad (2.5)$$

Amaç fonksiyonu (2.1) toplam kat edilen mesafenin en az yapılması gerektiğini ifade etmektedir. (2.2) ve (2.3) nolu kısıt, bir müşterinin mutlaka bir araç tarafından ziyaret edilmesi gerektiği ve bir müşteriyi ziyaret eden aracın aynı zamanda o müşteriden hareket etmek zorunda olduğunu belirtmektedir. (2.4) nolu kısıt, depoda başlamayan ve depoda bitmeyen turları elemekte kullanılmaktadır. (2.5) nolu kısıt ise araçlara yapılan yüklemelerin araç kapasite değeri Q 'yu geçmemesi gerektiğini belirtmektedir [7].

2.3. Araç Rotalama Probleminin Türleri

2.3.1. Kapasite kısıtlı ARP (KKARP)

KKARP, bir veya daha fazla depo bulunan bir işletmede, yükleme kapasiteleri kısıtlı araçların talepleri belli olan müşterilere servis yapmak için izlemesi gereken rotaların belirlenmesi problemidir [8]. Şekil 2.1'de görüleceği gibi KKARP'de her müşteriye sadece bir araç uğrar. Klasik ARP ile bir tutulmasının yanında tek farkı rota üzerindeki müşterilerin tüm taleplerinin toplamı araç kapasitesinden fazla olamamasıdır. ARP ailesinin en iyi bilinen ve üzerinde en çok çalışılan üyesidir.



Şekil 2.1. Kapasite kısıtlı ARP örneği

Şekil 2.1'de D_j , j müşterisinin talebini göstermektedir [9].

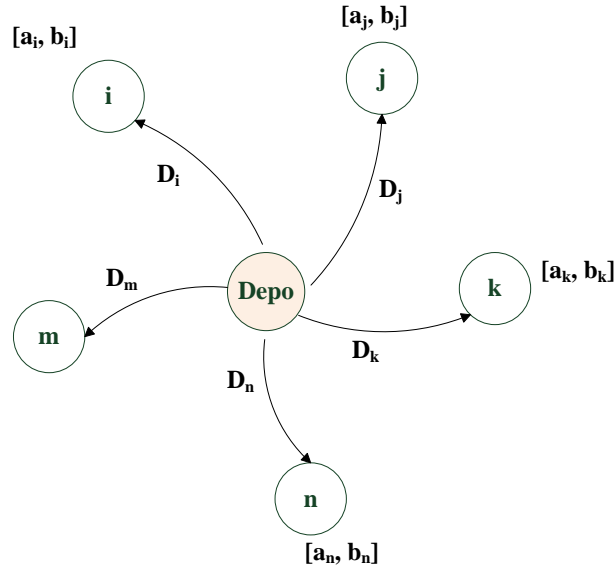
KKARP'nin farklı versiyonları bulunmaktadır. Örneğin bazı problemlerde her aracın bir sabit çalıştırma maliyeti bulunup bu değer amaç fonksiyonuna eklenmektedir. Bu durumda sabit maliyet unsurundan dolayı amaç fonksiyonunda kullanılan araç sayısı en az yapılmaya çalışılmakta ve çözümde sonuç olarak bazı araçların rotalaması yapılmamaktadır. Başka bir KKARP çeşidi ise problemde farklı tipte ve dolayısıyla yükleme kapasiteleri farklı olan araçların söz konusu olmasıdır [10].

2.3.2. Zaman pencereleli ARP (ZPARP)

Zaman Pencereleli Araç Rotalama Problemi (ZPARP), zaman penceresine uyularak bir depodan dađınık müşteriler kümesine en düşük maliyetli rotaların belirlenmesi problemi olarak tanımlanabilir. Rotalar, verilen bir zaman aralığında her müşteriye bir kere ve bir araç tarafından ziyaret edilecek biçimde tasarlanmalıdır. Ayrıca rota depoda başlayıp depoda son bulmalıdır [11].

Bu problem çeşitlerinde her müşteriye belirli bir $[a_i, b_i]$ zaman aralığında ulaşılması gerekir. $[a_i, b_i]$ ifadesine zaman penceresi denilmektedir. Problemden her araç depodan veya bir diđer müşteriden 0 zamanında ayrılacak ve bir i müşterisine uğradığında belirli bir servis süresi kadar müşteriye hizmet edecektir. Aracın talep noktasına gelme zamanı, belirli bir zaman aralığı içinde olmalıdır. Eğer araç müşteriye daha erken ulaşmış ise a_i başlangıç zamanına kadar beklemesi gerekmektedir. ZPARP'de amaç öncelikle müşterinin istediđi zaman aralığında hizmet vermektir. Sonrasında ise rotanın optimize edilmesi yoluyla çözüme ulaşılır [10].

Şekil 2.2'de zaman pencereleli araç rotalama probleminin örneđi verilmiştir.



Şekil 2.2. Zaman pencereleli ARP örneđi [9]

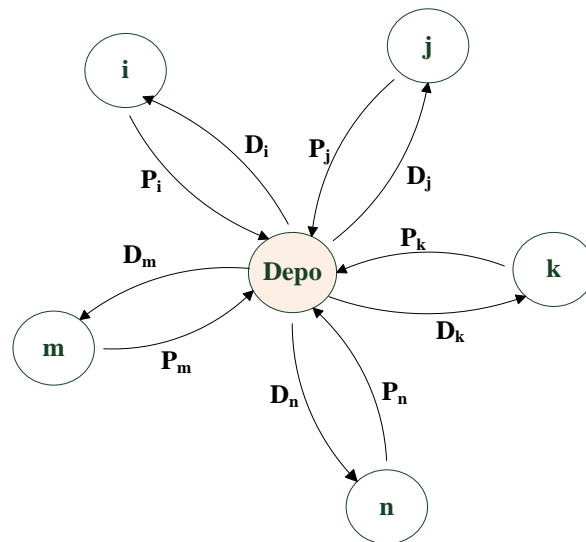
2.3.3. Dağıtım ve toplamalı ARP (DTARP)

Bu problem toplama ve dağıtım işlemlerini birlikte kapsayan bir araç rotalama problemi (DTARP) türüdür. Amaç; araç akışını ve toplam dolaşım süresini en aza indirmek, bunu yaparken de araçların malları müşterilere dağıtmak için ve müşterilerden topladığı malları depoya getirmek için yeterli kapasiteye sahip olmasına dikkat etmektir [12].

Araçlar bir veya birden fazla depodan başlayarak müşterilere uğrarlar ve tekrar başladığı depoya dönerler. Her müşteriye sadece bir araç uğrar. Aracın rotasındaki dağıtılacak ve toplanacak müşteri talepleri toplamı araç kapasitesinden fazla olamaz. Dağıtım ve toplama işlemi her müşteride de gerçekleştirilir [9].

DTARP'nde, i müşterisindeki dağıtım talebinin miktarını gösteren d_i ve i müşterisindeki toplama talebinin miktarını gösteren p_i ile her bir i müşteri ilişkilendirilmiştir. d_i ve p_i aynı türden mallardır. Bazen, her bir i müşteri için, dağıtım ve toplama talepleri arasındaki net farkı gösteren bir $d_i = d_i - p_i$ talep miktarı kullanılır. Her bir i müşteri için O_i dağıtım talebinin başlangıcı olan düğümü ve D_i toplama talebinin hedefi (varış noktası) olan düğümü ifade eder [13].

Şekil 2.3'te dağıtım ve toplamalı araç rotalama problemine örnek verilmiştir.



Şekil 2.3. Dağıtım ve toplamalı ARP örneği

Dağıtım ve toplamalı araç rotalama problemleri üçe ayrılır. Bunlar; önce dağıtım sonra topla araç rotalama problemleri, karışık dağıtım ve toplamalı araç rotalama problemleri ve eş zamanlı dağıtım ve toplamalı araç rotalama problemleridir.

Önce dağıtım sonra topla araç rotalama problemlerinde, depodan müşterilere dağıtılacak malzemelerin tamamı dağıtıldıktan sonra müşterilerden depoya geri getirilecek malzemelerin toplanma işlemi yapılır. Müşterilere birden fazla kez uğranır.

Karışık dağıtım ve toplamalı araç rotalama problemlerinde, araç kapasiteleri dikkate alınarak dağıtım ve toplama işlemi karışık olarak yapılır. Bu problemde de müşterilere birden fazla kez uğranabilir.

Eş zamanlı dağıtım ve toplamalı araç rotalama problemlerinde, araç kapasiteleri dikkate alınarak depodan müşteriye gönderilen malzeme bırakılmakta ve aynı anda müşteriden depoya geri gönderilecek malzeme alınmaktadır. Bu problemde, müşterilere sadece bir defa uğranır [9].

2.3.4. Mesafe kısıtlı ARP (MKARP)

Mesafe kısıtlı araç rotalama problemlerinde rotalara atanan her aracın gidebileceği en çok mesafe kısıtı bulunmaktadır. Mesafe kısıtları tek başlarına kullanılabilir gibi araçların kapasite kısıtları ile birlikte kullanılabilirler [14]. Mesafe kısıtı Denklem 2.6'daki gibi ifade edilmektedir. Bu denklemde bir aracın gidebileceği en uzun mesafe L ile gösterilmektedir.

$$i, j \in \{0, \dots, N\} \text{ için } \sum_{i=0}^N \sum_{j=0}^N d_{ij} \sum_{j=0, j=i}^N X_{ij} \leq L \quad (2.6)$$

Farklı çeşitlerde mesafe kısıtlı araç rotalama problemleri bulunabilmektedir. Örneğin farklı tipteki araçlar için farklı mesafe kısıtı söz konusu olabilir. Bunun yanında mesafe kısıtı yerine mesafeyle orantılı seyir süresi sınırlaması da konulabilir. Böyle

durumlarda araç her müşteriye uğradığında s_i servis süresi kadar bekleyecektir. Bu tip problemlerde mesafe kısıtı Denklem 2.7'deki gibi değişecektir.

$$i, j \in \{0, \dots, N\} \text{ için } \sum_{i=0}^N \sum_{j=0}^N d_{ij} \sum_{j=0, j=i}^N pX_{ij} + \sum_{i=0}^N \sum_{j=0}^N \frac{s_i + s_j}{2} \sum_{j=0, j=i}^N X_{ij} \leq T \quad (2.7)$$

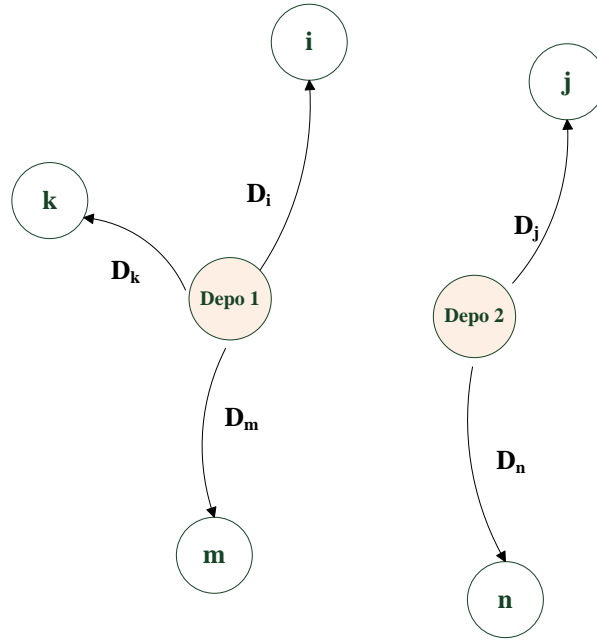
Burada p parametresi aracın seyir süresini kat edilen mesafe yoluyla hesaplanmaktadır. Servis sürelerinin (s_i) ikiye bölünerek hesaplanmasının sebebi ise aracın rotasında bulunan yolları sıralarken, rota içerisindeki bir müşterinin bir yolun başlangıcında diğer yolun ise sonunda yer alması ve bu sebeple denklemdeki toplam ifadesinde aynı müşteri ile iki defa karşılaşılmasıdır [10].

2.3.5. Çok depolu ARP (ÇDARP)

İşletmelerin birden çok depoya ve her deponun kendi müşteri bölgesine sahip olduğu durumlarda çok adet tek depolu ARP problemi ile karşı karşıya kalınmaktadır. Böyle durumlarda her müşteriye kendi bölgesine uygun depodan çıkan araç tarafından hizmet verilmektedir. Standart ÇDARP'de her araç rotası aynı depodan başlar ve aynı depoda sonlanır [15].

Depolarda bulunan araçlar, her rotadaki toplam talebi karşılamak zorundadır. Araçları rotalarken toplam uzaklığın en aza indirilmesi, taleplerin karşılanması, her müşterinin yalnızca bir kez ziyaret edilmesi ve araçların depolara geri dönmesi gibi ölçütler dikkate alınır. Çok depolu araç rotalama probleminde araçlar birden fazla depoda konumlanabilmektedir. Bu gibi durumlarda da toplam maliyeti minimize edecek ve toplam talepler karşılanacak şekilde araç sayısı minimize edilmeye çalışılır. Tüm araçlar aynıdır. Taleplerin bilindiği kabul edilmektedir. Problem dağıtım ve toplama operasyonlarına uyarlanabilir [5].

Şekil 2.4'te çok depolu araç rotalama problemine örnek verilmiştir.



Şekil 2.4. Çok depolu ARP örneği

Bu çalışmada ele alınan problem, kapasite kısıtlı araç rotalama problemidir. Araçların hepsi maksimum 40 palet kapasitelidir ve araçlara 32 ile 40 palet arasında yükleme yapılmaktadır. Bu ele alınan kapasite kısıtlı ARP, ürünlerin çeşidi ve dağıtım yapılacağı günlere göre 4 adet alt probleme ayrılmaktadır. Bu alt problemler; taze ürünün yoğun günde dağıtımı, kuru ürünün yoğun günde dağıtımı, taze ürünün sakin günde dağıtımı ve kuru ürünün sakin günde dağıtımı alt problemleridir. Bu alt problemlerden taze ürünün yoğun günde dağıtımı ve kuru ürünün yoğun günde dağıtımı alt problemlerinde 78 adet mağaza, taze ürünün sakin günde dağıtımı ve kuru ürünün sakin günde dağıtımı alt problemlerinde ise 66 adet mağaza mevcuttur. Taze ürünün yoğun günde dağıtımı alt probleminin talep miktarı 438, Kuru ürünün yoğun günde dağıtımı alt probleminin talep miktarı 724, taze ürünün sakin günde dağıtımı alt probleminin talep miktarı 402 ve kuru ürünün sakin günde dağıtımı alt probleminin talep miktarı ise 652 palettir.

2.4. Araç Rotalama Problemlerinin Çözüm Yöntemleri

Araç rotalama problemlerinin çözümü için yöneylem araştırmacıları tarafından çeşitli yöntemler ortaya koyulmuştur. Ancak araç rotalama problemi NP-zor problem sınıfından olduğu için çok değişkenli problemlerde çözüme kesin yöntemlerle

ulaşmak oldukça zor ve hatta bazen imkânsız olmaktadır. Problemin kısıtları arttıkça problem daha da karmaşık hale gelmekte ve optimum sonuca ulaşmak daha da zorlaşmaktadır. Bu nedenle araç rotalama problemlerinin çözümünde genellikle optimuma yakın değerler veren ve kısa sürede çözümü sağlayan yöntemler kullanılmaktadır [16].

Araç rotalama problemlerinin çözüm yöntemleri; kesin yöntemler, sezgisel yöntemler ve metasezgisel yöntemler olmak üzere üçe ayrılır.

2.4.1. Kesin yöntemler

Kesin yöntemler, problemin en iyi çözümünü bulan yöntemlerdir. Ancak kesin yöntemler, problemin kısıtları arttıkça çözüm süresi artar ve kısa sürelerde iyi çözümler bulamamaktadır. ARP'nin çözümü için dal ve sınır yöntemi, kesme düzlemi yöntemi, dal ve kesme yöntemi ve dinamik programlama yöntemleri kullanılmaktadır [17].

2.4.1.1. Dal ve sınır yöntemi

Dal ve sınır yöntemi, tamsayı düğümlerinde alt ve üst sınırlar kullanılarak çözüm uzayını tarayan bir yöntemdir. Herhangi bir aşamada bulunan düğümün alt ve üst sınırına bakılır ve o düğümden dallanma yapılıp yapılamayacağına karar verilir. Problemler, dallanmalar yaparak alt problemlere ayrılırlar. Üst sınır, çözümde o ana kadar elde edilmiş en iyi çözümdür. Eğer düğümden elde edilen değer üst sınırdan daha iyi ise üst sınır değeri güncellenir. Üst sınır güncellendiğinde, alt sınırı üst sınırına eşit veya daha büyük olan alt problemler dallanma kümesinden iptal edilir ve dallanma yapılmaz. Diğer alt problemler için dallanma yapılarak elde edilen yeni düğümler dallanma kümesine eklenir [2].

Bunu bir denklem üzerinden anlatacak olursak; S bizim çözüm uzayımız olsun. Dal ve sınır yöntemini kullanarak öncelikle tüm S çözüm uzayını değerlendirmiş oluruz. İlerleme ve sınırlama aşamalarında problemi esnek hale getiririz. Böyle yaparak S olası çözüm uzayında bulunmayan çözümleri de kabul etmiş oluruz. Bu esnek

değerleri çözmek, optimum çözüme daha düşük bir sınır değeri getirir. Bu değer S'nin bir elemanıysa veya s' ile eşit maliyete sahipse problem sona ermiştir, başka bir ifadeyle yeni çözüm olan s' optimaldir.

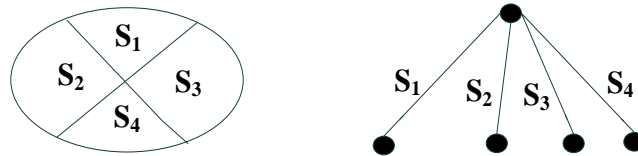
Aksi durumda S uzayında Denklem 2.8'den yola çıkarak S_1, \dots, S_n olarak n adet alt küme tanımlarız [18].

$$\bigcup_{i=1}^n S_i = S \quad (2.8)$$

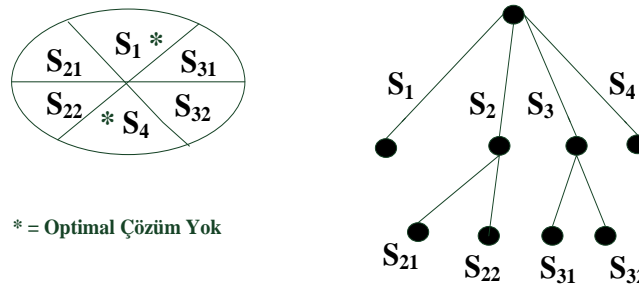
Şekil 2.5, Şekil 2.6 ve Şekil 2.7'de çözüm uzayı, dallanmalar ve alt çözümler gösterilmiştir.



Şekil 2.5. Çözüm uzayı ve araştırma ağacındaki gösterimi [19]



Şekil 2.6. Çözüm uzayının alt kümelere ayrılması ve araştırma ağacında gösterimi [19]



Şekil 2.7. Çözüm alt uzaylarının değerlendirilmesi ve yeni alt çözümlere ulaşılması [19]

Bu alt kümelerin her biri bir alt problemdir ve bunlar aday alt problemler listesine eklenir. Bu durum dallandırma olarak adlandırılır. Çözümüne devam etmek için aday alt problemlerden birini seçeriz ve ilerletiriz. Olası dört sonuç vardır. Bunlar:

- s' 'den daha iyi bir sonuç bulursak bunu s' ile değiştiririz ve devam ederiz.
- Alt problemin hiçbir sonuca ulaşamadığı ve alt problemi iptal etmemiz gereken durumlarla da karşılaşabiliriz.
- Aksi halde alt problemimizin alt sınırını üst global sınır ile karşılaştırırız. Mevcut üst sınırdan daha büyük ya da eşitse yine alt problemi iptal edebiliriz.
- Son olarak eğer alt problemi eleyemiyorsak, daha fazla dallanmaya ve bu alt probleme yeni alt problemler eklemeye zorlanırız. Alt problemlerin ortaya çıkmadığı bir noktaya gelene kadar yani optimum sonuca ulaşınca kadar algoritmaya devam ederiz [20].

2.4.1.2. Kesme düzlemi yöntemi

Doğrusal programlama problemlerinin tam sayılı çözümlerini sağlayan bu yöntem 1959 yılında R.E. Gomory tarafından geliştirilmiştir. Kesme düzlemi yöntemi de dal ve sınır yöntemindeki gibi sürekli doğrusal programlama probleminin optimum çözümüyle başlar. Ancak bu yöntemde kesme adı verilen özel kısıtlar ardı ardına oluşturularak çözüm uzayının düzenlenmesine gidilir [21]. Bu yöntemde takip edilecek aşamalar şunlardır:

- İlk aşama, eğer gerekli ise, orijinal sınırlamalar tam sayılı hale getirilir. Bunun anlamı katsayılar tam olsun diye, tüm sınırların değiştirilmesidir.
- Kesme düzlemi probleminin optimal çözüm tablosu bulunur. Eğer optimal çözüm değerleri tamsayı ise, çözüm elde edilmiştir. Eğer değerlerden en az biri tam sayı değilse bir sonraki aşamaya geçilir.
- Bu aşamada kesme bulunur. Bu amaçla optimal çözüm tablosundan tamsayı olmayan değişkenlerinden biri seçilir ve yeni bir kısıt elde edilir [22].

2.4.1.3. Dal ve kesme yöntemi

Dal ve kesme yöntemi doğrusal programlama problemlerinin tam sayılı çözümleri için oldukça etkili bir yöntemdir. Bu yöntem kesme düzlemi ve dal ve sınır yöntemlerinin bir birleşimidir. Dal ve kesme yöntemi de dal ve sınır ve kesme düzlemi yöntemleri gibi doğrusal programlama probleminin optimum çözümüyle başlar [23].

Genel bir tamsayı programlama problemini sadece kesme düzlemi yaklaşımı ile verimli olarak çözebilmek mümkün değildir, alternatif optimum çözümleri bulmak için dallandırma yapmak da gereklidir. Dal ve sınır yöntemi, kesme düzlemi yönteminin uygulanması ile oldukça hızlandırılabilir. Dallandırma yapılmadan kesme eklenebileceği gibi ağacın her düğümünün çözümünde de kesmeler kullanılabilir [24].

Bu yöntem tam sayı kısıtı olmayan doğrusal programı düzenli simpleks algoritma kullanarak çözer. Optimum bir çözüm elde edildiğinde tam sayı olarak tanımlanan değişken bu çözümde tam sayı değilse kesme düzlemi yöntemi kullanılır. Eğer böyle bir eşitsizlik bulunursa, bu doğrusal programa eklenir. Bu süreç tam sayılı çözüm bulunana kadar tekrar eder [25].

2.4.1.4. Dinamik programlama

Dinamik programlama, bir dizi karar verme işlemini optimize eden matematiksel işlemler bütünüdür. Dinamik programlamada n değişkenli bir problemin optimum çözümü, problemi n aşamaya ayrıştırarak ve her aşamada tek değişkenli bir alt problemi çözümlenir. Bunun hesaplanmasının avantajı, n değişkenli alt problemler yerine tek değişkenli alt problemleri optimum kılmasıdır. Dinamik programlama genellikle geriye doğru uygulanır. Bir alt problemin optimum çözümü bir sonraki alt problemin girdisidir. Son alt problem çözüldüğünde problemin tamamının optimum çözümüne ulaşılmış olur [26].

2.4.2. Klasik sezgisel yöntemler

Araç rotalama probleminin çözüm yöntemlerinden klasik sezgisel yöntemler üçe ayrılır. Bunlar; yapısal sezgisel algoritmalar, iyileştirme sezgisel algoritmalar ve iki aşamalı sezgisel yöntemlerdir.

- Yapısal sezgisel (tur kurucu) algoritmalar: Olası çözümü bulmaya çalışırken aşamalı olarak ilerlerler. Bu esnada çözüm maliyetlerini gözetmelerine rağmen kendi başlarına iyileştirme içermezler.
- İyileştirmeli sezgisel (tur geliştirici) algoritmalar: İyileştirme sezgiselleri, araç rotalarında bir sıra tepe noktası veya kenar değişimleri gerçekleştirerek herhangi bir olası çözümü iyileştirme eğilimindedir.
- İki aşamalı sezgisel yöntemler: Bu yöntemde problem köşelerin olası rotalar içine kümelenmesi ve gerçek rota yapılması olmak üzere iki bileşene ayrıştırılır [12].

2.4.2.1. Yapısal sezgisel (tur kurucu) algoritmalar

Bu yöntemlerde mümkün olmayan atamalarla çözüme başlanıp her defasında iki düğüm arasına bir dal ekleyerek mümkün olan çözüme ulaşılır. Eklenecek dal maliyet tasarrufuna göre seçilir ve dal eklenirken araç kapasite kısıtına uyup uymadığı kontrol edilir. Yapısal sezgisel yöntemler arasından en çok tercih edileni Clarke ve Wright'ın, Dantzig ve Ramser'in çalışmasından esinlenerek geliştirdikleri tasarruf algoritmasıdır [6]. Bunun dışında yerleştirme, en yakın komşu ve en kısa yol yöntemleri de yapısal sezgisel yöntemlerdendir [27].

Clarke ve Wright tasarruf algoritması: ARP'nin çözümü için en çok bilinen yöntemlerden biri Clarke ve Wright'ın 1964'de geliştirdikleri tasarruf algoritmasıdır. Bu yöntem ile rotaların bulunmasına ek olarak kaç adet aracın gerekeceği de bulunmaktadır. Bu yüzden tasarruf algoritması araç sayısının değişken olduğu problemlerde uygulanmaktadır. Tasarruf algoritması paralel ve sıralı olmak üzere iki çeşittir. Bu yöntemin adımları aşağıda belirtilmiştir [28]:

1. Adım: $s_{ij} = d_{i0} + d_{0j} - d_{ij}$ formülü ile her müşteri çifti için tasarruflar hesaplanır ve s_{ij} değerleri büyükten küçüğe doğru sıralanır. Bu formüldeki d_{i0} ; i müşterisi ile depo arasındaki uzaklık, d_{0j} ise depo ile j müşterisi arasındaki uzaklıktır.

2. Adım: Paralel tasarruf algoritmasında s_{ij} değerlerine göre büyükten küçüğe sıralanan müşteri ikilileri şu kurala göre birleştirilir: Eğer sıralamada xy ve yz müşteri çiftleri art arda geliyorsa araç kapasite kısıtı göz önüne alınarak 0-x-y-z-0 güzergâhında bir rota oluşturulur. Aksi takdirde xy ve tz formatında müşteri çifti art arda geliyorsa, bu müşteri çiftleri için ayrı ayrı rota oluşturulur. Sıralamada ele alınan müşteri çiftleri daha önce oluşturulmuş rotalar ile mukayese edilir, eğer uygunsuz rotaya dahil edilir. Sıralı tasarruf algoritmasında ise ele alınan xy ve tz formatındaki müşteri çiftleri kapasite kısıtı dikkate alınıp birleştirilerek 0-x-y-t-z-0 güzergâhında rota oluşturulur. Aynı rota için, sıralamada ele alınan müşteri çiftleri mukayese edilir, eğer kapasite kısıtına uygunsuz rotaya dahil edilir. Her iki yöntemde de kapasite kısıtına bakılarak bir rota tamamıyla oluşuncaya kadar müşteri çiftleri taranmaktadır. Müşteri çiftleri rotalara yerleştirilirken de toplam maliyetin minimum olmasına dikkat edilir.

Paralel tasarruf algoritması, Sıralı tasarruf algoritmasına göre daha iyi sonuçlar vermektedir. Bunun sebebi sıralı tasarruf algoritmasında sadece bir rota ele alınıp bu rotayı oluşturmak için s_{ij} tasarruf sıralamasının dikkate alınması, Paralel tasarruf algoritmasında ise s_{ij} sıralamasına bakılarak müşteri çiftinin birden fazla rota ile uyumluluğu kontrol edilmesi ve bu sebeple daha fazla alternatifin gözden geçirilebilmesidir [29].

Yerleştirme yöntemi: Yerleştirme yönteminde yerleştirme maliyetleri kullanılarak müşterilere araçlar atanmakta ve rotalar oluşturulmaktadır. Araç rotalama problemleri için kabul görmüş birkaç yerleştirme yöntemi olmasına rağmen çalışma sonuçları bu algoritmaların diğer yöntemlerle kıyaslanabilecek seviyede olmadığını göstermektedir [5].

En yakın komşu yöntemi: En yakın komşu yöntemi noktaları teker teker ekleyen bir yöntemdir. Her iterasyonda son eklenen noktaya en yakın nokta seçilir [29]. Algoritma bütün noktalar ziyaret edilene kadar devam eder ve bütün noktalar ziyaret edildikten sonra tekrar başlangıç noktasına döner. Bu yöntemin adımları şu şekildedir:

1. Adım: Herhangi bir noktadan başla.
2. Adım: Rotada olmayan ve en son eklenen noktanın en yakınındaki noktaya git.
3. Adım: 2. adımı bütün noktalar rotaya dahil olana kadar tekrarla. Daha sonra rotanın ilk ve son noktalarını birleştir.

Bu adımlar gezgin satıcı problemi içindir. Araç rotalama probleminin çözümü için uygulandığında başlangıç şehri depodur. Araç rota oluşturmaya depodan başlar ve tüm müşteriler ziyaret edildikten sonra tekrar depoya döner [30].

En kısa yol yöntemi: En kısa yol yöntemi tek deponun bulunduğu şebekelerde rotalama işlemi yapmak için kullanılmaktadır. Bu yöntemde bağlantılar ve düğümlerle temsil edilen bir şebeke bulunmaktadır. Şebeke içerisinde bulunan düğümler bağlantılar vasıtasıyla birbirine birleştirilir. Bu bağlantılar maliyet, zaman, mesafe ya da bunların bir karışımı olarak ifade edilebilir [16]. En kısa yol yönteminin uygulama adımları şöyledir:

1. Adım: Merkezi birimden rotalamaya başlanır. Merkezi noktaya en yakın müşteri ilk araca atanır.
2. Adım: Rotaya atanan müşteriye en yakın ve daha önce rotaya eklenmemiş noktalar incelenir. Eğer müşteriye en yakın iki nokta varsa her biri için süreç oluşturularak ayrı çözüm dalları oluşturulur.
3. Adım: Eğer oluşturulan rotada müşteri merkezi birimden sonra gelmiyorsa ve müşteriye en yakın başka müşteri ile merkezi birim aynı mesafede ise, süreç yine

ikiye ayrılarak yeni çözümler oluşturulur. Rotalamada araç kapasite kısıtı sağlanıyorsa ilk önce müşteri başka müşteriye bağlanır. Daha sonra çözüm ağacında yeni bir dal oluşturularak müşteri direkt olarak merkezi birime bağlanır.

4. Adım: Çözümler ayrı ayrı hesaplanarak en uygun çözüm seçilir [20].

2.4.2.2. İyileştirmeli sezgisel (tur geliştirici) algoritmalar

İyileştirmeli sezgisel algoritmalarda mevcut çözümde yer alan araç rotaları arasında müşteri veya yol değişimi yapılarak uygun çözüm iyileştirilmeye çalışılır. Bu algoritmalar isteğe göre seçilmiş bir çözümle başlar ve başka iyileştirmenin mümkün olmadığı bir yerel minimumda biter [31].

Tek rota iyileştirmeli sezgisel algoritmalar: ARP içinde yer alan tek bir rotanın iyileştirilmesi amacıyla kullanılan ve 1965 yılında Lin tarafından geliştirilen λ -opt metodu literatürde en fazla kullanılan yöntemlerden biridir. Bu yöntemde λ adet yol bir rotadan çıkartılarak mümkün olan tüm permütasyonlarda rotanın çeşitli noktalarına eklenmektedir. Mevcut çözümden daha iyi bir çözüm bulunması halinde yöntem, bu yeni rotayı çıktı olarak vermektedir. λ -opt yöntemi ile genellikle birbiri ile kesişmeyen doğruların bulunduğu bir rota oluşturulmaya çalışılır. λ -opt yönteminin bazı versiyonlarında başlangıç rotasından daha iyi sonucun elde edildiği durumda çözüm saklanarak algoritmadan çıkılmayıp, arama süreci devam ettirilmektedir. Genel olarak bir ARP'de λ -opt algoritmasının $O(n^\lambda)$ sürede sonuç bulması beklenmektedir. λ -opt yöntemi üzerinde değişiklikler yapılarak örneğin art arda gelen belli sayıda nokta yer değiştirilerek bu süre kısaltılabilmektedir [28].

Çok rota iyileştirmeli sezgisel algoritmalar: Çok rota iyileştirmeli sezgisel algoritmalarda bir ARP içinde yer alan rotalar arasında nokta alış verişi yaptırılmaktadır. Thompson ve Psaraftis'in 1993 yılında geliştirdikleri b-devirsel k-transfer yönteminde dairesel permütasyon ile sıralanan b rota ele alınıp her rotadan k adet müşteri diğer rotaya atanarak yeni çözümler elde edilmektedir. Breedam çok rota iyileştirmeli sezgisel algoritmaları dört grupta toplamıştır. Bunlar; yol değişimi yöntemi, nokta değişimi yöntemi, nokta atama yöntemi ve karma yöntemlerdir [32].

Bu dört yöntem Van Breedam tarafından test edilmiş ve nokta değişimi yönteminin hesaplama süresi ve çözüm kalitesi bakımından en iyi çözümler ürettiği gözlemlenmiştir [2].

Thomson ve Psaraftis sezgisel algoritması: Araç rotalama problemlerine uygulandıklarında periyodik transferlerin asıl amacı, küçük miktarlardaki taleplerin taşınmasının toplam maliyetini iyileştirmektir. Bu yüzden periyodik transfer algoritmalarında kümele ve rotala şeklinde iki aşamalı karar yapısı kullanılmaktadır. Periyodik transferler aşağıdaki formülle çözümlenebilmektedir:

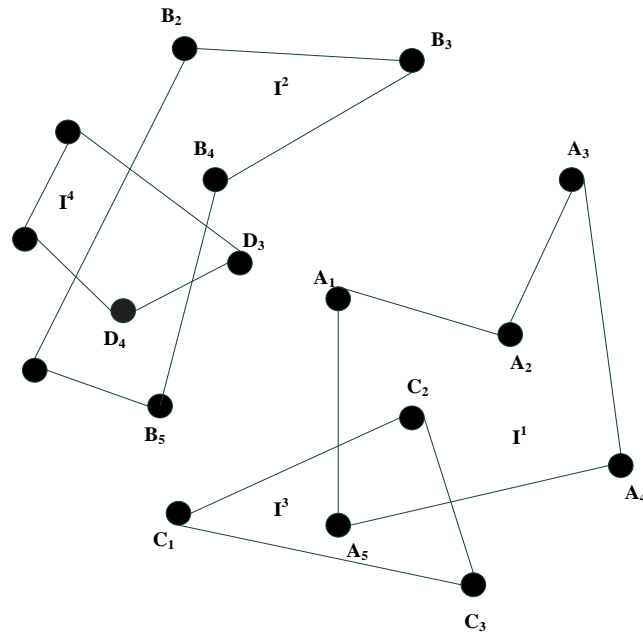
$[I^1, \dots, I^m]$ aralığındaki değerler kümesi araç rotalama problemi için uygun çözümler oluşturan rotalar olsun. P ise $[1, \dots, m]$ aralığının a_1 kümesinin periyodik permütasyonu olsun. Örnek olarak $p=(2\ 5\ 3)$ ise $p(2)=5$, $p(5)=3$ ve $p(3)=2$ dir. Her bir j için taleplerin I^j 'den $I^{p(j)}$ 'ye olan eş zamanlı taşınmalarına periyodik transfer denilmektedir. Thomson ve Psaraftis (1993) k 'nın sabit bir tamsayı olduğu durum ve her bir j için, k kadar talebin I^j rotasından $I^{p(j)}$ rotasına eş zamanlı taşınmalarını içeren ve periyodik k -transferleri kavramını temel alan bir yöntem önermektedirler. Thomson ve Psaraftis döngüsel permütasyonun bir sonraki rotasına, değiştirilen her rotadan k müşteri ve b rotanın dairesel permütasyonu içindeki genel bir b -periyodik, k -transfer düzeni tanımlamışlardır. Şekil 2.8 ve Şekil 2.9'da 3-periyot, 2-transfer içeren, I^m değerleri ile p katsayısı aşağıdaki gibi olan müşteriler ve uzaklıkların bu algoritma ile çözümü gösterilmektedir.

$$I^1 = |A_1, A_2, A_3, A_4, A_5|, I^2 = |B_1, B_2, B_3, B_4, B_5|, I^3 = |C_1, C_2, C_3|$$

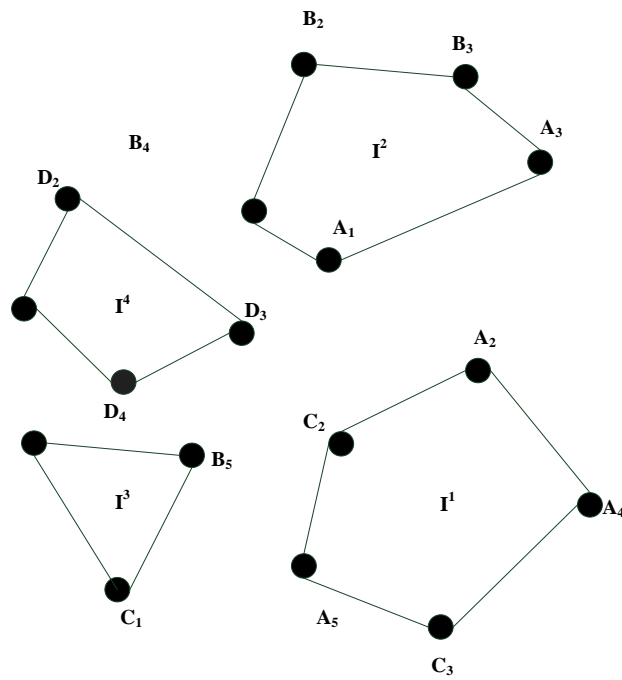
$$\text{ve } I^4 = |D_1, D_2, D_3, D_4|$$

$$p = (1\ 2\ 3)$$

Periyodik transfer eş zamanlı olarak I^1 'den I^2 'ye $|A_1, A_3|$ kullanılarak, I^2 'den I^3 'e $|B_1, B_5|$ kullanılarak, I^3 'den I^1 'e $|C_2, C_3|$ kullanılarak ve I^4 aynı kalarak oluşturulmaktadır [33].



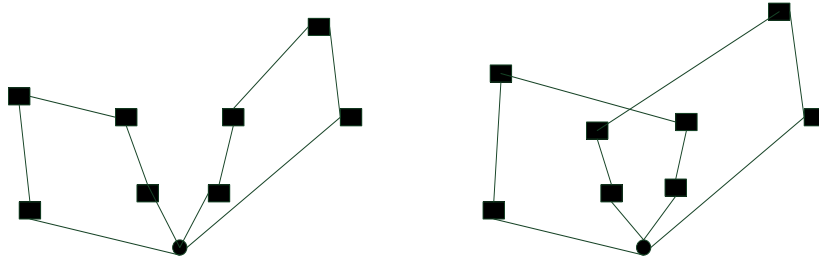
Şekil 2.8. Periyodik transfer uygulamadan önceki rotalar [33]



Şekil 2.9. Periyodik transfer uyguladıktan sonraki rotalar [33]

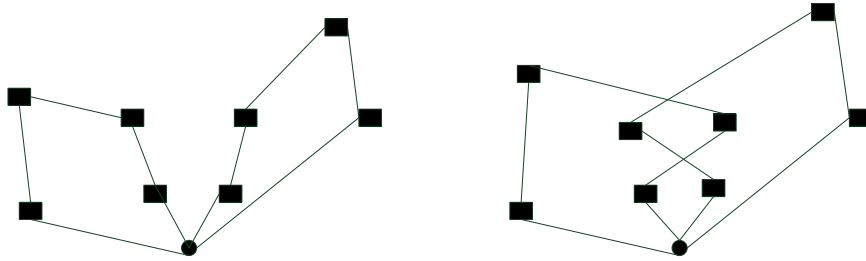
Van Breedam algoritması: Bu algoritmada 4 adet işlem vardır. Bunlar; zincir çaprazlama, zincir değiş-tokuşu, zincir yer değiştirme ve zincir bağdaştırma.

- Zincir Çaprazlama: İki farklı rotadaki iki müşterinin çaprazlanmasıyla oluşan yeni zincirlerdir. Zincir çaprazlama örneği Şekil 2.10'da gösterilmiştir.



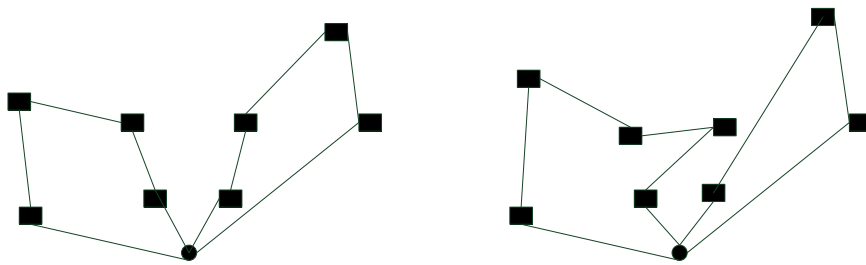
Şekil 2.10. Zincir çaprazlama örneği [32]

- Zincir Değiş-tokuşu: İki farklı rotadaki müşteri noktalarının değiştirilmesiyle komşuluk türetme işlemidir. Zincir değiş-tokuşu örneği Şekil 2.11'de gösterilmiştir.



Şekil 2.11. Zincir değiş-tokuşu örneği [32]

- Zincir Yer Değiştirme: Zincire ait müşteri noktasının bir rotadan diğerine nakil işlemidir. Zincir yer değiştirme örneği Şekil 2.12'de gösterilmiştir.



Şekil 2.12. Zincir yer değiştirme örneği [32]

- Zincir Bağdaştırma: Uygulama esnasında zincir deęiş-tokuşu veya zincir yer deęiştirmeden en iyisinin seçilmesi işlemidir [32].

Kinderwater ve Savelsbergh algoritması: Kinderwater ve Savelsbergh algoritmasında turlar izole edilmiş olarak düşünülmemektedir. Bu nedenle yollar ve müşteriler farklı turlar arasında deęiştirilebilir. Bu deęişikliği yapan operasyonlar şunlardır:

- Müşteri kaydırma: Bir rotada bulunan müşterilerin dięer rotaya kaydırılmasıdır.
- Çaprazlama: İki rotanın bir rotada kesiştirilmesidir.
- Müşteri Deęiş-Tokuşu: İki farklı rotada bulunan iki müşterinin birbirleri ile yer deęiştirmesidir [18].

2.4.2.3. İki aşamalı sezgisel yöntemler

Bu metotların birinci aşamasında, müşteriler kapasite kısıtına göre uygun gruplara ayrılmaktadır. İkinci aşamasında ise oluşturulan grupların içinde rotalama işlemi yapılmaktadır. İki aşamalı sezgisel algoritmalar hangi adımın önce ve sonra yapılacağına göre önce grupla sonra rotala ve önce rotala sonra grupla olmak üzere ikiye ayrılmaktadır. Önce grupla sonra rotala yöntemleri; Gillet ve Miller'in 1974 yılında geliştirdikleri süpürme algoritması, Fisher ve Jaikumar'ın 1981 yılında geliştirdikleri algoritma, Bramel ve Simichi-Levi algoritması, taç yaprağı algoritması, Taillard algoritması ve budanmış dal sınır algoritmasıdır [6].

- Önce grupla sonra rotala yöntemleri

Süpürme algoritması: 1974 yılında Gillet ve Miller tarafından geliştirilmiştir. Ancak algoritmanın ilk prensiplerinin Wren (1971) ve Wren-Holliday'in (1972) çalışmalarına dayandığı ileri sürülmektedir. Süpürme algoritmasında depodan başlanıp müşteri noktalarına rassal olarak ulaşmaya çalışılarak uygun kümeler belirlenir. Daha sonra ise her kümede gezgin satıcı problemi çözülür. Rota kapasitesi dolana kadar müşteri eklenerek uygun rotalar tespit edilir. İlk rotanın kapasitesi

dolduğunda ikinci rota için aynı işlem uygulanır. Bu süreç tüm düzlem taranana kadar sürdürülür [28].

Bu algoritmanın birinci ve ikinci aşamaları sıralıdır. ARP'nin öklid düzlemde gerçekleştiği varsayılır. Müşterilerin konumlarının kutupsal koordinatları (r_i, θ_i) , bağlı oldukları depolar $r_1 = 0$ ve rastgele bir müşteri $\theta_{i^*} = 0$ 'daki i^* olarak gösterilir. Müşteriler $\theta_2 \leq \dots \leq \theta_n$ göre tekrar sıralanırlar. Algoritmanın adımları şöyledir [34]:

Aşama 1:

1. Adım: Kullanılmayan k aracı seçilir.
2. Adım: Rotalara atanmamış ve en küçük açığa (θ_i) sahip müşteriden (i) başlanır, rotadaki ardışık müşteriler $i+1, i+2, \dots, k$ aracının kapasitesi dolana kadar dahil edilir.
3. Adım: Eğer tüm müşteriler işleme alınmış veya tüm araçlar döngüye girmiş ise 2. Aşamaya geçilir, yoksa 1. adıma geri dönülür.

Aşama 2:

4. Adım: Araçlara atanmış her müşteri kümesi için ayrı ayrı gezgin satıcı problemi çözülüp son rotalar bulunur.

Fisher ve Jaikumar atama tabanlı algoritması: Bu algoritmada; ilk olarak müşteri sayısı seçilir ve kalan her i müşterisini k müşterisi ile rotalamak için sezgisel maliyet (d_{ik}) hesaplanır. Sezgisel maliyet hesaplandıktan sonra genelleştirilmiş atama problemi (GAP) kullanılarak çözülür. Kapasite kısıtını karşılayan k adet grup oluşturulur. Her grup optimalliği sağlamak için gezgin satıcı problemi yardımıyla rotalanır [35]. Fisher ve Jaikumar Algoritmasının çözüm adımları şöyledir [30]:

Aşama 1:

1. Adım: Grupların çekirdeğini oluşturacak m müşteri seçilir ve her birine bir araç ayrılır.

2.Adım: Her i müşterisinin her k kümesine atanması sonucu oluşan maliyetler d_{ik} hesaplanır. $d_{ik} = \min\{c_{oi} + c_{ik} + c_{k0}, c_{0k} + c_{ki} + c_{i0}\} + (c_{0k} + c_{k0})$

3.Adım: Maliyet (d_{ik}), genelleştirilmiş atama problemi (GAP) ile çözülür.

Aşama 2:

4.Adım: GAP ile oluşturulan gruptaki müşteri kümeleri gezgin satıcı problemi yardımıyla rotalanır.

Taç yaprağı algoritması: Ryan, Hjorring ve Glover 1993 yılında süpürme algoritmasının doğal olarak geliştirilmiş halini taç yaprağı olarak isimlendirdi. Burada oluşturulan rotalara taç yaprağı denilmektedir. Bu algorithmada birbiri ile ortak noktaları kullanılabilen bir rota havuzu oluşturulmakta, araç rotalama probleminin çözümü içinde rotaların arasından seçim yapılmaktadır. Rotaların arasından seçim yapmak için kullanılan model şöyledir [28]:

$$\min \sum_{k \in S} d_k x_k \quad (2.9)$$

Şu kısıtlara göre;

$$\sum_{k \in S} a_{ik} x_k = 1 \quad i = 1, \dots, n \quad (2.10)$$

$$x_k = 0 \text{ yada } 1 \quad k \in S \quad (2.11)$$

$$a_{ik} = 0 \text{ yada } 1 \quad k \in S \quad i = 1, \dots, n \quad (2.12)$$

Bu modelde S rota setlerini, n müşteri sayısını, k oluşturulan rota sayısını ve d_k ise k rotasının toplam mesafesini ifade etmektedir. Eğer $a_{ik} = 1$ ise i nolu müşteri k rotasına aittir. Eğer $x_k = 1$ ise de k rotası çözüme aittir. Bu model ilk olarak Balinski ve Quandt tarafından 1964 yılında kurulmuş olup, büyük boyutlu problemlerde rota sayısı fazla olduğundan dolayı uygulanması zorlaşmaktadır [2].

Taillard algoritması: Bu algoritma, 1993 yılında araç rotalama problemleri için Taillard tarafından geliştirilmiştir. Bu algoritma Tabu rota algoritmasının bazı özelliklerini içeren tabu arama uygulamasıdır. Algoritmada komşu çözümler λ -yer değiştirme ($\lambda = 1$) tekniği ile bulunmakta ve uygun olmayan çözümler tabu rota algoritmasının tersine kabul edilmemektedir. Taillard algoritmasının yeni özelliği, ana problemi alt problemlere bölüp paralel hesaplamaya olanak sağlamasıdır. Bu bölme işlemi deponun merkezde bulunup müşterilerin düzgün olarak depo çevresine dağıldığı düzlemsel problemlerde şu şekilde yapılmaktadır: Alt problemler müşterilerin depo merkezli sektörler ayrılması ile elde edilir. Her bir alt problem paralel hesaplamalarla bağımsız olarak çözülür fakat noktalardan yan yana bulunanların bir sektörden diğerine periyodik olarak geçmesi gerekmektedir. Bu nedenle deponun merkezde ve noktaların da düzgün olarak dağılmış olması gerekir. Eğer depo merkezde değil ve müşteriler düzgün olarak dağılmamışsa bölme işlemi en kısa yollar göz önüne alınarak merkezi depoda başlayan ağaç yapısına göre yapılmaktadır [2].

Bu algoritmada yer alan iterasyonlarda genel olarak sırayla aşağıdaki işlemler yapılmaktadır:

- Noktaları sektörler bölme ve alt problemler elde etme.
- Her alt problemi standart tabu arama algoritması ile çözme.
- TA algoritmasında belirli bir iterasyon sayısından sonra bu alt problemlerde bulunan çözümleri birleştirerek probleme bütün bir çözüm üretme [36].

Taillard algoritmasının tabu rota algoritması ile benzer yönü, tabu etiketleri kullanması ve sürekli olarak çeşitlendirme stratejisini kullanmasıdır. Çeşitlendirme stratejisinde yer alan sık sık tekrarlanan rotalar arasında nokta alışverişini cezalandırmaya ek olarak bu yöntemde, alt problemlerin sınırları periyodik olarak güncellenmektedir. Bu algoritma, literatürde tabu arama yöntemleri arasında en iyi sonuçları veren algoritma olarak gösterilmektedir [31].

Bramel ve Simichi-Levi lokasyon tabanlı algoritması: Bramel ve Simichi-Levi sabit sayıda araç kullanarak 1995 yılında yeni bir iki aşamalı algoritma geliştirmişlerdir.

Bu algoritmada Fisher ve Jaikumar'ın geliştirdikleri atama tabanlı yönteminden farklı olarak rota kökleri bulunurken kapasiteli lokasyon problemi çözülmektedir. Kapasiteli lokasyon problemi, müşteri koordinatları ve talep değerleri bilinen bir düzeneğin, n adet ve C kapasiteli tesisin bağlı olduğu müşterilere en yakın konumda olması için hangi koordinatlara kurulacağını bulduğu problemdir [2].

Algoritmanın işleyiş adımları aşağıda belirtilmektedir [10]:

1.Adım: Müşteriler, toplam talepleri C araç kapasitesini geçmeyecek şekilde gruplanarak n adet sanal tesise bağlanırlar. Aynı sanal tesise bağlı talep noktaları ile bir grup oluşturur. Sanal tesisin koordinatları belirlenirken, oluşturulan grup içinde yer alan müşterilere olan toplam mesafenin minimum olması sağlanır.

2.Adım: Her grupta bulunan müşterilere ek olarak merkez depo da eklenir ve bir rota oluşturulur. Aracın müşterilere ziyaret sırası, ekleme mantığı ile belirlenir. Bir k grubunda yer alan tüm noktalar $T_k = \{0, i_1, \dots, i_L\}$ ise rastgele bir nokta seçilerek başlangıç rotası oluşturulur. Daha sonra, rotaya eklenecek talep noktası i'nin sırası daha önce rotalanmış noktalar ile aşağıdaki maliyet fonksiyonlarından biri seçilerek belirlenir:

Direkt maliyet: $d_{ik} = \min_k = 1, \dots, 1\{2c_{iik}\}$

En yakın nokta maliyeti: $d_{ik} = \min_k = \{c_{iki} + c_{iik+1} - c_{iki+1}\}$

Budanmış dal ve sınır algoritması: Chritofides, Mingozi ve Toth tarafından 1979'da geliştirilen bu algoritmada basit bir arama ağacı yapısı kullanılmaktadır. Bu arama ağacı araç rotası olarak birçok seviye içerir. Her seviye ise olası ve egemen olunamayan araç rotaları kümesini içerir. Arama ağacında her seviyede tek bir dal bulunmakta ve her iterasyonda rotalanmamış bir müşteri seçilerek onun içinde yer alması mümkün rotalar üzerinden sonuç bulunmaya çalışılmaktadır [37].

Bu yöntemde, tüm olası dallanma olanakları kullanılmaz. Dallanma, sezgisel kurallar yardımıyla daraltılır. Böylece arama ağacının çok daha az sayıda düğüm içermesi ve

daha kısa sürede bir sonuca ulaşılması sağlanır. Ulaşılan sonucun en iyi olma garantisi yoktur fakat iyi bir çözüm elde edilmeye çalışılır [38].

- Önce rotala sonra grupla yöntemleri

Önce rotala sonra grupla yöntemlerinin ilk aşamasında çevre kısıtlarını dikkate almadan büyük bir gezgin satıcı problemi turu oluşturulur. İkinci aşamasında ise bu tur araç rotalarına ayrılır. Araç sayısının sınırlı olmadığı problemlere uygulanan bu yöntem ilk olarak, ikinci aşamadaki problemin standart bir en kısa yol problemi olduğunu ve $O(n^2)$ zamanında çözülebileceğini araştırmış olan Beasley tarafından öne sürülmüştür. En kısa yol probleminde i ve j düğümleri arasındaki dolaşım maliyeti d_{ij} , $c_{0i} + c_{0j} + l_{ij}$ ye eşittir (l_{ij} , gezgin satıcı problemi turunda i 'den j 'ye gitme maliyetidir.). Önce rotala sonra grupla sezgiselinin diğer yaklaşımlarla rekabet edebilir özellikte olduğunu kanıtlayan bulgular bulunmamaktadır [37].

2.4.3. Metasezgisel yöntemler

“Metasezgisel” terimi literatürde ilk defa 1986 senesinde Glover tarafından kullanılmıştır. "Meta" öneki Yunancada “daha ilerisi, üst derecede olanı” anlamına gelmektedir [10].

Klasik sezgisel yöntemlerde çözüm uzayında arama yapılırken belirlenen komşuluk yapısından daha iyi bir komşu çözüm bulunamazsa arama durdurulmaktadır. Bu nedenle klasik sezgisel yöntemler yerel minimum noktalara takılmakta ve yapılan arama kör bir arama olmaktadır [18]. Metasezgisel yöntemler ise çözüm uzayının en umut verici noktalarında arama yapıp yerel optimum noktalara takılmadan optimum çözüme yakın sonuçlar elde etmeyi amaçlamaktadır. Bu yöntemlerle elde edilen sonuçlar, klasik sezgisel yöntemlerle üretilen sonuçlara göre daha iyidir fakat hesaplama işlemi klasik sezgisel yöntemlere kıyasla daha uzun sürmektedir [10].

2.4.3.1. Tavlama benzetimi

Tavlama Benzetimi'nin kesikli en iyileme problemlerinde kullanılması 1980'li yıllara dayanmaktadır. Kombinatoriyal problemler için TB kullanımı ilk olarak Kirckpatrick ve arkadaşları (1983) ile Cerny (1985) tarafından önerilmiştir. Bu çalışmalar Metropolis ve arkadaşlarının 1953 yılında metalleri ısıtarak enerji seviyelerinin hesaplanmasına dayanmaktadır [10].

Tavlama terimi fiziksel olarak, ısıtılan bir katıdan düşük enerji durumları elde etme sürecidir. Bu süreçte ilk olarak katının sıcaklığı eriyebileceği bir değere kadar yükseltilir. Daha sonra ise katıların parçacıkları düşük sıcaklıkta daha düzenli olduğu için parçacıklar kendini düzenleyene kadar sıcaklığı giderek azaltılır.

Katıların tavlama sürecinde geçirdiği durumları benzetebilmek için 1953 yılında Metropolis ve arkadaşları tarafından "Metropolis Algoritması" geliştirilmiştir. Metropolis benzetim yardımı ile enerjideki değişimi hesaplamıştır. Metropolis algoritmasının temel prensibi, soğutma işlemindeki enerji değişimine göre belirlenir. Algoritmada i durumunda bulunan katının enerjisi E_i iken, bir sonraki j durumuna geçen katının enerjisi E_j 'dir. Eğer j durumundaki enerji i durumundaki enerjiden küçük veya eşitse, j durumu yeni mevcut çözüm olarak kabul edilir. Aksi halde j durumu Denklem 2.13'teki formüle göre elde edilen olasılık değeri ile kabul edilir. Bu olasılık değeri "Metropolis Kriteri" olarak adlandırılmaktadır [39].

$$\exp \left[\frac{E_i - E_j}{k_B * T} \right] \quad (2.13)$$

Algoritmanın çalışma şekli şöyledir: Bir olurlu çözümle ve bir T sıcaklık değeriyle başlanır. Bir sonraki adımda bu olurlu çözüm ile komşu çözüm değerlendirilir. Eğer yeni çözümün amaç fonksiyonunun değeri daha iyi ise algoritma bu çözümle devam eder. Fakat daha kötü bir sonuç elde ediliyorsa TB algoritması bu çözümü "Metropolis Kriteri" ile elde edilen olasılık değerine göre kabul eder. Söz konusu T sıcaklığı başta büyük alınır, böylece kötü çözümlerin kabul edilme olasılığı yüksektir,

fakat ilerledikçe bu sıcaklık değeri de bir soğutma oranına bağlı olarak azalır, böylece kötü çözümler daha düşük olasılıkla kabul edilmiş olur [40].

Osman'ın 1993 yılında yaptığı TB uygulamasının anlaşılması zor olmasına rağmen diğer tavlama benzetimi uygulamalarına göre daha başarılıdır. Bu algoritma daha iyi bir çözümle çalışmaya başlar ve deneme fazındaki bazı parametreleri ayarlandığı için zengin çözüm komşulukları keşfedilir. Soğutma çizelgesi de diğerine göre daha karmaşıktır [41].

2.4.3.2. Tabu arama algoritması

Tabu arama algoritması, çözüm uzayını araştırmak için yerel araştırma sürecine rehberlik eden bir algoritmadır ve ilk defa Glover (1986) tarafından insan hafızasının çalışmasından esinlenilerek önerilmiştir [42]. Tabu arama algoritmasında iki önemli yapı bulunmaktadır. Bunlardan ilki komşuluk arama yapısıdır. Oluşturulan mevcut komşulukta birbirini izleyen seçilmiş çözümlerden daha iyi olanı bir sonraki çözüm olarak atanır. İkincisi ise arama bölgesinde sınırlandırılmış yol izlenip yerel optimuma uğramadan ve arama çeşitliliğini sağlayarak aramaya rehberlik yapmaktır. Mevcut en iyi çözüm en iyi değerlerin kaydedildiği listede bulunmuyorsa seçilebilir. Geçmişteki en iyi değeri belirli sayıya kadar tutan listeye tabu listesi denir [43]. Tabu arama algoritmasının adımları şu şekildedir [44]:

1. Adım: Rastgele olarak bir başlangıç çözümü seçilir.
2. Adım: Seçilen başlangıç çözümünün komşu çözümleri belirlenir. Bir amaç fonksiyonu göz önüne alınarak komşu çözümleri değerlendirilir. Bu değerlendirme sırasında arama listesi hafızada tutulur. Değerlendirilen komşular tabu arama listesinde yoksa bulunmuş en iyi çözüm ile karşılaştırılır ve arama listesine eklenir.
3. Adım: Son çözüm o ana kadar bulunmuş en iyi sonuç ise yeni başlangıç çözümü olarak seçilir.
4. Adım: Durdurma kriteri sağlanana kadar 2. ve 3. adımlar tekrar edilir.

2.4.3.3. Parçacık sürü optimizasyonu

Parçacık Sürü Optimizasyonu (PSO), popülasyon temelli ve sürü zekası esasına dayalı bir optimizasyon yöntemidir. 1995-1996 yıllarında Kennedy ve Eberhart tarafından tasarlanmıştır [45]. Kuş ve balık sürülerinin yiyecek arayışları ve tehlikeden kaçışları esnasındaki toplu hareketlerinden esinlenerek geliştirilmiştir.

PSO'da popülasyonu oluşturan bireyler parçacık olarak adlandırılır ve her parçacığın kendine ait bir hızı vardır. Her nesilde bireyler kendi hızlarını, en iyi bireye göre yenilerler. Bu sayede yeni nesildeki bireyler bir öncekine göre daha iyi pozisyona gelirler. Tabu arama algoritmasının adımları şu şekildedir [46]:

1. Adım: Rastgele üretilen başlangıç pozisyonları ve hızları ile birlikte parçacıklar oluşturulur.
2. Adım: Popülasyondaki tüm bireylerin uygunluk değerleri hesaplanır.
3. Adım: Her nesilde bütün bireyler bir önceki neslin en iyisi ile karşılaştırılır ve daha iyi birey varsa yer değiştirilir.
4. Adım: Nesildeki en iyi değer global en iyi değerden daha iyi ise yer değiştirilir.

5. Adım:

$$V_{id} = W * V_{id} + c_1 * rand_1 * (P_{id} - X_{id}) + c_2 * rand_2 * (P_{gd} - X_{id}) \quad (2.14)$$

$$X_{id} = X_{id} + V_{id} \quad (2.15)$$

Denklem 2.14 ve Denklem 2.15'te X_{id} pozisyonu, V_{id} hız değerlerini, $rand_1$ ve $rand_2$ değerleri ise 0-1 arasında rastgele üretilmiş sayıları göstermektedir. W atalet ağırlık değeridir. c_1 , c_2 ivme katsayılarıdır ve genellikle 2'ye yakın bir değer kabul edilirler.

6. Adım: Durdurma kriteri sağlanana kadar 2. 3. 4. ve 5. adımlar tekrar edilir.

2.4.3.4. Karınca kolonisi algoritması

Karınca kolonisi algoritması (KKA), gerçek karıncaların beslenme davranışından esinlenilerek Dorigo ve arkadaşları (1991) tarafından geliştirilmiştir [47].

Karıncalar yuvaları ile besin kaynakları arasında gidebilecekleri yolları belirlemektedirler. Belirlenen yolların herhangi birinden ilk geçen karınca yola feromon deneni ve etrafa koku yayan bir salgı bırakmaktadır. Bir karınca yolu üzerindeki feromon miktarı yiyecek ile yuva arasındaki mesafeye ve yiyeceğin kalitesine bağlı olmaktadır. Eğer yol kısa ise bu koku daha yoğun olmaktadır ve diğer karıncalar da bu yoldan devam etmektedirler. Karıncalar iki yolun kesiştiği yerde hangi yolu seçeceklerine ilk önce koku miktarının yoğunluğuna, ikinci olarak ise gelişigüzel bir ölçüte göre karar vermektedirler. Bu gelişigüzel seçimin amacı ise bütün karıncaların aynı yolda gitmesini engelleyerek yeni ve daha kısa yollar keşfetmektir [48]. KKA üçüncü bölümde daha ayrıntılı bir şekilde incelenecektir.

2.4.3.5. Genetik algoritma

Genetik algoritma (GA) doğadaki evrim süreçlerini kullanan bir arama yöntemidir. Michigan üniversitesinde bulunan John Holland 1975 yılında yeni çocuklar oluşturmak için ebeveynlerden gelen bilgileri farklı kombinasyonlara göre birleştirip genetik algoritma tekniğini bulmuştur.

Genetik algoritmalar çözüm uzayındaki her noktayı kromozom adı verilen ikili dizi olarak kodlar ve her noktanın bir uygunluk değeri vardır. Genetik algoritma her nesilde mutasyon ve çaprazlama gibi GA operatörlerini kullanarak yeni bir popülasyon oluşturmaktadır. Bir kaç nesil sonunda popülasyonda uygunluk değerleri daha iyi bireyler yer alacaktır. Özetle; genetik algoritmada çözümlerin kodlanması, uygunluk değerlerinin hesaplanması, çoğalma, çaprazlama ve mutasyon işlemleri uygulanmaktadır [34]. GA üçüncü bölümde daha ayrıntılı bir şekilde incelenecektir.

2.4.3.6. Yapay arı kolonisi algoritması

Yapay arı kolonisi algoritması (YAKA) Karaboğa ve arkadaşları tarafından 2005 yılında arıların yiyecek arama davranışlarından esinlenilerek geliştirilmiş bir tür optimizasyon algoritmasıdır [49].

Yapay arı kolonisi algoritmasında bazı kabuller yapılmaktadır. Bunlardan ilki her bir kaynağın nektarının sadece bir işçi arı tarafından alınıyor olmasıdır. Buradan yola çıkarak işçi arı sayısı toplam kaynak sayısına eşittir diyebiliriz. Diğer kabullenme ise işçi arıların sayısının gözcü arıların sayısına eşit olmasıdır. Kaynağında nektar biten işçi arı kâşif arı haline dönüşmektedir. Bu algorithmada kaynakların yerleri bize olası çözümleri, kaynaklardaki nektar miktarları ise çözümün uygunluğunu göstermektedir. Dolayısıyla yapay arı kolonisi algoritması en fazla nektara sahip kaynağı bulmaya çalışarak optimum çözümü bulmaya odaklı çalışmaktadır [50]. YAKA üçüncü bölümde daha ayrıntılı bir şekilde incelenecektir.

BÖLÜM 3. ÇALIŞMADA KULLANILAN ALGORİTMALAR

3.1. Literatür Araştırması

Bu tez çalışmasında karınca kolonisi algoritması, genetik algoritma ve yapay arı kolonisi algoritması kullanılmıştır. Bu kullanılan algoritmalar ile yapılmış bazı ARP çalışmaları Tablo 3.1’de özetlenmiştir.

Tablo 3.1. Çalışmada kullanılan algoritmalar ile yapılmış bazı ARP çalışmaları

Problemin Tipi	Yazarı	Çözüm Yöntemleri		
		KKA	GA	YAKA
Klasik ARP	(Bullnheimer ve diğ., 1997) [51]	X		
	(Bullnheimer ve diğ., 1999) [52]	X		
	(Baker ve Ayechev, 2003) [53]		X	
	(Lucic ve Teodorovic, 2003) [54]			X
	(Kuşçu, 2009) [55]	X		
Kapasite Kısıtlı ARP	(Mazzeo ve Loiseau, 2004) [56]	X		
	(Gajpal ve Abad, 2009) [57]	X		
	(Keskintürk, 2009) [58]	X		
	(Wang ve Zhang, 2009) [59]		X	
	(Ekizler, 2011) [16]	X		
	(Brajevic, 2011) [60]			X
	(Szeto ve diğ., 2011) [61]			X
(Reed ve diğ., 2014) [62]	X			
Çok Depolu ARP	(Gambardella, 2004) [63]	X		
	(Kurt, 2008) [5]	X		
	(Yılmaz, 2008) [12]	X		
	(Ho ve diğ., 2008) [64]		X	
	(Önder, 2011) [34]		X	
	(Narasimha ve diğ., 2013) [65]	X		
Heterojen Filolu ARP	(Liu ve diğ., 2009) [66]	X		
Dinamik ARP	(Montemanni ve diğ., 2005) [67]		X	
Dağıtım ve Toplamalı ARP	(Li ve Zhang, 2007) [68]		X	
	(Göksal, 2010) [17]		X	
	(Tasan ve Gen, 2012) [69]		X	

Tablo 3.1. Çalışmada kullanılan algoritmalar ile yapılmış bazı ARP çalışmaları (Devamı)

	(Berger ve Barkaoui, 2004) [70]		X
	(Rizzoli ve diğ., 2005) [71]	X	
	(Donati, 2006) [72]	X	
	(Ma ve diğ., 2006) [73]		X
	(Alvarenga ve diğ., 2007) [74]		X
Zaman Pencere li ARP	(Şeker, 2007) [18]		X
	(Kılıç, 2008) [75]	X	
	(Wang ve diğ., 2008) [76]		X
	(Wang ve diğ., 2008) [77]		X
	(Dursun, 2009) [40]		X
	(Kemer, 2010) [42]		X
	(Vidal ve diğ., 2013) [78]		X

Bu çalışmada ele alınan problem, kapasite kısıtlı ARP'dir. Araçların hepsi en çok 40 palet kapasitelidir ve araçlara 32 ile 40 palet arasında yükleme yapılmaktadır. Bu ele alınan kapasite kısıtlı ARP, ürünlerin çeşidi ve dağıtım yapılacağı günlere göre 4 adet alt probleme ayrılmaktadır. Bu alt problemler; taze ürünün yoğun günde dağıtımı, kuru ürünün yoğun günde dağıtımı, taze ürünün sakın günde dağıtımı ve kuru ürünün sakın günde dağıtımı problemidir. Bu alt problemlerden taze ürünün yoğun günde dağıtımı ve kuru ürünün yoğun günde dağıtımı alt problemlerinde 78 adet mağaza, taze ürünün sakın günde dağıtımı ve kuru ürünün sakın günde dağıtımı alt problemlerinde ise 66 adet mağaza mevcuttur.

Ürünlerin yoğun günde dağıtımı alt problemleri için kurulan matematiksel model aşağıda verilmiştir.

Değişkenler:

$$x_{ij} = \begin{cases} 1, & \text{eğer araç } i \text{ müşterisinden } j \text{ müşterisine gidiyorsa} \\ 0, & \text{aksi takdirde} \end{cases}$$

$$i \neq j, \quad i, j \in \{0, \dots, \dots, 78\} \text{ ve } 0 \text{ ana depo}$$

Amaç Fonksiyonu:

$$\text{En az } Z = \sum_{i=0}^{78} \sum_{j=0, i \neq j}^{78} d_{ij} x_{ij} \quad (3.1)$$

Kısıtlar:

$$\sum_{i=1, i \neq j}^{78} x_{ij} = 1, \quad \forall j, \quad j \in \{1, \dots, 78\} \quad (3.2)$$

$$\sum_{j=1, j \neq i}^{78} x_{ij} = 1, \quad \forall i, \quad i \in \{1, \dots, 78\} \quad (3.3)$$

$$\sum_{i=0}^{78} \sum_{j=0, i \neq j}^{78} x_{ij} + x_{ji} \leq 1, \quad (3.4)$$

$$\sum_{i=1}^{78} q_i \sum_{j=0, i \neq j}^{78} x_{ij} \leq 40, \quad (3.5)$$

Ürünlerin sakin günde dağıtımı alt problemleri için kurulan matematiksel model aşağıda verilmiştir.

Değişkenler:

$$x_{ij} = \begin{cases} 1, & \text{eğer araç } i \text{ müşterisinden } j \text{ müşterisine gidiyorsa} \\ 0, & \text{aksi takdirde} \end{cases}$$

$$i \neq j, \quad i, j \in \{0, \dots, \dots, 66\} \text{ ve } 0 \text{ ana depo}$$

Amaç Fonksiyonu:

$$\text{En az } Z = \sum_{i=0}^{66} \sum_{j=0, i \neq j}^{66} d_{ij} x_{ij} \quad (3.6)$$

Kısıtlar:

$$\sum_{i=1, i \neq j}^{66} x_{ij} = 1, \quad \forall j, \quad j \in \{1, \dots, 66\} \quad (3.7)$$

$$\sum_{j=1, j \neq i}^{66} x_{ij} = 1, \quad \forall i, \quad i \in \{1, \dots, 66\} \quad (3.8)$$

$$\sum_{i=0}^{66} \sum_{j=0, i \neq j}^{66} x_{ij} + x_{ji} \leq 1 \quad (3.9)$$

$$\sum_{i=1}^{66} q_i \sum_{j=0, i \neq j}^{66} x_{ij} \leq 40, \quad (3.10)$$

Yukarıdaki modellerde, d_{ij} ; i müşterisi ile j müşterisi arasındaki uzaklığı, q_i ise i müşterisinin talep miktarını göstermektedir.

Amaç fonksiyonu (3.1 ve 3.6) toplam kat edilen mesafenin en az yapılması gerektiğini ifade etmektedir. (3.2), (3.3), (3.7) ve (3.8) nolu kısıtlar, bir müşterinin mutlaka bir araç tarafından ziyaret edilmesi gerektiği ve bir müşteriyi ziyaret eden aracın aynı zamanda o müşteriden hareket etmek zorunda olduğunu belirtmektedir. (3.4) ve (3.9) nolu kısıtlar, depoda başlamayan ve depoda bitmeyen (alt) turları elemekte kullanılmaktadır. (3.5) ve (3.10) nolu kısıtlar ise araçlara yapılan yüklemelerin araç kapasite değeri 40'ı geçmemesi gerektiğini belirtmektedir.

Bu çalışmada ilk olarak 78 mağazalık ve 66 mağazalık alt problemler için ayrı ayrı gezgin satıcı problemi uygulanmaktadır. 66 mağazaya sahip alt problemler için çözüm uzayı $66! = 5,44344939E+92$, 78 mağazaya sahip alt problemler için çözüm uzayı ise $78! = 1,13242812E+115$ 'dir. Ele alınan problemde şehir sayısı arttıkça çözüm karmaşıklığı da artmaktadır. Bu tip problemler NP-zor problem sınıfındadır. Alt problemlerin çözüm uzayı çok büyük olduğu için pratik olarak kesin çözüm yöntemleri kullanılarak çözülmesi imkânsızdır. Bu nedenle metasezgisel yöntemlerden karınca kolonisi, genetik ve yapay arı kolonisi algoritmaları kullanılmış ve bir mağazadan harekete başlayıp her mağazaya sadece bir defa uğradıktan sonra başladığı mağazaya geri dönen en kısa tur oluşturulmuştur. Daha sonra alt problemlerin haftalık talepleri göz önüne alınarak, elde edilen rotadaki ilk mağazadan başlanıp kamyonların palet kapasite kısıtını sağlayacak şekilde kümeler oluşturulmaktadır. Oluşturulan her küme dağıtım yapacak olan aracı temsil etmektedir. Son olarak ise oluşturulan kümelerin her birinin rotaları ana depodan başlayıp ve yine ana depoda son bulacak şekilde dal ve sınır yöntemiyle oluşturulup kat edilen mesafeler en az yapılmaya çalışılmıştır.

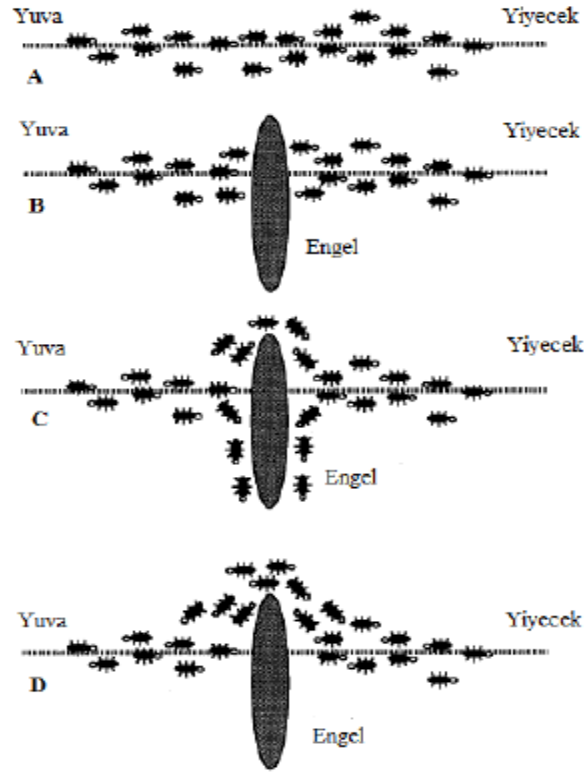
3.2. Karınca Kolonisi Algoritması

3.2.1. Gerçek karıncalar

Gerçek karıncalar yuvaları ile yiyecek kaynakları arasındaki en kısa yolu bulabilme yeteneğine sahiptirler. Yollarına bir engel konulduğunda veya coğrafi olaylar nedeniyle kullandıkları yolda bir değişiklik olduğunda, kendilerine yeni bir kısa yol bulurlar ve bu yeni duruma adapte olurlar [79].

Dorigo ve Gambardella karıncaların davranışlarını yorumlayabilmek için 1996 yılında yaptıkları deneyde, karıncaların yollarına engeller koyarak davranışlarını gözlemlemiştir. Tasarladıkları deney Şekil 3.1'de gösterilmektedir.

Şekil 3.1A karıncaların yuvalarıyla yiyecek kaynağı arasındaki izledikleri yolu göstermektedir. Karıncaların şekildeki gibi tek bir hat oluşturup ilerlemeleri yol üzerine bıraktıkları feromon sayesinde mümkün olmaktadır. Karıncaların bir sonraki aşamada nasıl davranacaklarını gözlemleyebilmek için Şekil 3.1B'de yollarının üzerine bir engel koyulmuştur. Karıncalar yolları üzerine konulan engelle karşılaştıklarında takip edecekleri feromon olmadığından dolayı sağ ve sol alternatiflerinden birini rastgele Şekil 3.1C'deki gibi seçmişlerdir. Burada tesadüfen kısa yolu tercih eden karıncaların birim zamanda yol üzerine bırakacakları feromon miktarı daha çok olacak ve belirli bir süre sonra Şekil 3.1D'de görüldüğü gibi tüm karıncalar kısa yolu kullanmaya başlayacaklardır [80].



Şekil 3.1. Gerçek karıncaların en kısa yolu bulma adımları [80]

3.2.2. Yapay karıncalar ve karınca kolonisi algoritması

Karıncaların yol seçim özelliği, bazı özellikleri aynen kalarak ve bir takım eklemeler yapılarak gerçek problemlerin çözümünde kullanılabilir hale getirilmiştir. Gerçek karıncalardan aynen alınan özellikler şöyledir:

- Feromon aracılığı ile karıncalar arasında kurulan iletişim,
- Feromon miktarının fazla olduğu yolların öncelikle tercih edilmesi,
- Kısa yollar üzerinde bulunan feromon miktarının daha hızlı artması.

Aynen alınan özelliklerin yanı sıra karıncalardan daha iyi bir performans alabilmek için bir takım özellikler eklenmiştir. Bu özellikler şöyledir [12]:

- Yapay karıncalar sürekli olmayan ayrık bir dünyada yaşarlar.
- Yapay karıncalar tamamen kör olmayıp problemle ilgili detaylara erişebilirler.

- Yapay karıncalar problemin çözümü için oluşturdukları bilgileri tutabilecekleri hafızaya sahiptirler.

Yapay karıncalar bulunan çözümün kalitesini belirleyen feromon miktarını biriktirirler. Yapay karıncaların feromon yayma zamanı probleme bağlıdır ve bu nedenle gerçek karınca davranışlarını yansıtmaz. Sistemin verimliliğini iyileştirmek için, karınca kolonisi algoritması ileriye bakma, yerel optimizasyon, gerileme gibi gerçek karıncalarda bulunmayan özelliklerle geliştirilebilir.

Karınca kolonisi algoritması, yapay feromon izlerinin güncelleştirilmesiyle tekrarlanan bir yapıya sahiptir. Bu algoritma çeşitli varyasyonlara sahiptir. Bunlar; karınca sistemi, elitist karınca sistemi, rank temelli karınca sistemi, min-maks karınca sistemi, karınca-Q sistemi ve karınca kolonisi sistemidir [60].

Karınca kolonisi algoritmasının çalışma sürecindeki temel işlemler; yapay karıncaların turları sonunda geçmiş oldukları yolların feromon miktarlarının artırılması, belirli bir oranda feromon buharlaşmasının gerçekleştirilmesi, en iyi çözümün bulunması, buna bağlı olarak global feromon güncellemesinin yapılması ve karıncaların yenilenen bu feromon miktarlarına bağlı olarak yol tercihlerini gerçekleştirmeleridir [81]. Yol tercih kuralları geçiş kuralı olarak adlandırılmaktadır [60].

3.2.2.1. Geçiş kuralı

Karınca kolonisi algoritmasında yol tercihi yapılırken iki seçenek mevcuttur. Bunlardan ilki, q_0 olasılıkla feromonun en fazla olduğu yolun seçilmesidir. q_0 parametresi genellikle %90 olarak belirlenir. Bir i noktasında bulunan k karıncasının gideceği u noktası Denklem 3.11'deki gibi seçilmektedir.

$$j = \operatorname{argmax}\{\tau(i, u)^\alpha [\eta(i, u)]^\beta\} \quad u \in j_k(i) \quad \text{eğer } q < q_0 \quad (3.11)$$

$\tau(i, j)$, i ve j noktaları arasındaki feromon miktarını, $\eta(i, j)$ ise i ve j noktaları arasındaki mesafeyle ters orantılı olan seçilebilirlik parametresini göstermektedir. α ve β ise ayarlanabilir parametrelerdir.

Denklem 3.11'de $j_k(i)$, i noktasından gidilebilecek ve henüz k karıncası tarafından ziyaret edilmemiş noktalar kümesidir.

İkinci seçenek ise, gidilecek yolu, yollardaki feromon iziyle orantılı olarak seçmektir. $p_k(i, j)$ yolların seçilme olasılığıdır ve Denklem 3.12'deki gibi hesaplanmaktadır.

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in j_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta}, & \text{eğer } j \in j_k(i) \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.12)$$

Bu olasılıklara bağlı olarak gidilecek noktalar belirlenmektedir. Feromonun yoğun olduğu yolların seçilme olasılığı daha yüksektir [60].

3.2.2.2. Feromon güncellemesi

Çözüm uzayının taranabilmesi için feromon güncellemesi yapılmaktadır. İlk olarak tüm yollardaki feromonlar, belirlenen buharlaşma oranına göre buharlaştırılmaktadır. Daha sonra karıncaların geçtikleri yollardaki feromon miktarları, o yolu kullanan karıncanın toplam yol uzunluğuyla ters orantılı olarak arttırılmaktadır. Böylelikle kısa yola sahip karıncaların kullandıkları yollardaki feromon miktarları daha fazla olacaktır [82].

Feromon güncellemesi lokal feromon güncellemesi ve global feromon güncellemesi olmak üzere iki yolla yapılabilmektedir.

- Lokal feromon güncellemesi

Lokal feromon güncellemesinin amacı; her karıncanın kendi turunu tamamlarken geçtiği kenarlardaki feromon miktarını azaltarak, arkadan gelen karıncaları diğer

kenarları seçmeleri konusunda cesaretlendirmek ve farklı çözümlerin oluşmasını sağlamaktır. Lokal feromon güncellemesi işlemi Denklem 3.13'teki gibi yapılmaktadır:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + (\rho)\tau_0 \quad (3.13)$$

ρ parametresi buharlaşma oranını, τ_0 bütün kenarlara başlangıçta atanmış olan feromon miktarını göstermektedir [83].

- Global feromon güncellemesi

Global feromon güncellemesi önceden belirlenmiş olan m kadar karıncanın uygun rota oluşturmasından sonra, bulunan en iyi rota içerisindeki bütün kenarlara feromon ekleyerek gerçekleştirilmektedir. Global feromon güncellemesi işlemi Denklem 3.14'teki gibi yapılmaktadır:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + (\rho)L^{-1} \quad (3.14)$$

L bulunan en uygun rotanın tur uzunluğudur [77].

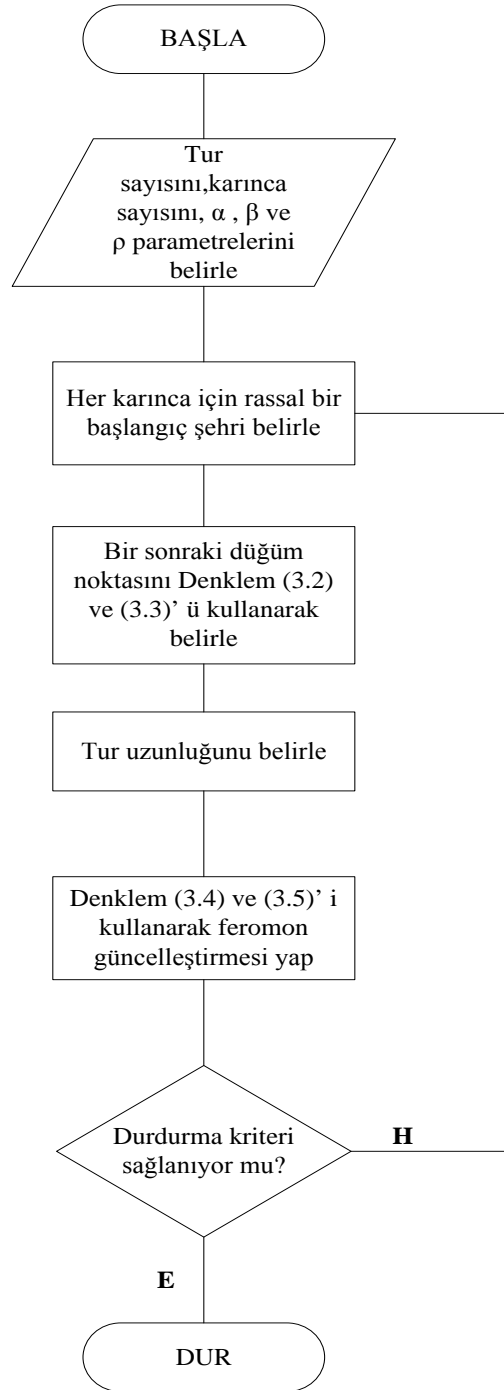
Bir iterasyon boyunca gerçekleştirilen işlemler yukarıda anlatıldığı gibidir. Aşağıda karınca koloni algoritmasının adımları görülmektedir [81]:

1. Adım: Tur sayısı, karınca sayısı, α , β ve ρ parametreleri belirlenir.
2. Adım: Karıncalar her düğüme (şehre) rastsal olarak yerleştirilir.
3. Adım: Her karınca, sonraki düğüm noktasını denklemde verilen seçilme olasılığına bağlı olarak seçer ve turunu tamamlar.
4. Adım: Her karıncanın kat ettiği yolların uzunluğu hesaplanır ve lokal feromon güncellemesi yapılır.

5. Adım: En iyi çözüm hesaplanır ve global feromon güncellemesinde kullanılır.

6. Adım: Maksimum iterasyon sayısı ya da yeterlilik kriteri sağlanana kadar 2. Adım'a gidilir.

Karınca kolonisi algoritmasının akış diyagramı Şekil 3.2 'de gösterilmiştir.



Şekil 3.2. Karınca kolonisi algoritmasının akış diyagramı

3.3. Genetik Algoritma

3.3.1. Genetik algoritmanın tanımı

Genetik algoritma, seçilme amacına en uygun bireyin hayatta kalıp, bazı genlerini başka bireylere aktararak diğer bireyleri de kendisi gibi amaca en uygun duruma getiren, tüm bunları yaparken de doğal seçim ve gen değişimlerine dayanan bir optimizasyon problemidir. Genetik algoritma ilk defa John Holland tarafından 1975 yılında geliştirilmiştir.

Genetik algoritma, çok fazla teorik bilgiye ihtiyaç duyulmadan uygulanabilmesi ve uygulamadan elde edilen sonuçların da direkt kullanabilmesi gibi avantajları sayesinde yaygın olarak kullanılmaktadır.

Genetik algoritma, rastgele seçilmiş bireylerden oluşan bir nesille optimizasyona başlamaktadır. GA'nın amacı; nesildeki zayıf bireyleri ortadan kaldırıp, güçlü bireylerden oluşan yeni nesiller oluşturmaktır. Bu amaç doğrultusunda, nesildeki güçlü bireyler kendilerine eş seçip genlerini çaprazlamaktadırlar. Oluşturulan her yeni nesilde bireyleri oluşturan değişkenlerden herhangi biri veya bir kaç mutasyona uğratarak farklı bireylerin oluşması sağlanmaktadır. Her yeni nesilde bu işlemler devam etmekte ve bireylerin birbirine benzeme oranı olarak kabul edilen durdurma kriteri sayesinde de algoritmanın çalışması sonlandırılmaktadır [84].

3.3.2. Genetik algoritmanın terimleri

Kromozom: Bir ya da birden fazla gen yapısının bir araya gelmesiyle oluşan ve problemin çözümü için gerekli tüm bilgiyi içeren dizilere kromozom denir. Her bir kromozom bir çözümü temsil eder [85].

Gen: Kromozom içerisinde kendi başına genetik bilgi taşıyan en küçük yapı birimine gen denir [86].

Popülasyon: Çözüm bilgilerini içeren kromozomların bir araya gelerek oluşturdukları topluluğa popülasyon denir. Genetik algoritmada popülasyondaki kromozom sayısı ile ilgili genel bir kural yoktur ve genelde sabit tutulur [87].

Popülasyon büyüklüğü arttıkça popülasyondaki çeşitlilikte artış gösterecektir. Bu artış sonucunda daha fazla uygunluk değeri hesaplanacağından problemin çözüm süresi de uzayacaktır [88].

Kodlama: Genetik algoritmanın uygulanmasındaki ilk adım değişkenlerin kodlanmasıdır. Probleme ait bilgilerin GA'nın anlayabileceği dile çevrilmesine kodlama denir [42].

Her probleme göre kromozomların kodlama şekli değişmektedir. İyi bir kodlama sayesinde iyi bir genetik yapı oluşturulabilir. Uygulamalarda kullanılan kodlama çeşitleri şunlardır [61]:

- İkili Kodlama: İkili kodlama GA uygulamalarında en çok tercih edilen kodlama türüdür. Her kromozom bit karakter dizilerinden oluşmaktadır. Genler sadece 0 veya 1 değerini alır [89]. Aşağıda ikili kodlamaya örnek gösterilmiştir.

Kromozom 1	1	0	0	1	0	0	1	1	1	0	0	1
Kromozom 2	0	0	0	1	1	0	1	1	0	0	0	0

Bu kodlama biçiminin gezgin satıcı, araç rotalama ve çizelgeleme gibi problemlerde kullanılması uygun değildir. Bunun nedeni; ikili kodlama ile tüm olası çözümlerin temsil edilememesidir [42].

- Permütasyon Kodlama: Bu kodlama çeşidi gezgin satıcı problemi ve çizelgeleme problemleri gibi permütasyon problemleri için kullanışlıdır. Burada her kromozom, sayıları bir sırada temsil etmektedir. Şekilde permütasyon kodlamaya örnek gösterilmiştir.

Kromozom 1	7	8	9	4	1
Kromozom 2	8	7	9	1	4

- Değer Kodlama: Değer kodlama yönteminde her dizi, bir değerler kümesinden oluşmaktadır. Değerler probleme göre herhangi bir sayı veya karakter olabilir.
- Ağaç Kodlama: Bu yöntem gelişen, değişen programlar veya ifadeler için kullanılır. Burada her kromozom, nesnelere ve nesnelere arası işlemleri içeren bir ağaç yapısından oluşmaktadır [89].

3.3.3. Genetik algoritmanın parametreleri

3.3.3.1. Popülasyon büyüklüğü: Popülasyon büyüklüğünün belirlenmesi genetik algoritmanın önemli aşamalarından biridir. Popülasyon küçük olduğunda iterasyon daha hızlı olacak fakat algoritmanın yerel optimuma takılma olasılığı artacaktır. Popülasyon çok büyük olduğunda ise çözüm kalitesi artacak ancak optimum çözüme ulaşma zamanı uzayacaktır. Goldberg 1985 yılında yalnızca kromozom uzunluğuna bağlı bir popülasyon büyüklüğü hesaplama yöntemi önermiştir [88]. Popülasyon büyüklüğünü hesaplama yöntemi Denklem 3.15'te gösterilmiştir.

$$N = 1,65 * 2^{0,21 * l} \quad (3.15)$$

N = popülasyon büyüklüğü

l = kromozom uzunluğu

Ayrıca Schaffer ve arkadaşları (1989), çok sayıda test fonksiyonu üzerinde yaptıkları araştırmalar sonucunda 20–30 arası bir popülasyon büyüklüğünün iyi sonuçlar verdiğini belirtmişlerdir [90].

3.3.3.2. Üreme parametresi: Genetik algoritmada üreme aşamasında bireyler oluşturulan popülasyondan seçilir ve yeni kuşağı oluşturacak çocuk bireyleri ortaya çıkarmak üzere tekrar yapılandırılır. Uyum değeri yüksek olan bireyler birçok kez seçilebilirken uyum değeri düşük olan bireyler elenebilmektedir. İki bireyin seçimi sonrası bireylerin kromozomları çaprazlama ve mutasyon mekanizmaları kullanılarak tekrar yapılandırılır [34].

3.3.3.3. Çaprazlama olasılığı: Mevcut en iyi kromozomların özelliklerini birleştirerek daha iyi özellikli kromozomlar elde etmek amacıyla yapılan çaprazlama operatörünün temel parametresi çaprazlama olasılığıdır ve P_c ile gösterilir. Çaprazlama olasılığı çaprazlamanın ne sıklıkla yapılacağını belirler [91].

Çaprazlama olasılığı eşleşme havuzuna girecek kromozomların sayısını belirler. Bu oranın yüksek olması, iyi özellikteki bireylerin yeni popülasyonda bulunma olasılığını azaltır. Düşük olması ise yeterli sayıda yeni bireyin oluşmasını engeller [92].

3.3.3.4. Mutasyon olasılığı: Amacı popülasyondaki genetik çeşitliliği korumak olan mutasyonun önemli parametresi, kromozomda ne kadar değişiklik olacağını gösteren mutasyon olasılığıdır ve P_m ile gösterilir. Mutasyon olasılığı 0 olduğunda yeni birey mutasyona uğramadan oluşur. %100 olduğunda ise kromozom tamamen değişmektedir [91].

Bazı çalışmalar popülasyon büyüklüğü 20'den büyükken $P_m > 0,05$ olarak seçildiği durumda veya popülasyon büyüklüğü 20'den küçükken $P_m < 0,002$ olduğu durumda performansın arttığını ortaya koymuştur. Orta büyüklükteki bir popülasyon için ise $P_m = 0,001$ oranının seçilmesi uygun olacaktır [42].

3.3.3.5. Kuşak farkı: Her kuşakta oluşan yeni kromozom oranına kuşak farkı denir. Kuşak aralığı yüksek bir değer olduğunda birçok kromozomun değiştiği anlamına gelmektedir.

3.3.3.6. Seçim stratejisi: Eski kuşağı yenilemenin çeşitli yöntemleri bulunmaktadır. Bunlar; kuşaksal strateji, en uygun (elitist) strateji ve denge durumu stratejisidir. Kuşaksal stratejide, mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirir. Popülasyonun en iyi kromozomu da yenilediğinden dolayı bir sonraki kuşağa aktarılamaz. En iyi kromozomu geri kazanmak için bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, popülasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir. Denge durumu stratejisinde ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir.

3.3.3.7. Uygunluk fonksiyonunun ölçeklenmesi: Uygunluk fonksiyonunun ölçeklenmesinde rank, oransal, doğrusal ve üstsel ölçekleme gibi yöntemler mevcuttur. Problemin yapısına göre en uygun ölçekleme yönteminin seçilmesi genetik algoritmanın etkin işlemesi açısından önemli kontrol parametrelerinden biridir [93].

3.3.4. Genetik algoritmanın çalışma prensibi

1.Adım: Olası çözümlerin kodlandığı bir çözüm grubu oluşturulmaktadır. Çözüm grubu denilen bu yapıya biyolojideki benzerliğinden dolayı popülasyon, çözüm kodlarına ise kromozom ismi verilmektedir. Bu adım popülasyondaki birey sayısının belirlenmesiyle başlar. Popülasyondaki birey sayısı için kesin bir değer olmamakla birlikte genellikle 100-300 aralığında bir büyüklük önerilir. Popülasyon bu aşamadan sonra rastgele oluşturulmaktadır [94].

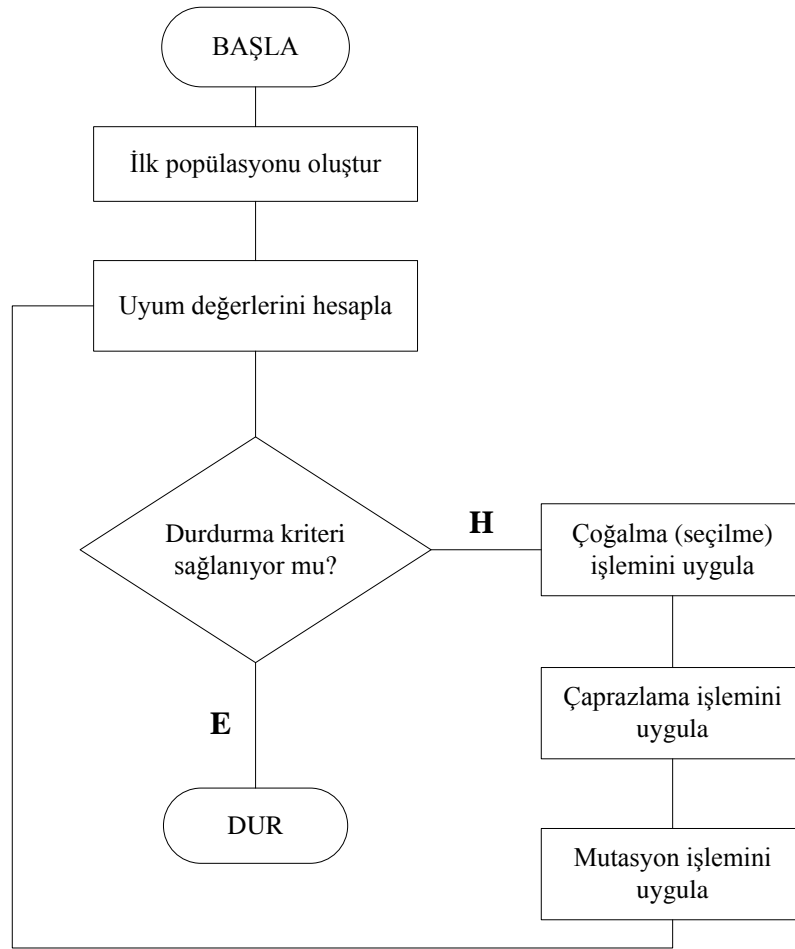
2.Adım: Bu adımda uygunluk fonksiyonu yardımıyla her kromozomun ne kadar iyi olduğu bulunur. Uygunluk fonksiyonu ayrıca genetik algoritmanın beynini oluşturmaktadır. Genetik algoritmanın probleme özel çalışan tek kısmı bu fonksiyondur. Çoğu zaman genetik algoritmanın başarısı bu fonksiyonun verimli ve hassas olmasına göre değişebilmektedir [95].

3.Adım: Bu adımda kromozomları uygunluk değerine göre eşleyerek yeniden kopyalama, seçim ve çaprazlama operatörleri uygulanır. Bu sayede yeni bir popülasyon elde edilmiş olur. Seçim yapılırken de rulet tekerleği seçimi ve turnuva seçimi gibi seçme yöntemleri kullanılmaktadır.

4. Adım: Bu adımda ise eski kromozomlar elenerek sabit büyüklükte yeni bir popülasyon oluşturulur. Yeni popülasyonun başarısını bulmak amacıyla tüm kromozomlar için uygunluk değerleri yeniden hesaplanır. Eğer yeni popülasyonun oluşum süresi dolmamışsa 3. adıma gidilir [96].

5.Adım: Popülasyonların hesaplanmasında en iyi bireyler saklandığı için o ana kadar bulunan en iyi kromozom bize sonucu vermektedir [94].

Genetik algoritmanın akış diyagramı Şekil 3.3'te gösterilmiştir.



Şekil 3.3. Genetik algoritmanın akış diyagramı [97]

3.4. Yapay Arı Kolonisi Algoritması

3.4.1. Arılar ve sürü zekâsı

Bir sürüde kendi başına organize olma ve iş bölümü olmak üzere iki önemli işlev bulunmaktadır. Kolonideki arılar hiç bir otorite olmaksızın kendilerine ait olan işleri başarıyla gerçekleştirirler. Arıların farklı işleri eş zamanlı olarak gerçekleştiriyor olması arılardaki iş bölümünü göstermektedir. Bu iş bölümü esnasında hiç bir otoritenin etki etmemesi de arılarda kendi başına organize olma yeteneğinin varlığını göstermektedir. Arı kovanının içerisinde farklı görevlerde birçok arı bulunmaktadır. Yapay arı kolonisi algoritması üç grup arı içermektedir. Bunlar işçi arılar, gözcü arılar ve kâşif arılardır. İşçi arılar, dışarıya çıkıp besin kaynağı ararlar. Kaynak

arayışını tamamlayınca da kovandaki dans alanına dönüp kaynağın uzaklığı, yönü ve kalitesi hakkında gözcü arılara bilgi verirler.

Gözcü arılar, işçi arıların dans alanında kendilerine aktardığı bilgileri değerlendirerek istediği bir kaynağa işçi arı olarak gidebilirler veya kaynak seçmek için biraz daha bekleyebilirler.

Kâşif arıların görevleri ise yeni besin kaynaklarını keşfetmektir.

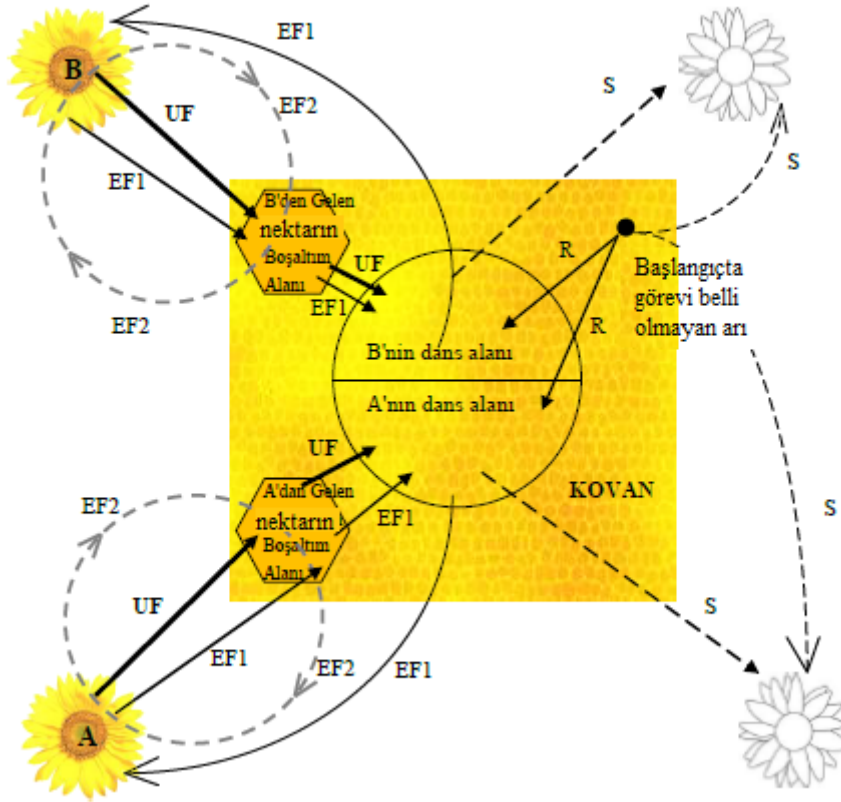
3.4.2. Arılarda besin kaynağı bulma

Arılar arasında bilgi alışverişi koloniye ait ortak bilginin oluşmasında çok önemlidir. Değişik bölümleri bulunan arı kovanının en önemli yeri dans alanıdır. Arılar besin kaynağının uzaklığını yönünü ve kalitesini dans alanında birbirlerine iletirler. Kaynağın kovanla arasındaki mesafe, nektar miktarı, nektarın tatlılığı, kolay çıkartılması gibi arılara kolaylık veya zorluk sağlayabilecek her şey dansın çeşidini ve hızını etkilemektedir. Dans çeşitleri; dairesel dans, kuyruk dansı ve titreme dansıdır. Daire ve kuyruk dansı; kaynağın kovan ile arasındaki mesafeyi belirtmek için yapılır. Daire dansı 50-100 m mesafede bir kaynak olduğunu, kuyruk dansı ise 100 m'den 10 km'ye kadar geniş mesafede bir kaynak olduğunu ifade etmek için yapılan dans türüdür. Titreme dansı da bulunan nektar çok kaliteli olduğunda ve miktar bakımından çok olduğunda yapılan dans türüdür. Gözcü arılar dans alanında yapılan dansları izleyerek kendilerine en uygun kaynağı belirlerler [98].

Arıların besin kaynağı arama davranışları Şekil 3.4'te özetlenmiştir. Burada A ve B keşfedilmemiş iki kaynak olarak kabul edilmiştir. Başlangıçta görevi belli olmayan arı için iki seçenek mevcuttur. Bunlardan ilki; S ile gösterilen kâşif arı olarak besin kaynağı aramaya başlayabilir. İkincisi ise dans eden arıları izleyerek ilgili besin kaynaklarına giden bir gözcü arı olabilir. Bu arıda R ile gösterilmiştir.

Besin kaynağını bulan arılar kovanlara nektar getirmeye başlarlar. Bu durumda bu arılar işçi arı haline gelmişlerdir. Bu arılar için bundan sonra üç durum söz konusudur. Bunlardan ilki; gittiği kaynağı bırakarak tekrar gözcü arı olma durumudur

ve UF ile gösterilmiştir. İkincisi; dans ederek besin kaynağına yönelmesi durumudur. Bu durum EF1 ile gösterilmiştir. Üçüncüsü ise EF2 ile gösterilen dans etmeden doğrudan besin kaynağına yönelmesi durumudur [49].



Şekil 3.4. Arıların besin kaynağı arama davranışları [99]

3.4.3. Yapay arı kolonisi algoritması

Yapay arı kolonisi algoritması, doğadaki arıların besin kaynağı arama davranışlarından ilham alınarak 2005 yılında Karaboğa ve arkadaşları tarafından geliştirilmiştir [49].

Yapay arı kolonisi algoritmasında arılar besin kaynağının bulunduğu alana bağlı olarak çalışırlar ve buldukları konumlar ile temsil edilirler. Bu konumlar problemin çözümü için gerekli olan parametrelerdir. Bu nedenle araştırma uzayının boyutunu parametre sayısı belirlemektedir. Arı kolonisi içerisinde en iyi konumda bulunan arı, o adımdaki en iyi çözümü ifade eder. Arıların buldukları konum

uygunluk fonksiyonu ile tanımlanır. Arılar bir önceki konumunu ve önceki adımlardaki iyi çözümlerin konumlarını da dikkate alarak araştırma uzayında optimum çözümü bulmaya çalışırlar [100].

Yapay arı kolonisi algoritmasının işlem adımları aşağıda görülmektedir:

1. Adım: İlk besin kaynaklarının oluşturulması

Yapay arı kolonisi algoritmasında besin kaynağının konumu optimizasyon problemi için olası bir çözümü ifade eder. Bu nedenle yapay arı kolonisi algoritmasına rastgele yiyecek kaynakları oluşturularak başlanır. Başlangıçtaki besin kaynakları parametrelerin sınırları arasında rastgele olarak üretilir. (Denklem 3.16)

$$x_{ij} = x_j^{\min} + \text{rand}(0,1)(x_j^{\max} - x_j^{\min}) \quad (3.16)$$

$$i = 1 \dots \dots SN \quad (3.17)$$

$$j = 1 \dots \dots \dots D \quad (3.18)$$

Denklem 3.16'da x_j^{\min} j. parametrenin alt sınırı ve x_j^{\max} ise j. parametrenin üst sınırıdır. Denklem 3.17'de SN besin kaynaklarının sayısı, Denklem 3.18'de ise D optimize edilecek parametre sayısını göstermektedir. Başlangıç kaynaklarına ait deneme sayaçları da (failure_i) bu aşamada sıfırlanmaktadır.

2. Adım: İşçi arıların besin kaynaklarına gönderilmesi

Her kaynakta bir işçi arı bulunmaktadır. Dolayısıyla kaynak sayısı işçi arı sayısına eşittir. İşçi arılar görevli oldukları kaynağın komşuluğunda olan bir kaynağı rastgele seçerek bu kaynağın kalitesini seçime tabi tutarlar. İşçi arıların bu davranışı Denklem 3.19'daki gibi formülize edilmiştir.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (3.19)$$

x_{ij} işçi arının görevli olduğu kaynağı, x_{kj} rastgele seçilen kaynağı, φ_{ij} ise $[-1,1]$ arasında üretilen rastgele sayıdır. Her bir x_i kaynağı için $[1,D]$ arasında rastgele seçilen j parametresinin değiştirilmesi ile v_{ij} kaynağı bulunur [101].

Denklem 3.19'da üretilen değerler parametreye ait alt ve üst sınırları aşarsa bu sınırları aşan değer belirli olan alt ve üst sınır değerlerine eşitlenir. Eğer $x_i > x_i^{\max}$ ise $x_i = x_i^{\max}$ ve $x_i < x_i^{\min}$ ise $x_i = x_i^{\min}$ olarak değiştirilir. Bu yapılan kontrolden sonra bulunan kaynağın uygunluk değeri en az problemleri için Denklem 3.20'deki gibi hesaplanmaktadır.

$$\text{fitness}_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0 \\ 1 + \text{abs}(f_i), & f_i < 0 \end{cases} \quad (3.20)$$

Bu denklemde f_i değeri v_i kaynağının maliyetidir. En çok problemlerinde ise uygunluk fonksiyonu f_i fonksiyonudur.

Kaynakların uygunluk değerleri hesaplandıktan sonra bunlar arasında açgözlü seçme işlemi uygulanır. Eğer bulunan v_i kaynağının uygunluk değeri mevcut x_i kaynağının uygunluk değerinden daha iyi ise işçi arı eski kaynağı (x_i) hafızasından siler ve yeni kaynağı (v_i) hafızasına alarak deneme sayacını sıfırlar. Aksi halde x_i hafızada kalmaya devam eder ve deneme sayacı arttırılır [102].

3. Adım: Gözcü arılar için besin kaynaklarının olasılıklarının hesaplanması

Kovana geri dönen işçi arılar buldukları kaynaklarla ilgili bilgileri dans alanında gözcü arılarla paylaşırlar. Gözcü arılar tüm bu bilgileri değerlendirir ve besin kaynağının miktarıyla ilişkili olarak bir kaynak seçerler. Uygunluk tabanlı bu seçimde rulet çarkı yöntemi kullanılır. Bir kaynağın uygunluk değerinin, diğer bütün kaynakların uygunluk değerlerinin toplamına oranı olasılık değeri olarak belirlenmiştir ve p_i ile gösterilmektedir. (Denklem 3.21)

$$p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i} \quad (3.21)$$

Denklem 3.21'de fitness_i uygunluk deęerini (kaynaęın kalitesini), SN ise kaynak sayısını göstermektedir.

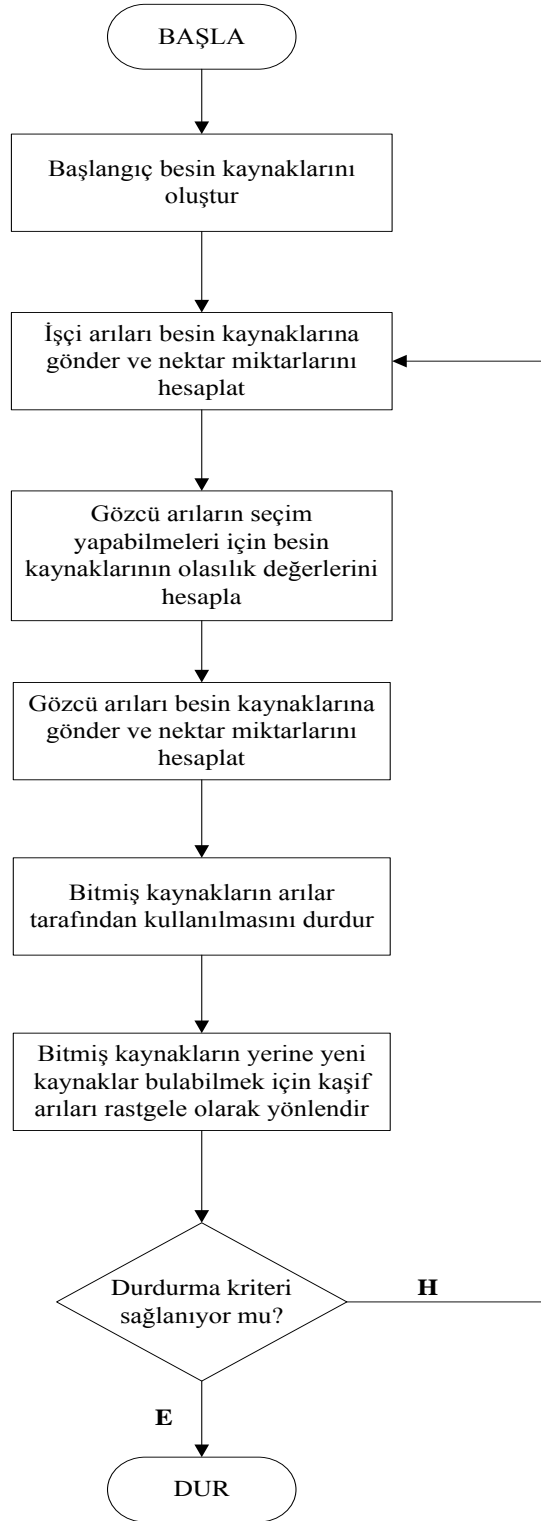
4. Adım: Gözcü arıların besin kaynaęı seçme işlemi

Yapay arı kolonisi algoritmasının bu aşamasında her kaynak için [0,1] arasında rastgele bir sayı üretilir. Bir önceki adımda hesaplanan p_i olasılık deęeri üretilen rastgele sayıdan büyükse gözcü arı yeni bir çözüm üreterek kalitesini hesaplar. Eğer yeni çözüm mevcut çözümden daha iyi ise gözcü arı eski çözümü hafızadan siler ve deneme sayacını sıfırlar. Diğer durumda ise mevcut çözüm kullanılır ve deneme sayacı bir arttırılır.

5. Adım: Besin kaynaęının bırakılması ve kâşif arı üretilmesi

Arıların yaptığı faaliyetlerin döngüsü tamamlandıktan sonra kaynaęın bitip bitmedięine bakılır. Kaynaęın bitip bitmedięi deneme sayacı "limit" parametresi olarak tanımlanan deęerden büyükse bu kaynaęın artık bitmiş olduęuna karar verilir ve kaynak bırakılır. Görevli arı besin kaynaęını bıraktıktan sonra besin kaynaęı arayacak olan kâşif arıya dönüşür [101].

Yapay arı kolonisi algoritmasının akış diyagramı Şekil 3.5'te gösterilmiştir.



Şekil 3.5. Yapay arı kolonisi algoritmasının akış diyagramı

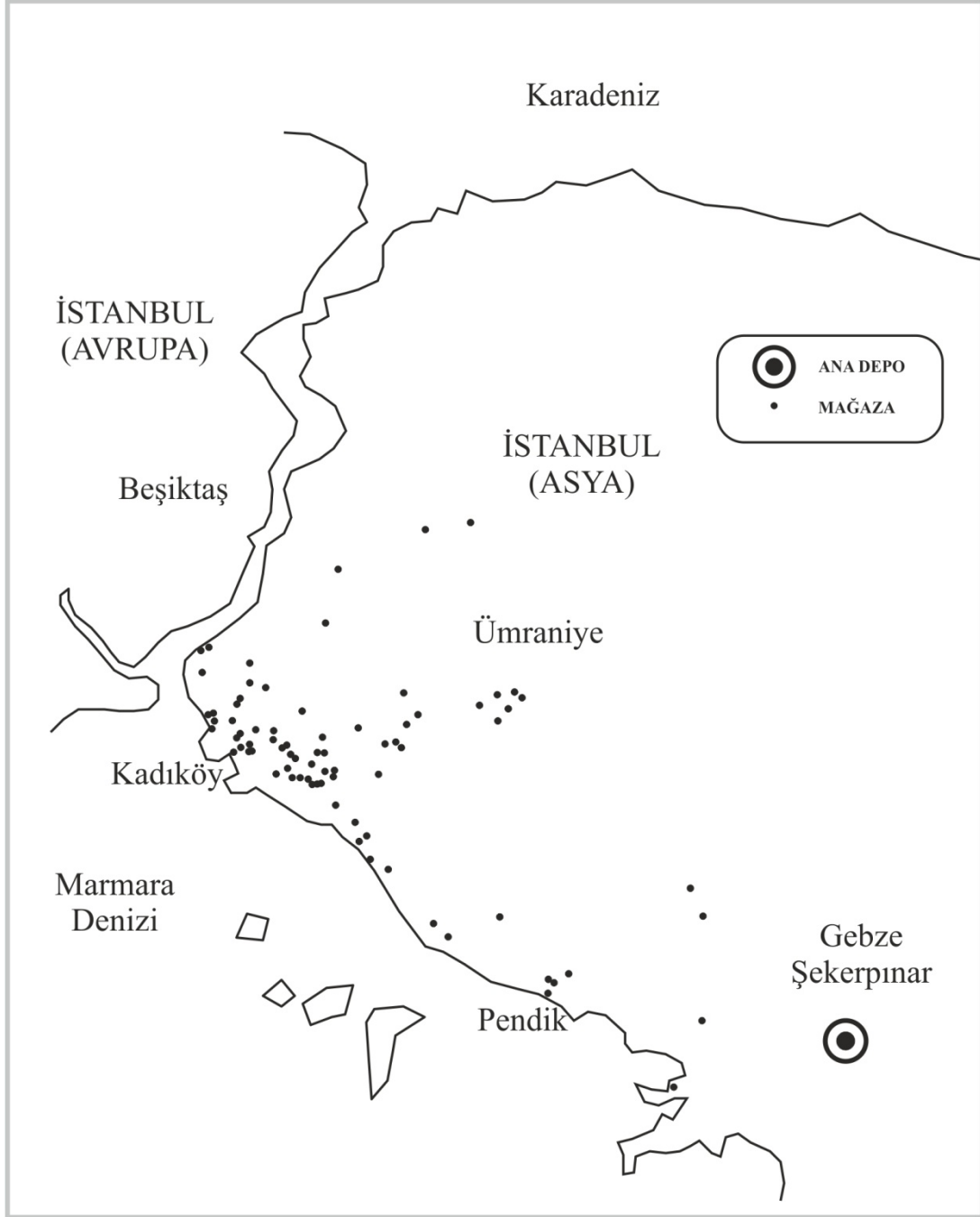
BÖLÜM 4. BİR SÜPERMARKET ZİNCİRİNDE ROTALAMA PROBLEMİNİN METASEZGİSEL ALGORTİMALAR İLE ÇÖZÜLMESİ

4.1. Problemin Tanımı

Bu çalışmada bir süpermarket zinciri ele alınarak haftalık taleplerinin karşılanabilmesi için en uygun araç rotalarının belirlenmesi problemi çözülmüştür.

Ele alınan süpermarket zincirinde mağazalara yapılan dağıtım sadece hafta içi günlerde gerçekleştirilmektedir. Dağıtımı gerçekleştiren araç filosu homojen araçlardan oluşmaktadır. Araçların hepsi maksimum 40 palet kapasitelidir ve araçlara 32 ile 40 palet arasında yükleme yapılmalıdır. Bu dağıtım filusunda iki çeşit kamyon mevcuttur. Bunlar taze ürün ve kuru ürün kamyonlarıdır. Taze ürün kamyonları soğuk ortamda muhafaza edilmesi gereken ürün gruplarını taşıyan kamyonlardır. Kuru ürün kamyonları ise klimasız (soğutucusuz) olduğundan taze ürün grubunda olmayan ürünleri taşımaktadır.

Bu çalışmada bir adet ana depo ve ürün dağıtımının yapılacağı 78 adet mağaza mevcuttur. Şekil 4.1'de ana deponun ve mağazaların haritadaki görüntüsü gösterilmektedir.



Şekil 4.1. Ana depo ve mağazaların haritadaki gösterimi

Ürün dağıtımının yapıldığı 78 adet mağaza kendi içinde ikiye ayrılmaktadır. Bunlardan ilki hafta içi her gün taze ve kuru ürün talep eden mağazalardır ve günlük mağazalar olarak adlandırılmaktadır. İkincisi ise sadece haftanın üç günü (pazartesi, çarşamba ve cuma) taze ve kuru ürün talep eden talep eden mağazalardır ve yedek mağazalar olarak adlandırılmaktadır. 78 adet mağazadan 66 adeti günlük, 12 adeti ise yedek mağazadır. Mağazaların numaraları ve çeşitleri Tablo 4.1'de gösterilmiştir.

Tablo 4.1. Mağazaların numaraları ve çeşitleri

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

Çalışmada günlük ve yedek mağazalara dağıtım yapılan günler yoğun gün, sadece günlük mağazalara dağıtım yapılan günler ise sakin gün olarak belirtilmiştir. Bütün bu bilgilerden yola çıkacak olursak, hem yedek hem de günlük olmak üzere toplam 78 mağazaya dağıtım yapıldığı için pazartesi, çarşamba ve cuma günleri yoğun günlerdir. Salı ve perşembe günleri ise sadece günlük mağazalara dağıtım yapılmaktadır. Bu nedenle salı ve perşembe günleri sakin günlerdir.

Ürün dağıtımının yapıldığı günlere göre 4 adet alt problem ortaya çıkmaktadır. Bu alt problemler şunlardır:

- Alt problem 1: Taze ürünün yoğun günde dağıtım problemi
- Alt problem 2: Kuru ürünün yoğun günde dağıtım problemi
- Alt problem 3: Taze ürünün sakin günde dağıtım problemi
- Alt problem 4: Kuru ürünün sakin günde dağıtım problemi

Bu ortaya çıkan 4 adet alt problemin her biri için 21 haftalık talep verileri mevcuttur. Bu talep verileri EK-1'de verilmiştir. Alt problemlerin her birinin çözümü için karınca kolonisi, genetik ve yapay arı kolonisi algoritmaları kullanılmıştır.

4.2. Üç Farklı Algoritma ile Problemin Çözümü

Alt problemlerin çözüm aşamaları Tablo 4.2'de özetlenmiştir.

Tablo 4.2. Önerilen yöntemin çözüm aşamaları

Aşama 1	Karınca kolonisi, genetik ve yapay arı kolonisi algoritması ile gezgin satıcı probleminin çözülmesi
Aşama 2	Haftalık talepler ve araç kapasiteleri dikkate alınarak kümeleme yapılması
Aşama 3	Araç rotalama probleminin dal ve sınır yöntemi ile çözülmesi

Aşama 1: Karınca kolonisi, genetik ve yapay arı kolonisi algoritmaları kullanılarak, 78 mağazaya sahip yoğun gün ve 66 mağazaya sahip sakin gün için ayrı ayrı gezgin satıcı problemi uygulanır. Yani bir mağazadan harekete başlayıp her mağazaya sadece bir defa uğradıktan sonra başladığı mağazaya geri dönen en kısa tur oluşturulur.

- Karınca kolonisi algoritması için GSP çözümü;

1. Adım: Yapılan denemeler sonucunda feromon miktarının önemini belirleyen sabit olan α parametresi 1, sezgisel bilginin önemini belirleyen sabit olan β parametresi 5 ve buharlaşma oranını gösteren ρ parametresi ise 0,65 olarak belirlenir.

2. Adım: Karıncalar her şehre rastsal olarak yerleştirilir.

3. Adım: Denklem 3.12'deki seçilme olasılığı ile bir sonraki gidilecek şehir belirlenir ve tur tamamlanır.

4. Adım: Tamamlanan turların toplam mesafeleri hesaplanır ve Denklem 3.13'e göre lokal feromon güncellemesi yapılır.

5. Adım: En iyi çözüm hesaplanır ve Denklem 3.14'e göre global feromon güncellemesi yapılır.

6. Adım: 10000 iterasyon sonunda yapılan denemelerde iyileştirme görülmediği için algoritma durdurulur.

- Genetik algoritma için GSP çözümü;

1. Adım: Denklem 3.15'e göre popülasyon büyüklüğü hesaplanarak başlangıç popülasyonu oluşturulur.

2. Adım: Popülasyondaki çözümlere bakılarak toplam mesafeler hesaplanır ve en az mesafeye sahip çözüm seçilir.

3. Adım: %80 çaprazlama oranı ile çaprazlama işlemi gerçekleştirilir. (Çaprazlama oranı yapılan denemeler sonucunda %80 olarak belirlenmiştir.)

4. Adım: % 0,1 mutasyon oranı ile mutasyon işlemi gerçekleştirilir. (Mutasyon oranı yapılan denemeler sonucunda %0,1 olarak belirlenmiştir.)

5. Adım: Yeni oluşan çözümlere göre tekrardan toplam mesafeler hesaplanır ve en az mesafeye sahip çözüm seçilir.

6. Adım: 10000 iterasyon sonunda yapılan denemelerde iyileştirme görülmediği için algoritma durdurulur.

- Yapay arı kolonisi algoritması için GSP çözümü;

1. Adım: Yapay arı kolonisi algoritmasında D optimize edilecek parametre sayısı yani problemin boyutudur. 78 mağazalı alt problemler için 78, 66 mağazalı alt problemler

için 66 olarak alınmıştır. SN besin kaynağını temsil etmektedir, bizim ele aldığımız problemde müşteri sayısıdır. Bu parametre de aynı şekilde 78 mağazalı alt problemler için 78, 66 mağazalı alt problemler için 66 olarak alınır. Burada 78 mağazalı alt problem için 39 tane işçi arı 39 tane de gözcü arı çalışacaktır. 66 mağazalı alt problem için ise 33 tane işçi arı 33 tane de gözcü arı çalışacaktır. Yapay arı kolonisi algoritmasında durdurma kriteri iterasyon sayısıdır ve 10000 olarak alınmıştır. Limit parametresi ise iterasyon sayısı/50=200 olarak belirlenmiştir.

2. Adım: Denklem 3.16'ya göre başlangıç besin kaynakları oluşturulur ve Denklem 3.19'a göre tüm işçi arılar besin kaynaklarına gönderilir. Her işçi arı komşuluğundaki bir besin kaynağını rastgele seçerek işlemeye başlar. Yani seçtiği besin kaynağının komşuluğundaki kaynakları belirleyerek bu besin kaynağının kalitesini seçime tabi tutar. Besin kaynağı işlendikten sonra bu besine ait yeni bir besin kalitesi (çözüm değeri) hesaplanır. Ele alınan problemde müşteriler ziyaret edildiğinde oluşan toplam yol uzunluğu besin kaynağının kalitesini temsil etmektedir Hesaplanan bu değer önceki değerlerden daha iyi ise bu besin kaynağı ile ilgili bilgiler hafızaya alınır ve limit değeri sıfırlanır.

3. Adım: İşçi arılardan sonra gözcü arılar devreye girerler. Denklem 3.20'ye göre hesaplanan besinlerin uygunluk değerine göre bir besin kaynağı seçilir. Gözcü arılar bu besin kaynağı üzerinde çalışmaya başlarlar. Aynı şekilde elde edilen çözüm değeri önceki çözüm değerinden daha iyi ise bu besin ve besinle ilgili bilgiler hafızaya alınırlar. Eğer besin kaynağında iyileşme sağlanırsa limit değeri sıfırlanır aksi takdirde limit değeri bir arttırılır. Bu safhada gözcü arılar işçi arılardan farklı olarak uygunluk değerine göre seçim yaparlar.

4. Adım: İşçi arı ve gözcü arılardan sonra devreye kâşif arı girer. Bu adımın esas nedeni algoritmanın yerel minimum ya da maksimumda takılmasına engel olmaktır. Dolayısıyla elde edilmiş olan çözümü bozarak yani limit değerlerini tamamen sıfırlayarak yeni bir çözüm değerinin elde edilmesini sağlar. Elde edilen çözüm değeri ile önceden hafızaya alınmış olan çözüm değeri karşılaştırılır. Bu iki çözüm değerinden iyi olanının hafızada tutulmasına devam edilir.

5. Adım: 10000 iterasyon sonunda yapılan denemelerde iyileştirme görülmediği için algoritma durdurulur.

Aşama 2: Taze ürünün yoğun günde dağıtımı, kuru ürünün yoğun günde dağıtımı, taze ürünün sakin günde dağıtımı ve kuru ürünün sakin günde dağıtımı alt problemlerinin haftalık talepleri göz önüne alınarak, elde edilen rotadaki ilk mağazadan başlanıp kamyonların palet kapasite kısıtını sağlayacak şekilde kümeler oluşturulur.

Aşama 3: Oluşturulan kümelerin her birinin rotaları ana depodan başlayıp ve yine ana depoda son bulacak şekilde dal ve sınır yöntemiyle oluşturulur ve kat edilen mesafeler hesaplanır.

Taze ürünün yoğun günde dağıtımı alt probleminin 1. haftadaki talep miktarları ele alınarak karınca kolonisi, genetik ve yapay arı kolonisi algoritmalarıyla çözümü detayları ile aşağıda verilmiştir. Diğerlerinin sonuçları özet şeklinde Tablo 4.13, 4.17, 4.20 ve 4.25’de verilmiştir.

- Karınca kolonisi algoritması ile çözümü;

Aşama 1: Ele alınan problem yoğun gün problemi olduğu için mağaza sayısı 78dir. Bu aşamada karınca kolonisi algoritması kullanılarak 78 adet mağazaya gezgin satıcı problemi uygulanmış ve bir mağazadan harekete başlayıp yine aynı mağazada sonlanan bir rota elde edilmiştir. Elde edilen rota Tablo 4.3'te gösterilmektedir:

Tablo 4.3. Karınca kolonisi algoritması ile elde edilen rota

Düğüm	Mağaza Numarası	Düğüm	Mağaza Numarası	Düğüm	Mağaza Numarası	Düğüm	Mağaza Numarası
1	73	21	64	41	37	61	45
2	67	22	65	42	32	62	42
3	71	23	61	43	35	63	22
4	69	24	59	44	47	64	41
5	2	25	62	45	15	65	44
6	6	26	63	46	26	66	38
7	5	27	60	47	29	67	25
8	4	28	68	48	19	68	21

Tablo 4.3. Karınca kolonisi algoritması ile elde edilen rota (Devamı)

9	1	29	70	49	28	69	16
10	55	30	11	50	39	70	48
11	30	31	66	51	31	71	27
12	56	32	12	52	33	72	40
13	52	33	10	53	76	73	43
14	57	34	7	54	3	74	24
15	54	35	9	55	36	75	13
16	53	36	8	56	23	76	74
17	51	37	75	57	46	77	14
18	50	38	77	58	20	78	78
19	49	39	72	59	17	79	73
20	58	40	34	60	18		

Aşama 2: Taze ürünün yoğun günde dağıtım alt probleminin 1. haftadaki talepleri Tablo 4.4'te gösterilmiştir.

Tablo 4.4. Taze ürünün yoğun günde dağıtım alt probleminin 1. haftadaki talepleri

Müşteri Numarası	Talep Miktarı (Palet)	Müşteri Numarası	Talep Miktarı (Palet)	Müşteri Numarası	Talep Miktarı (Palet)
1	5	27	3	53	5
2	7	28	5	54	9
3	9	29	5	55	5
4	7	30	3	56	5
5	9	31	5	57	7
6	3	32	5	58	5
7	7	33	3	59	7
8	9	34	5	60	7
9	7	35	5	61	5
10	5	36	3	62	7
11	5	37	5	63	9
12	7	38	5	64	5
13	5	39	3	65	5
14	5	40	5	66	5
15	7	41	5	67	5
16	5	42	3	68	9
17	5	43	5	69	5
18	7	44	5	70	7
19	5	45	3	71	3
20	5	46	5	72	5
21	9	47	7	73	7
22	5	48	7	74	3
23	5	49	5	75	5

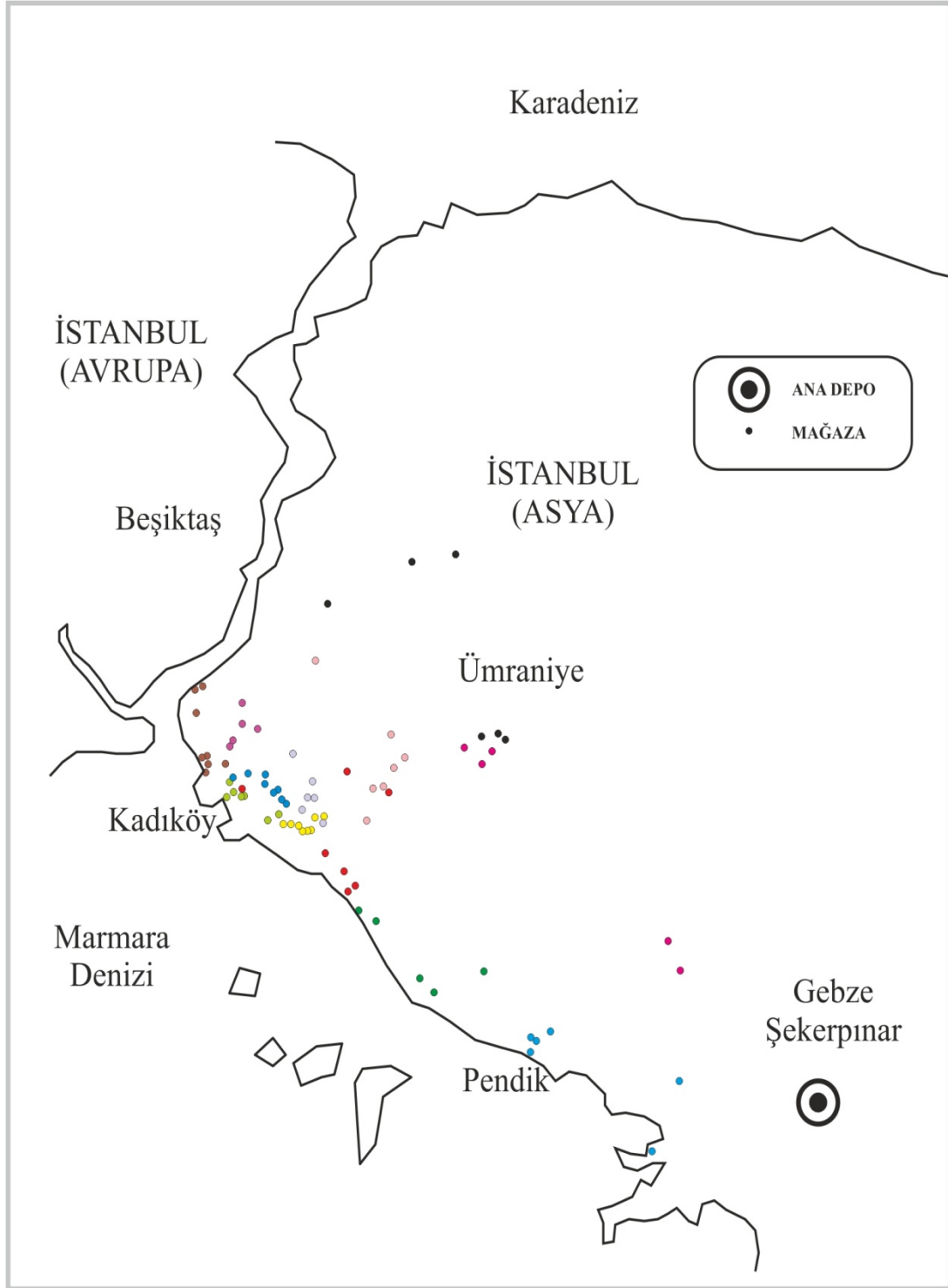
Tablo 4.4. Taze ürünün yoğun günde dağıtım alt probleminin 1. haftadaki talepleri (Devamı)

24	3	50	9	76	9
25	5	51	9	77	3
26	5	52	5	78	7

Tablo 4.4'teki 1. haftanın talepleri göz önüne alınarak, oluşturulan rotadaki ilk mağaza olan 73 numaraları mağazadan başlanıp kamyonların 32-40 palet kısıtı sağlanacak şekilde kümeler oluşturulmuştur. Bu kümeler Tablo 4.5'te gösterilmiştir. Burada her araç bir kümeyi temsil etmektedir. Oluşturulan kümelerin haritadaki gösterimi ise Şekil 4.2'de verilmiştir. Bu şekilde her renk bir kümeyi temsil etmektedir.

Tablo 4.5. KKA sonucunda oluşan kümeler ve uğranacak mağazalar

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



Şekil 4.2. KKA ile oluşturulan kümelerin haritadaki gösterimi

Şekil 4.2’de 12 farklı renk bulunmakta ve her renk bir kümeyi temsil etmektedir.

Aşama 3: Problemin çözümünün bu aşamasında ise oluşturulan bu araçların her birinin izleyeceği rotalar, ana depodan başlayıp ana depoda son bulacak şekilde dal

ve sınır yöntemiyle oluşturulmuştur. Araçların izleyeceği rotalar ve kat ettikleri mesafeler Tablo 4.6'da gösterilmiştir. Bu oluşturulan rotalardaki 79 numaralı mağaza ana depodur.

Tablo 4.6. KKA sonucunda araçların izleyeceği rotalar ve kat ettikleri mesafeler

Araçlar	Araçların İzleyeceği Rotalar	Toplam Km
1	79-69-71-67-73-2-6-5-79	92,3
2	79-4-1-55-30-56-52-57-79	88,1
3	79-49-50-51-53-54-79	81,4
4	79-64-58-61-59-62-65-79	60
5	79-60-68-70-11-63-79	82,4
6	79-10-66-12-7-9-8-79	124,3
7	79-47-35-32-37-34-72-75-77-79	105,7
8	79-39-28-19-29-15-26-33-31-79	92,4
9	79-36-3-76-46-23-20-79	91,2
10	79-17-18-45-42-22-41-38-44-79	90,2
11	79-25-21-16-48-27-40-43-79	96,6
12	79-78-14-24-13-74-79	96,6
TOPLAM KM		1101,2

- Genetik algoritma ile çözümü;

Aşama 1: İlk aşamada genetik algoritma kullanılarak 78 adet mağazaya gezgin satıcı problemi uygulanmış ve bir mağazadan harekete başlayıp yine aynı mağazada sonlanan bir rota elde edilmiştir. Elde edilen rota Tablo 4.7'de gösterilmektedir:

Tablo 4.7. Genetik algoritma ile elde edilen rota

Düğüm	Mağaza Numarası	Düğüm	Mağaza Numarası	Düğüm	Mağaza Numarası	Düğüm	Mağaza Numarası
1	30	21	5	41	46	61	13
2	56	22	6	42	20	62	74
3	54	23	2	43	17	63	14
4	51	24	69	44	18	64	78
5	50	25	71	45	45	65	75
6	49	26	67	46	42	66	77
7	58	27	70	47	22	67	72
8	65	28	68	48	41	68	34
9	64	29	11	49	44	69	37
10	60	30	10	50	38	70	32
11	63	31	66	51	25	71	35

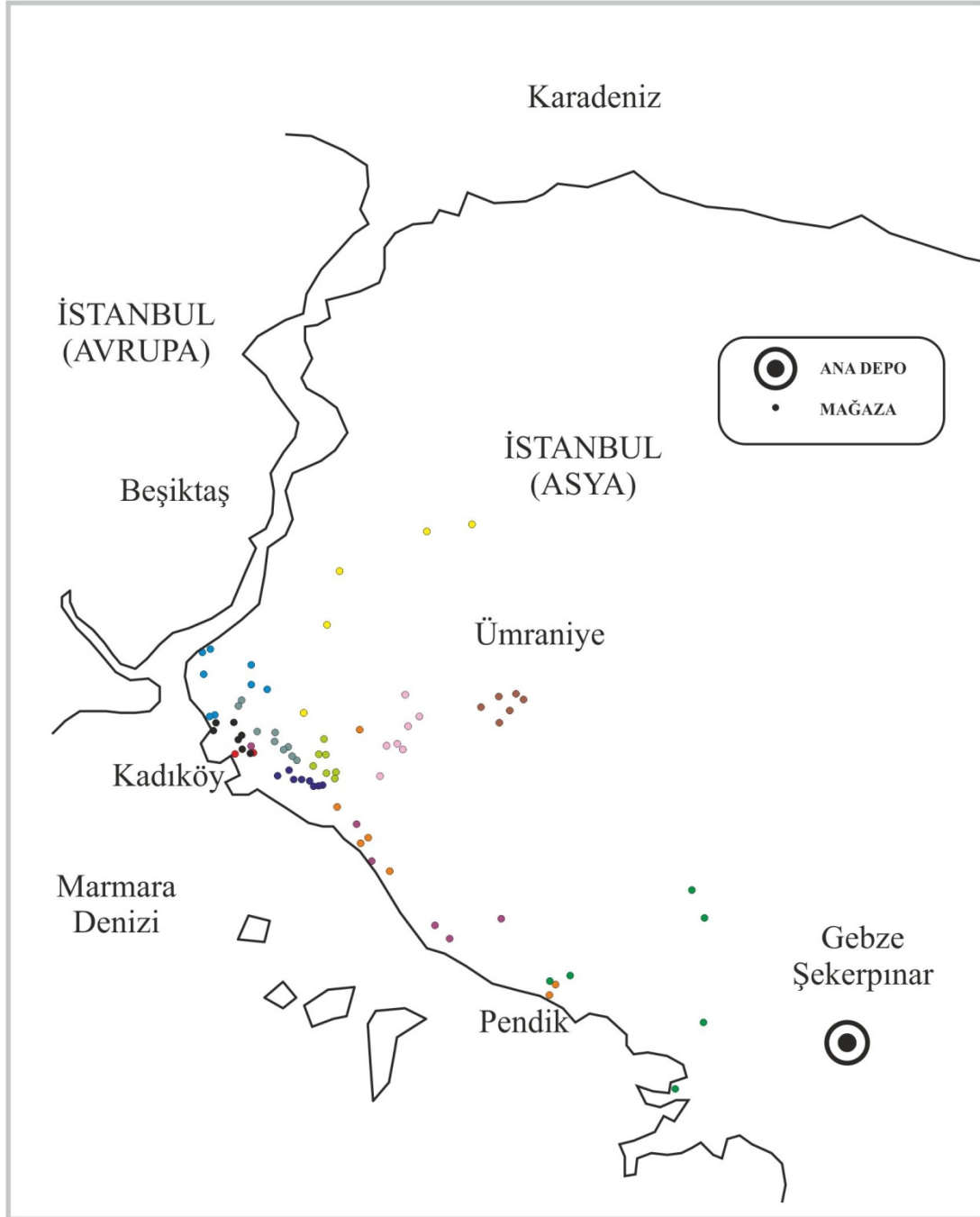
Tablo 4.7. Genetik algoritma ile elde edilen rota (Devamı)

12	62	32	12	52	21	72	47
13	59	33	7	53	39	73	15
14	61	34	9	54	28	74	43
15	53	35	8	55	33	75	48
16	57	36	73	56	31	76	27
17	52	37	76	57	19	77	40
18	55	38	3	58	29	78	16
19	1	39	36	59	26	79	30
20	4	40	23	60	24		

Aşama 2: Bu aşamada Tablo 4.4'teki Taze ürünün yoğun günde dağıtımını alt probleminin 1. haftadaki talepleri göz önüne alınarak, oluşturulan rotadaki ilk mağaza olan 30 numaraları mağazadan başlanıp kamyonların 32-40 palet kısıtı sağlanacak şekilde kümeler oluşturulmuştur. Bu kümeler Tablo 4.8'de gösterilmiştir. Burada her araç bir kümeyi temsil etmektedir. Oluşturulan kümelerin haritadaki gösterimi ise Şekil 4.3'te verilmiştir. Bu şekilde her renk bir kümeyi temsil etmektedir.

Tablo 4.8. GA sonucunda oluşan kümeler ve uğranacak mağazalar

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



Şekil 4.3. GA ile oluşturulan kümelerin haritadaki gösterimi

Şekil 4.3'te 12 farklı renk bulunmakta ve her renk bir kümeyi temsil etmektedir.

Aşama 3: Son aşamada ise oluşturulan bu araçların her birinin izleyeceği rotalar, ana depodan başlayıp ana depoda son bulacak şekilde dal ve sınır yöntemiyle oluşturulmuştur. Araçların izleyeceği rotalar ve kat ettikleri mesafeler Tablo 4.9'da gösterilmiştir. Bu oluşturulan rotalardaki 79 numaralı mağaza ana depodur.

Tablo 4.9. GA sonucunda araçların izleyeceği rotalar ve kat ettikleri mesafeler

Araçlar	Araçların İzleyeceği Rotalar	Toplam Km
1	79-56-30-54-51-50-49-79	85,1
2	79-60-63-62-58-64-65-79	63,8
3	79-1-55-52-57-53-59-61-79	88,4
4	79-69-71-67-2-6-5-4-79	87,5
5	79-70-10-12-66-11-68-79	83,1
6	79-76-73-7-9-8-79	129,8
7	79-20-3-36-23-46-18-17-79	93
8	79-25-21-38-44-45-42-22-41-79	91,6
9	79-31-33-29-26-24-13-19-28-39-79	95,7
10	79-74-14-78-77-75-72-37-34-79	107,3
11	79-15-27-48-43-47-35-32-79	99,6
12	79-40-16-79	96,9
TOPLAM KM		1121,8

- Yapay arı kolonisi algoritması ile çözümü;

Aşama 1: İlk aşamada yapay arı kolonisi algoritması kullanılarak 78 adet mağazaya gezgin satıcı problemi uygulanmış ve bir mağazadan harekete başlayıp yine aynı mağazada sonlanan bir rota elde edilmiştir. Elde edilen rota Tablo 4.10'da gösterilmektedir:

Tablo 4.10. Yapay arı kolonisi algoritması ile elde edilen rota

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

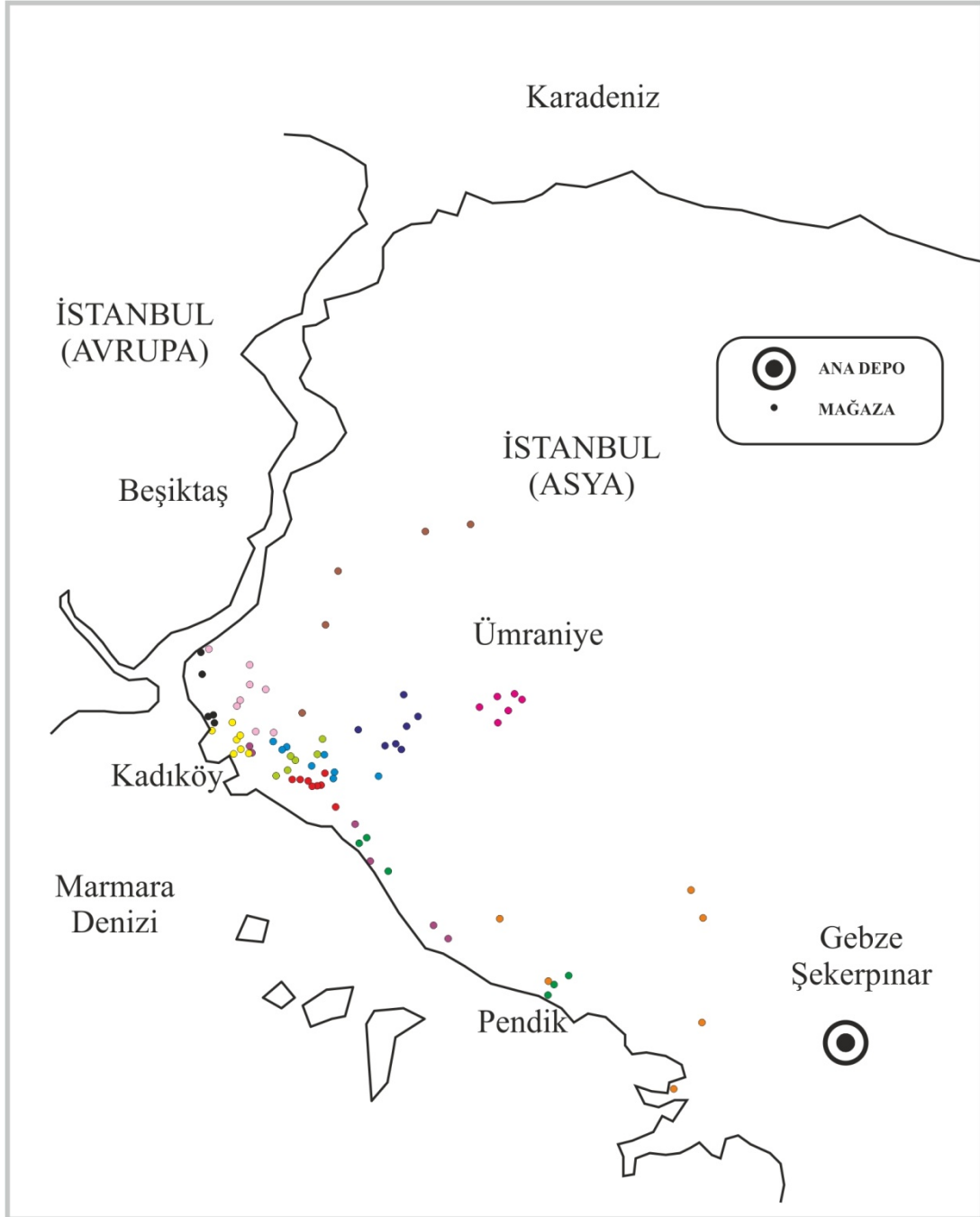
Tablo 4.10. Yapay arı kolonisi algoritması ile elde edilen rota (Devamı)

17	64	37	39	57	6	77	34
18	60	38	36	58	2	78	37
19	63	39	3	59	20	79	35
20	62	40	76	60	17		

Aşama 2: Bu aşamada Tablo 4.4'teki taze ürünün yoğun günde dağıtımını alt probleminin 1. haftadaki talepleri göz önüne alınarak, oluşturulan rotadaki ilk mağaza olan 35 numaraları mağazadan başlanıp kamyonların 32-40 palet kısıtı sağlanacak şekilde kümeler oluşturulmuştur. Bu kümeler Tablo 4.11'de gösterilmiştir. Burada her araç bir kümeyi temsil etmektedir. Oluşturulan kümelerin haritadaki gösterimi ise Şekil 4.4'te verilmiştir. Bu şekilde her renk bir kümeyi temsil etmektedir.

Tablo 4.11. YAKA sonucunda oluşan kümeler ve uğranacak mağazalar

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



Şekil 4.4. YAKA ile oluşturulan kümelerin haritadaki gösterimi

Şekil 4.4'te 12 farklı renk bulunmakta ve her renk bir kümeyi temsil etmektedir.

Aşama 3: Son aşamada ise oluşturulan bu araçların her birinin izleyeceği rotalar, ana depodan başlayıp ana depoda son bulacak şekilde dal ve sınır yöntemiyle oluşturulmuştur. Araçların izleyeceği rotalar ve kat ettikleri mesafeler Tablo 4.12'de gösterilmiştir. Bu oluşturulan rotalardaki 79 numaralı mağaza ana depodur.

Tablo 4.12. YAKA sonucunda araçların izleyeceği rotalar ve kat ettikleri mesafeler

Araçlar	Araçların İzleyeceği Rotalar	Toplam Km
1	79-15-40-27-48-43-47-35-79	99,7
2	79-56-16-30-54-51-50-79	89,6
3	79-64-65-49-58-63-60-79	72,9
4	79-61-62-59-53-57-52-79	86,4
5	79-55-42-22-41-38-44-45-18-79	92,4
6	79-25-21-36-28-3-36-79	93,6
7	79-76-73-7-9-8-79	129,8
8	79-70-10-12-66-11-68-79	83,1
9	79-69-67-71-6-5-4-1-79	88,1
10	79-2-20-17-23-46-19-33-31-79	93,1
11	79-78-75-14-74-13-24-29-26-79	103,6
12	79-32-37-34-72-77-79	102,4
TOPLAM KM		1134,7

Bu çalışmada her bir alt problem için 21 haftalık talep verileri mevcuttur. Diğer haftalar içinde aynı aşamalar tekrarlanmış ve her bir alt problem için 21 adet sonuç elde edilmiştir.

4.3. Sonuçların ANOVA Testi ile Yorumlanması

4.3.1. Taze ürünün yoğun günde dağıtım alt problemi için ANOVA testi

Taze ürünün yoğun günde dağıtım alt probleminde her hafta 78 mağazaya dağıtım yapılmaktadır. Her mağazanın 21 haftalık talep verileri mevcuttur. Mağazalar tarafından her hafta talep edilen palet miktarları 3-9 aralığında değişmekte ve toplamları 438 palet olmaktadır.

Bu alt problem karınca kolonisi, genetik ve yapay arı kolonisi algoritması ile çözüldüğünde elde edilen sonuçlar Tablo 4.13'te verilmiştir.

Tablo 4.13. Taze ürünün yoğun günde dağıtım alt problemi için KKA, GA ve YAKA sonuçları

Talepler	Araç Sayıları	KKA ile Bulunan Toplam Mesafe (km)	GA ile Bulunan Toplam Mesafe (km)	YAKA ile Bulunan Toplam Mesafe (km)
1	12	1101,2	1121,8	1134,7
2	12	1100,8	1116,5	1151,1
3	12	1140,7	1159,6	1152
4	12	1157,7	1137,3	1222,2
5	12	1120,9	1129,3	1131,7
6	12	1163,3	1134,7	1152,6
7	12	1106,5	1128,4	1152
8	12	1141,2	1128,7	1149,2
9	12	1107,8	1148,7	1145,6
10	12	1175	1142,1	1137,8
11	12	1139,4	1147	1136,4
12	12	1143,7	1164,7	1152,5
13	12	1114,2	1110,1	1116,6
14	12	1095,4	1115,8	1152,5
15	12	1142,9	1160,3	1145,5
16	12	1114,7	1129	1153,4
17	12	1112,8	1120,4	1129,3
18	12	1119,1	1123	1146,4
19	12	1112,2	1120	1153,4
20	12	1134,4	1136,2	1141
21	12	1116,8	1122,2	1120

Tablo 4.13'teki sonuçlara ANOVA testi uygulanmış ve SPSS sonuçları aşağıda verilmiştir. Bu çalışmada;

$H_0: \mu_1 = \mu_2 = \mu_3$ (Üç algoritma ile bulunan toplam mesafelerin ortalaması arasında fark yoktur.)

H_1 : En az bir algoritma ile bulunan toplam mesafelerin ortalaması farklıdır.

Olacak şekilde hipotezler tanımlanmıştır. ANOVA testi uygulandığında ilk olarak varyansların homojen olup olmadığı incelenmektedir. Tablo 4.14'teki p (sig.) değeri $0,151 > 0,05$ olduğundan dolayı grup varyanslarının homojen olduğu sonucuna varılır. Grup varyansları homojen olduğu için Tablo 4.15'teki F testinin sonuçları anlamlı olacaktır. Varyans analizi tablosundaki p (sig.) değeri $0,006 < 0,05$ olduğundan dolayı H_1 hipotezi kabul edilir. H_1 hipotezi kabul edildiği için üç algoritma ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir fark vardır diyebiliriz. Tablo 4.16'da

taze ürünün yoğun günde dağıtım alt probleminin üç algoritma ile çözümü sonucunda bulunan toplam mesafelerin ortalaması verilmiştir. Toplam mesafelerin ortalaması karınca kolonisi algoritması ile 1126,7 km, genetik algoritma ile 1133,133 km ve yapay arı kolonisi algoritması ile 1146,471 km olarak bulunmuştur. Bu sonuçlar bize karınca kolonisi algoritmasının diğer iki algoritmaya göre daha iyi sonuç verdiğini göstermektedir.

Tablo 4.14. Taze ürünün yoğun günde dağıtım alt problemi için varyansların homojenliği testi

Levene Statistic	df1	df2	Sig.
1,954	2	60	,151

Tablo 4.15. Taze ürünün yoğun günde dağıtım alt problemi için varyans analizi tablosu

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4271,414	2	2135,707	5,547	,006
Within Groups	23100,810	60	385,013		
Total	27372,223	62			

Tablo 4.16. Taze ürünün yoğun günde dağıtım alt problemi için tanımlayıcı istatistikler

	N	Mean	Std. Deviation	Std. Error
1	21	1126,700	22,1030	4,8233
2	21	1133,133	15,5659	3,3968
3	21	1146,471	20,5961	4,4944
Total	63	1135,435	21,0116	2,6472
Model				
Fixed Effects			19,6218	2,4721
Random Effects				5,8224

4.3.2. Kuru ürünün yoğun günde dağıtım alt problemi için ANOVA testi

Kuru ürünün yoğun günde dağıtım alt probleminde her hafta 78 mağazaya dağıtım yapılmaktadır. Her mağazanın 21 haftalık talep verileri mevcuttur. Mağazalar tarafından her hafta talep edilen palet miktarları 6-14 aralığında değişmekte ve toplamları 724 palet olmaktadır.

Bu alt problem karınca kolonisi, genetik ve yapay arı kolonisi algoritması ile çözüldüğünde elde edilen sonuçlar Tablo 4.17'de verilmiştir.

Tablo 4.17. Kuru ürünün yoğun günde dağıtımı alt problemi için KKA, GA ve YAKA sonuçları

Talepler	Araç Sayısı	KKA ile Bulunan Toplam Mesafe (km)	Araç Sayısı	GA ile Bulunan Toplam Mesafe (km)	Araç Sayısı	YAKA ile Bulunan Toplam Mesafe (km)
1	21	1843,1	21	1840	20	1764
2	20	1780	21	1857,8	21	1860,5
3	21	1844,6	21	1835,7	21	1865,2
4	21	1873,6	21	1853,3	21	1879,7
5	21	1878,3	20	1778,2	20	1773,9
6	21	1847,7	21	1855,9	21	1871,9
7	20	1753,1	20	1761,8	20	1786,8
8	20	1743,8	21	1850,4	21	1834
9	21	1878	21	1862,3	20	1812,1
10	20	1776	21	1853,5	21	1867,8
11	21	1856,9	22	1962,4	21	1866,2
12	21	1862,3	21	1876,3	20	1790,9
13	21	1838	21	1859,3	21	1860,8
14	20	1780,1	21	1860,7	21	1862,6
15	20	1791,9	20	1783	20	1791
16	21	1851,5	20	1786,5	21	1866
17	21	1847,8	21	1874,4	21	1876,3
18	21	1846,3	21	1883,2	20	1774,9
19	21	1867	21	1861,2	21	1872,3
20	21	1844,4	21	1835,7	21	1852,9
21	21	1849,1	21	1862,3	21	1878

Tablo 4.17'deki sonuçlara ANOVA testi uygulanmış ve SPSS sonuçları aşağıda verilmiştir. Bu çalışmada;

$H_0: \mu_1=\mu_2=\mu_3$ (Üç algoritma ile bulunan toplam mesafelerin ortalaması arasında fark yoktur.)

H_1 : En az bir algoritma ile bulunan toplam mesafelerin ortalaması farklıdır.

Olacak şekilde hipotezler tanımlanmıştır. ANOVA testi uygulandığında ilk olarak varyansların homojen olup olmadığı incelenmektedir. Tablo 4.18'deki p (sig.) değeri $0,655 > 0,05$ olduğundan dolayı grup varyanslarının homojen olduğu sonucuna varılır. Grup varyansları homojen olduğu için Tablo 4.19'daki F testinin sonuçları anlamlı olacaktır. Varyans analizi tablosundaki p (sig.) değeri $0,461 > 0,05$ olduğundan dolayı H_0 hipotezi kabul edilir. H_0 hipotezi kabul edildiği için üç algoritma ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir fark yoktur diyebiliriz. Elde edilen

sonuçlar arasında istatistiksel olarak anlamlı bir fark olmadığı için bu alt problemin çözümünde üç algıtmada tercih edilebilir.

Tablo 4.18. Kuru ürünün yoğun günde dağıtım alt problemi için varyansların homojenliği testi

Levene Statistic	df1	df2	Sig.
,427	2	60	,655

Tablo 4.19. Kuru ürünün yoğun günde dağıtım alt problemi için varyans analizi tablosu

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2766,887	2	1383,443	,785	,461
Within Groups	105803,342	60	1763,389		
Total	108570,229	62			

4.3.3. Taze ürünün sakin günde dağıtım alt problemi için ANOVA testi

Taze ürünün sakin günde dağıtım alt probleminde her hafta 66 mağazaya dağıtım yapılmaktadır. Her mağazanın 21 haftalık talep verileri mevcuttur. Mağazalar tarafından her hafta talep edilen palet miktarları 5-9 aralığında değişmekte ve toplamları 402 palet olmaktadır.

Bu alt problem karınca kolonisi, genetik ve yapay arı kolonisi algoritması ile çözüldüğünde elde edilen sonuçlar Tablo 4.20'de verilmiştir.

Tablo 4.20. Taze ürünün sakin günde dağıtım alt problemi için KKA, GA ve YAKA sonuçları

Talepler	Araç Sayısı	KKA ile Bulunan Toplam Mesafe (km)	GA ile Bulunan Toplam Mesafe (km)	YAKA ile Bulunan Toplam Mesafe (km)
1	11	1016,5	1017,4	1026,9
2	11	1046,4	1034,6	1053
3	11	1005	1033,7	1022,5
4	11	1029,1	1034,9	1045,1
5	11	1039,7	1027,3	1039,8
6	11	1033,6	1027,2	1058,5
7	11	1020,5	1043,6	1046,7
8	11	1019,9	1035	1020,8
9	11	1013,1	1035,9	1026
10	11	1026,7	1028,2	1038,4

Tablo 4.20. Taze ürünün sakın günde dağıtımı alt problemi için KKA, GA ve YAKA sonuçları (Devamı)

11	11	1030,9	1035,6	1039,2
12	11	1036,5	1040,9	1036,8
13	11	1016,3	1021,8	1043
14	11	1007,6	1020,5	1039,1
15	11	1017,7	1039,6	1030
16	11	1064,3	1027,3	1053,4
17	11	1025,1	1039,7	1043,2
18	11	1051,6	1036	1036,1
19	11	1018,8	1023,6	1017,6
20	11	1033,4	1032,2	1028,8
21	11	1013,7	1027,6	1032,7

Tablo 4.20'deki sonuçlara ANOVA testi uygulanmış ve SPSS sonuçları aşağıda verilmiştir. Bu çalışmada;

$H_0: \mu_1=\mu_2=\mu_3$ (Üç algoritma ile bulunan toplam mesafelerin ortalaması arasında fark yoktur.)

H_1 : En az bir algoritma ile bulunan toplam mesafelerin ortalaması farklıdır.

Olacak şekilde hipotezler tanımlanmıştır. ANOVA testi uygulandığında ilk olarak varyansların homojen olup olmadığı incelenmektedir. Tablo 4.21'deki p (sig.) değeri $0,027 < 0,05$ olduğundan dolayı grup varyanslarının homojen olmadığı sonucuna varılır. Grup varyansları homojen olmadığı için Tablo 4.22'deki F testinin sonuçları anlamlı olmayacaktır. Bu nedenle ANOVA testinin alternatifi olan Welch ve Brown-Forsythe testleri uygulanmıştır. Bu testlerin sonuçları Tablo 4.23'te gösterilmektedir. Brown-Forsythe testine baktığımızda p (sig.) değeri $0,024 < 0,05$ olduğundan dolayı H_1 hipotezi kabul edilir. Aynı şekilde Welch testine baktığımızda da p (sig.) değeri $0,047 < 0,05$ olduğundan dolayı H_1 hipotezi kabul edilir. H_1 hipotezi kabul edildiği için üç algoritma ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir fark vardır diyebiliriz. Tablo 4.24'te taze ürünün sakın günde dağıtımı alt probleminin üç algoritma ile çözümünü sonucunda bulunan toplam mesafelerin ortalaması verilmiştir. Toplam mesafelerin ortalaması karınca kolonisi algoritması ile 1026,971 km, genetik algoritma ile 1031,552 km ve yapay arı kolonisi algoritması ile 1037,029 km olarak bulunmuştur. Bu sonuçlar bize karınca kolonisi algoritmasının diğer iki algoritmaya göre daha iyi sonuç verdiğini göstermektedir.

Tablo 4.21. Taze ürünün sakın günde dağıtımı alt problemi için varyansların homojenliği testi

Levene Statistic	df1	df2	Sig.
3,831	2	60	,027

Tablo 4.22. Taze ürünün sakın günde dağıtımı alt problemi için varyans analizi tablosu

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1064,839	2	532,420	4,043	,023
Within Groups	7900,698	60	131,678		
Total	8965,537	62			

Tablo 4.23. Taze ürünün sakın günde dağıtımı alt problemi için Welch ve Brown-Forsythe testi

	Statistic ^a	df1	df2	Sig.
Welch	3,318	2	36,718	,047
Brown-Forsythe	4,043	2	46,776	,024

Tablo 4.24. Taze ürünün sakın günde dağıtımı alt problemi için tanımlayıcı istatistikler

	N	Mean	Std. Deviation	Std. Error
1	21	1026,971	14,8892	3,2491
2	21	1031,552	7,1376	1,5576
3	21	1037,029	11,0635	2,4143
Total	63	1031,851	12,0252	1,5150
Model				
Fixed Effects			11,4751	1,4457
Random Effects				2,9071

4.3.4. Kuru ürünün sakın günde dağıtımı alt problemi için ANOVA testi

Kuru ürünün yoğun günde dağıtımı alt probleminde her hafta 66 mağazaya dağıtım yapılmaktadır. Her mağazanın 21 haftalık talep verileri mevcuttur. Mağazalar tarafından her hafta talep edilen palet miktarları 8-14 aralığında değişmekte ve toplamları 652 palet olmaktadır.

Bu alt problem karınca kolonisi, genetik ve yapay arı kolonisi algoritması ile çözüldüğünde elde edilen sonuçlar Tablo 4.25'te verilmiştir.

Tablo 4.25. Kuru ürünün sakin günde dağıtım alt problemi için KKA, GA ve YAKA sonuçları

Talepler	Araç Sayısı	KKA ile Bulunan Toplam Mesafe (km)	Araç Sayısı	GA ile Bulunan Toplam Mesafe (km)	Araç Sayısı	YAKA ile Bulunan Toplam Mesafe (km)
1	19	1654,4	18	1561,8	18	1596,6
2	19	1694,1	19	1676,6	19	1703,4
3	19	1655,4	19	1659,5	19	1703,6
4	19	1678,2	18	1590,2	19	1708,8
5	19	1656,8	19	1673,4	19	1647,6
6	19	1675,9	19	1676,8	19	1673,7
7	19	1667,3	19	1670,9	18	1594,9
8	19	1664,4	19	1681,5	18	1609,2
9	19	1658,9	18	1605,6	20	1759,8
10	19	1668,5	18	1578	19	1686,5
11	18	1592,4	19	1666,1	19	1675,2
12	18	1589,9	19	1661,6	18	1605,1
13	19	1652,5	19	1657,9	18	1553,8
14	19	1673	19	1656,7	19	1677
15	18	1606	19	1648,4	19	1632,8
16	19	1702,2	19	1690	20	1761,8
17	19	1678,1	19	1683,6	19	1649,4
18	20	1777,7	19	1679,8	18	1604,6
19	18	1612,2	19	1681,7	18	1631,9
20	19	1668,2	19	1677,9	19	1677,7
21	19	1665,1	19	1661,5	19	1704,2

Tablo 4.25'teki sonuçlara ANOVA testi uygulanmış ve SPSS sonuçları aşağıda verilmiştir. Bu çalışmada;

$H_0: \mu_1=\mu_2=\mu_3$ (Üç algoritma ile bulunan toplam mesafelerin ortalaması arasında fark yoktur.)

H_1 : En az bir algoritma ile bulunan toplam mesafelerin ortalaması farklıdır.

Olacak şekilde hipotezler tanımlanmıştır. ANOVA testi uygulandığında ilk olarak varyansların homojen olup olmadığı incelenmektedir. Tablo 4.26'daki p (sig.) değeri $0,061 > 0,05$ olduğundan dolayı grup varyanslarının homojen olduğu sonucuna varılır. Grup varyansları homojen olduğu için Tablo 4.27'deki F testinin sonuçları anlamlı olacaktır. Varyans analizi tablosundaki p (sig.) değeri $0,861 > 0,05$ olduğundan dolayı H_0 hipotezi kabul edilir. H_0 hipotezi kabul edildiği için üç algoritma ile elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir fark yoktur diyebiliriz. Elde edilen

sonuçlar arasında istatistiksel olarak anlamlı bir fark olmadığı için bu alt problemin çözümünde üç algortmada tercih edilebilir.

Tablo 4.26. Kuru ürünün sakın günde dağıtımı alt problemi için varyansların homojenliği testi

Levene Statistic	df1	df2	Sig.
2,933	2	60	,061

Tablo 4.27. Kuru ürünün sakın günde dağıtımı alt problemi için varyans analizi tablosu

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	604,595	2	302,297	,150	,861
Within Groups	120693,541	60	2011,559		
Total	121298,136	62			

4.3.5. Alt problemlerin ANOVA testi sonuçlarının karşılaştırılması

Taze ürünün yoğun günde dağıtımı, kuru ürünün yoğun günde dağıtımı, taze ürünün sakın günde dağıtımı ve kuru ürünün sakın günde dağıtımı alt problemleri karınca kolonisi, genetik ve yapay arı kolonisi algoritmalarıyla çözülmüş elde edilen sonuçlar ANOVA testi yardımıyla karşılaştırılmıştır. Alt problemlerin ANOVA testi sonuçları Tablo 4.28'de özetlenmiştir.

Tablo 4.28. Alt problemlerin ANOVA testi sonuçları

Alt Problem	ANOVA Testi Sonucuna Göre Algoritmaların Üstünlükleri
Taze ürünün yoğun günde dağıtımı alt problemi	Karınca kolonisi algoritması
Kuru ürünün yoğun günde dağıtımı alt problemi	Algoritmalar arasında fark yok
Taze ürünün sakın günde dağıtımı alt problemi	Karınca kolonisi algoritması
Kuru ürünün sakın günde dağıtımı alt problemi	Algoritmalar arasında fark yok

Alt problemlerin haftalık talep miktarlarını incelediğimizde taze ürünün yoğun günde dağıtımı alt probleminin 438, Kuru ürünün yoğun günde dağıtımı alt probleminin 724, taze ürünün sakın günde dağıtımı alt probleminin 402 ve kuru ürünün sakın günde dağıtımı alt probleminin ise 652 palet olduğunu görüyoruz. Taze ürünün dağıtıldığı alt problemlerde talep miktarları birbirlerine çok yakın ve kuru ürünün dağıtıldığı alt problemlerin talep miktarlarına göre daha azdır. Talep miktarları

arttıkça aracın kapasitesi sınırlı olduđu için bir aracın uğrayacağı müşteri sayısı da az olmakta, buna bađlı olarak da araç sayısı ve toplam kat edilen mesafeler artmaktadır.

ANOVA testi sonuçlarını incelediğimizde taze ürünün dağıtıldığı alt problemlerde karınca kolonisi algoritmasının daha iyi sonuç verdiği, kuru ürünün dağıtıldığı alt problemlerde ise üç algoritma arasında istatistiksel olarak anlamlı bir farkın olmadığı görölmektedir. Bundan dolayı üç algoritmanın da uygulanabileceğini düşünöyoruz. Bu ortaya çıkan sonuç yukarıda da anlatıldığı gibi alt problemlerin haftalık talep miktarlarından kaynaklanmaktadır.

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

İşletmeler, dünyadaki rekabet ortamına ayak uydurabilmek için gelişen teknolojiye uygun şekilde ürettikleri hizmet ve ürünleri müşterilerine ulaştırırken hem müşteri memnuniyetini üst seviyede tutmak hem de taşıma maliyetlerini ve harcanan zamanı en aza indirmek zorundadırlar.

Taşıma maliyetlerinin ve harcanan zamanın minimuma indirilmesi, araç rotalama işleminin etkin bir şekilde gerçekleştirilmesiyle mümkündür. Birçok farklı kısıta sahip olabilen ARP için üç çeşit çözüm yöntemi mevcuttur. Bunlar; kesin, klasik sezgisel ve metasezgisel yöntemlerdir. ARP NP-zor problem sınıfından olduğu için problemin boyutu büyüdüğünde, küçük boyutlu problemlerin çözümünde etkili sonuçlar veren kesin çözüm yöntemleri yetersiz kalmaktadır. Bu nedenle büyük boyutlu problemlerde klasik sezgisel ve metasezgisel yöntemler tercih edilmektedir. Klasik sezgisel yöntemler, metasezgisel yöntemlere göre daha kısa sürede çözüm üretirler. Fakat metasezgisel yöntemlerle elde edilen sonuçlar klasik sezgisel yöntemlerle elde edilen sonuçlara göre daha kalitelidir.

Bu çalışmada perakende sektöründe hizmet veren bir işletmenin haftalık taleplerinin karşılanabilmesi için araç rotalarının belirlenmesi problemi ele alınmıştır. Bu problem kapasite kısıtlı ARP'dir. Araçların hepsi maksimum 40 palet kapasitelidir ve araçlara 32 ile 40 palet arasında yükleme yapılmaktadır. Bu ele alınan kapasite kısıtlı ARP, ürünlerin çeşidi ve dağıtım yapılacağı günlere göre 4 adet alt probleme ayrılmaktadır. Bu alt problemler; taze ürünün yoğun günde dağıtımı, kuru ürünün yoğun günde dağıtımı, taze ürünün sakin günde dağıtımı ve kuru ürünün sakin günde dağıtımı problemidir. Bu alt problemlerden taze ürünün yoğun günde dağıtımı ve kuru ürünün yoğun günde dağıtımı alt problemlerinde 78 adet mağaza, taze ürünün sakin günde dağıtımı ve kuru ürünün sakin günde dağıtımı alt problemlerinde ise 66 adet mağaza mevcuttur. Taze ürünün yoğun günde dağıtımı alt probleminin talep

miktarı 438, Kuru ürünün yoğun günde dağıtımını alt probleminin talep miktarı 724, taze ürünün sakın günde dağıtımını alt probleminin talep miktarı 402 ve kuru ürünün sakın günde dağıtımını alt probleminin talep miktarı ise 652 palettir. Bu çalışmada ilk olarak karınca kolonisi, genetik ve yapay arı kolonisi algoritmaları kullanılarak 78 mağazalık ve 66 mağazalık alt problemler için ayrı ayrı gezgin satıcı problemi uygulanmıştır. Bir mağazadan harekete başlayıp her mağazaya sadece bir defa uğradıktan sonra başladığı mağazaya geri dönen en kısa tur oluşturulmuştur. Daha sonra alt problemlerin haftalık talepleri göz önüne alınarak, elde edilen rotadaki ilk mağazadan başlanıp kamyonların palet kapasite kısıtını sağlayacak şekilde kümeler oluşturulmuştur. Oluşturulan her küme dağıtım yapacak olan aracı temsil etmektedir. Kümeler oluşturulduktan sonra ise her birinin rotaları ana depodan başlayıp ve yine ana depoda son bulacak şekilde dal ve sınır yöntemiyle oluşturulup kat edilen mesafeler hesaplanmıştır. Son olarak ise elde edilen sonuçlar ANOVA testi yardımıyla karşılaştırılmıştır.

ANOVA testi sonuçlarını incelediğimizde taze ürünün dağıtıldığı alt problemlerde karınca kolonisi algoritmasının daha iyi sonuç verdiği, kuru ürünün dağıtıldığı alt problemlerde ise üç algoritma arasında istatistiksel olarak anlamlı bir farkın olmadığı görülmektedir. Bundan dolayı üç algoritmanın da uygulanabileceğini düşünüyoruz. Bu ortaya çıkan sonuç alt problemlerin haftalık talep miktarlarından kaynaklanmaktadır. Taze ürünün dağıtıldığı alt problemlerde talep miktarları birbirlerine çok yakın ve kuru ürünün dağıtıldığı alt problemlerin talep miktarlarına göre daha azdır. Talep miktarları arttıkça aracın kapasitesi sınırlı olduğu için bir aracın uğrayacağı müşteri sayısı da az olmakta, buna bağlı olarak da toplam araç sayısı ve kat edilen mesafeler artmaktadır. Dolayısıyla kuru ürünün dağıtımını alt problemlerinde üç algoritmanın da elde ettiği sonuçlar birbirine yakın olmakta bu nedenle aralarında istatistiksel olarak anlamlı bir farkın oluşmadığını düşünmekteyiz.

Bu çalışmada ele alınan problemde ürünlerin teslim zamanı ile ilgili kısıt dikkate alınmamıştır. Bundan sonraki çalışmamıza bu kısıtın da eklenmesi düşünülmektedir. Ayrıca bu problemde ürün dağıtımını yapan araçların sayısı sınırsız olarak ele alınmıştır. Zaman kısıtına ilave olarak araç sayısının sabit tutulması kısıtı da modele

eklenebilir. Böylece ele alınan problemin gerçek hayat problemlerine yaklaştırılması düşünülmektedir.

KAYNAKLAR

- [1] DEMİRCİOĞLU, M., Araç Rotalama Probleminin Sezgisel Bir Yaklaşım ile Çözümlemesi Üzerine Bir Uygulama, Çukurova Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 2009.
- [2] TOTH, P., VIGO, D., An Overview of Vehicle Routing Problems-Chapter 1, The vehicle routing problem, SIAM, Philadelphia, pp.1-26, 2002.
- [3] AYDEMİR, E., Esnek Zaman Pencereli Araç Rotalama Problemi Ve Bir Uygulama, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2006.
- [4] DÜZAKIN, E., DEMİRCİOĞLU, M., Araç Rotalama Problemleri ve Çözüm Yöntemleri, Çukurova Üniversitesi İİBF Dergisi, 13(1):68-87, 2009.
- [5] KURT, M., Çoklu Depolu Araç Rotalama Problemleri için Bir Karınca Kolonisi Optimizasyonu Algoritmasının Tasarımı ve Uygulaması, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2008.
- [6] ERYAVUZ, M., GENCER, C., Araç Rotalama Problemine Ait Bir Uygulama, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 6(1):139-155, 2001.
- [7] RUIZ R., MAROTO C., ALCARAZ J., A Decision Support System for a Real Vehicle Routing Problem, European Journal of Operational Research, 153:593-606, 2004.
- [8] AI, T.J., KACHITVICHYANUKUL, V., Particle Swarm Optimization and Two Solution Representations for Solving the Capacitated Vehicle Routing Problem, Computer and Industrial Engineering, 56:380-387, 2009.
- [9] GERDAN, O., Müşteriler Arası Malzeme Akışlı Eş Zamanlı Dağıtım-Toplama Yapılan Araç Rotalama Problemi ve Sezgisel Çözümü, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 2007.

- [10] EROL, V., Araç Rotalama Problemleri için Popülasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2006.
- [11] BRÄYSY, O., GENDREAU, M., “Metaheuristics for the Vehicle Routing Problem with Time Windows”, SINTEF Report, SINTEF Applied Mathematics, Research Council of Norway, Norway, 2001.
- [12] YILMAZ, Ş., Çok Depolu Araç Rotalama Probleminin Karınca Kolonisi Optimizasyonu ile Modellenmesi ve Bir Çözüm Önerisi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2008.
- [13] TEZER, T., Toplama ve Dağıtım Zaman Pencereci Araç Rotalama Problemi için Kesin Çözüm Yaklaşımı ve Örnek Uygulamalar, Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2009.
- [14] YURTKURAN, A., Araç Rotalama Problemlerinin Çözümü için Yeni Bir Meta-Sezgisel Yaklaşım: Elektromanyetik Algoritma, Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2009.
- [15] CREVIER, B., CORDEAU, J.F., LAPORTE, G., The Multi-Depot Vehicle Routing Problem with Inter-Depot Routes, European Journal of Operational Research, 176:756-773, 2007.
- [16] EKİZLER, H., Araç Rotalama Probleminin Çözümünde Karınca Kolonisi Optimizasyonu Algoritmasının Kullanılması, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2011.
- [17] GÖKSAL, F.P., Eşzamanlı Topla-Dağıt Araç Rotalama Problemi için Sezgisel Yaklaşımlar: Genetik Algoritma ve Kuş Sürüsü Eniyileme, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2010.
- [18] ŞEKER, Ş., Araç Rotalama Problemleri ve Zaman Pencereci Stokastik Araç Rotalama Problemine Genetik Algoritma Yaklaşımı, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2007.
- [19] CLAUSEN, J., Branch and Bound Algorithms - Principles and Examples, Department of Computer Science, University of Copenhagen, 1999.
- [20] <http://neo.lcc.uma.es/radi-aeb/WebVRP/>, Erişim Tarihi: 25.03.2014.
- [21] TAHA, H., Operations Research And Introduction, Copyright by Prentice Hall. Inc. Simon&Schuster/A Viacom Company, 2005.
- [22] ÇEVİK, O., Tam Sayılı Doğrusal Programlama ile İşgücü Planlaması ve Bir Uygulama, Afyon Kocatepe Üniversitesi İİBF Dergisi, 8(1):157-171, 2006.

- [23] BAŞKAYA, Z., ÖZTÜRK, B., Tamsayılı Programlamada Dal Kesme Yöntemi ve Bir Ekmek Fabrikasında Oluşturulan Araç Rotalama Problemine Uygulanması, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 24(1):101-114, 2005.
- [24] MITCHELL, J.E., Branch-and-Cut Algorithms for Integer Programming, *Encyclopedia of Optimization*, 9:1650-1657, 1998.
- [25] http://en.wikipedia.org/wiki/Branch_and_cut, Erişim Tarihi: 20.03.2014.
- [26] ÇETİN, E., Dinamik Programlama ile Sınır Tenörü Optimizasyonu, *Doğu ve Güneydoğu Madenlerinin Değerlendirilmesi Sempozyumu*, s.139-143, 2005.
- [27] EMEL, G., TAŞKIN, Ç., Araç Rotalama Problemlerinin İki Aşamalı Çözümünde Genetik Algoritma Kullanımı, *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 7(1):1-17, 2005.
- [28] LAPORTE, G., GENDREAU, B., POTVIN, J., SEMET, F., Classical and Modern Heuristics for the Vehicle Routing Problem, *International Transactions in Operational Research*, 7:285- 300, 2000.
- [29] BREEDAM, A.V., A Parametric Analysis of Heuristics for the Vehicle Routing Problem with Side-Constraints, *European Journal of Operational Research*, 137:348-370, 2002.
- [30] MARINAKIS, Y., MIGDALAS A., Heuristic Solutions of Vehicle Routing Problems in Supply Chain Management, DSS Laboratory, Department of Production Engineering and Management, Technical University of Crete, Greece, 2001.
- [31] CORDEAU, J.F., GENDREAU, M., LAPORTE, G., POTVIN, J.Y., SEMET, F., A Guide To Vehicle Routing Heuristics, *Journal of the Operational Research Society*, 53(5):512-522, 2002.
- [32] BREEDAM, A.V., Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem, *Computers & Operations Research*, 28:289-315, 2001.
- [33] THOMPSON, P.M., PSARAFTIS, H.N., Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems, *Operations Research*, 41(5):935-946, 1993.
- [34] ÖNDER, E., Araç Rotalama Problemlerinin Parçacık Sürü ve Genetik Algoritma ile Optimizasyonu, *İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi*, 2011.

- [35] ROPKE, S., Heuristic and Exact Algorithms for Vehicle Routing Problems, University of Copenhagen, Department of Computer Science, Doktora Tezi, Danimarka, 2005.
- [36] TARANTILIS, C.D., IOANNOU, G., PRASTACOS, G., Advanced Vehicle Routing Algorithms for Complex Operations Management Problems, Journal of Food Engineering, 70:455-471, 2005.
- [37] LAPORTE, G., SEMET, F., Classical Heuristics for the Vehicle Routing Problem, Les Cahiers du GERAD, G-98- 54, Canada, 1999.
- [38] ULUSOY, G., Proje Planlamada Kaynak Kısıtlı Çizelgeleme, Mühendislik ve Doğa Bilimleri Fakültesi, Sabancı Üniversitesi, İstanbul, 2002.
- [39] GÜDEN, H., VAKVAK, B., ÖZKAN, B. E., ALTIPARMAK, F., DENGİZ, B., Genel Amaçlı Arama Algoritmaları ile Benzetim Eniyilemesi: En İyi Kanban Sayısının Bulunması, Endüstri Mühendisliği Dergisi, 16(1):2-15, 2005.
- [40] DURSUN, P., Zaman Pencereli Araç Rotalama Problemi'nin Genetik Algoritma ile Modellenmesi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2009.
- [41] GENDREAU, M., LAPORTE, G., POTVIN, J.Y., Metaheuristics for the Vehicle Routing Problem, Les Cahiers du GERAD, G-98-52, Canada, 1999.
- [42] KEMER, B., Araç Rotalama Problemlerine Genetik Algoritma Yaklaşımı: Bir Gıda Dağıtım Firması Uygulaması, Karadeniz Teknik Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 2010.
- [43] AL-ANZI, F.S., ALLAHVERDİ, A., A Self-Adaptive Differential Evolution Heuristic for Two-Stage Assembly Scheduling Problem to Minimize Maximum Lateness with Setup Times, European Journal of Operational Research, 182:80-94, 2007.
- [44] ÇETİN, M., Gezgin Satıcı Örnek Problemlerinin Optimum Sonuçlarının Grid Aracılığı ile Hesaplanması, Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2007.
- [45] KENNEDY, J., EBERHART, R., Particle Swarm Optimization, in Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp.1942–1948, 1995.
- [46] ÖZSAĞLAM, M.Y., Parçacık Sürü Optimizasyonu Algoritmasının Gezgin Satıcı Problemine Uygulanması ve Performansının İncelenmesi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2009.

- [47] DORIGO, M., MANIEZZO, V., COLORNI, A., Ant System: An Autocatalytic Optimizing Process, Technical Report TR91-016, Politecnico di Milano, 1991.
- [48] KARABOGA, D., Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, 2011.
- [49] KARABOGA, D., An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [50] KARABOGA, D., AKAY, B., “A Comparative Study of Artificial Bee Colony Algorithm”, Applied Mathematics and Computation, 214:108-132, 2009.
- [51] BULLNHEIMER, B., HARTL, R.F., STRAUSS, C., Applying the Ant System to the Vehicle Routing Problem, 2nd International Conference on Metaheuristics - MIC97, 1997.
- [52] BULLNHEIMER, B., HARTL, R.F., STRAUSS, C., An Improved Ant System Algorithm for the Vehicle Routing Problem, Annals of Operations Research, 89:319–28, 1999.
- [53] BAKER, B.M., AYECHIEW, M.A., A Genetic Algorithm for the Vehicle Routing Problem, Computers & Operations Research, 30:787-800, 2003.
- [54] LUCIC, P., TEODOROVIC, D., Computing with Bees: Attacking Complex Transportation Engineering Problems, International Journal on Artificial Intelligence Tools, 12(3):375-394, 2003.
- [55] KUŞÇU, Ö., Araç Rotalama Sistemlerinde Sezgisel Yöntemler, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2009.
- [56] MAZZEO, S., LOISEAU, I., An Ant Colony Algorithm for the Capacitated Vehicle Routing , Electronic Notes in Discrete Mathematic, 18:181–186, 2004.
- [57] GAJPAL, Y., ABAD, P.L., Multi-Ant Colony System (MACS) for a Vehicle Routing Problem with Backhauls, European Journal of Operational Research, 196: 102–117, 2009.
- [58] KESKİNTÜRK, T., Araç Rotalama Problemlerinin Global Karınca Koloni Optimizasyonu ile Çözümü, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 2009.
- [59] WANG, C.-H., LU, J.-Z., A Hybrid Genetic Algorithm that Optimizes Capacitated Vehicle Routing Problems, Expert Systems with Applications, 36(2921–2936), 2009.

- [60] BRAJEVIC, I., Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem, Proceedings of the 5th European conference on European computing conference, pp. 239-244, 2011.
- [61] SZETO, W.Y., WU, Y., HO, S.C., An Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem, European Journal of Operational Research, 215:126-135, 2011.
- [62] REED, M., YIANNAKOU, A., EVERING, R., An Ant Colony Algorithm for the Multi-Compartment Vehicle Routing Problem, Applied Soft Computing, 15:169–176, 2014.
- [63] GAMBARDELLA, L.M., RIZZOLI, A.E., OLIVERIO, F., CASAGRANDE, N., DONATI, A.V., MONTEMANNI, R., LUCIBELLO, E., Ant Colony Optimization for Vehicle Routing in Advanced Logistics Systems, IDSIA and AntOptima via Fusoni 4, Switzerland, 2004.
- [64] HO, W., HO, G.T.S., JI, P., LAU, H.C.W., A Hybrid Genetic Algorithm for the Multi-Depot Vehicle Routing Problem, Engineering Applications of Artificial Intelligence, 21:548–557, 2008.
- [65] NARASIMHA, K.V., KIVELEVITCH, E., SHARMA, B., KUMAR, M., An Ant Colony Optimization Technique for Solving Min–Max Multi-Depot Vehicle Routing Problem, Swarm and Evolutionary Computation, 2013.
- [66] LIU, S., HUANG, W., MA, H., An Effective Genetic Algorithm for the Fleet Size and Mix Vehicle Routing Problems, Transportation Research Part E, 45(434–445), 2009.
- [67] MONTEMANNI, R., GAMBARDELLA, L.M., RIZZOLI, A.E., DONATI, A.V., Ant Colony System for a Dynamic Vehicle Routing Problem, Journal of Combinatorial Optimization, 10:327–343, 2005.
- [68] LI, J., ZHANG, J., A Genetic Algorithm to Vehicle Routing Problem in Reverse Logistics, 14th International Conference on Management Science & Engineering, August 20-22, pp.573-578, 2007.
- [69] TASAN, A.S., GEN, M., A Genetic Algorithm Based Approach to Vehicle Routing Problem with Simultaneous Pick-Up and Deliveries, Computers & Industrial Engineering, 62:755-761, 2012.
- [70] BERGER, J., BARKAOUI, M., A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows, Computers & Operations Research, 31:2037–2053, 2004.

- [71] RIZZOLI, A.E., OLIVERIO, F., MONTEMANNI, R., GAMBARDELLA, L.M., Ant Colony Optimisation for Vehicle Routing Problems: from Theory to Applications, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA) and AntOptima via Fusoni 4, CH-6900, Switzerland, 2005.
- [72] DONATI, A. V., MONTEMANNI, R., CASAGRANDE, N., RIZZOLI, A.E., GAMBARDELLA, L.M., Time Dependent Vehicle Routing Problem with a Multi Ant Colony System, European Journal of Operational Research, 185:1174-1191, 2006.
- [73] MA, J., ZOU, H., GAO, L.-Q, LI, D., Immune Genetic Algorithm for Vehicle Routing Problem with Time Windows, Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006, pp.3465-3469, 2006.
- [74] ALVARENGA, G.B., MATEUS, G.R., TOMI, G., A Genetic and Set Partitioning Two-Phase Approach for the Vehicle Routing Problem with Time Windows, Computers & Operations Research, 34:1561-1584, 2007.
- [75] KILIÇ, S., Bulanık Karar Ortamında Karınca Kolonisi Optimizasyonu Yöntemiyle Araç Rotalama, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 2008.
- [76] WANG W., WANG, Z., QIAO, F., An Improved Genetic Algorithm for Vehicle Routing Problem with Time-Window, International Symposium on Computer Science and Computational Technology, 2008.
- [77] WANG, X.-P., XU, C.-L., HU, X.-P., Genetic Algorithm for Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles, 15th International Conference on Management Science & Engineering, 10-12 September 2008, pp.128-133, 2008.
- [78] VIDAL, T., CRAINIC, T.G., GENDREAU, M., PRINS, C., A Hybrid Genetic Algorithm with Adaptive Diversity Management for a Large Class of Vehicle Routing Problems with Time-Windows, Computers & Operations Research, 40:475-489, 2013.
- [79] GOSS., S., ARON, S., DENEUBOURG, J.L. VE PASTEELS, J.M., Self-Organized Shortcuts in the Argentine Ant, Naturwissenschaften, 76:579-581, 1989.
- [80] DORIGO, M., GAMBARDELLA, L.M., Ant Colonies for the Travelling Salesman Problem, BioSystems, 43:73-81, 1997.
- [81] KESKİNTÜRK, T., SÖYLER, H., Global Karınca Kolonisi Optimizasyonu, Gazi Üniv. Müh. Mim. Fak. Der., 21(4):689-698, 2006.

- [82] STUTZLE, T., HOOS, H.H., MAX-MIN Ant System, Future Generation Computer Systems, 16:889-914, 2000.
- [83] DORIGO, M., SOCHA, K., 2007, An Introduction to Ant Colony Optimization, IRIDIA, Bruxelles, ISSN: 1781-3794.
- [84] DURUKAN, D., Doğrusal ve Doğrusal Olmayan Kafes Sistemlerin Genetik Algoritma ile Optimum Tasarımı, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2013.
- [85] ZEYVELİ, M., Genetik Algoritma İle Hız Kutusu Dişli Tasarımı, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 2005.
- [86] ELMAS, Çetin; Yapay Zeka Uygulamaları Yapay Sinir Ağları – Bulanık Mantık – Genetik Algoritma, Seçkin Yayıncılık, 2010.
- [87] AKTÜRK, F., Genetik Algoritma ile Büyükbaş Süt Hayvanlarında Süt Verimi Maksimizasyonu, Gazi Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 2013.
- [88] SIRIWARDENE, N.R., PERERA, B.J.C., Selection of Genetic Algorithm Operators for Urban Drainage Model Parameter Optimization, Mathematical and Computer Modelling, 44:415-429, 2006.
- [89] NABIYEV, V.V., Yapay Zeka: Problemler Yöntemler Algoritmalar, 1. Baskı, Ankara: Seçkin Yayınevi, 2003.
- [90] BALIN, A., Genetik Algoritma ile Hat Dengeleme, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2010.
- [91] SIVANANDAM, S.N., DEEPA, S.N., Introduction to Genetic Algorithm, Berlin: Springer-Verlag, 2008.
- [92] HAUPT, R.L., HAUPT, S.E., Practical Genetic Algorithms, Second Edition, New Jersey: Wiley-Interscience, 2004.
- [93] EMEL, G., TAŞKIN, Ç., Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21(1):129-152, 2002.
- [94] ENGİN, T., Genetik Algoritma ile Toplu Ulaşım Sistemi Hareket Çizelgesi Optimizasyonu: Çanakkale Örneği, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 2013.
- [95] UÇANER, M.E., ÖZDEMİR, O.N., Genetik Algoritmalar ile İçme Suyu Şebekelerinde Ek Klorlama Optimizasyonu, Gazi Üniv. Müh. Mim. Fak. Der, 17(4):157-170, 2002.

- [96] İŞÇİ, Ö., KORUKOĞLU, S., Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama, Celal Bayar Üniversitesi İİBF Yönetim ve Ekonomi Dergisi, 10(2):191-208, 2003.
- [97] www.ibrahimcayiroglu.com, Erişim Tarihi: 10.02.2014.
- [98] AKÇA, M.R., Yapay Arı Kolonisi Algoritması Kullanılarak Gezgin Satıcı Probleminin Türkiyedeki İl ve İlçe Merkezlerine Uygulanması, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2011.
- [99] TOPLU, M.N., Retinaya Ait Oftalmoskop Görüntüleri Üzerinde Yapay Arı Kolonisi Algoritması Kullanılarak Görüntü Kayıtlama, Ege Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2013.
- [100] TOSUN, Ö., Yapay Arı Kolonisi Algoritması ve Permütasyon Akış Tipi Çizelgeleme Problemine Uygulanması, Akdeniz Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 2012.
- [101] TOKMAK, M., Yapay Arı Kolonisi Algoritması ile Ders Çizelgeleme Probleminin Çözümü, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2011.
- [102] AKAY, B., KARABOGA, D., A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization, Information Sciences, 192:120–142, 2012.

EKLER

EK-1 Alt problemlerin Haftalık Talep Miktarları

TAZE ÜRÜNÜN YOĞUN GÜNDE DAĞITIMI ALT PROBLEMİ																					
Mağaza Numaraları	1. Hafta	2. Hafta	3. Hafta	4. Hafta	5. Hafta	6. Hafta	7. Hafta	8. Hafta	9. Hafta	10. Hafta	11. Hafta	12. Hafta	13. Hafta	14. Hafta	15. Hafta	16. Hafta	17. Hafta	18. Hafta	19. Hafta	20. Hafta	21. Hafta
1	5	3	4	5	3	3	4	3	3	3	3	3	7	3	8	3	8	3	4	5	4
2	7	4	5	5	4	3	3	3	3	8	4	3	9	3	3	4	8	4	7	8	9
3	9	5	5	5	3	3	4	3	3	4	5	3	9	4	7	6	3	5	7	3	5
4	7	5	5	6	4	8	7	3	7	4	3	4	3	5	6	7	3	3	9	7	8
5	9	5	5	5	5	6	5	3	4	5	9	4	6	6	3	8	4	4	4	4	3
6	3	5	6	6	6	9	7	4	5	8	9	3	7	3	4	7	8	4	5	6	7
7	7	5	7	3	7	6	4	4	7	9	3	8	3	6	4	9	5	6	4	4	4
8	9	5	7	4	8	9	7	5	8	4	4	5	4	8	3	7	6	5	3	6	6
9	7	9	7	7	9	9	3	8	7	3	5	6	5	3	4	9	7	9	8	3	4
10	5	9	8	8	3	4	5	9	6	3	6	8	7	7	3	8	8	7	3	6	6
11	5	9	8	9	4	9	6	6	3	4	7	6	3	4	5	3	6	4	4	4	3
12	7	9	9	9	5	5	3	8	3	5	8	9	3	9	4	6	9	3	7	3	6
13	5	8	9	9	5	8	9	6	4	6	7	7	3	3	8	9	3	4	3	5	3
14	5	8	4	8	5	3	5	9	5	7	9	8	9	4	6	4	3	5	7	8	6
15	7	8	5	6	3	7	3	7	6	8	7	9	4	7	7	7	8	6	7	9	8
16	5	6	6	7	4	4	7	8	3	9	8	9	3	6	8	8	4	3	3	9	9
17	5	6	3	5	3	6	5	9	3	3	8	9	4	5	3	3	5	3	4	6	8
18	7	6	4	4	3	4	3	9	3	5	9	3	6	4	4	3	4	6	7	9	6
19	5	3	5	3	3	6	4	9	3	4	3	7	8	4	3	7	3	4	5	8	8
20	5	3	7	5	3	3	4	7	4	3	8	3	8	3	3	4	6	5	4	4	8
21	9	3	8	6	4	6	3	9	4	3	9	4	7	3	9	7	7	9	4	6	7
22	5	5	9	7	5	3	4	3	5	9	7	8	3	9	6	8	3	8	5	3	3

23	5	5	9	8	6	6	4	9	6	7	3	7	7	6	8	3	7	5	6	9	7
24	3	6	9	5	7	8	4	7	3	4	3	5	3	8	8	3	8	4	5	8	3
25	5	7	8	6	8	9	9	3	3	3	3	6	4	8	7	4	4	3	4	3	4
26	5	8	7	7	9	8	9	8	7	5	4	8	8	7	8	4	9	4	8	9	8
27	3	9	6	4	3	6	4	8	3	8	4	8	6	8	8	8	4	5	4	9	6
28	5	9	5	5	3	9	7	7	3	9	5	6	3	8	6	5	5	4	3	8	3
29	5	6	4	6	3	7	3	5	4	9	6	7	3	7	4	6	5	3	6	3	3
30	3	7	5	3	4	7	3	5	3	6	3	8	8	4	7	3	5	5	7	9	8
31	5	5	6	3	4	7	8	5	7	9	3	9	3	7	7	3	4	4	8	4	3
32	5	4	7	3	3	3	7	5	8	8	4	3	4	7	9	4	8	3	4	3	4
33	3	4	7	4	4	8	4	5	6	4	5	7	3	9	4	5	6	5	4	4	3
34	5	4	5	3	5	9	4	7	6	6	6	3	4	5	5	6	7	4	3	6	4
35	5	3	4	7	6	3	9	5	5	3	7	4	3	8	3	7	4	9	8	8	3
36	3	4	3	5	6	3	4	3	9	9	6	9	4	9	7	6	9	8	4	8	4
37	5	5	3	3	7	4	3	3	8	8	7	5	3	5	3	7	4	3	6	7	3
38	5	5	4	3	7	4	9	3	4	3	8	6	4	4	3	3	7	9	3	3	6
39	3	5	6	3	6	5	4	4	4	9	9	4	3	6	4	8	8	9	4	7	3
40	5	5	4	3	5	6	5	7	5	9	3	4	4	3	5	7	4	8	6	3	4
41	5	6	6	3	4	7	8	3	6	8	4	5	3	4	6	3	4	8	4	4	3
42	3	6	6	5	3	8	5	6	6	3	6	7	4	4	7	3	3	3	7	8	6
43	5	6	7	4	4	7	6	6	7	7	9	6	3	5	8	8	4	4	6	6	3
44	5	6	7	5	4	7	7	8	9	8	3	8	4	5	8	3	5	8	8	3	9
45	3	7	3	6	4	7	7	7	9	5	5	6	3	6	8	8	6	5	6	3	7
46	5	7	3	7	3	8	8	6	6	3	6	5	6	3	4	6	3	5	9	8	3
47	7	8	3	7	3	8	5	3	6	3	7	6	9	7	3	3	6	4	7	3	6
48	7	9	3	8	4	8	5	3	8	7	7	7	8	7	9	3	7	3	9	4	3
49	5	9	9	5	8	7	4	7	9	3	8	4	5	3	7	4	9	8	9	6	9
50	9	8	3	6	9	8	8	9	9	9	9	8	4	3	3	5	9	7	4	7	3
51	9	7	7	4	9	9	9	3	6	7	3	4	6	6	3	6	3	9	8	3	9
52	5	8	6	6	8	3	8	8	4	6	9	5	7	7	4	3	3	6	4	6	5
53	5	7	5	7	8	9	3	4	6	9	8	5	6	7	5	9	4	9	3	3	9
54	9	8	4	3	9	5	3	4	9	3	7	6	9	9	6	3	8	7	4	4	7
55	5	4	3	4	9	3	4	7	3	6	9	4	9	6	9	3	8	3	5	3	9
56	5	5	3	5	9	4	4	6	7	9	3	3	4	9	6	6	5	9	7	5	3
57	7	3	3	7	8	3	5	7	8	6	8	7	5	6	3	3	5	6	5	5	4

58	5	4	3	8	4	7	7	7	9	7	7	7	7	7	7	4	6	7	4	4	9
59	7	5	5	8	5	3	8	6	7	5	3	3	8	9	9	3	7	6	4	6	9
60	7	6	6	3	8	3	6	8	9	8	3	4	5	6	3	9	8	8	9	7	4
61	5	7	3	3	8	4	9	4	9	3	7	5	4	3	4	9	4	9	9	8	4
62	7	9	3	3	9	4	6	4	9	3	3	6	9	4	4	9	9	8	9	7	7
63	9	3	9	3	9	7	8	4	4	4	3	7	9	9	5	7	9	8	5	3	9
64	5	3	4	4	3	3	7	6	3	4	3	3	9	5	4	9	7	6	4	3	5
65	5	4	3	5	3	3	9	6	5	3	3	3	7	5	3	3	7	9	4	6	6
66	5	3	9	6	9	3	9	5	6	5	4	4	9	3	9	4	3	3	9	4	3
67	5	3	3	8	4	3	4	3	9	5	5	5	4	3	4	9	3	4	3	7	9
68	9	9	9	9	4	7	3	3	9	3	3	3	6	7	3	6	3	8	4	5	4
69	5	5	5	9	4	6	7	5	5	6	3	4	7	8	3	4	4	3	3	5	9
70	7	3	5	9	9	5	4	5	7	3	7	4	6	7	4	3	7	4	5	5	7
71	3	4	9	9	5	3	3	4	3	4	7	8	9	8	5	7	5	7	7	6	6
72	5	3	9	9	6	5	7	3	8	9	6	9	9	3	9	8	5	7	3	4	7
73	7	3	9	9	6	3	7	3	4	4	9	4	5	9	7	3	5	4	9	3	5
74	3	3	8	9	9	3	4	3	6	3	5	3	7	3	6	8	6	7	7	9	7
75	5	4	5	8	8	3	5	8	4	6	7	3	8	3	7	7	4	8	5	9	3
76	9	5	4	4	9	6	8	7	4	4	3	7	5	4	9	4	3	4	7	7	4
77	3	3	4	4	7	5	4	9	4	6	3	8	7	3	9	6	9	4	9	8	6
78	7	5	4	3	8	5	9	4	3	5	3	4	5	4	8	8	3	5	7	4	7

KURU ÜRÜNÜN YOĞUN GÜNDE DAĞITIMI ALT PROBLEMİ																					
Mağaza Numaraları	1. Hafta	2. Hafta	3. Hafta	4. Hafta	5. Hafta	6. Hafta	7. Hafta	8. Hafta	9. Hafta	10. Hafta	11. Hafta	12. Hafta	13. Hafta	14. Hafta	15. Hafta	16. Hafta	17. Hafta	18. Hafta	19. Hafta	20. Hafta	21. Hafta
1	8	7	13	13	12	11	8	9	14	13	13	14	7	9	8	14	8	13	10	13	8
2	12	6	6	7	10	9	13	8	12	8	11	9	9	12	12	10	8	13	7	6	8
3	14	9	12	10	9	13	10	14	10	13	6	11	9	10	7	9	10	11	9	12	12
4	12	7	7	10	12	9	7	9	7	13	13	7	11	8	6	7	13	12	7	7	11
5	14	9	9	7	9	6	11	11	12	11	9	10	6	12	10	10	9	8	11	9	7
6	6	11	10	11	11	13	7	7	10	12	9	7	7	7	8	7	8	11	13	10	11
7	12	8	11	10	7	6	6	10	11	14	10	8	10	6	13	9	12	8	10	11	12
8	14	11	7	11	11	11	8	7	7	10	12	11	10	8	13	7	6	13	10	7	7
9	12	11	9	7	12	11	7	8	7	8	9	6	11	13	11	9	7	9	8	9	7
10	8	6	13	7	7	10	11	11	6	13	13	8	7	13	12	8	8	7	10	13	6
11	8	9	7	6	11	11	13	6	10	9	11	6	12	11	12	12	9	12	9	7	11
12	12	7	9	6	10	9	6	8	10	8	10	9	12	12	13	7	6	13	7	9	11
13	8	11	9	10	9	13	9	6	14	12	7	7	13	12	8	9	12	6	10	8	7
14	8	13	6	7	12	9	11	9	13	13	11	8	9	13	12	11	11	9	7	6	7
15	12	7	10	13	9	13	7	7	12	6	7	9	12	8	7	8	8	6	7	10	7
16	8	10	7	6	8	6	7	8	9	9	8	9	12	12	10	7	12	13	12	7	13
17	8	7	14	9	9	11	10	9	11	6	8	9	12	7	6	11	7	11	10	14	11
18	12	12	7	11	9	11	8	9	7	11	9	12	6	10	8	14	11	6	7	7	11
19	8	12	8	7	6	10	10	12	11	6	10	7	8	7	11	7	7	12	10	8	12
20	8	11	10	11	9	12	12	7	6	11	8	11	8	8	9	12	6	8	10	10	12
21	14	8	12	6	7	13	12	11	11	8	9	13	7	11	9	11	7	9	10	8	9
22	8	7	8	11	12	12	9	13	7	9	7	8	10	9	6	9	13	7	13	8	8
23	8	7	8	7	7	6	12	8	6	7	11	7	7	6	8	9	8	12	6	8	11
24	6	10	7	7	8	8	6	7	10	14	12	10	12	8	8	10	8	8	11	7	13
25	8	14	11	10	12	9	10	10	11	8	10	8	14	8	7	13	10	12	11	11	8
26	8	6	13	11	7	7	10	8	7	12	7	8	8	7	8	11	9	8	8	11	9
27	6	14	7	7	8	6	10	8	9	8	8	8	6	8	8	8	10	8	12	7	8
28	8	9	6	9	10	12	7	8	13	8	13	12	11	8	6	11	11	9	12	6	9
29	8	9	9	13	14	7	8	12	8	9	8	11	11	7	13	11	12	7	6	9	8
30	6	7	14	8	8	8	10	11	12	7	10	10	8	13	7	10	10	9	7	13	8
31	8	10	8	12	12	14	7	10	7	9	10	10	10	7	7	12	8	8	8	8	13
32	8	8	9	8	11	13	7	10	14	8	7	12	11	7	9	11	8	12	11	9	8

33	6	8	7	8	9	8	11	12	12	12	12	7	7	9	11	11	6	12	8	7	9
34	8	12	8	12	9	9	11	7	11	12	9	11	9	11	8	7	7	8	9	8	8
35	8	6	7	11	10	13	9	11	12	8	7	11	8	8	10	7	11	7	8	7	7
36	6	9	10	12	6	10	10	11	9	9	12	9	7	10	7	13	9	9	11	10	12
37	8	14	10	9	11	10	13	9	8	8	13	10	11	7	12	9	12	11	6	10	12
38	8	7	6	8	7	10	9	10	12	11	9	13	7	12	13	7	7	9	11	6	13
39	6	8	13	12	10	8	11	13	10	9	7	7	7	13	14	8	8	9	9	13	10
40	8	7	11	10	9	12	11	8	8	9	6	11	9	11	10	7	12	8	6	11	11
41	8	13	6	8	9	11	8	7	6	8	8	8	8	10	11	10	9	8	11	6	7
42	6	7	9	6	10	8	14	6	13	11	6	7	11	10	6	12	13	11	7	9	7
43	8	11	12	13	8	7	11	6	13	7	9	6	7	7	10	8	13	7	6	12	9
44	8	7	8	13	12	7	11	8	11	8	13	8	12	10	10	9	13	8	8	8	10
45	6	8	7	11	7	7	7	6	11	11	8	6	9	14	8	8	9	11	6	7	7
46	8	10	13	11	11	8	6	6	6	12	8	6	13	8	11	6	12	12	9	13	10
47	12	13	7	14	13	8	13	10	10	12	7	10	6	6	7	8	6	12	7	7	9
48	12	10	10	10	14	8	10	11	8	10	7	11	9	7	9	11	7	10	9	10	14
49	8	10	13	8	8	7	12	7	7	8	8	6	8	9	7	11	9	8	9	13	6
50	14	9	13	7	9	8	8	9	9	7	14	8	6	7	10	10	9	7	11	14	7
51	14	8	9	9	7	10	9	9	6	9	8	12	7	7	11	7	10	9	8	9	8
52	8	10	11	6	8	11	8	6	10	6	10	8	12	11	11	9	11	6	13	11	8
53	8	11	7	10	9	14	12	6	6	9	8	13	10	8	8	9	6	9	12	7	9
54	14	6	10	6	9	9	11	14	12	7	7	13	6	8	6	13	11	14	13	10	12
55	8	14	13	12	14	7	8	6	9	10	13	11	7	14	9	8	6	10	14	13	14
56	8	12	13	14	7	6	14	9	6	9	9	9	10	9	6	6	6	9	14	13	14
57	12	9	13	7	9	7	13	7	7	6	14	7	6	6	12	7	8	6	12	13	14
58	8	13	6	6	14	7	7	6	9	14	7	7	12	7	7	6	11	7	11	6	10
59	12	10	6	9	8	13	13	6	7	6	11	6	7	8	9	8	14	14	10	6	10
60	12	6	11	7	6	9	6	8	9	8	8	6	7	14	12	7	7	8	9	11	6
61	8	7	9	9	6	12	9	13	7	12	7	6	14	9	9	6	11	11	14	9	6
62	12	14	11	14	6	8	6	11	9	8	14	7	14	10	10	14	9	6	9	11	6
63	14	6	9	9	6	7	11	11	9	7	8	11	12	9	12	11	14	8	7	9	6
64	8	10	7	7	9	8	7	14	8	6	7	6	8	6	6	6	6	6	6	7	6
65	8	7	8	6	7	6	7	14	8	7	7	8	8	6	6	6	9	12	6	8	14
66	8	8	11	8	14	7	6	14	7	12	12	14	10	9	8	6	9	6	6	11	14
67	8	14	6	11	6	14	12	11	10	8	6	14	14	10	14	6	8	6	6	6	13

68	14	13	6	10	13	7	14	8	11	8	7	8	11	10	6	14	6	7	6	6	8
69	8	7	12	11	6	6	14	12	6	6	6	12	12	7	12	11	11	6	6	12	9
70	12	6	9	14	6	7	7	10	7	11	7	10	14	7	7	14	7	6	7	10	7
71	6	8	6	9	6	6	6	9	9	8	6	8	9	13	13	9	14	8	8	6	8
72	8	13	14	8	7	12	7	14	8	7	6	8	6	14	9	10	10	9	7	14	6
73	12	10	6	12	9	6	6	14	12	10	12	14	6	7	7	14	10	14	14	9	6
74	6	10	10	6	6	9	9	10	9	7	14	8	8	13	14	14	6	12	14	13	6
75	8	7	7	14	6	14	9	6	7	8	7	6	7	10	7	9	6	12	12	7	8
76	14	6	12	6	12	6	6	6	14	6	10	14	7	6	6	6	6	6	10	12	9
77	6	9	7	7	14	10	6	9	6	10	14	14	6	6	7	6	14	6	6	7	8
78	12	8	9	6	14	8	7	13	6	14	10	14	14	14	14	6	14	14	14	9	8

TAZE ÜRÜNÜN SAKIN GÜNDE DAĞITIMI ALT PROBLEMİ																					
Mağaza Numaraları	1. Hafta	2. Hafta	3. Hafta	4. Hafta	5. Hafta	6. Hafta	7. Hafta	8. Hafta	9. Hafta	10. Hafta	11. Hafta	12. Hafta	13. Hafta	14. Hafta	15. Hafta	16. Hafta	17. Hafta	18. Hafta	19. Hafta	20. Hafta	21. Hafta
1	5	8	6	6	8	3	8	8	4	6	9	5	7	7	4	3	3	6	4	6	9
2	7	7	5	7	8	9	3	4	6	9	8	5	6	7	5	9	4	9	9	3	9
3	9	8	4	3	9	5	9	4	9	3	7	6	9	7	6	3	8	7	4	4	7
4	7	4	3	4	9	3	4	7	3	3	9	4	9	6	9	3	8	3	5	3	9
5	9	5	4	5	9	4	4	6	7	9	3	3	4	9	6	6	5	9	7	5	3
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	8	9	6	9	8	8	9	9	7	9	8	4	7	3	5	9	7	4	7	3
8	9	7	7	4	9	9	9	3	6	9	3	4	6	6	9	6	3	9	8	9	3
9	7	8	6	6	8	8	8	8	8	6	9	5	7	7	4	8	9	6	4	6	5
10	5	7	5	7	8	5	3	4	6	9	8	5	6	7	5	9	4	9	9	3	5
11	5	8	4	3	9	8	3	4	9	3	7	6	9	7	6	5	8	7	5	4	4
12	7	4	3	4	9	8	4	7	7	3	9	4	9	6	9	5	8	3	5	3	3
13	5	5	4	5	9	8	9	6	7	9	7	9	4	9	6	6	9	9	7	5	3
14	5	3	5	7	8	5	5	7	8	6	8	7	5	6	3	7	5	6	5	5	3
15	7	4	4	8	4	7	7	7	9	7	7	7	7	7	7	5	6	7	5	9	8
16	5	5	5	8	5	5	8	6	7	3	8	3	8	3	9	5	7	6	5	6	7
17	5	6	6	3	8	5	6	8	9	8	9	4	5	6	8	9	8	8	9	7	9
18	7	7	3	3	8	5	9	4	9	3	7	5	4	3	4	9	4	9	9	8	4
19	5	9	3	3	9	4	6	4	9	3	3	6	9	4	4	9	9	8	9	7	3
20	5	3	4	3	9	7	8	4	4	4	3	7	9	3	5	7	9	8	5	9	7
21	9	3	4	4	3	3	7	6	3	6	9	3	9	5	9	9	7	6	5	3	7
22	5	4	3	5	3	3	9	6	5	9	3	3	7	5	3	5	7	9	5	6	5
23	5	3	4	6	9	3	9	5	6	5	4	4	9	4	9	5	3	3	9	4	5
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	5	9	3	9	4	7	3	3	9	8	3	8	6	7	3	6	3	8	5	5	8
26	5	5	5	9	4	6	7	5	5	4	3	3	7	8	3	5	4	5	9	5	9
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	5	3	9	9	9	9	4	3	6	5	5	3	7	3	6	8	6	7	7	9	9
29	5	4	5	8	8	3	5	8	4	6	7	3	8	3	7	7	4	8	5	9	5
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	5	7	5	7	8	9	3	4	6	9	8	5	6	7	5	9	4	9	9	5	9
32	5	8	4	3	9	5	3	4	9	3	7	6	9	7	6	3	8	7	5	5	7

33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	5	9	9	3	9	7	8	4	4	5	3	7	9	3	5	7	9	8	5	5	9
35	5	9	4	4	3	9	7	6	3	6	3	3	9	5	4	9	7	6	5	5	5
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	5	9	5	6	3	7	3	7	6	8	7	9	4	7	7	7	8	6	7	5	8
38	5	9	6	7	4	4	7	8	3	9	8	9	3	6	8	8	4	5	9	5	9
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	5	9	9	7	5	3	4	3	5	9	7	8	3	9	6	8	3	8	5	5	5
41	5	9	9	8	6	6	4	9	6	7	3	7	7	6	8	3	7	5	6	9	7
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	5	9	6	5	3	6	5	8	3	8	4	8	6	8	8	8	4	5	5	9	6
44	5	9	5	5	3	9	7	7	3	9	5	6	3	8	6	5	5	5	5	8	5
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	5	9	7	7	5	9	5	8	7	9	5	6	5	3	4	9	7	9	8	5	4
47	7	9	8	8	5	9	7	9	6	5	6	8	7	7	9	8	8	7	5	6	6
48	7	5	8	9	5	9	6	6	5	5	7	6	5	4	5	5	5	4	5	5	5
49	5	5	9	9	5	9	5	8	5	5	8	9	3	9	4	6	9	5	7	9	5
50	9	5	9	9	5	8	9	6	5	6	7	7	3	3	8	5	5	5	9	5	5
51	9	5	4	8	5	9	9	9	5	7	9	8	5	4	6	5	9	5	9	8	6
52	5	5	5	9	5	9	5	7	6	8	8	9	5	7	7	5	8	5	7	9	8
53	5	5	6	7	5	9	7	8	9	9	8	9	5	6	8	5	5	5	5	9	5
54	9	5	9	5	5	5	5	4	5	5	8	9	5	5	3	5	5	3	5	6	8
55	5	6	4	4	5	5	9	9	9	5	5	8	5	4	4	5	4	6	5	9	6
56	5	7	9	9	5	5	5	3	8	5	5	7	8	4	3	5	3	4	5	8	8
57	7	8	7	5	5	5	5	7	5	5	8	3	8	6	8	5	6	5	9	5	8
58	5	5	8	6	7	5	9	9	5	5	9	4	7	3	9	7	7	5	5	9	5
59	7	5	9	7	7	5	5	8	5	5	7	8	3	9	6	5	9	5	5	6	5
60	7	5	9	8	5	5	9	9	5	5	5	7	5	6	8	5	7	5	9	9	5
61	5	5	9	5	5	5	5	7	5	5	5	5	5	8	8	5	6	5	5	5	5
62	7	5	8	6	4	5	9	3	5	5	5	6	5	8	7	5	4	9	5	5	5
63	9	5	7	7	4	5	5	3	5	5	5	8	5	7	8	5	9	5	5	9	5
64	5	5	6	4	5	5	5	8	5	8	5	8	5	8	8	5	4	5	5	5	5
65	5	5	5	5	5	5	5	7	5	8	5	6	5	8	6	5	5	4	9	5	5
66	5	5	8	6	5	5	5	2	5	9	5	7	3	7	4	5	5	5	6	5	5
67	5	6	5	9	5	5	5	5	5	6	5	8	8	4	7	5	5	5	7	9	8

68	9	4	6	8	5	5	5	5	5	5	5	9	7	7	7	4	5	5	5	5	9
69	5	4	7	6	4	5	5	5	5	5	5	5	4	7	9	5	8	5	5	5	5
70	7	4	7	4	4	5	5	5	9	5	5	7	3	9	4	5	6	5	5	5	9
71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	5	9	8	8	6	5	5	9	6	5	5	8	7	7	3	7	8	5	3	5	5
73	7	9	8	9	6	7	6	6	9	5	5	6	3	4	5	9	6	5	5	5	5
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	5	4	7	5	6	5	5	7	5	6	5	3	8	9	3	5	5	5	5	5	8
76	9	4	8	6	6	5	9	9	8	5	5	4	7	3	9	7	5	5	5	5	7
77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	7	5	7	4	5	9	7	5	8	5	5	6	9	8	7	9	5	5	5	5	5

KURU ÜRÜNÜN SAKIN GÜNDE DAĞITIMI ALT PROBLEMİ																					
Mağaza Numaraları	1. Hafta	2. Hafta	3. Hafta	4. Hafta	5. Hafta	6. Hafta	7. Hafta	8. Hafta	9. Hafta	10. Hafta	11. Hafta	12. Hafta	13. Hafta	14. Hafta	15. Hafta	16. Hafta	17. Hafta	18. Hafta	19. Hafta	20. Hafta	21. Hafta
1	8	14	7	6	6	10	11	14	10	8	10	7	13	9	12	8	11	6	8	9	12
2	12	5	11	11	8	7	7	10	12	11	10	8	13	14	14	13	7	11	14	12	6
3	14	3	12	11	7	8	7	8	9	14	11	13	11	9	7	9	11	14	10	7	8
4	12	14	7	10	11	11	6	13	13	8	7	13	12	8	8	7	6	11	8	11	14
5	14	7	11	11	13	6	10	9	11	6	12	11	12	12	9	12	11	8	9	13	7
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	12	6	13	7	14	12	12	11	11	8	9	13	7	11	9	11	9	7	9	11	7
8	14	14	7	14	10	6	9	13	7	14	7	8	10	9	6	9	8	9	14	11	9
9	12	12	10	8	8	8	12	8	6	7	11	9	7	6	14	9	6	14	10	10	9
10	8	9	10	7	13	8	6	7	10	14	12	10	12	8	8	10	7	7	11	7	10
11	8	13	7	11	9	7	10	10	11	8	10	8	14	14	7	13	12	11	11	9	11
12	12	12	11	13	8	10	10	8	7	12	7	14	8	7	8	11	10	8	8	9	14
13	8	9	10	6	12	7	10	8	9	8	8	8	6	8	14	8	6	8	6	13	8
14	8	14	11	14	13	12	7	8	13	8	13	12	11	8	6	11	7	14	14	13	14
15	12	11	7	9	6	14	8	12	8	9	8	11	11	7	13	11	10	9	6	6	6
16	8	12	14	7	9	8	10	11	12	7	10	12	8	13	7	10	6	6	12	7	14
17	8	14	6	9	6	6	7	10	7	9	10	10	10	7	14	12	12	7	7	13	11
18	12	8	6	7	11	11	7	10	14	8	7	12	11	14	9	11	7	8	9	8	7
19	8	11	10	14	14	11	11	12	12	12	12	7	7	9	11	11	7	14	12	7	8
20	8	9	7	9	11	8	11	7	11	12	9	11	9	11	8	7	14	9	9	7	11
21	14	8	13	8	8	10	9	11	12	8	7	14	8	14	10	7	14	10	10	13	9
22	8	11	6	8	9	11	10	11	9	9	12	9	7	10	7	13	7	9	12	9	14
23	8	10	9	3	7	7	13	9	8	8	13	10	11	7	12	9	9	6	6	13	7
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	8	13	7	7	11	10	13	9	11	12	8	7	11	8	8	10	7	11	7	8	7
26	8	14	11	10	12	6	10	10	11	9	9	12	9	7	10	7	13	9	9	11	10
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	8	6	8	9	11	8	11	6	8	8	8	8	10	11	10	9	8	11	6	14	14
29	8	8	14	10	8	14	6	13	11	6	7	11	10	14	12	13	11	7	9	7	5
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	8	11	10	9	13	10	14	10	13	6	11	9	10	7	9	10	11	9	12	12	14
32	8	8	10	12	9	7	9	7	13	13	7	11	8	14	7	13	12	7	7	11	14

33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
34	8	10	12	7	11	9	11	8	9	7	11	9	12	6	10	8	14	11	14	7	7
35	8	8	12	8	7	6	10	10	12	11	6	10	7	8	7	11	7	7	12	10	8
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	8	12	11	11	7	11	11	11	8	7	7	10	12	11	10	8	13	14	14	13	10
38	8	6	12	11	9	7	12	11	7	8	14	8	9	6	11	13	11	9	7	9	8
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	8	9	7	7	13	14	8	9	11	8	11	6	8	8	8	8	10	11	10	9	8
41	8	5	9	7	7	9	6	10	8	14	6	13	11	6	7	11	10	6	12	13	11
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	8	14	11	9	10	9	13	9	14	14	12	7	7	13	12	8	9	12	6	10	14
44	8	7	13	14	7	12	9	11	9	13	13	11	8	9	13	12	11	11	9	7	9
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	8	12	7	6	11	13	14	7	8	12	8	9	8	11	11	7	13	11	12	7	14
47	12	11	10	8	14	8	8	14	10	11	12	7	10	10	8	13	7	10	10	9	9
48	12	12	7	11	7	12	12	14	7	10	9	9	10	10	10	7	7	12	8	8	8
49	8	8	8	9	12	8	11	13	7	10	14	8	7	12	11	7	9	11	14	12	11
50	14	12	11	9	11	8	9	8	11	12	12	12	12	7	7	9	11	11	6	12	8
51	14	10	12	14	10	14	7	8	6	7	11	12	9	11	14	11	8	7	7	8	14
52	8	14	8	11	10	14	9	12	8	14	12	8	14	11	8	8	10	7	11	7	8
53	8	8	12	12	7	12	11	11	12	11	9	9	12	9	7	10	7	13	9	9	11
54	14	5	10	14	7	7	7	7	6	9	8	8	13	10	11	7	12	9	12	11	9
55	8	9	14	8	13	13	14	14	8	10	12	11	9	13	7	12	13	14	7	9	11
56	8	14	8	11	14	9	10	10	9	13	10	9	7	7	14	13	14	8	8	9	11
57	12	8	13	9	7	7	9	13	14	8	8	14	6	11	9	11	10	7	12	8	6
58	8	10	9	8	13	14	14	7	8	11	6	8	14	8	8	10	11	10	9	8	11
59	12	11	14	11	8	14	7	14	14	6	13	11	6	14	11	10	6	12	13	11	9
60	12	9	7	10	6	6	14	11	6	6	13	7	9	14	7	7	10	8	13	7	6
61	8	8	10	8	14	7	7	9	6	14	11	8	13	8	12	10	14	14	13	8	14
62	12	11	8	12	7	9	9	7	14	6	11	11	8	6	9	14	8	8	9	11	9
63	14	9	11	14	14	12	11	10	6	14	6	12	14	14	13	8	11	14	12	12	11
64	8	12	10	11	7	10	8	6	13	10	10	12	7	10	6	6	7	8	6	12	9
65	8	10	12	10	6	14	12	6	10	11	8	10	7	11	14	7	9	11	9	10	11
66	8	9	14	9	14	7	10	14	12	9	7	8	8	9	8	9	14	11	9	8	11
67	8	9	9	10	7	14	8	9	8	9	9	7	14	8	6	7	10	7	14	14	11

68	14	9	8	11	12	9	8	8	9	12	14	14	8	12	14	14	11	7	10	9	9
69	8	9	12	14	14	14	13	12	12	13	9	7	14	11	14	14	6	14	11	11	14
70	12	9	6	8	9	9	9	9	14	9	8	13	14	14	7	8	7	13	8	12	6
71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	8	10	10	14	9	12	14	7	10	7	9	10	10	10	14	6	12	8	8	8	12
73	12	11	8	9	8	11	13	7	10	14	14	7	12	11	7	9	14	14	12	11	5
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	8	9	8	13	12	10	8	11	13	10	8	7	7	5	13	14	9	8	9	9	9
76	14	8	14	11	10	14	12	14	8	14	14	14	14	9	11	10	14	12	8	14	11
77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	12	9	10	13	11	11	8	6	6	6	12	8	6	13	14	11	14	12	14	9	9

ÖZGEÇMİŞ

Serap Ercan, 21.10.1990'da Sakarya'da doğdu. İlk, orta ve lise eğitimini Sakarya'da tamamladı. 2008 yılında Arifiye Anadolu Öğretmen Lisesi'nden mezun oldu. 2008 yılında Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nü kazandı ve okulu birincilik ile bitirdi. 2012 yılında aynı üniversitede yüksek lisans eğitimi almaya başladı. Halen 2012 yılında çalışmaya başladığı Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nde araştırma görevlisi olarak görev yapmaktadır.