

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

**ÇOK AMAÇLI ELEKTRİKLİ ARAÇ ROTALAMA PROBLEMİNİN  
HİYERARŞİK BİR METASEZGİSEL ALGORİTMA İLE ÇÖZÜMÜ**

**DOKTORA TEZİ**

**Serap ERCAN CÖMERT**

**Endüstri Mühendisliği Anabilim Dalı**

**SUBAT 2023**



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

**ÇOK AMAÇLI ELEKTRİKLİ ARAÇ ROTALAMA PROBLEMİNİN  
HİYERARŞİK BİR METASEZGİSEL ALGORİTMA İLE ÇÖZÜMÜ**

**DOKTORA TEZİ**

**Serap ERCAN CÖMERT**

**Endüstri Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Prof. Dr. Harun Reşit YAZĞAN**

**ŞUBAT 2023**



Serap ERCAN CÖMERT tarafından hazırlanan “ÇOK AMAÇLI ELEKTRİKLİ ARAÇ ROTALAMA PROBLEMİNİN HİYERARŞİK BİR METASEZGİSEL ALGORİTMA İLE ÇÖZÜMÜ” adlı tez çalışması 08.02.2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda Doktora tezi olarak kabul edilmiştir.

### Tez Jürisi

<b>Jüri Başkanı :</b>	<b>Prof. Dr. Cemil ÖZ</b> Sakarya Üniversitesi	.....
<b>Jüri Üyesi :</b>	<b>Prof. Dr. Harun Reşit YAZĞAN</b> Sakarya Üniversitesi	.....
<b>Jüri Üyesi :</b>	<b>Doç. Dr. Özer UYGUN</b> Sakarya Üniversitesi	.....
<b>Jüri Üyesi :</b>	<b>Prof. Dr. Alpaslan FIĞLALI</b> Kocaeli Üniversitesi	.....
<b>Jüri Üyesi :</b>	<b>Prof. Dr. Serol BULKAN</b> Marmara Üniversitesi	.....



## **ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ**

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “ÇOK AMAÇLI ELEKTRİKLİ ARAÇ ROTALAMA PROBLEMİNİN HİYERARŞİK BİR METASEZGİSEL ALGORİTMA İLE ÇÖZÜMÜ” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(08/02/2023)

Serap ERCAN CÖMERT





*Aileme*



## TEŐEKKÜR

Lisans, yüksek lisans ve doktora eđitimim boyunca deđerli bilgi ve tecrübelerinden yararlandığım, yardımlarını esirgemeyen, beni teşvik eden ve kendimi geliőtirmemde büyük katkısı olan kıymetli danışman hocam Prof. Dr. Harun Reşit YAZĞAN'a sonsuz teşekkürlerimi sunarım.

Tez izleme jürimde yer alan Prof. Dr. Cemil ÖZ ve Doç. Dr. Özer UYGUN hocalarıma yaptıkları katkı, yorum ve yönlendirmeler için çok teşekkür ederim.

Her ihtiyaç duyduğumda bilgi ve tecrübelerini benden esirgemeyerek bana destek olan deđerli hocam Dr. Öğr. Üyesi Sena KIR ve akademisyen arkadaşlarım Arş. Gör. Meryem Nur MORGÜL TUMBAZ ve Arş. Gör. Furkan YENER'e de teşekkürlerimi sunarım.

Bu zorlu süreçte her zaman destekçim olan sevgili eşim Semih CÖMERT'e, hayattaki en büyük motivasyon kaynađım biricik ođlum Muhammed Mert CÖMERT'e ve tüm hayatım boyunca her zaman arkamda olan, beni sürekli destekleyen ve bugünlere gelmemde en büyük emeđe sahip olan canım anneme, babama ve kardeşime çok teşekkür ederim.

Son olarak doktora tezimi 2211/E Yurt İçi Doğrudan Doktora Bursu ile maddi olarak destekleyen TÜBİTAK'a teşekkürlerimi sunarım.

Serap ERCAN CÖMERT



## İÇİNDEKİLER

### Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ .....	v
TEŞEKKÜR .....	ix
İÇİNDEKİLER .....	xi
KISALTMALAR .....	xiii
SİMGELER .....	xv
TABLO LİSTESİ .....	xvii
ŞEKİL LİSTESİ .....	xix
ÖZET .....	xxi
SUMMARY .....	xxiii
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1. Tezin Amacı ve Kapsamı .....	3
1.2. Tezin Literatüre Katkısı .....	4
1.3. Tezin Organizasyonu .....	5
<b>2. ÇOK AMAÇLI OPTİMİZASYON PROBLEMİ</b> .....	<b>7</b>
2.1. Çok Amaçlı Optimizasyon .....	7
2.2. Çok Amaçlı Optimizasyon Problemlerinin Çözüm Yöntemleri .....	9
2.2.1. Geleneksel yöntemler .....	9
2.2.1.1. Ağırlıklı toplam yöntemi (ATY) .....	10
2.2.1.2. Konik skalerleştirme yöntemi (KSY) .....	11
2.2.2. Çok amaçlı evrimsel algoritmalar .....	12
2.2.2.1. Çok amaçlı parçacık sürü optimizasyonu (ÇAPSO) .....	13
<b>3. PROBLEMİN TANIMI</b> .....	<b>17</b>
3.1. Çok Amaçlı Elektrikli Araç Rotalama Problemi .....	19
3.1.1. Senaryo 1 .....	19
3.1.2. Senaryo 2 .....	25
3.1.3. Senaryo 3 .....	27
<b>4. LİTERATÜR ARAŞTIRMASI</b> .....	<b>31</b>
4.1. Elektrikli Araç Rotalama Problemi ile İlgili Çalışmalar .....	31
4.2. Çok Amaçlı Yeşil Araç Rotalama Problemi ile İlgili Çalışmalar .....	37
<b>5. ÇÖZÜM METODU</b> .....	<b>39</b>
5.1. Karınca Kolonisi Optimizasyonu (KKO) .....	39
5.1.1. Çözüm oluşturma aşaması .....	40
5.1.2. Feromon Güncelleme Aşaması .....	40
5.1.2.1. Lokal feromon güncellemesi .....	41
5.1.2.2. Global feromon güncellemesi .....	41
5.2. Yapay Arı Kolonisi Algoritması (YAKA) .....	42
5.2.1. Başlangıç aşaması .....	42
5.2.2. İşçi arı aşaması .....	43
5.2.3. Gözcü arı aşaması .....	44

5.2.4. Kaşif arı aşaması .....	44
5.3. Geliştirilen Hiyerarşik Yöntem (HKKO-YAKA) .....	45
5.3.1. 1. aşama: HKKO .....	46
5.3.1.1. Çözüm oluşturma aşaması.....	46
5.3.1.2. Yerel arama .....	47
5.3.1.3. Kabul kriteri .....	50
5.3.1.4. Feromon güncelleme aşaması .....	51
5.3.2. 2. aşama: YAKA .....	51
5.3.2.1. Başlangıç aşaması .....	51
5.3.2.2. Uygunluk değerlerinin hesaplanması .....	52
5.3.2.3. İşçi arı aşaması .....	52
5.3.2.4. Gözcü arı aşaması.....	53
5.3.2.5. Kaşif arı aşaması .....	53
5.3.3. HKKO-YAKA'nın parametreleri.....	53
5.3.4. HKKO-YAKA'nın sözde kodu.....	55
5.3.5. Bir ÇAEARP örneğinin geliştirilen HKKO-YAKA ile çözümü .....	57
<b>6. SAYISAL ANALİZLER.....</b>	<b>61</b>
6.1. Taguchi Yöntemi ile Parametre Tasarımı.....	61
6.2. Geliştirilen HKKO-YAKA'nın Sınanması.....	66
<b>7. ÖRNEK OLAY ÇALIŞMASI.....</b>	<b>73</b>
7.1. Senaryoların HKKO-YAKA ile Çözümü.....	74
7.2. Senaryoların ÇAPSO ile Çözümü .....	84
7.3. Senaryoların Uyarlanmış ÇAPSO ile Çözümü.....	87
7.4. Parametrik Analizler.....	89
7.4.1. Yöntemlerin elde ettiği sonuçlar açısından karşılaştırılması.....	89
7.4.2. Senaryolardan elde edilen sonuçların karşılaştırılması .....	92
<b>8. SONUÇ VE DEĞERLENDİRME .....</b>	<b>95</b>
<b>KAYNAKLAR.....</b>	<b>97</b>
<b>EKLER.....</b>	<b>105</b>
<b>ÖZGEÇMİŞ.....</b>	<b>113</b>

## KISALTMALAR

<b>ANOVA</b>	: Varyans analizi
<b>ARP</b>	: Araç rotalama problemi
<b>AYA</b>	: Alternatif yakıtlı araçlar
<b>ATY</b>	: Ağırlıklı toplam yöntemi
<b>CNG</b>	: Sıkıştırılmış doğal gaz
<b>ÇAEARP</b>	: Çok amaçlı elektrikli araç rotalama problemi
<b>ÇAKKO</b>	: Çok amaçlı karınca kolonisi optimizasyonu
<b>ÇAPSO</b>	: Çok amaçlı parçacık sürü optimizasyonu
<b>ÇAYARP</b>	: Çok amaçlı yeşil araç rotalama problemi
<b>DOŞFEARP</b>	: Doğrusal olmayan şarj fonksiyonlu elektrikli araç rotalama problemi
<b>EA</b>	: Elektrikli araç
<b>EARP</b>	: Elektrikli araç rotalama problemi
<b>GA</b>	: Genetik algoritma
<b>HKKO</b>	: Hibrit karınca kolonisi optimizasyonu
<b>KKO</b>	: Karınca kolonisi optimizasyonu
<b>KSY</b>	: Konik skalerleştirme yöntemi
<b>LNG</b>	: Sıvılaştırılmış doğal gaz
<b>NP</b>	: Non polinomial
<b>NSGA-II</b>	: Bastırılmamış sıralama genetik algoritması II
<b>PSO</b>	: Parçacık sürü optimizasyonu
<b>S/G</b>	: Sinyal gürültü oranı
<b>TB</b>	: Tavlama bnetimi
<b>UGKA</b>	: Uyarlanabilir geniş komşuluk arama yöntemi
<b>YAKA</b>	: Yapay arı kolonisi algoritması
<b>YARP</b>	: Yeşil araç rotalama problemi
<b>ZPEARP</b>	: Zaman pencereleli elektrikli araç rotalama problemi
<b>ZPKFEARP</b>	: Zaman pencereleli ve karma filolu elektrikli araç rotalama problemi





## SİMGELER

$\Omega$	: Arama uzayındaki uygun çözümler kümesi
$<$	: Baskınlık
$C$	: Müşteri noktaları kümesi
$R$	: Şarj istasyonlarına ait noktalar kümesi
$\{0\}$	: Depo
$V$	: Dğümler kümesi
$T$	: Bataryaların farklı şarj edilme teknolojileri seti
$K$	: Araçlar kümesi
$d_{ij}$	: $i$ noktasından $j$ noktasına olan uzaklık (km)
$as_{ij}$	: $i$ ve $j$ noktaları arasındaki beklenen ortalama hız (km/sa)
$t_{ij}$	: $i$ noktasından $j$ noktasına olan seyahat süresi (sa)
$E_{max}$	: Aracın batarya kapasitesi (KWsa)
$T_{max}$	: Rota süresi (sa)
$q_i$	: $i$ müşterisinin talep miktarı (ton)
$Q$	: Aracın yük kapasitesi (ton)
$s_i$	: $i$ müşterisinin servis süresi (sa)
$\gamma^*$	: Depodaki şarj ünitesinin birim ücreti (TL/KWsa)
$\gamma$	: Şarj ünitesinin birim ücreti (TL/KWsa)
$\gamma_t$	: $t$ tipi teknolojiye sahip şarj ünitesinin birim ücreti (TL/KWsa)
$\vartheta$	: Şarj etme işleminin sabit maliyeti (TL)
$g^*$	: Şarj ünitesinin ortalama şarj etme hızı (KWsa/sa)
$gt$	: $t$ tipi teknolojiye sahip şarj ünitesinin ortalama şarj etme hızı (KWsa/sa)
$\rho_0$	: EA'ların tam yüklü iken tüketeceği enerji oranı (KWsa/km)
$\rho^*$	: EA'ların yüksüz olarak tüketeceği enerji oranı (KWsa/km)
$p_k(i, j)$	: Yolların seçilme olasılığı
$\tau(i, j)$	: $i$ ve $j$ noktaları arasındaki feromon miktarı ve
$\eta(i, j)$	: $i$ ve $j$ noktaları arasındaki mesafeyle ters orantılı olan seçilebilirlik parametresi

$J_k(i)$	: i noktasından gidilebilecek ve henüz k karıncası tarafından ziyaret edilmemiş noktalar kümesi
$\alpha$	: Feromon miktarının önemini belirleyen sabit
$\beta$	: Sezgisel bilginin önemini belirleyen sabit
$\rho$	: Buharlaşma oranı
$T_0$	: Başlangıç sıcaklık
$T_f$	: Son sıcaklık
$\epsilon$	: Soğuma katsayısı
$SN$	: Besin kaynaklarının sayısı
$D$	: Optimize edilecek parametre sayısı
$x_j^{\min}$	: Parametrelerin alt sınırı
$x_j^{\max}$	: Parametrelerin üst sınırı
$failure_i$	: Geliştirilememe sayacı
$fitness_i$	: Besin kaynağının uygunluk değeri

## TABLO LİSTESİ

### Sayfa

<b>Tablo 2.1.</b> ÇAPSO'nun sözde kodu. ....	15
<b>Tablo 3.1.</b> Senaryoların karşılaştırılması.....	19
<b>Tablo 3.2.</b> ÇAEARP modeline ait notasyonlar, parametreler ve karar değişkenleri. ....	21
<b>Tablo 4.1.</b> EARP literatür incelemesi.....	35
<b>Tablo 5.1.</b> HKKO-YAKA'nın sözde kodu.....	55
<b>Tablo 5.2.</b> Müşterilerin talep miktarları (ton).....	57
<b>Tablo 5.3.</b> Ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki uzaklık miktarları (km). ....	57
<b>Tablo 5.4.</b> Örnek probleme ait bilgiler. ....	58
<b>Tablo 5.5.</b> HKKO sonucunda elde edilen rotaya ait amaç fonksiyonu değerleri. ....	59
<b>Tablo 5.6.</b> YAKA sonucunda elde edilen rotaya ait amaç fonksiyonu değerleri. ....	60
<b>Tablo 6.1.</b> S/G oranlarının hesaplanması. ....	62
<b>Tablo 6.2.</b> Belirlenen parametreler ve seviyeleri.....	63
<b>Tablo 6.3.</b> L18 ortogonal dizisi ve deney sonuçları. ....	64
<b>Tablo 6.4.</b> Parametrelerin en uygun seviyeleri ve değerleri.....	65
<b>Tablo 6.5.</b> S/G oranlarının varyans analizi.....	66
<b>Tablo 6.6.</b> Ortalama toplam maliyet ve hesaplama süresinin karşılaştırılması. ....	68
<b>Tablo 6.7.</b> Detaylı toplam maliyetlerin karşılaştırılması. ....	69
<b>Tablo 7.1.</b> Mağazaların 1. hafta talepleri (ton).....	73
<b>Tablo 7.2.</b> Ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki mesafeler (km). ....	73
<b>Tablo 7.3.</b> Senaryo 1 için amaç fonksiyonu değerleri. ....	75
<b>Tablo 7.4.</b> Senaryo 2 için amaç fonksiyonu değerleri. ....	75
<b>Tablo 7.5.</b> Senaryo 3 için amaç fonksiyonu değerleri. ....	76
<b>Tablo 7.6.</b> Rastgele seçilen amaç fonksiyonu ağırlık değerleri.....	77
<b>Tablo 7.7.</b> Farklı ağırlık değerleri için ATY ile elde edilen Pareto-optimal çözümler. ....	78
<b>Tablo 7.8.</b> Farklı ağırlık değerleri için KSY ile elde edilen Pareto-optimal çözümler. ....	79
<b>Tablo 7.9.</b> Senaryo 1'in HKKO-YAKA ile çözümünden elde edilen sonuçlar. ....	81
<b>Tablo 7.10.</b> Senaryo 2'nin HKKO-YAKA ile çözümünden elde edilen sonuçlar. ....	82
<b>Tablo 7.11.</b> Senaryo 3'ün HKKO-YAKA ile çözümünden elde edilen sonuçlar. ....	83
<b>Tablo 7.12.</b> Senaryo 1'in ÇAPSO ile çözümünden elde edilen sonuçlar.....	84
<b>Tablo 7.13.</b> Senaryo 2'nin ÇAPSO ile çözümünden elde edilen sonuçlar.....	85
<b>Tablo 7.14.</b> Senaryo 3'ün ÇAPSO ile çözümünden elde edilen sonuçlar. ....	86
<b>Tablo 7.15.</b> Senaryo 1'in uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar. ....	87
<b>Tablo 7.16.</b> Senaryo 2'nin uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar. ....	88

<b>Tablo 7.17.</b> Senaryo 3'ün uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar. .....	89
<b>Tablo 7.18.</b> Dört yöntem için varyansların homojenliği testi sonuçları. ....	90
<b>Tablo 7.19.</b> Dört yöntem için ANOVA Welch testi sonuçları. ....	90
<b>Tablo 7.20.</b> Dört yöntem için Tamhane post-hoc testi sonuçları.....	91
<b>Tablo 7.21.</b> Senaryolar için varyansların homojenliği testi sonucu. ....	92
<b>Tablo 7.22.</b> Senaryolar için ANOVA testi sonucu. ....	93
<b>Tablo 7.23.</b> Senaryolar için Tamhane post-hoc testi sonucu.....	93

## ŞEKİL LİSTESİ

### Sayfa

Şekil 2.1. İki amaçlı bir en küçükleme probleminin Pareto cephesi örneği (Peng, 2021). .....	9
Şekil 2.2. Dışbükey Pareto cephesinde ATY'nin gösterimi (Burke ve Kendall, 2014). .....	11
Şekil 2.3. Dışbükey olmayan görüntü kümesinin iki amaçlı bir problemde gösterimi (Sipahioğlu ve Saraç, 2010). .....	12
Şekil 2.4. Parçacığın arama uzayındaki hareketi (Faryad Aghjeh Kand, 2014). .....	14
Şekil 3.1. ÇAEARP'nin örnek gösterimi (Senaryo 1). .....	20
Şekil 3.2. ÇAEARP'nin örnek gösterimi (Senaryo 2). .....	26
Şekil 3.3. ÇAEARP'nin örnek gösterimi (Senaryo 3). .....	28
Şekil 5.1. Geliştirilen yöntemin çalışma prosedürü. ....	46
Şekil 5.2. Ters çevirme operatörünün işleyişinin gösterimi. ....	48
Şekil 5.3. Yer değiştirme operatörünün işleyişinin gösterimi. ....	48
Şekil 5.4. Ekleme operatörünün işleyişinin gösterimi. ....	49
Şekil 5.5. 2-opt algoritmasının işleyişinin gösterimi. ....	49
Şekil 5.6. 3-opt algoritmasının işleyişinin gösterimi. ....	50
Şekil 5.7. HKKO sonucunda oluşan EA'ların rotası. ....	59
Şekil 5.8. YAKA sonucunda oluşan EA'ların rotası. ....	60
Şekil 6.1. Parametrelerin seviyeleri için S/G oranları. ....	65



# ÇOK AMAÇLI ELEKTRİKLİ ARAÇ ROTALAMA PROBLEMİNİN HİYERARŞİK BİR METASEZGİSEL ALGORİTMA İLE ÇÖZÜMÜ

## ÖZET

Gerçek hayattaki rotalama planlarında, tek bir amaca dayanmadan birden fazla amacın optimize edilmeye çalışıldığı bir gerçektir. Bu doğrultuda, tez çalışmasında ürünlerin müşterilere dağıtım işleminin elektrikli araçlar (EA) ile yapıldığı araç rotalama probleminin (ARP) güncel bir çeşidi olan elektrikli araç rotalama problemi (EARP) çok amaçlı olarak ele alınmıştır. Ele alınan çok amaçlı elektrikli araç rotalama problemi (ÇAEARP), birbiriyle çelişen beş adet amaç fonksiyonu olacak şekilde düşünülmüş, buna ilaveten ÇAEARP, farklı şarj politikaları (tam veya kısmi) ve şarj teknolojileri (yavaş, orta veya hızlı) göz önünde bulundurularak üç farklı senaryo halinde modellenmiştir.

Ele alınan ÇAEARP'nin çözümü için ilk aşamada Hibrit Karınca Kolonisi Optimizasyonu (HKKO) ve ikinci aşamada ise Yapay Arı Kolonisi Algoritmalarının (YAKA) kullanıldığı iki aşamadan oluşan hiyerarşik bir yöntem (HKKO-YAKA) geliştirilmiştir. YAKA'da başlangıç çözümü rastgele üretildiği için problemin çözüm süresi olumsuz yönde etkilenebilmektedir. Problemin çözüm süresini hem kısaltmak hem de daha iyi sonuçlar elde edebilmek için geliştirilen yöntemin ilk aşamasında, yerel arama yöntemleriyle desteklenmiş ve kabul kriteri olarak da tavlama benzetimi kullanılan HKKO kullanılmıştır. İkinci aşamada ise birinci aşamadan elde edilen çözüm YAKA için başlangıç çözümü olarak ele alınıp EA'lar için nihai rotalar oluşturulmuştur.

HKKO-YAKA'nın performansının değerlendirilmesi için Felipe ve ark., (2014) tarafından geliştirilen test problemleri kullanılmıştır. Elde edilen sonuçlar geliştirilen yöntemin makul sürede iyi çözümler bulmada etkili olduğunu göstermektedir. Ayrıca, geliştirilen HKKO-YAKA'nın bir gerçek hayat ÇAEARP üzerinde uygulanabilirliği gösterilmiştir. HKKO-YAKA, Pareto-optimal çözümleri bulmak amacıyla ağırlıklı toplam yöntemi (ATY) ve konik skalerleştirme yöntemi (KSY) için arama işlemi olarak kullanılmıştır. KKO-YAKA'nın gerçek hayat problemi üzerindeki performansını değerlendirmek için çok amaçlı evrimsel algoritmalarından olan çok amaçlı parçacık sürü optimizasyonu (ÇAPSO) ve uyarlanmış ÇAPSO adını verdiğimiz başka bir ÇAPSO kullanılmıştır. Sonuç olarak, çok amaçlı geleneksel yöntemler çok amaçlı evrimsel algoritmalara göre daha etkili sonuçlar verdiği görülmekte ve kısmi şarj politikası ile çoklu şarj teknolojisi seçeneklerinin lojistik şirketlerinin rota kararlarını önemli ölçüde iyileştirebileceği görülmüştür.





# **THE SOLUTION TO THE MULTI-OBJECTIVE ELECTRIC VEHICLE ROUTING PROBLEM USING A HIERARCHICAL METAHEURISTIC ALGORITHM**

## **SUMMARY**

In real-life routing plans, more than one objective is tried to be optimized without relying on a single objective. In this direction, the electric vehicle routing problem (EVRP), which is a current version of the vehicle routing problem (VRP), in which the products are distributed to the customers with electric vehicles (EV), was discussed in the thesis study. The multi-objective electric vehicle routing problem (MOEVRP) was considered to have five conflicting objective functions. The first is minimizing the total charging cost, which consists of fixed and variable costs. Because batteries allow a limited number of recharge cycles during their operating life, a fixed cost is associated with each recharge. On the other hand, variable cost is the sum of the cost of charging the batteries during the day and at night. The others minimize the number of vehicles required; the total traveled distance, the load-dependent energy consumption, and the number of charging stations required. In addition, MOEVRP was modeled in three different scenarios, considering different charging policies (full or partial) and charging technologies (slow, medium, or fast).

A hierarchical approach consisting of two stages (HACO-ABCA) was developed, in which a hybrid ant colony optimization (HACO) was used in the first stage, and an artificial bee colony algorithm (ABCA) was used in the second stage for the solution of the MOEVRP. Since the initial solution is randomly generated in ABCA, the solution time of the problem may be adversely affected. HACO was used in the first stage of the developed approach to shorten the solution time and obtain better results. Local search methods supported the HACO, and simulated annealing was used as the acceptance criterion. In the second stage, the solution obtained from the first stage was considered the initial solution for ABCA, and final routes were created for the EVs.

To test the performance of the developed HACO-ABCA, we used the benchmark instances presented by Felipe et al. (2014). The benchmark instances consist of 60 data sets. The results were compared with the TB-based algorithm and the metaheuristic method proposed by Felipe et al. (2014) and Keskin and Çatay (2018), respectively. Comparisons are made in terms of total charging costs. The results show that our HACO-ABCA improves all the solutions and forty-seven instances found by Felipe et al. (2014) and Keskin and Çatay (2018), respectively.

Also, the applicability of the developed HACO-ABCA on a real-life MOEVRP was demonstrated. The objective is to determine the most feasible routes for vehicles to meet the weekly store demands of a supermarket chain in Turkey. HACO-ABCA was used as the search engine for the weighted-sum method (WSM) and the conic method (CM) to find Pareto-optimal solutions. First, five objective functions were

scalarized separately with WSM and CM, and then Pareto-optimal solutions were found with HACO-ABCA. Multi-objective particle swarm optimization (MOPSO), one of the multi-objective evolutionary algorithms, and another MOPSO, which we call the modified MOPSO, were used to evaluate the performance of the HACO-ABCA on the real-life problem. In MOPSO, the initial solution is randomly generated. This situation negatively affects the solution quality. In the modified MOPSO, the initial solution is determined by the developed hierarchical approach, and then the solution is continued with the MOPSO, and the final solution is found. The obtained results showed that the results obtained by the developed hierarchical approach produced better results than MOPSO and modified MOPSO in three scenarios. If we evaluate the hierarchical approach, better performance was obtained when using CM compared to WSM. Apart from comparing the methods, it can also be concluded that the partial recharge and multiple recharge technology options can significantly improve the route decisions of logistic companies.

## 1. GİRİŞ

Yöneylem araştırması perspektifinden bakıldığında, geleneksel lojistik faaliyeti esas olarak maliyetleri düşürmeye odaklanmaktadır. Geleneksel lojistik üzerine yapılan araştırmalar ve uygulamalar geçtiğimiz yıllarda daha kabul edilebilir sonuçlar elde ederken, ekonomik büyümenin getirdiği çevresel sorunlardan özellikle dünyadaki toplam sera gazı emisyonlarının yaklaşık %25'inden taşımacılık sektörünün sorumlu olduğu ve bu orana en büyük katkıyı %75 oranla karayolu taşımacılığının sağladığı düşünüldüğünde çeşitli farklı bakış açılarından olayların incelenmesi zarureti doğmuştur (Muller, 2022; IEA, 2019).

Bu doğrultuda petrol bazlı yakıtların çevreye verdiği zararları önlemek ve daha yaşanabilir bir dünya oluşturmak için dünya genelinde hükümetler güneş, elektrik, biyodizel, sıvılaştırılmış doğal gaz (LNG), sıkıştırılmış doğal gaz (CNG) araçları gibi alternatif yakıtlı araçların (AYA) kullanımını teşvik etmek için yeni düzenlemeler getirmektedir. Çok sayıdaki belediye, kamu kurum ve kuruluşları ve özel şirketler gönüllü olarak bu harekete destek vererek araç filolarına AYA'ları dâhil etmektedirler (Erdoğan ve Miller-Hooks, 2012). Örnek vermek gerekirse, lojistik sektöründe öncü bir şirket olan FedEx bunlardan bir tanesidir (FedEx, 2018).

Diğer yandan, çevrenin korunması ve enerji tasarrufundaki artan endişeler sonucunda, dünyadaki elektrikli araç (EA) teknolojisinin gelişimi bu ihtiyaçları karşılamak için hızlanmıştır. EA'lar karbon emisyonlarını azaltmak için iyi bir alternatif ulaşım çözümü sunmaktadır. EA'ların sıfır emisyon, düşük gürültü seviyesi, düşük enerji maliyetleri ve düşük bakım maliyetleri gibi sebeplerle alanında benzersiz oluşu, birçok araştırmacının ilgilisi çeken sıcak bir konu haline gelmiştir (Dinger ve ark, 2010). Bununla birlikte günümüz teknolojisinde EA'ların sınırlı pil ömrü, yüksek maliyet ve çeşitli şarj teknolojileri gibi sınırlamaları olmaktadır. Dolayısıyla rotalama problemlerinde kullanılan bir EA, konvansiyonel araçlara göre filo büyüklüğü ve rota kararlarında farklılıklara neden olmaktadır. Mevcut durumda batarya şarj istasyonlarının konvansiyonel benzin istasyonlarına göre çok az olması

ve batarya şarj miktarının sınırlı mesafelere izin vermesi nedeniyle EA'ların rota kararlarında bu kısıtın özellikle ele alınması zorunluluk gerektirmektedir.

Lojistik alanında, müşterilere yapılacak dağıtım işlemleri veya tedarikçilerden fabrikaya malzeme taşınması gibi taşıma işlemleri genellikle Araç Rotalama Problemleri (ARP) olarak ele alınmaktadır. İlk olarak Dantzig ve Ramser (1959) tarafından literatüre kazandırılmış olan ARP, bir araç filosu tarafından müşterilere hizmet vermek için depoda başlayan ve depoda biten en uygun rotaları bulmayı amaçlamaktadır. Son 50 yılda, ARP'nin çeşitleri ve çözüm yöntemlerine ilişkin kapsamlı çalışmalar yapılmıştır (Clarke ve Wright, 1964; Laporte, 1992; Toth ve Vigo, 2001; Golden ve ark, 2008). Elektrikli Araç Rotalama Problemi (EARP), her biri talep ve hizmet süresine sahip bir EA filosu, şarj istasyonları ve müşterilerden oluşur. EA'lara alternatif olan AYA'ları içeren ARP, literatürde ilk olarak Erdoğan ve Miller-Hooks (2012) tarafından sunulmuş ve yeşil araç rotalama problemi (YARP) olarak adlandırılmıştır. YARP'de, araçların sınırlı bir yakıt kapasitesi ve turu sırasında alternatif yakıt istasyonlarında yakıt ikmali yapma olasılığı göz önünde bulundurulmaktadır. Çözdükleri problemlerinin amacı, müşteriler arasında seyahat edilen toplam mesafeyi en küçükmektir. Daha sonraki çalışmalarda, müşterilerin zaman aralığı, farklı şarj teknolojilerinin kullanılma ve kısmi şarjlara izin verilmesi gibi bazı gerçekçi kısıtlamalar ele alınarak farklı varyasyonlar dikkate alınmıştır (Schneider ve ark, 2014; Felipe ve ark, 2014; Keskin ve Çatay, 2016).

Birçok gerçek dünya problemi, birden fazla ve birbiriyle çelişen hedeflerin varlığı ile karakterize edilmesine rağmen EARP çalışmalarında çoklu nesnellik kavramı dikkate alınmamıştır. Operasyonel, taktik ve stratejik hedeflere ulaşılırken gerçek hayattaki dağıtım ağlarının çok daha karmaşık ve çok yönlü olduğunu kabul etmek önemlidir (Melian-Batista ve ark, 2014). Müşterilere ürünlerin dağıtım faaliyetlerinde toplam katedilen mesafe dolayısıyla da dağıtım maliyetinin en küçüklenmesi işletmelerin temel amacı olsa dahi amaçlar sadece mesafe veya maliyeti en küçükmek ile sınırlı değildir. Örneğin, müşterilere ürün dağıtımını yapacak araç sayısı azaldığında toplam katedilen mesafe artmaktadır. Gidilen mesafedeki bu artış enerji tüketimini de artıracaktır. Dolayısıyla bu araçların bir rota üzerinde bir veya daha fazla şarj istasyonunu ziyaret etmesi gerekmektedir. Böylece toplam şarj maliyetinin artması beklenir. Bu durum, ARP'nin çok amaçlı olarak ele alınıp çözülmesi gerektiğini

göstermektedir. Çok amaçlı problemlerde amaçlar birbiriyle çeliştiği için bir amaçtaki iyileşme diğer amacın kötüleşmesine neden olabilir. Bu nedenle, bu problemlerde tek bir optimal çözüm yerine Pareto-optimal çözüm kümesi bulunarak bu küme içerisinde karar vericinin seçim yapmasına imkan verilmektedir.

Literatürde ARP'lerin çözümü için geliştirilen yöntemler kesin, klasik sezgisel ve metasezgisel olmak üzere üçe ayrılmaktadır. Klasik ARP'ler NP-zor problem sınıfından olduğu için EARP'lerde NP-zor problem olarak ele alınmaktadır (Lenstra ve Kan, 1981). Bazı küçük boyutlu problemler kesin yöntemler kullanılarak çözülebilmekte, ancak problemin boyutu büyüdükçe çözüme kesin yöntemlerle ulaşmak oldukça zor hatta bazen imkânsız olmakta ve çözüm süresinde önemli ölçüde artmaktadır (Desrochers ve ark, 1992). Ancak, tüm problemleri en iyi şekilde çözebilen tek bir algoritma mevcut değildir. Her algoritmanın kendine göre avantajları olmakla birlikte bazı dezavantajları da bulunmaktadır. Son zamanlarda metasezgisel algoritmalarda farklı teknikleri birleştirerek, yani "hibrit" ve kademeli, yani "hiyerarşik" algoritmalar geliştirmek yeni bir trend haline gelmiştir. Çeşitli algoritmaların avantajlarını kullanarak ve bunları uygun şekilde birleştirerek, algoritmanın dezavantajlarının etkisi en aza indirilebilmekte ve böylece geliştirilen yeni algoritmanın genel etkisi iyileştirilebilmektedir. Genel olarak, hibrit algoritmaların mevcut fikri, daha karmaşık optimizasyon problemlerini çözebilmek için farklı optimizasyon yöntemlerinin avantajlarını kullanarak birleştirmektir (Zhou ve ark, 2011).

### **1.1. Tezin Amacı ve Kapsamı**

Bu çalışmanın amacı farklı gerçekçi kısıtları göz önünde bulundurarak çok amaçlı elektrikli araç rotalama problemini (ÇAEARP) literatüre kazandırmaktır. ÇAEARP, EA'ların şarj politikaları (tam veya kısmi) ve şarj teknolojileri (yavaş, orta veya hızlı) açısından 3 farklı senaryo olarak ele alınmaktadır. ÇAEARP'nin birbiri ile çelişen beş adet amaç fonksiyonu mevcuttur. Bunlar; toplam şarj etme maliyetinin en küçüklenmesi, gerekli olan EA sayısının en küçüklenmesi, toplam seyahat mesafesinin en küçüklenmesi, enerji tüketiminin en küçüklenmesi ve uğranması gereken şarj istasyonu sayısının en küçüklenmesi şeklinde planlanmıştır. Çalışmada problemin çözümü için iki aşamadan oluşan yeni bir hiyerarşik yöntem (HKKO-

YAKA) (Hibrit Karınca Kolonisi Optimizasyonu (HKKO) ve Yapay Arı Kolonisi Algoritması (YAKA)) geliştirilmiştir. Geliştirilen hiyerarşik yöntem (HKKO-YAKA), ÇAEARP'nin çözümünde Pareto-optimal çözümleri bulmak amacıyla Ağırlıklı Toplam Yöntemi (ATY) ve Konik Skalerleştirme Yöntemi (KSY) için arama işlemi olarak kullanılmıştır.

## 1.2. Tezin Literatüre Katkısı

Bu tez çalışmasının dört farklı temel katkısı mevcuttur. Bunlardan birincisi, EARP'nin beş adet birbiri ile çelişen amaç fonksiyonlu varyantının, farklı şarj politikaları (tam veya kısmi) ve şarj teknolojileri (yavaş, orta veya hızlı) göz önünde bulundurularak modellenmesidir.

İkincisi ise beş farklı amaç fonksiyonundan biri olan enerji tüketimini en küçükmek hedefini sağlarken aracın yük durumunun da ele alınmasıdır. Literatürde enerji tüketimi hesaplanırken çoğu EARP çalışması araçtaki yük miktarını dikkate almamaktadır. Bu tür çalışmalar, kat edilen mesafeye bağlı olarak sabit enerji tüketim oranını ele almaktadır. Ancak gerçek hayatta enerji tüketimi hesaplanırken araçtaki yük miktarı dikkate alınması daha gerçekçi olacaktır.

Üçüncü olarak, geliştirilen problemin çözümü için iki aşamadan oluşan hiyerarşik bir yöntem (HKKO-YAKA) geliştirilmiştir. Problemin başlangıç çözümünün elde edildiği birinci aşamada HKKO önerilmiştir. Önerilen HKKO, karınca kolonisi optimizasyonu çerçevesi altında komşuluk arama kapasitesini genişletmek için yerel arama metotlarıyla desteklenerek ve karar verme aşamasında ise Tavlama Benzetimi (TB) yöntemini kullanarak geliştirilmiştir. Yerel aramanın algoritmaya etkisinin analizi Ek A'da verilmiştir. Geliştirilen hiyerarşik yöntemin ikinci aşamasında ise birinci aşamadan elde edilen çözümler dikkate alınarak problem YAKA ile çözülmüştür. Başlangıç çözümünün seçimi, sonuçların kalitesini doğrudan etkilediği için algoritmanın önemli bir adımıdır. HKKO, daha iyi bir çözümle aramaya başlayan YAKA'nın hesaplama süresini önemli ölçüde azalttığı Ek B'de ayrıntılı bir şekilde verilmiştir. Ek olarak, Felipe ve ark. (2014) tarafından geliştirilen test problemleri üzerinde HKKO-YAKA'nın etkinliği doğrulandı ve çoğu problem için en iyi yeni sonuçlar elde edilmiştir.

Son olarak geliştirilen yöntem gerçek hayat problemleri üzerinde test edilmiştir. Örnek olay çalışması sadece geleneksel yöntemler ile çok amaçlı evrimsel algoritmaların karşılaştırılması açısından kapsamlı bir çalışma sunmakla kalmaz, aynı zamanda ÇAEARP’de kısmi şarj politikasının ve çoklu şarj teknolojisinin kullanımının işletmelerin yönlendirme kararları üzerindeki etkilerini göstermesi açısından da literatüre katkılar sağlamaktadır.

### **1.3. Tezin Organizasyonu**

Yapılan tez çalışması 8 başlıkta ele alınmıştır.

Bölüm 2’de çok amaçlı optimizasyon konusu ele alınmış çok amaçlı optimizasyon problemi ve çözüm yöntemleri hakkında genel bilgiler verilmiştir.

Bölüm 3’te ilk olarak EARP irdelenmekte daha sonra ÇAEARP’nin çıkış noktalarından biri olan çok amaçlı yeşil araç rotalama problemi (ÇAYARP) açıklanmıştır. Son olarak ise tezin konusu olan ÇAEARP üç farklı senaryo altında incelenmiş ve matematiksel modellerine yer verilmiştir.

Bölüm 4’te EARP ve ÇAYARP ile ilgili literatür araştırması sunulmuştur.

Bölüm 5’te geliştirilen hiyerarşik yöntem ve çözüm aşamaları açıklanmıştır.

Bölüm 6’da, parametre tasarımı ve seçilen test problemlerinin geliştirilen yöntem ile çözümüne ilişkin sonuçlar verilmiştir.

Bölüm 7’de ise geliştirilen yöntem gerçek hayat problemleri üzerinde test edilmiştir. Örnek olay çalışması ile geleneksel yöntemler ve çok amaçlı evrimsel algoritmaların karşılaştırılmasının yanında üç farklı senaryonun da karşılaştırılması sunulmuştur.

Çalışmanın son bölümünde ise elde edilen bulgular tartışılmış ve gelecekte yapılacak çalışmalar için önerilerde bulunulmuştur.





## 2. ÇOK AMAÇLI OPTİMİZASYON PROBLEMİ

Bu bölümde, çok amaçlı optimizasyon problemlerinin temel kavramları ve çok amaçlı optimizasyon problemlerinin çözüm yöntemleri anlatılacaktır. Tez çalışmasında kullanılan çözüm yöntemleri ile ilgili ise detaylı bilgiler verilecektir.

### 2.1. Çok Amaçlı Optimizasyon

Gerçek hayatta özellikle mühendislik alanında optimizasyon problemlerinin çoğu birden fazla çatışan amacın eş zamanlı olarak sağlanmasını gerektirmektedir. Bu durum bir amaç değerinin iyileştirilmesinin diğer hedeflerin bozulmasına neden olabileceği anlamına gelmektedir. Tek amaçlı optimizasyonda olduğu gibi tek bir optimal çözüm bulmak yerine, çok amaçlı optimizasyon problemleri, birbiriyle çatışan tüm amaçları aynı anda sağlayabilmek için bir dizi uzlaşmayı temsil eden takas çözümüne sahiptir. Çok amaçlı bir optimizasyon problemi matematiksel olarak Denklem 2.1.'deki gibi ifade edilmektedir (Rao, 1991):

$$\begin{aligned} & \text{(Enküçük) } F_m(X) \quad m = 1,2,3, \dots, M \\ & g_i(x) \leq 0, \quad i = 1,2,3, \dots, I \\ & h_j(x) = 0, \quad j = 1,2,3, \dots, J \\ & X = (x_1, x_2, x_3, \dots, x_N)^T \\ & x_n^{(L)} \leq x_n \leq x_n^{(U)} \end{aligned} \tag{2.1}$$

Denklem 2.1'de verilen  $F_m(X)$  M adet amaç fonksiyonunu,  $g_i(x)$  ve  $h_j(x)$  sağlanması gereken kısıtları,  $X$  karar değişkenleri vektörünü,  $x_n^{(L)}$  ve  $x_n^{(U)}$  ise sırasıyla (n. karar değişkeni için) alt ve üst sınırları ifade etmektedir.

Aşağıda, çok amaçlı optimizasyon problemleri için dört önemli tanım verilmiştir (Aharon, 1980).

- Pareto-baskınlık:

$x^0 = (x_1^0, x_2^0, \dots, x_n^0)$  ve  $x^1 = (x_1^1, x_2^1, \dots, x_n^1)$  iki karar vektörü olsun.  $x^0$ 'ın  $x^1$ 'e göre baskın olabilmesi için aşağıdaki iki koşulun sağlanması gerekmektedir. Bu koşullar aşağıda tanımlanmıştır.

- (1) Tüm amaç fonksiyonları için  $f(x^0)$  değerleri  $f(x^1)$  'den büyük değildir.
- (2)  $f(x^0)$ 'ın  $f(x^1)$ 'den küçük olduğu en az bir amaç fonksiyonu vardır.

Baskınlık durumunun matematiksel ifadesi denklem 2.2.'de verilmiştir.

$$\text{Eğer} \quad \begin{array}{l} \forall i \in \{1, 2, \dots, M\} \quad f_i(x^0) \leq f_i(x^1) \\ \wedge \exists j \in \{1, 2, \dots, M\} \quad f_j(x^0) < f_j(x^1) \end{array} \quad x^0 < x^1 \quad (2.2)$$

- Pareto-optimal çözüm:

$\Omega$  kümesinin hiçbir elemanı  $x^0$  çözümünü baskılayamıyorsa  $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$  çözüm vektörüne Pareto-optimal çözüm denir ve denklem 2.3.'teki gibi ifade edilmektedir.

$$\neg \exists x^1 \in \Omega: x^1 < x^0 \quad (2.3)$$

- Pareto-optimal çözüm seti:

$\Omega$  kümesindeki Pareto-optimal çözümlerin, karar vektörlerinden oluşan kümedir ve denklem 2.4.'teki gibi ifade edilmektedir.

$$PS = \{x^0 \in \Omega \mid \neg \exists x^1 \in \Omega: x^1 < x^0\} \quad (2.4)$$

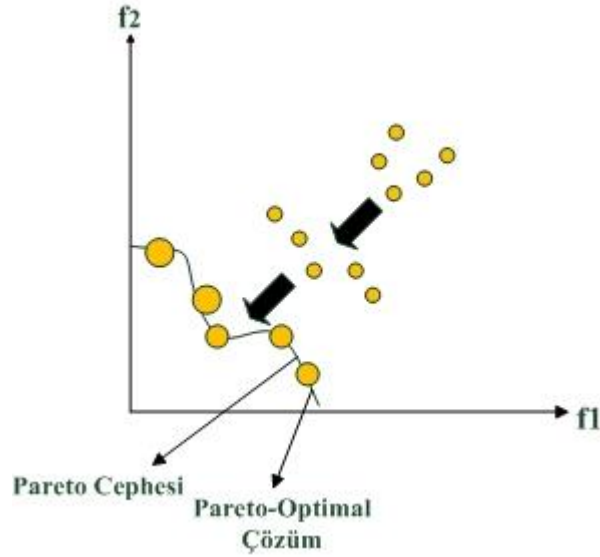
- Pareto Cephe:

Amaç uzayındaki Pareto-optimal çözümlerin kümesi Pareto cephesi olarak adlandırılır ve denklem 2.5.'teki gibi edilmektedir.

$$PF = \{F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \mid x \in PS\} \quad (2.5)$$

Pareto-optimal kümesindeki çözümler, karar uzayındaki başka herhangi bir çözüm tarafından baskılanamayan çözümlerdir. Çok amaçlı optimizasyonun amacı, daha iyi yakınsama ve çeşitlilik ile baskılanamamış nihai çözümler elde etmektir. Burada yakınsama, hedef uzayda elde edilen baskılanamayan çözümlerin Pareto cepheye

mümkün olduğunca yakın çözümler bulması anlamına gelmektedir. Çeşitlilik ise baskılanamayan çözümlerin Pareto cephesinde mümkün olduğunca düzgün dağılması anlamına gelmektedir. Şekil 2.1.'de, iki amaçlı bir en küçükleme problemi için bir Pareto cephe örneği gösterilmektedir.



**Şekil 2.1.** İki amaçlı bir en küçükleme probleminin Pareto cephesi örneği (Peng, 2021).

## 2.2. Çok Amaçlı Optimizasyon Problemlerinin Çözüm Yöntemleri

Çok amaçlı optimizasyon problemlerinin çözüm yöntemleri literatürde iki grup altında incelenmektedir. Bunlar; geleneksel yöntemler ve çok amaçlı evrimsel algoritmalarıdır.

### 2.2.1. Geleneksel yöntemler

Geleneksel Yöntemler karar vericinin tercihlerini ifade eden karar verme sürecine dahil olduğu aşamaya göre dört kategoriye ayrılmaktadır. Bunlar; tercih bilgisinin kullanılmadığı, tercih bilgisini çözüm sürecinin sonunda, başında, çözüm süreci esnasında isteyen yöntemlerdir (Hwang ve Masud, 1979; Miettinen, 1999; Branke ve ark, 2008).

Tercih bilgisinin kullanılmadığı yöntemler:

Bu yöntemlerde çözüm sürecinde karar vericinin tercihleri dikkate alınmaz. Çok amaçlı optimizasyon problemi nispeten basit bir yöntem kullanılarak çözülür ve Pareto-optimal kümesinin tipik olarak 'ortasında' bir uzlaşma

çözümü bulmak amaçlanır. Bu yöntemlere örnek olarak global kriter yöntemi ve çok amaçlı yakınsal yığın yöntemi verilebilir.

Tercih bilgisini çözüm sürecinin sonunda isteyen yöntemler:

Bu yöntemler, Pareto-optimal kümenin olası tüm çözümlerini bulur. Pareto-optimal kümedeki en uygun çözüm karar vericinin tercihi ile belirlenir. Bu yöntemlere örnek olarak ağırlıklı toplam yöntemi, konik skalerleştirme yöntemi ve  $\epsilon$ -kısıt yöntemi verilebilir.

Tercih bilgisini çözüm sürecinin başında isteyen yöntemler:

Bu yöntemlerde, karar verici çözüm sürecinden önce tercihlerini ve beklentilerini belirtmelidir. Elde edilen çözüm tatmin edici ise, karar vericinin çözüm sürecine çok fazla zaman harcaması gerekmez. Bu yöntemlerde karşılaşılan zorluk ise problemin sonucu önceden bilinmediği için karar vericinin beklentilerinin ne kadar gerçekçi olduğunun önceden bilinmemesidir. Bu yöntemlere örnek olarak hedef programlama yöntemi, değer fonksiyonu yöntemi ve sözlüksel sıralama yöntemi verilebilir.

Tercih bilgisini çözüm süreci esnasında isteyen yöntemler:

Bu yöntem sınıfına genellikle etkileşimli yöntemler denir. Çözüm uzayında arama yaparken aynı anda karar vericilerin tercihleri hakkındaki bilgilerine güvenirler. Karar verici problemin çözümü devam ederken problem hakkında daha fazla bilgi sahibi oldukça tercih bilgisini kullanır. Bu yöntemlere örnek olarak STEM yöntemi, Tchebycheff yöntemi, adım yöntemi, referans noktası yöntemi ve NIMBUS yöntemi verilebilir.

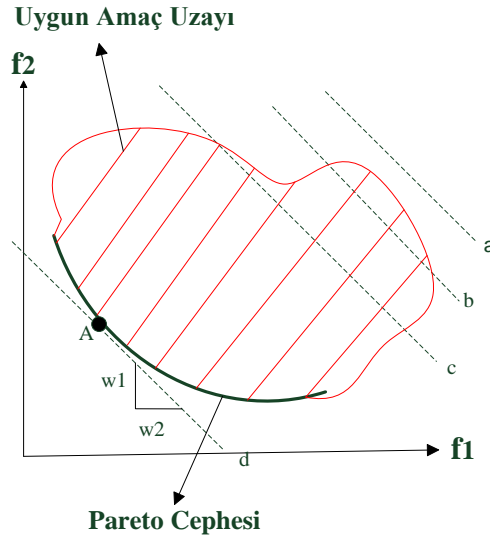
### **2.2.1.1. Ağırlıklı toplam yöntemi (ATY)**

Çok amaçlı optimizasyon problemlerinin çözümünde en yaygın kullanılan yöntemlerden biri olan ATY, 1955 yılında Gass ve Saaty tarafından geliştirilmiştir. Her bir amaç fonksiyonu bir ağırlık katsayısı ile ilişkilendirilir ve amaç fonksiyonlarının ağırlıklandırılmış toplamı ile bir fonksiyon oluşturularak optimize edilmeye çalışılır (Gass ve Saaty, 1955).

Her  $w_i$  ( $1 \geq w_i \geq 0$ ) ağırlığı bir  $f_i(x)$  ( $i = 1, \dots, n$ ) amaç fonksiyonu ile ilişkilendirildiğinde, skaler problem denklem 2.6.'da verilmiştir.

$$\text{Enk} \sum_{i=1}^n w_i f_i(x) \quad (2.6)$$

İki amaçlı bir optimizasyon probleminin Pareto-optimal çözümlerinin ağırlıklı toplam yöntemi ile nasıl bulunabileceği Şekil 2.2.'de gösterilmektedir. Uygun amaç uzayı ve karşılık gelen Pareto-optimal çözüm kümesi gösterilmiştir. İki amaç fonksiyonu olduğu için  $w_1$  ve  $w_2$  olmak üzere iki ağırlık vardır, ancak yalnızca biri bağımsızdır. Herhangi biri bilindiğinde diğeri basit çıkarma ile hesaplanabilmektedir ( $w_1 + w_2 = 1$ ). Bu seçilen ağırlık vektörü ile Pareto-optimal cephede önceden belirlenmiş bir optimal çözüm olan A noktasındaki çözüme karşılık geldiği görülmektedir. Ağırlık vektörü değiştirilerek farklı bir Pareto-optimal nokta elde edilebilir. Ayrıca ATY'yi uygulayabilmemiz için Şekil 2.2.'deki gibi en uygun çözüm alanının dışbükey küme ve  $f(x)$  amaç fonksiyonlarının da bu kümede dışbükey fonksiyonlar olması gerekmektedir (Ehrgott, 2005).



Şekil 2.2. Dışbükey Pareto cephesinde ATY'nin gösterimi (Burke ve Kendall, 2014).

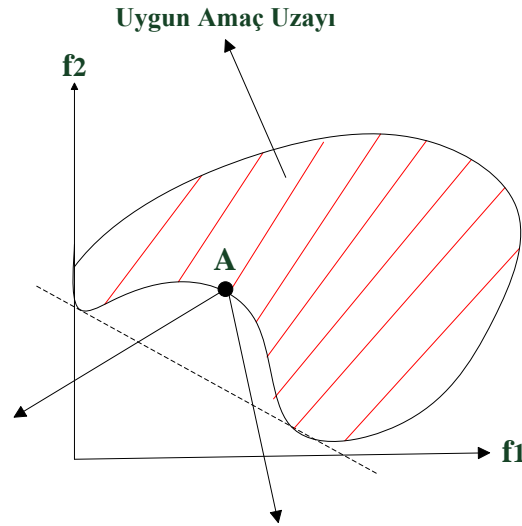
### 2.2.1.2. Konik skalerleştirme yöntemi (KSY)

KSY, Gasimov tarafından çok amaçlı optimizasyon problemlerinin çözümü için 2001 yılında geliştirilmiştir (Gasimov, 2001). Konik skalerleştirme tekniğinin dayandığı temel fikir, Pareto-optimal çözümlerin bulunmasında destek konilerin

kullanılmasıdır. Bu yöntem ile karar vericinin tercihlerini yansıtan, amaç fonksiyonu ağırlıkları ile referans noktası dikkate alınır ve yöntem referans noktasının seçimi konusunda bir kısıtlama getirmez. Referans noktaları ( $a_i$ ) isteğe bağlı olarak seçilebilir.  $\alpha$ , koninin tepe açısını belirleyen parametredir.  $0 \leq \alpha \leq \text{Enk } w_i$  şeklinde belirlenmektedir. Bu yöntemle göre skaler optimizasyon problemi denklem 2.7.'de verilmiştir (Üstün, 2007).

$$\text{Enk } \sum_{i=1}^n w_i (f_i(x) - a_i) + \alpha \sum_{i=1}^n |(f_i(x) - a_i)| \quad (2.7)$$

Şekil 2.3'te iki amaçlı bir örnek problemin dışbükey olmayan görüntü kümesi görülmektedir. Bu dışbükey olmayan kümedeki A noktasının bir doğru yardımı ile bulunması mümkün değildir. Fakat aynı nokta bir koni ile bulunabilmektedir. KSY'de noktalar konilerle desteklendiğinden, ATY ile bulunamayan Pareto-optimal çözümlerin elde edilmesi mümkün olabilmektedir. KSY, dışbükeylik koşulu gerektirmedikinden çok geniş bir problem sınıfına başarıyla uygulanabilmektedir (Gasimov, 2001).



**Şekil 2.3.** Dışbükey olmayan görüntü kümesinin iki amaçlı bir problemde gösterimi (Sipahioğlu ve Saraç, 2010).

### 2.2.2. Çok amaçlı evrimsel algoritmalar

Çok amaçlı evrimsel algoritmalar stokastik optimizasyon teknikleridir. Diğer optimizasyon algoritmalarına benzer şekilde, çok amaçlı evrimsel algoritmalar belirli

bir probleme ait Pareto-optimal çözümlerini bulmak için kullanılır, ancak popülasyona dayalı bir yaklaşım kullanarak farklılık gösterir. Çok amaçlı evrimsel algoritmalarda, her bir yinelemede objektif değerler her birey için hesaplanır ve daha sonra, yavru popülasyonunun üretimi için potansiyel olarak daha iyi çözümler seçmek amacıyla, toplum içindeki hakimiyet ilişkilerini belirlemek için kullanılır. Bu popülasyon, gelecek nesillere popülasyon üretmek için ana popülasyonla birleştirilebilir (Coello, 1999).

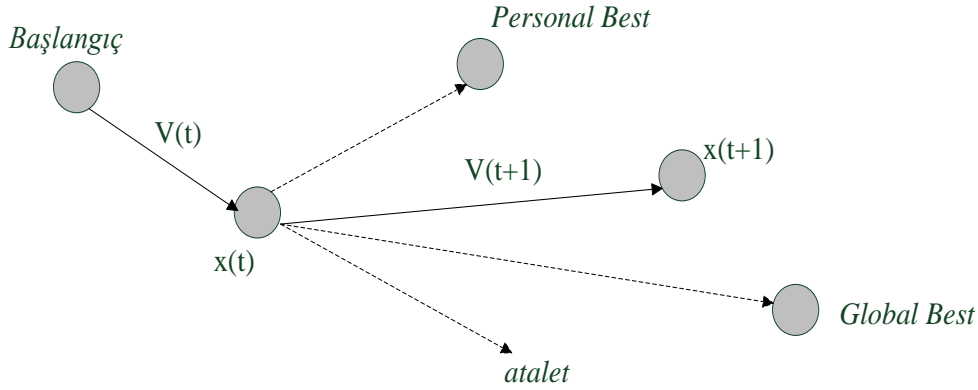
Çok amaçlı evrimsel algoritmalar kendi özelliklerinden dolayı, çok amaçlı optimizasyon problemlerini çözmek için diğer geleneksel matematiksel tekniklere göre daha uygundur. 1990'ların başından beri, araştırmacılar çok sayıda çok amaçlı evrimsel algoritma önermişlerdir. Sıkça kullanılan çok amaçlı evrimsel algoritmalara örnek olarak; vektör değerlendirmeli genetik algoritma, bastırılmamış sıralama genetik algoritması, çok amaçlı stokastik arama tekniği, bastırılmamış sıralama genetik algoritması II (NSGA-II), hücrelendirilmiş pareto genetik algoritması, çok amaçlı parçacık sürü optimizasyonu (ÇAPSO), çok amaçlı karınca kolonisi optimizasyonu (ÇAKKO), ayrıştırmaya dayalı çok amaçlı evrimsel algoritma ve çok amaçlı diferansiyel gelişim algoritması verilebilir (Qu ve ark, 2018).

#### **2.2.2.1. Çok amaçlı parçacık sürü optimizasyonu (ÇAPSO)**

ÇAPSO, 2004 yılında Coello ve arkadaşlarının üzerinde yaptığı çalışmalar sonucunda literatüre kazandırılmıştır (Coello ve ark, 2004). Tek amaçlı optimizasyonu problemlerinin çözümünde kullanılan parçacık sürü optimizasyonu (PSO) gibi ÇAPSO'da çok amaçlı optimizasyon problemlerini çözen genelleştirilmiş bir PSO dur. ÇAPSO, PSO ile aynı ilkeleri takip etmekle birlikte, bazı temel farkları bulunmaktadır. Bu nedenle ilk olarak PSO'nun anlaşılması daha uygun olacaktır.

PSO, Kennedy ve Eberhart (1995) tarafından önerilen popülasyon temelli metasezgisel bir optimizasyon tekniğidir. PSO'da, çözüm uzayındaki aramalar popülasyonu temel olarak yapılmaktadır. Popülasyona sürü, sürü elemanlarına ise parçacık denir. Popülasyondaki her bir parçacık probleme ait çözümü temsil etmektedir. Arama uzayında yeni çözümler bulmak için birtakım kurallar ve ilkeler tanımlanmıştır. Bunlardan en önemlisi hareket kavramıdır. Parçacıkların önceden gösterdiği tepkiler, parçacıkların arama uzayında şimdiye kadar buldukları en iyi

pozisyon ve tüm sürü elemanlarının şimdiye kadar buldukları en iyi pozisyon hareketin kaynağını bize göstermektedir.



**Şekil 2.4.** Parçacığın arama uzayındaki hareketi (Faryad Aghjeh Kand, 2014).

Şekil 2.4.'te bir parçacığın arama uzayındaki hareketi gösterilmiştir. Parçacık başlangıç pozisyonunda olduğunu varsayalım.  $v(t)$  hız vektörü ile parçacığın pozisyonu  $x(t)$  olarak değişmiştir. Parçacığın  $x(t)$ 'den sonra pozisyonunun değiştirilmesi istenirse üç farklı hareket gösterebilir. Bunlardan ilki  $x(t)$  tepkisine uyarak hareket ederse bu hareket *atalet* doğrultusunda olacaktır. İkincisi eğer kendi tecrübelerine göre hareket ederse parçacıkların arama uzayında şimdiye kadar buldukları en iyi pozisyon olan *Personal Best* doğrultusunda devam edecektir. Üçüncü olarak da eğer sürüdeki diğer parçacıkların tecrübelerine göre hareket ederse bu hareket tüm sürü elemanlarının şimdiye kadar buldukları en iyi pozisyon olan *Global Best* doğrultusunda olacaktır. Burada uygun görülen hareket yukarıda bahsedilen üç çeşit hareketin bir bileşimidir. Bu pozisyonda  $x(t+1)$ 'dir. Parçacığı  $x(t+1)$  pozisyonuna ulaştıracak  $v(t+1)$  hızının hesabı denklem 2.8.'te verilmiştir (Eberhart ve Kennedy, 1995; Shi ve Eberhart, 1999).

$$v_i(t + 1) = w * v(t) + c_1 * rand_1 * (P_i(t) - x_i(t)) + c_2 * rand_2 * (G(t) - x_i(t)) \quad (2.8)$$

Burada  $rand_1$  ve  $rand_2$  değerleri 0 ile 1 arasında rastgele üretilen sayıdır.  $w$  ise atalet ağırlık değerini göstermektedir.  $c_1$  ile  $c_2$  hızlandırma sabiti olarak adlandırılan pozitif iki sabit sayıdır.  $c_1$ , parçacığın kendi tecrübelerine göre hareket etmesini,  $c_2$  ise sürüdeki diğer parçacıkların tecrübelerine göre hareket etmesini sağlar. Hız



vektörünün güncellenmesinin ardından *i*. parçacığın pozisyon vektörünün elamanları denklem 2.9'a göre güncellenmektedir (Eberhart ve Kennedy, 1995; Shi ve Eberhart, 1999).

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2.9)$$

ÇAPSO ve PSO arasındaki fark en iyi *Personal Best* ve *Global Best*'in bulunmasıdır. ÇAPSO'ya depo kavramı eklenmiştir. Bu depo, baskın çözümleri kaydetmek için kullanılır. *Global Best* artık sabit bir nokta değildir ve her parçacık her hareket ettiğinde rastgele olarak depodan bir elemanını lider olarak seçmektedir. PSO'da tek bir amaç fonksiyonu vardır ve bir lider olarak kabul edilir. Ancak ÇAPSO birbirine üstünlüğün olmadığı birçok baskılanmamış vektör ve durumdan oluşmaktadır. Bu nedenle, lider olmaları için onlara eşit şans verilmelidir. Liderin aramaya rehberlik etmesi beklenir. Ayrıca her parçacığın farklı bir lider seçebileceği bir mekanizma uygulanmaktadır. Mekanizma, keşfedilen arama uzayının hiperküplere bölünmesine dayanmaktadır. Hiperküplere içerdikleri parçacıkların sayısıyla ters orantılı olarak bir uygunluk değeri atanır. Daha sonra klasik rulet çarkı seçimi, bir hiperküp ve ondan bir lider seçmek için kullanılır (Knowles ve Corne, 2000). ÇAPSO'nun sözde kodu, Tablo 2.1'de verilmiştir.

**Tablo 2.1.** ÇAPSO'nun sözde kodu.

---

---

1:	<b>Başlat</b>
2:	<b>for</b> parçacık=1 to parçacık sayısı
3:	Parçacıkları rastgele konum ve hızla başlat
4:	Konum ve hız bilgilerine göre uygunluk değerlerini hesapla
5:	<b>end for</b>
6:	Depodaki parçacıklar için baskın çözümleri depola
7:	<b>while</b> maksimum iterasyon sayısına ulaşılmadı <b>do</b>
8:	<b>for</b> parçacık=1 to parçacık sayısı
9:	Depodan bir lider seç
10:	Parçacığın hızını ve konumunu güncelle
11:	Mutasyon uygula
12:	Yeni uygunluk değerlerini hesapla ve değerlendir

---

**Tablo 2.1. (Devamı) ÇAPSO'nun sözde kodu.**

---

13:     *Personal best*'i güncelle  
14:     **end for**  
15:     Depodaki yeni parçacıklar için baskın çözümleri güncelle  
16:     **end while**  
17:     **End**

---

### 3. PROBLEMİN TANIMI

ARP, bir veya birkaç depoya sahip ve her biri farklı bir yerleşimde bulunan belirli sayıda müşterilerin taleplerini karşılayabilecek en uygun araç rotalarını bulmayı amaçlayan problemlerdir. En önemli kombinatorial optimizasyon problemlerinden biri olan ARP, ilk olarak Dantzig ve Ramser (1959) tarafından literatüre kazandırılmıştır. Daha sonra Clarke ve Wright (1964) tarafından ARP'nin çözümü için tasarruf algoritması önerilmiş ve bu çalışmadan sonra ARP'ye ilgi daha da artmıştır.

ARP'lere marketlerden müşterilere siparişlerin teslimatı, endüstriyel atık toplama, okul servislerinin güzergâhının belirlenmesi, kargoların dağıtımı gibi çeşitli gerçek hayat uygulamalarında karşılaşılabilmektedir. Gerçek hayattaki ihtiyaçlar doğrultusunda klasik ARP'ye yeni değişken ve kısıtlar eklenerek ARP'nin farklı türleri literatüre kazandırılmıştır.

ARP'lerde kısıtların yanında diğer önemli konuda problemin amacının belirlenmesidir. Klasik ARP, bir firmanın araç filosu tarafından izlenen rotaların maliyet etkisine odaklanmaktadır. Oysaki lojistik politikaları tasarlanırken, maliyet etkisine ek olarak, çevresel ve sosyal etkiler de dikkate alınmalıdır. Sürdürülebilir lojistik klasik ARP modeline katkıda bulunarak çeşitli uygulamalarla yeni modeller ortaya çıkarmıştır. Yeşil lojistik bunlardan biridir. Yeşil lojistik, farklı yönlendirme stratejilerinden oluşan çevresel etkileri ölçmeyi, yakıt veya enerji tüketimini azaltmayı ve atık bertarafını yönetmeyi içermektedir (Sbihi ve Eglese, 2007). Mesafeye veya yolculuk süresine dayalı olarak çevresel tasarrufları tahmin eden önceki çalışmalardan farklı olarak Palmer tarafından 2007 yılında literatürde ilk kez çevresel sorunlar ARP'de ele alınmıştır. Bu ARP türü genel ismiyle yeşil araç rotalama problemi olarak adlandırılmaktadır. Bu problemlerden bazıları, enerji tüketimini hesaplayabilmek için karmaşıklık seviyelerine sahip elektrikli araçları hedeflemektedir. Geri kalanları ise dizel araçları hedeflemekte ve emisyon modellerini rotalamaya dahil etmektedir (Palmer, 2007).

Taşımacılığın iklim değışikliđi, çevre ve hava kirliliđi üzerindeki etkisi genellikle sera gazı emisyonları ile ilişkilendirilmektedir. Yıllar içinde içten yanmalı motorlardan kaynaklanan emisyonları azaltmak için çok çaba sarf edilmiştir. Bu çabaların bir sonraki adımını ise içten yanmalı motorları elektrikli aktarma organlarıyla değıştirerek egzoz emisyonlarını tamamen azaltmaktır. Elektrikli taşıtlar egzoz emisyonu yaymazlar, daha sessizdirler, daha basittirler, dolayısıyla bu da daha az bakıma ihtiyaç duyar anlamına gelmektedir. Ayrıca daha düşük yakıt maliyeti, hükümetlerin vergi teşvikleri ve şirketin “yeşil” imajını destekleme potansiyeli açısından uzun vadeli ekonomik ve çevresel faydalar sağlamaktadır (Pelletier ve ark, 2016).

EARP, her biri talep ve hizmet süresine sahip müşteriler ile şarj istasyonları ve elektrikli araçlardan oluşan bir filoyu içerir. Araç ilerledikçe bataryadaki şarj miktarı alınan yol ile orantılı olarak azalmaktadır. Bu nedenle, aracın rotasına devam edebilmesi için şarj istasyonlarına uğrayıp bataryasını şarj etmesi gerekebilmektedir. Elektrikli araçlara dönüşümü başlatmak, şarj istasyonları kurmak için yapılan yatırımlar, batarya kapasitesi, zamanlama ve rotaların yanı sıra trafik koşulları ve hava durumu da dâhil olmak üzere bir dizi ana faktöre bağlıdır. Literatürde bu dönüşümün gerçek hayattaki bir uygulaması Lee ve ark. (2013) tarafından yapılmıştır. Yapılan çalışmada New York şehir içinde bir ürün dağıtım problemi için elektrikli araçlara alternatif olarak dizel araçlar ele alınmış ve sonuçların karşılaştırılması yapılmıştır. Çalışma sonuçlarına göre elektrikli araçlar kullanıldığında sera gazlarında önemli bir azalma olduğu, daha az enerji tüketildiđi ve hatta farklı senaryolara göre toplam maliyette önemli bir azalma olduğu gözlenmiştir. Elektrikli araçlara olan dönüşüm zorlu fakat uzun vadede sağlayacağı birçok fayda açısından çok önemli bir süreç olduğunu söyleyebiliriz.

Tez kapsamında EARP beş adet birbiriyle çelişen amaç fonksiyonu ele alınarak çok amaçlı olarak sunulmuş ve ÇAEARP’de üç farklı senaryo şeklinde incelenmiştir. Bu bölümde ÇAEARP’nin tanımını ve varsayımları verilecek, probleme ait matematiksel model sunulacaktır.

### 3.1. Çok Amaçlı Elektrikli Araç Rotalama Problemi

Bu çalışmada, ÇAEARP’de birbiri ile çelişen beş adet amaç fonksiyonu kabul edilmiştir. Amaç fonksiyonları;

- Toplam şarj etme maliyetini en küçükmek,
- Gerekli olan EA sayısını en küçükmek,
- Toplam seyahat mesafesini en küçükmek,
- Enerji tüketimini en küçükmek,
- Uğranması gereken şarj istasyonu sayısını en küçükmektir.

Problem EA’ların şarj politikaları ve şarj teknolojileri açısından 3 adet senaryo olarak ele alınmıştır. Senaryoların birbirlerinden farkları aşağıdaki Tablo 3.1’de özetlenmiştir.

**Tablo 3.1.** Senaryoların karşılaştırılması.

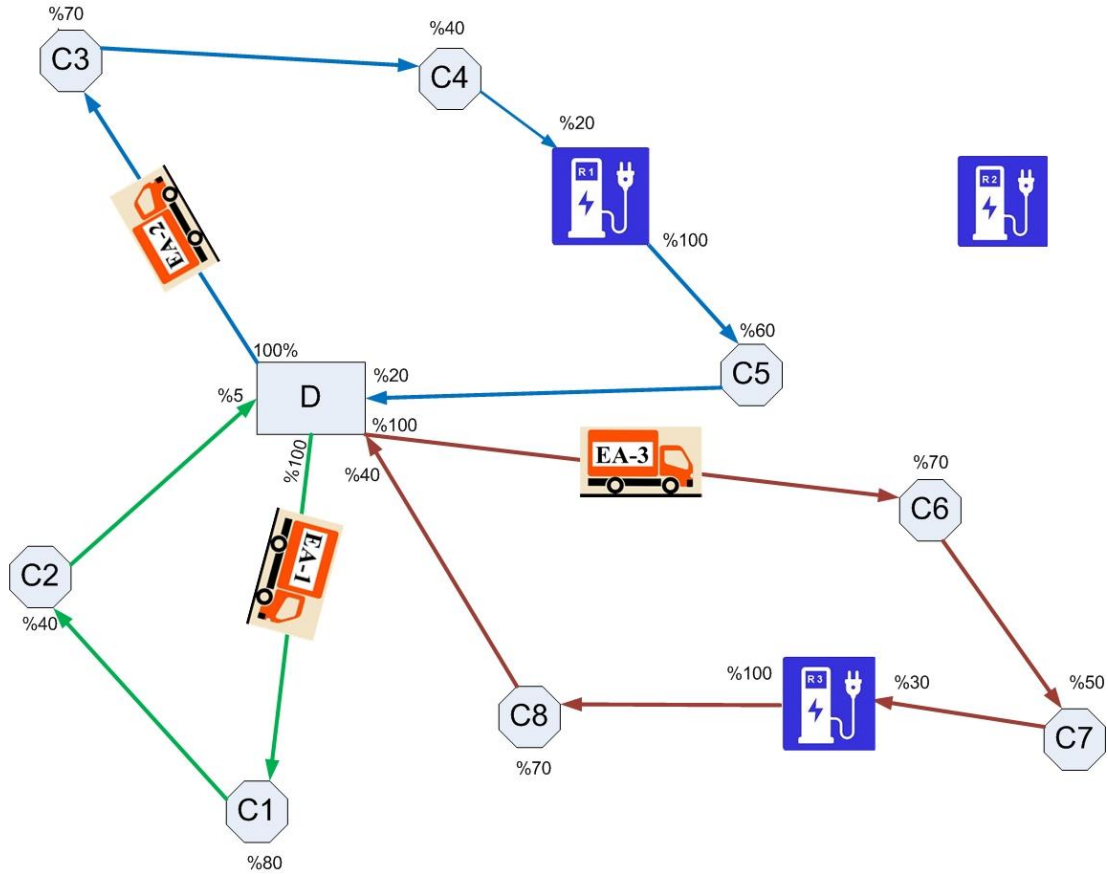
	Şarj Politikası	Şarj Teknolojisi
<b>Senaryo 1</b>	Tam	Normal
<b>Senaryo 2</b>	Tam veya Kısmi	Normal
<b>Senaryo 3</b>	Tam veya Kısmi	Hızlı ve Normal

#### 3.1.1. Senaryo 1

Senaryo 1 de ele alınan ÇAEARP; talep ve hizmet süresi olan müşterileri, şarj istasyonlarını ve bir elektrikli araç filosunu içermektedir. Her tur araçların bataryalarının tam dolu şarja sahip olduğu ( $E_{max}$ ) depoda başlar ve her tur  $T_{max}$  zaman sınırı içerisinde depoda son bulur. Aracın bataryasındaki şarj seviyesi sadece kat edilen mesafeye göre değil, aracın taşıdığı yük miktarına bağlı olarak da azalmaktadır. Dolayısıyla aracın rotasına devam etmesi için şarj istasyonlarını ziyaret etmesi gerekebilmektedir. Aracın depodan tam dolu şarj ile ayrıldığı gibi şarj istasyonundan da tam dolu batarya ile ayrılmaktadır. Bu çalışmada, Erdoğan ve Miller-Hooks (2012) tarafından önerilen matematiksel modelden farklı olarak şarj etme süreleri sabit değildir ve aracın şarj istasyona vardığında doğrusal olarak yeniden doldurulması gereken şarj miktarına bağlı olduğu varsayılmaktadır.

Şekil 3.1’de sekiz müşteriye (C1–C8), üç şarj istasyonunu (R1–R3) ve depoyu (D) içeren Senaryo 1’e ait bir ÇAEARP örneği gösterilmektedir. Rotalardaki yüzde

değerleri, araç bir müşteriye veya istasyona vardığında ve bataryasını doldurup şarj istasyonundan ayrıldığındaki bataryasının doluluk oranını göstermektedir. 1 numaralı elektrikli aracın (EA1) bataryasındaki şarj miktarı rotası boyunca tükettiği enerji miktarından daha fazla olduğu için herhangi bir şarj istasyonuna uğramasına gerek kalmamıştır. 2 numaralı elektrikli araç (EA2), 3. ve 4. müşteriye (C3 ve C4) hizmet verdikten sonra 1 numaralı şarj istasyonuna (R1) uğrayıp 5. müşteriye (C5) hizmet vermeden önce bataryasını tamamen şarj etmektedir. 3 numaralı elektrikli araç (EA3), 6. ve 7. müşteriye (C6 ve C7) hizmet verdikten sonra 3 numaralı şarj istasyonuna (R3) uğrayıp 8. müşteriye (C8) hizmet vermeden önce bataryasını tamamen şarj etmektedir.



**Şekil 3.1.** ÇAEARP'nin örnek gösterimi (Senaryo 1).

Aşağıda sunulan ÇAEARP matematiksel modelin bazı karar değişkenleri, notasyonları ve kısıtları Erdoğan ve Miller-Hooks (2012) ve Felipe ve ark. (2014) tarafından önerilen modellerde kullanılanları genişletmektedir. Geliştirilen ÇAEARP modelin parametreleri ve karar değişkenleri Tablo 3.2'de özetlenmiştir.

**Tablo 3.2.** ÇAEARP modeline ait notasyonlar, parametreler ve karar değişkenleri.

<b>Notasyonlar</b>	
C	Müşteri noktaları kümesi
R	Şarj istasyonlarına ait noktalar kümesi
{0}	Depo
V	Düğüm kümesi, $V = \{0\} \cup C \cup R$
T	Bataryaların farklı şarj edilme teknolojileri seti
K	Araçlar kümesi
<b>Parametreler</b>	
$d_{ij}$	i noktasından j noktasına olan uzaklık (km)
$as_{ij}$	i ve j noktaları arasındaki beklenen ortalama hız (km/sa)
$t_{ij}$	i noktasından j noktasına olan seyahat süresi (sa) ( $t_{ij} = d_{ij}/as_{ij}$ )
$E_{max}$	Aracın batarya kapasitesi (KWsa)
$E_{max}$	Rota süresi (sa)
$q_i$	i müşterisinin talep miktarı (ton)
Q	Aracın yük kapasitesi (ton)
$s_i$	i müşterisinin servis süresi (sa)
$\gamma^*$	Depodaki şarj ünitesinin birim ücreti (TL/KWsa)
$\gamma$	Şarj ünitesinin birim ücreti (TL/KWsa)
$\gamma_t$	t tipi teknolojiye sahip şarj ünitesinin birim ücreti (TL/KWsa)
$\theta$	Şarj etme işleminin sabit maliyeti (TL)
$g^*$	Şarj ünitesinin ortalama şarj etme hızı (KWsa/sa)
$g_t$	t tipi teknolojiye sahip şarj ünitesinin ortalama şarj etme hızı (KWsa/sa)
$\rho_0$	EA'ların tam yüklü iken tüketeceği enerji oranı (KWsa/km)
$\rho^*$	EA'ların yüksüz olarak tüketeceği enerji oranı (KWsa/km)
<b>Karar değişkenleri</b>	
$x_{ijk}$	Eğer EA i noktasından j noktasına giderse 1, aksi halde 0
$u_{ik}$	k aracının i düğüm noktasından ayrılırken ki yük seviyesi (ton)
$dt_{ik}$	k aracının i düğümüne varış zamanı (sa)
$f_{ik}$	k aracının i düğüm noktasından ayrılmadan önceki batarya şarj oranı (KWsa)

**Tablo 3.2. (Devamı)** ÇAEARP modeline ait notasyonlar, parametreler ve karar değişkenleri.

<b>Karar değişkenleri</b>	
$z_{jkt}$	k aracının bataryasının j düğümünde t tipi teknoloji kullanılarak şarj edildiği miktarı (KWsa)
$z_{jk}^*$	k aracının bataryasının j düğümünde şarj edildiği miktarı (KWsa)
$g_k$	k aracının bataryasının depoda şarj edildiği miktarı (KWsa)
$\Delta Rt_{ik}$	$(E_{\max} - f_{ik})/g^*$ olarak hesaplanan k aracının şarj edilme süresi fonksiyonu
$\Delta Rt_{ikt}$	$(E_{\max} - f_{ik})/gt$ olarak hesaplanan k aracının t tipi teknolojiye sahip şarj ünitesinde şarj edilme süresi fonksiyonu

Senaryo 1 de ele alınan ÇAEARP modelinin amaç fonksiyonları ve kısıt denklemleri şöyledir:

**Amaç fonksiyonları:**

$$\text{Enküçük} \sum_{k \in K} \gamma^* g_k + \sum_{k \in K} \sum_{j \in R} \gamma z_{jk}^* + \vartheta \sum_{k \in K} \sum_{i \in V, j \in R \cup \{0\}} x_{ijk} \quad (3.1)$$

$$\text{Enküçük} \sum_{k \in K} \sum_{j \in V/\{0\}} x_{0jk} \quad (3.2)$$

$$\text{Enküçük} \sum_{k \in K} \sum_{i, j \in V} d_{ij} x_{ijk} \quad (3.3)$$

$$\text{Enküçük} \sum_{k \in K} \sum_{i, j \in V} (\rho_0 + (\frac{\rho^* - \rho_0}{Q} u_{ik})) d_{ij} x_{ijk} \quad (3.4)$$

$$\text{Enküçük} \sum_{k \in K} \sum_{i \in V, j \in R} x_{ijk} \quad (3.5)$$

**Kısıtlar:**

$$\sum_{k \in K} \sum_{j \in V, i \neq j} x_{ijk} = 1 \quad \forall i \in C \quad (3.6)$$



$$\sum_{k \in K} \sum_{j \in V, i \neq j} x_{ijk} \leq 1 \quad \forall i \in R \quad (3.7)$$

$$\sum_{i \in V, i \neq j} x_{ijk} - \sum_{i \in V, i \neq j} x_{jik} = 0 \quad \forall i \in V, \forall k \in K \quad (3.8)$$

$$u_{jk} \leq u_{ik} - q_i x_{ijk} + Q(1 - x_{ijk}) \quad (3.9)$$

$$\forall i \in V \setminus \{0\}, \forall j \in V \setminus \{0\}, i \neq j, \forall k \in K$$

$$0 \leq u_{ik} \leq Q \quad \forall i \in \{0\}, \forall k \in K \quad (3.10)$$

$$u_{jk} \geq 0 \quad \forall j \in V \setminus \{0\}, \forall k \in K \quad (3.11)$$

$$\sum_{k \in K} \sum_{j \in V/\{0\}} x_{ojk} \leq K \quad (3.12)$$

$$\sum_{k \in K} \sum_{j \in V/\{0\}} x_{jok} \leq K \quad (3.13)$$

$$dt_{jk} \geq dt_{ik} + ((t_{ij} + s_i)x_{ijk}) - (T_{\max}(1 - x_{ijk})) \quad (3.14)$$

$$\forall i \in C, \forall j \in V, i \neq j, \forall k \in K$$

$$dt_{jk} \geq dt_{ik} + ((t_{ij} + \Delta R t_{ik})x_{ijk}) - (T_{\max}(1 - x_{ijk})) \quad (3.15)$$

$$\forall i \in R, \forall j \in V, i \neq j, \forall k \in K$$

$$dt_{jk} \geq t_{ij} - (T_{\max}(1 - x_{ijk})) \quad (3.16)$$

$$\forall i \in \{0\}, \forall j \in V, i \neq j, \forall k \in K$$

$$0 \leq dt_{ik} \leq T_{\max} \quad \forall i \in \{0\}, \forall k \in K \quad (3.17)$$

$$g_k \leq E_{\max} \quad \forall k \in K \quad (3.18)$$

$$f_{jk} \leq f_{ik} - (rd_{ij}x_{ijk}) + (E_{\max}(1 - x_{ijk})) \quad \forall i \in V, \forall j \in C \quad (3.19)$$

$$f_{jk} = E_{\max} \quad \forall j \in R \cup \{0\} \quad (3.20)$$

$$f_{jk} \geq \min\{rd_{j0}, r(d_{ji} + d_{i0})\} \quad \forall j \in C, \forall i \in R \quad (3.21)$$

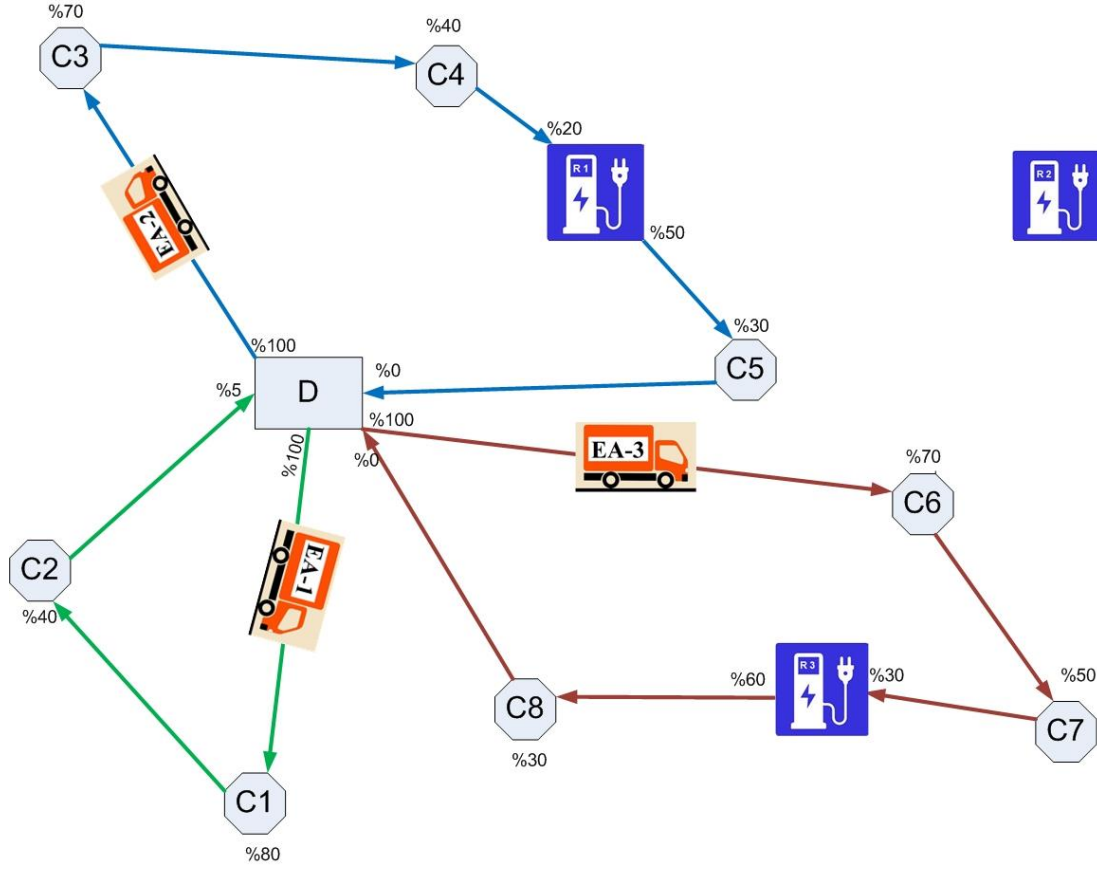
$$x_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (3.22)$$

Denklem 3.1 yani 1. amaç fonksiyonu toplam şarj etme maliyetinin en küçüklenmesini sağlamaktadır. Bu amaç fonksiyonundaki birinci terim gece boyunca depoda şarj edilen miktarın maliyetini, ikinci terim ise tüm istasyonlarda (depo dahil) gün boyunca şarj edilen miktarın maliyetini ve üçüncü terim, batarya çevrimiyle ilgili sabit maliyeti ifade etmektedir. 2. amaç fonksiyonu olan denklem 3.2, gerekli olan EA sayısının en küçüklenmesini, 3. amaç fonksiyonu olan denklem 3.3, toplam seyahat mesafesinin en küçüklenmesini, 4. amaç fonksiyonu olan denklem 3.4 yüke bağlı bir fonksiyon olarak kabul edilen enerji tüketiminin en küçüklenmesini ve 5. amaç fonksiyonu olan denklem 3.5 ise uğranması gereken toplam şarj istasyonu sayısının en küçüklenmesini sağlamaktadır. Denklem 3.6, tüm müşterilere bir kez hizmet verilmesini sağlamaktadır. Denklem 3.7, her bir şarj istasyonundan en fazla bir kez ana depo, müşteri veya şarj istasyonuna gidilmesini garantilemektedir. Denklem 3.8, ana depo, müşteriler ve şarj istasyonlarına gelen yolların sayısının ana depo, müşteriler ve şarj istasyonlarına giden yolların sayısına eşit olduğunu belirten bir akış kısıtıdır. Denklem 3.9, denklem 3.10 ve denklem 3.11 kapasiteye ilişkin kısıtlardır. Bu kısıtlarla birlikte herhangi bir rotada bir aracın en fazla kapasitesi kadar yük almasını sağlamaktadır. Bu kısıtlar aynı zamanda, problem için uygun olmayan turların (alt turların) oluşumunu da engelleme görevi görmektedirler. Denklem 3.12, gerekli olan araç sayısını hesaplayan kısıttır. Denklem 3.13, ana depodan müşterilere ve şarj istasyonuna olan akışın en az araç sayısı kadar olması gerektiğini ifade etmektedir. Denklem 3.14, denklem 3.15 ve denklem 3.16, belirli bir noktadan hareket eden aracın her bir noktaya (müşteri, şarj istasyonu ve depo) varış zamanını belirlemektedir. Denklem 3.17, her aracın  $T_{max}$  tur süresi içinde depoya geri dönmesini sağlamaktadır. Denklem 3.18, gece depoda şarj edilen miktarın batarya kapasitesini aşmamasını sağlamaktadır. Denklem 3.19, aracın turu boyunca şarj seviyesini belirlemektedir. Denklem 3.20, aracın depodan veya şarj istasyonundan tam şarj ( $E_{max}$ ) ile ayrılmasını sağlamaktadır. Denklem 3.21, her müşteri noktasından, komşu bir şarj istasyonuna ya da depoya ulaşmak için yeterli şarj miktarının olmasını sağlamaktadır. Denklem 3.22 ise  $x$  değişkeninin 0-1 tamsayı değerini almasını sağlamaktadır.

### 3.1.2. Senaryo 2

Senaryo 2’de ele alınan ÇAEARP, Senaryo 1’dekinin aksine şarj istasyonlarında kısmi şarja izin vermektedir. Yani, EA’lar depodan bataryaları tam dolu şarj seviyesiyle ayrılmakta fakat şarj istasyonlarından herhangi bir şarj seviyesiyle ayrılabilirler. Ayrıca araç bulunduğu rotada en az bir kez şarj istasyonuna uğrayıp bataryasını şarj edecekse boş bir batarya ile depoya dönecektir.

Şekil 3.2’de sekiz müşteriye (C1–C8), üç şarj istasyonunu (R1–R3) ve depoyu (D) içeren Senaryo 2’ye ait bir ÇAEARP örneği gösterilmektedir. Rotalardaki yüzde değerleri, araç bir müşteriye veya istasyona vardığında ve bataryasını doldurup şarj istasyonundan ayrıldığındaki bataryasının doluluk oranını göstermektedir. 1 numaralı elektrikli aracın (EA1) bataryasındaki şarj miktarı rotası boyunca tükettiği enerji miktarından daha fazla olduğu için herhangi bir şarj istasyonuna uğramasına gerek kalmamıştır. 2 numaralı elektrikli araç (EA2), 3. ve 4. müşteriye (C3 ve C4) hizmet verdikten sonra 1 numaralı şarj istasyonuna (R1) uğrayıp 5. müşteriye (C5) hizmet vermeden önce bataryasını kısmi olarak şarj etmektedir. Aynı şekilde, 3 numaralı elektrikli araçta (EA3) 6. ve 7. (C6 ve C7) müşteriye hizmet verdikten sonra 2 numaralı şarj istasyonunda (R3) bataryasını kısmi olarak şarj edip 8. müşteriye (C8) hizmet vermek için yola çıkmaktadır.



Şekil 3.2. ÇAEARP'nin örnek gösterimi (Senaryo 2).

Senaryo 2’de ele alınan ÇAEARP modeli, bazı eklemeler ve değişikliklerle Senaryo 1’deki matematiksel modelin amaç fonksiyonlarını ve kısıtlamalarını içermektedir. Denklem 3.20 ve denklem 3.21 kaldırılmakta ve aşağıdaki denklemler modele eklenmektedir.

$$f_{jk} = E_{\max} \quad \forall j \in \{0\}, \forall k \in K \quad (3.23)$$

$$f_{jk} \leq E_{\max} \quad \forall j \in R, \forall k \in K \quad (3.24)$$

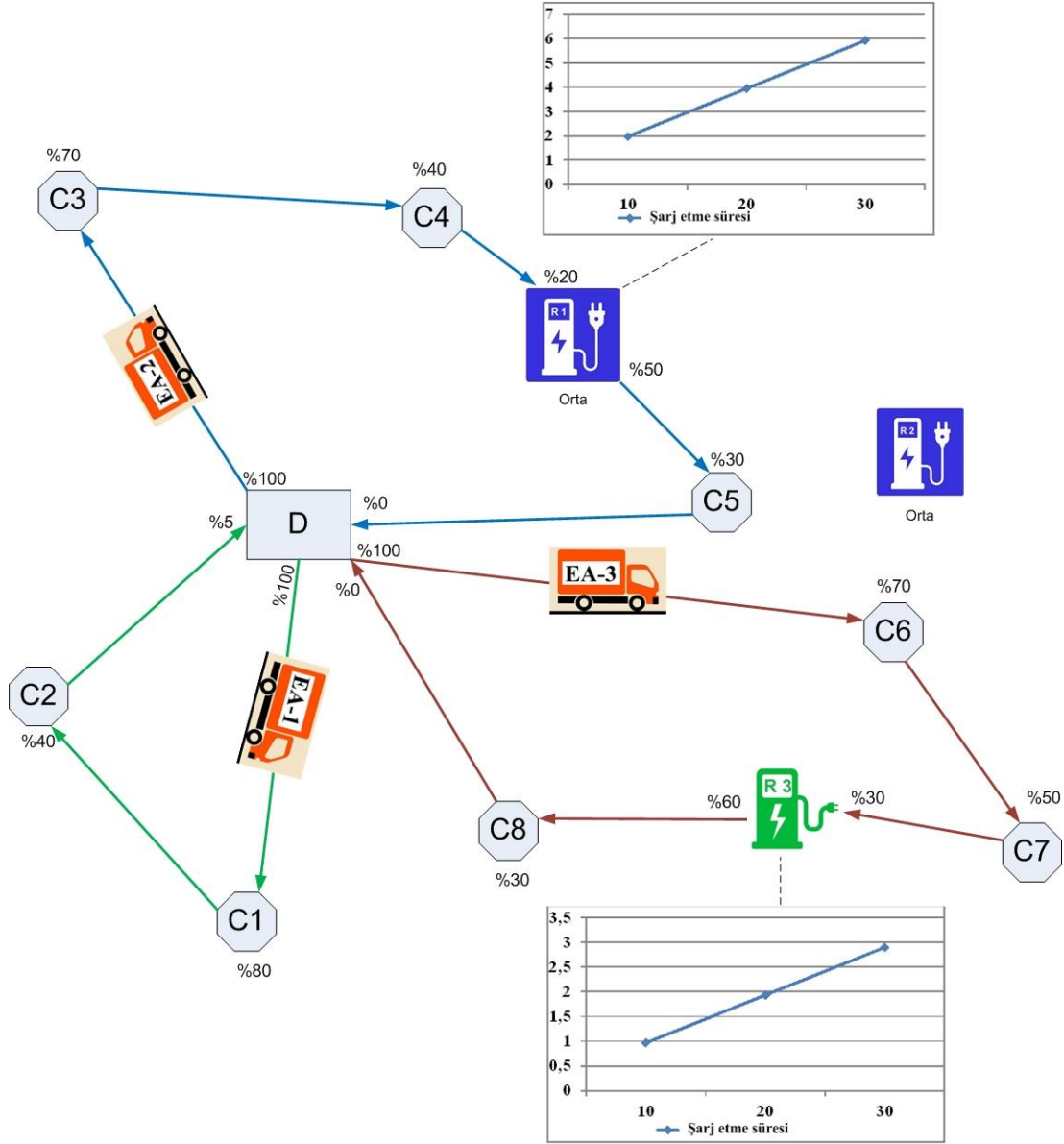
$$f_{jk} = f_{ik} - (rd_{ij}x_{ijk}) + z_{jk}^* \quad \forall j \in R, \forall i \in C, \forall k \in K \quad (3.25)$$

Denklem 3.23, aracın depodan tam şarj ( $E_{\max}$ ) ile ayrılmasını sağlamaktadır. Denklem 3.24, aracın şarj istasyonundan ayrılmadan önceki şarj seviyesinin batarya kapasitesini aşmamasını garantilemektedir. Denklem 3.25 ise aracın bir şarj istasyonundan ayrılırken ki şarj seviyesinin, vardığındaki şarj seviyesi ve şarj istasyonundaki şarj edilen miktarın toplamına eşit olmasını sağlamaktadır.

### 3.1.3. Senaryo 3

Senaryo 3'te ele alınan ÇAEARP, Senaryo 2'deki ÇAEARP'ye ek olarak birden fazla teknolojiye sahip şarj istasyonlarını da değerlendirerek çoklu şarj teknolojisini probleme dâhil etmiştir. Her şarj teknolojisinin, pratikte karşılaşılabileceğimiz çeşitli şarj cihazlarını yansıtan farklı bir şarj hızı vardır. Bu şarj teknolojileri yavaş (3-22 kW), orta (20-25 kW) ve hızlı (50 kW) olarak üç tipte ele alınmaktadır (www.etmd.org.tr). Yavaş şarj teknolojisine geleneksel bir ev tipi priz örnek verilebilir. Bu teknolojiye şarj işlemi diğer tüm teknolojilerden çok daha yavaştır ve genellikle 4 ile 20 saat arasında sürmektedir. Bu nedenle gün içinde kullanılamaz genellikle depodaki araçları gece boyunca şarj etmek için kullanılır. Yavaş şarj teknolojisi mevcut en ucuz teknoloji olduğu içinde araçlar depodan tam dolu olarak ayrılırlar. Diğer şarj teknolojileri daha hızlıdır ancak daha pahalıdır.

Şekil 3.3'te sekiz müşteriye (C1–C8), üç şarj istasyonunu (R1–R3) ve depoyu (D) içeren Senaryo 3'e ait bir ÇAEARP örneği gösterilmektedir. Rotalardaki yüzde değerleri, araç bir müşteriye veya istasyona vardığında ve bataryasını doldurup şarj istasyonundan ayrıldığındaki bataryasının doluluk oranını göstermektedir. 1 numaralı elektrikli aracın (EA1) bataryasındaki şarj miktarı rotası boyunca tükettiği enerji miktarından daha fazla olduğu için herhangi bir şarj istasyonuna uğramasına gerek kalmamıştır. Şekil 3.3'teki şarj istasyonları farklı şarj teknolojilerine sahiptir (orta ve hızlı). Orta ve hızlı şarj teknolojisine sahip istasyonların şarj hızları sırasıyla 22 kWsa/sa ve 45 kWsa/sa 'dır. Şekil 3.3'teki grafikler, şarj istasyonlarındaki araçların bataryalarının şarj olma sürelerini göstermektedir. EV2 ve EV3'ün pili kısmen aynı oranda (%30) şarj edilmektedir. Ancak EV2, turunu daha kısa sürede tamamlayarak depoya geri dönecektir.



Şekil 3.3. ÇAEARP'nin örnek gösterimi (Senaryo 3).

Senaryo 3'te ele alınan ÇAEARP modeli, bazı eklemeler ve değişikliklerle Senaryo 2'deki matematiksel modelin amaç fonksiyonlarını ve kısıtlamalarını içermektedir. Denklem 3.1 (1. amaç fonksiyonu), denklem 3.15 ve denklem 3.25 kaldırılmakta ve aşağıdaki denklemler modele eklenmektedir.

Amaç fonksiyonu:

$$\text{Enküçük} \sum_{k \in K} \gamma^* g_k + \sum_{k \in K} \sum_{j \in R} \sum_{t \in T_j} \gamma_t z_{jkt} + \vartheta \sum_{k \in K} \sum_{i \in V, j \in RU\{0\}} x_{ijk} \quad (3.1)$$

Kısıtlar:

$$dt_{jk} \geq dt_{ik} + \left( \left( t_{ij} + \sum_{t \in T_i} \Delta R t_{ikt} \right) x_{ijk} \right) - (T_{\max}(1 - x_{ijk})) \quad (3.26)$$

$$\forall i \in R, \forall j \in V, i \neq j, \forall k \in K$$

$$f_{jk} = f_{ik} - (rd_{ij}x_{ijk}) + \sum_{t \in T_j} z_{jkt} \quad (3.27)$$

$$\forall j \in R, \forall i \in C, \forall k \in K$$

Denklem 3.1 ile ifade edilen 1. amaç fonksiyonu toplam şarj etme maliyetinin en küçüklenmesini sağlamaktadır. Bu amaç fonksiyonundaki birinci terim depoda gece boyunca depoda şarj edilen miktarın maliyetini, ikinci terim ise tüm istasyonlarda (depo dahil) gün boyunca şarj edilen miktarın maliyetini ve üçüncü terim, batarya çevrimiyle ilgili sabit maliyeti ifade etmektedir. Denklem 3.26, şarj istasyonlarından ayrılan araçların her bir noktaya (müşteri, şarj istasyonu ve depo) varış zamanını belirlemektedir. Denklem 3.27, aracın bir şarj istasyonundan ayrılırken ki şarj seviyesinin, vardığındaki şarj seviyesi ve şarj istasyonundaki şarj edilen miktarın toplamına eşit olmasını sağlamaktadır.





## **4. LİTERATÜR ARAŞTIRMASI**

Tezin konusu olan çok amaçlı elektrikli araç rotalama problemi, öncelikle iki araştırma akışını birleştirmektedir. Bunlardan ilki AYA'lardan EARP'ye odaklanmaktadır. İkinci araştırma akışı ise ÇAYARP ile ilgilidir. Buna istinaden bu bölümde literatür araştırması, EARP'ler ve ÇAYARP'ler olmak üzere iki ana başlık altında verilecektir.

### **4.1. Elektrikli Araç Rotalama Problemi ile İlgili Çalışmalar**

EARP, elektrikli araçların sınırlı sürüş menziline şarj altyapılarını da içererek hesaba katan ARP'nin bir çeşididir. Conrad ve Figliozzi (2011), sınırlı menzile sahip araçların önceden belirlenmiş bir müşteri grubuna hizmet vermesi gerektiği, ancak araçların tura devam edebilmesi için müşteri lokasyonlarında maksimum pil seviyesinin %80'ine kadar hızlı şarj edebildiği en eski EARP'lerden birini sunmuşlardır.

Erdoğan ve Miller-Hooks (2012), sadece müşteri lokasyonlarında yakıt ikmali yapmak yerine yol üzerindeki alternatif yakıt istasyonlarında yakıt ikmali yapan AYA'ların rotalanmasını YARP bağlamında ele alan bir matematiksel modeli formülize etmişlerdir. Model toplam seyahat mesafesinin en küçüklenmesini amaçlamaktadır. AYA'ların yakıt ihmal sürelerinin eşit olduğu ve yakıt doldurulduktan araçların deposunun tam olarak dolduğu varsayılmaktadır. Erdoğan ve Miller-Hooks (2012), YARP'yi çözmek için iki farklı sezgisel yöntem önermişlerdir. Bu yöntemler; uyarlanmış Clarke ve Wright tasarruf algoritması ve yoğunluk tabanlı kümeleme algoritmasıdır. Bu iki sezgisel yöntemin performansını test etmek için iki çeşit örnek problem tasarlamışlardır. Bunlardan ilki 20 müşteriye sahip 40 küçük boyutlu örnekten oluşurken, ikincisi 500'e kadar müşteriye sahip 12 örnekten oluşmaktadır. Zhang ve ark. (2018a), Erdoğan ve Miller-Hooks (2012)'un çalışmasında da olduğu gibi AYA'ların rotalanmasını ve tüm AYA'ların toplam rota uzunluğunun en küçüklenmesini amaçlamışlardır. Ele aldıkları problemi çözmek için iki aşamalı bir sezgisel yöntem ile karınca kolonisi sistemine dayalı bir metasezgisel önermişlerdir.

Zhang ve ark. (2018b), seyahat mesafesi, araç ağırlığı, araç hızı ve motor verimliliği dikkate alınarak toplam enerji tüketiminin en küçüklenmesi için bir EARP modeli geliştirmişlerdir. Geliştirdikleri modelin çözümünde karınca kolonisi algoritması ve uyarlanabilir geniş komşuluk arama yöntemi (UGKA) kullanmışlardır.

Schneider ve ark. (2014), EARP'ye araç kapasite kısıtı ve zaman penceresi kısıtı ekleyerek zaman pencereli elektrikli araç rotalama problemini (ZPEARP) tanımlamıştır. ZPEARP'de araçlar bataryalarını şarj noktalarında tamamen doldurulmaktadır. Şarj etme süresi artık sabit olarak varsayılmamakta, aktarılan enerji ile orantılıdır. Yazarlar kullanılan araç sayısı ve toplam katedilen mesafeyi en küçüklemeyi amaçlayarak hibrit bir değişken komşuluk arama ve tabu arama yöntemi önermişlerdir. Önerilen yöntemlerin performansını çeşitli test problemleri üzerinde denemişlerdir. Desaulniers ve ark. (2016), araçların bataryalarının sadece tam dolu şarj edilmesi yerine kısmi olarak da şarj edilmesine izin vererek aynı problemi farklı senaryolar ile çalışmışlardır. Problemlerin çözümü için ise dal-fiyat-kesme algoritması kullanmışlardır. Cortes-Murcia ve ark. (2019), Schneider ve ark. (2014)'nın geliştirdiği probleme uydu müşterileri kavramını ilave ederek ZPEARP'nin yeni bir varyasyonunu literatüre kazandırmışlardır. Bu varyant, EA bir şarj istasyonundayken bisiklet veya dron gibi alternatif ulaşım ile sürekli müşteri ziyaretlerine izin verir. Bu müşterilere uydu müşteri denir. Problemi çözmek için ise yinelemeli yerel arama algoritması önermişlerdir. Keskin ve ark. (2019), ZPEARP'yi şarj istasyonlarındaki zamana bağlı bekleme sürelerini göz önüne alarak yeniden modellemişlerdir. Zaman pencereleri esnektir, yani müşteri lokasyonlarına ve depoya geç gelmelere izin verilmekte fakat ceza maliyeti uygulanmaktadır. Ayrıca şarj istasyonlarında doğrusal olmayan şarj işlevi uygulanmaktadır. Ele aldıkları bu problemin çözümü için ise UGKA yöntemi ile karışık tamsayı doğrusal programlamanın birleşiminden oluşan bir yöntem geliştirmişlerdir. Zhang ve ark. (2020) çalışmalarında ZPARP'yi belirsizlik ortamında ele alarak yeni bir zaman pencereli bulanık elektrikli araç rotalama problemi önermişler ve önerdikleri bu problem için güvenilirlik teorisine dayalı bir bulanık optimizasyon modeli oluşturmuşlardır. Oluşturulan modelde hizmet süresi, enerji tüketimi ve seyahat süresi bulanık sayılar olarak alınmıştır. Modelin çözümü için ise bulanık simülasyon yöntemi destekli UGKA yöntemi kullanmışlardır.

Felipe ve ark. (2014) tam dolu bataryaya ulaşmak için gereken sürenin yoldaki şarj işlemleri sırasında önemli olacağını göz önüne alarak EARP'de kısmi şarj politikasını geliştirmişlerdir. Yazarlar sabit bir şarj etme süreci varsayarak birden fazla teknolojiye sahip şarj istasyonlarını ele almışlardır. Ele aldıkları problemi çözmek için hem yerel arama algoritmalarını hem de tavlama benzetimi tabanlı bir algoritma kullanmışlardır. Keskin ve Çatay (2016)'da ZPARP modeline kısmi şarj politikasını ekleyerek yeniden formülize etmişler ve çözümü için UGKA yöntemi önermişlerdir. Keskin ve Çatay (2018), ilk kez Felipe ve ark. (2014) tarafından ele alınan şarj istasyonlarındaki çoklu şarj teknolojisi üzerinde çalışarak kesin çözüm yöntemi ve UGKA yöntemini bir arada kullanarak bir mat-sezgisel yöntem geliştirmişlerdir.

Montoya ve ark. (2017), bataryalara doğrusal olmayan şarj etme işlevini ekleyerek doğrusal olmayan şarj fonksiyonlu elektrikli araç rotalama problemini (DOŞFEARP) literatüre kazandırmışlardır. Toplam seyahat süresini ve yeniden şarj olma süresini en küçükmeyi amaçlayarak yinelemeli yerel arama algoritmasından oluşan iki aşamalı hibrit bir metasezgisel algoritma geliştirmişlerdir. Froger ve ark. (2017) çalışmalarında şarj istasyonlarının kapasitelerinin olduğu ve sınırlı sayıda araca hizmet verebileceğini düşünerek DOŞFEARP'yi kapasiteli şarj istasyonlarını kapsayacak şekilde modellemişlerdir. Froger ve ark. (2019), DOŞFEARP için iki yeni formülasyon önermişlerdir. Literatürdeki 120 test problemlerinden 23'ünün sonucunu iyileştirmişlerdir. Zuo ve ark. (2019), dışbükey doğrusal olmayan şarj fonksiyonu üzerinde verimli bir doğrusallaştırma yaklaşımını geliştirerek ve zaman penceresi kısıtını da dikkate alarak DOŞFEARP için yeni bir model önermişlerdir. Önerdikleri modeli doğrulamak için Solomon'un test problemlerini kullanmışlardır. Kancharla ve Ramadurai (2020), DOŞFEARP'ye yüke bağlı enerji tüketimi ve kapasiteli şarj istasyonları kısıtlarını ekleyerek iki yeni EARP tanımlamışlardır. Tanımladıkları problemlerin çözümü için UGKA yöntemine beş yeni operatör önermişlerdir. Önerdikleri yöntemlerin etkinliğini DOŞFEARP test problemleri üzerinde ölçülmüş ve test problemlerin %63'ünde daha iyi sonuçlar elde etmişlerdir. Karakatic (2021), DOŞFEARP'yi çok depolu olarak ele alıp yeniden modellemiş ve çözümü için çoklu çapraz operatörler ile yeni bir iki katmanlı genotipin dikkate

alındığı genetik algoritma (GA) geliştirmiştir. Önerilen yöntemi en iyi bilinen test problemlerini kullanarak çeşitli metasezgisel yöntemlerle karşılaştırmışlardır.

Goeke ve Schneider (2015), çalışmalarında EA'lardan ve geleneksel içten yanmalı araçlardan oluşan karma bir filoyu içerecek şekilde zaman pencereli ve karma filolu elektrikli araç rotalama problemini (ZPKFEARP) önermişlerdir. Ele aldıkları problemi çözmek için yerel aramalı UGKA yöntemini kullanmışlardır. Hiermann ve ark. (2016) çalışmalarında farklı yük kapasitesine, pil boyutuna ve satın alma maliyetine sahip elektrikli araçlardan oluşan heterojen filoyu ele alarak ZPKFEARP'yi yeniden modellemişlerdir. Problemin çözümü için ise UGKA yöntemi temelli hibrit bir algoritma geliştirmişlerdir. Hiermann ve ark. (2019), EA'ları, geleneksel içten yanmalı araçları ve plug-in hibrit elektrikli araçları birleştiren bir ZPKFEARP literatüre sunmuşlardır. Problemin çözümünde kullanmak için tamsayı programlama ile hibritleştirilmiş genetik algoritmaya dayalı metasezgisel bir yöntem geliştirmişlerdir.

Lu ve ark. (2020), araçların rotalarının her bir düğümünde hareket saatlerinin ve hızlarının belirlendiği yeni bir EARP modeli olan zamana bağlı elektrikli araç rotalama problemini modellemişlerdir ve çözümü için ise yinelemeli değişken komşu arama yöntemi geliştirmişlerdir.

Keskin ve ark. (2021), şarj istasyonlarında bulunan sınırlı sayıdaki şarj cihazı nedeniyle elektrikli araçların şarj işlemine başlamadan önce bir kuyrukta bekleyebileceği zaman pencereli elektrikli araç yönlendirme probleminin bir uzantısı olan zaman pencereli ve şarj istasyonlarında stokastik bekleme süreli elektrikli araç rotalama problemini literatüre kazandırmışlardır. Önerdikleri problemin çözümü için UGKA yöntemini kullanarak iki aşamalı simülasyon tabanlı bir sezgisel yöntem geliştirmişlerdir. Tablo 4.1'de literatürdeki EARP ile ilgili çalışmalar özetlenmiştir.

**Tablo 4.1.** EARP literatür incelemesi.

Çalışma	Şarj Cihazı Tipi		Şarj Edilme Miktarı		Şarj Edilme Süresi				YBEF	Araç filosu		Amaç fonksiyonu		
	ZP	RS	Çoklu	Tek	Tam	Kısmi	Sıfır	Sabit		D	DO	HO	HE	Çoklu
Conrad ve Figliozzi (2011)				✓	✓			✓				✓		✓
Erdoğan ve Miller-Hooks (2012)		✓			✓			✓				✓		✓
Felipe ve ark. (2014)		✓	✓			✓		✓				✓		✓
Schneider ve ark. (2014)	✓			✓	✓					✓		✓		✓
Goeke ve Scheneider (2015)	✓			✓		✓				✓	✓			✓
Desaulniers ve ark. (2016)	✓			✓	✓	✓				✓		✓		✓
Hiermann ve ark. (2016)	✓			✓	✓					✓		✓		✓
Keskin ve Çatay (2016)	✓			✓		✓						✓		✓
Froger ve ark. (2017)		✓		✓		✓					✓	✓		✓
Montoya ve ark. (2017)		✓		✓	✓	✓					✓	✓		✓
Keskin ve Çatay (2018)	✓		✓			✓				✓		✓		✓
Zhang ve ark. (2018a)				✓	✓		✓					✓		✓
Zhang ve ark. (2018b)				✓	✓		✓					✓		✓
Cortés-Murcia ve ark. (2019)	✓			✓		✓				✓		✓		✓
Froger ve ark. (2019)		✓		✓		✓					✓	✓		✓
Hiermann ve ark. (2019)	✓			✓	✓					✓			✓	✓
Keskin ve ark. (2019)	✓			✓		✓					✓	✓		✓
Zuo ve ark. (2019)	✓			✓	✓						✓	✓		✓

**Tablo 4.1. (Devamı) EARP literatür incelemesi.**

Çalışma	Şarj Cihazı		Şarj Edilme		Şarj Edilme Süresi				YBEF	Araç filosu		Amaç fonksiyonu		
	ZP	RS	Çoklu	Tek	Tam	Kısmi	Sıfır	Sabit	D	DO	HO	HE	Çoklu	Tek
Kancharla ve Ramadurai (2020)		✓		✓		✓				✓	✓			✓
Lu ve ark. (2020)		✓		✓	✓			✓				✓		✓
Zhang ve ark. (2020)	✓			✓		✓			✓			✓		✓
Karakatic (2021)	✓			✓		✓				✓		✓		✓
Keskin ve ark. (2021)	✓			✓		✓			✓			✓		✓
Mevcut Çalışma	Senaryo 1	✓		✓	✓				✓		✓	✓	✓	
	Senaryo 2	✓		✓	✓	✓			✓		✓	✓	✓	
	Senaryo 3	✓	✓		✓	✓			✓		✓	✓	✓	

ZP: Zaman penceresi, RS: Rota Süresi, D: Doğrusal şarj süresi fonksiyonu, DO: Doğrusal olmayan şarj süresi fonksiyonu, YBEF: Yüke bağlı enerji fonksiyonu, HO: Homojen filo, HE: Heterojen filo

#### 4.2. Çok Amaçlı Yeşil Araç Rotalama Problemi ile İlgili Çalışmalar

ÇAYARP, sadece toplam seyahat mesafesini değil aynı zamanda ortaya çıkan gaz emisyonu, yakıt tüketimi gibi değişkenleri dikkate alan ve birden fazla amacı optimize etmeye çalışan ARP türüdür.

ÇAYARP'yi geçmişe dönük olarak incelediğimizde Jemai ve ark. (2012)'nin iki amaçlı YARP tanımladığı görülmektedir. Tanımlanan iki amaçlı YARP, toplam seyahat mesafesinin ve CO<sub>2</sub> emisyonlarının en küçüklenmesini amaçlamaktadır. Problemin çözümünde ise NSGA-II algoritmasını kullanmışlardır.

Siu ve ark. (2012), geleneksel maliyetin en küçüklenmesi amacının yanında CO<sub>2</sub> emisyon seviyelerinin de en küçüklenmesini problemin ikinci bir amacı olarak ele alan ÇAYARP önermişler ve çözümü için ise GA tabanlı bir yaklaşım geliştirmişlerdir. Geliştirdikleri yöntemin sonuçlarını uyarlanmış Martins' algoritmasının sonuçları ile karşılaştırmışlardır.

Molina ve ark. (2014) çalışmalarında, araçların farklı kapasiteler, maliyetler ve emisyon faktörleri ile karakterize edildiği, heterojen bir filoya sahip YARP için Tchebycheff yöntemlerine dayanan çok amaçlı bir model önermişlerdir. Önerdikleri model toplam iç maliyetleri, CO<sub>2</sub> ve NO<sub>x</sub> emisyonlarını en küçüklemeyi amaçlayan üç amaç fonksiyonuna sahiptir. Problemin çözümü için Clarke and Wright tasarruf sezgiseli temelli bir algoritma geliştirmişlerdir.

Jabir ve ark. (2015), kapasite kısıtlı araç rotalama problemine CO<sub>2</sub> emisyon ölçümlerini ekleyerek ÇAYARP modeli geliştirmişlerdir. Geliştirilen çok amaçlı model CO<sub>2</sub> emisyonunu azaltırken aynı zamanda tedarik maliyetlerini de en küçükleyen bir dizi Pareto-optimal çözüm üretmektedir. Problemin çözümü için ise karınca kolonisi optimizasyonu ve değişken komşuluk arama yöntemine dayanan hibrit bir sezgisel algoritma geliştirmişlerdir.

Norouzi ve ark. (2017) çalışmalarında iki amaçlı zamana bağlı YARP için bir matematiksel model önermişlerdir. Önerilen model, yolların fiziksel durumu, her bir aracın fiziksel özellikleri, trafik koşulları, araçların ağırlıkları ve yükü dikkate alınarak toplam seyahat zamanı ve toplam yakıt tüketimini en küçükleyen iki amaçlı

bir matematiksel modeldir. Ele alınan problemin çözümü için ise uyarlanmış PSO kullanmışlardır.

Poonthalir ve Nadarajan (2018) değişken hız sınırlamasının rota maliyeti ve yakıt tüketimi üzerindeki etkisini incelemek için iki amaçlı yakıt verimli YARP'yi tartışmışlardır. Ele alınan problemin çözümünde PSO'ya açgözlü mutasyon operatörünü ve zamanla değişen hızlanma katsayısını dâhil ederek kullanmışlardır.

Li ve ark. (2019) geliri en büyüleyen, maliyetleri, zamanı ve emisyonu en küçükleyen çok amaçlı çok depolu bir YARP sunmuşlardır. Çözüm için feromon güncellemesinde yenilikçi bir yaklaşıma sahip geliştirilmiş karınca kolonisi optimizasyonu kullanmışlardır.

Ganji ve ark. (2020) çok amaçlı heterojen filolu zaman pencereli YARP'yi önermişler ve üç adet çok amaçlı metasezgisel algoritma ile çözümünü yapmışlardır. Bu algoritmalar; ÇAPSO, NSGA-II ve ÇAKKO'dur.



## 5. ÇÖZÜM METODU

Çalışmada problemin çözümü için iki aşamadan oluşan yeni bir hiyerarşik yöntem (HKKO-YAKA) geliştirilmiştir. Problemin başlangıç çözümünün elde edildiği ilk aşamada kullanılmak üzere KKO hibritleştirilerek HKKO elde edilmiştir. Geliştirilen hiyerarşik yöntemin ikinci aşamasında ise birinci aşamadan elde edilen çözümler dikkate alınarak problem YAKA ile çözülmektedir.

Bu bölümde öncelikle genel olarak KKO ve YAKA'lar hakkında detaylı bilgiler verilecek, daha sonra geliştirilen HKKO-YAKA'nın yapısı ve aşamaları anlatılacaktır. Ayrıca, açıklayıcı bir örnek üzerinden HKKO-YAKA detaylandırılacaktır.

### 5.1. Karınca Kolonisi Optimizasyonu (KKO)

KKO, Dorigo ve ark. (1991) tarafından gerçek karıncaların beslenme davranışlarından esinlenilerek geliştirilen popülasyon tabanlı bir metasezgisel algoritmadır. Karıncalar yuvaları ile besin kaynakları arasındaki mümkün olan en kısa yolu belirlemektedirler. Karıncalar, besin kaynağından yuvaya veya tam tersi yönde giderken, feromon adı verilen kimyasal bir karışım salarak bir feromon izi oluştururlar. Yoldaki feromon miktarlarının yoğunluğu, besin kaynağının yuvaya yakınlığının bir göstergesidir. Feromon miktarının yoğunluğu ne kadar yüksekse, diğer karıncaların da bu yoldan takip etme olasılığı o kadar yüksektir. Bu durum en kısa yolu bulmak için tüm karıncaların ortak etkileşimini temsil etmektedir (Dorigo ve ark, 1991).

Gerçek karıncaların yol seçim özelliği karınca kolonisi algoritmasına ilham olmuştur. Algoritmada tanımlanan yapay karıncaların birçok özelliği gerçek karıncalarla aynıdır. Aynı kalan özellikler aşağıdaki gibi özetlenmiştir (Dorigo ve ark, 1999):

- Karıncalar arasında bir iletişim mekanizması bulunmaktadır.
- En kısa yolun bulunması amaçlanmaktadır.
- Mevcut bilgilere dayalı olarak gelecekteki durumu tahmin etmenin bir yolu olan olasılıksal bir seçim stratejisi kullanarak yollarını belirlemektedirler.

Gerçek karıncaların davranışları ile yapay karıncaların davranışları arasındaki temel farklar ise şunlardır:

- Yapay karıncalar hafızaları sayesinde gittikleri yolu hatırlayarak algoritmanın verimliliğini artırabilmektedirler.
- Yapay karıncalar bir yolu seçerken kör değildirler, yani bilinçli olarak en kısa yolu aramaktadırlar.
- Yapay karıncalarda problemin çözüm uzayı ayrıkken, gerçek karıncalar sürekli-zaman ortamında yaşamaktadırlar.

KKO'nun iki temel aşaması vardır. Bunlar çözüm oluşturma ve feromon güncelleme aşamalarıdır (Dorigo ve Birattari, 2011).

### 5.1.1. Çözüm oluşturma aşaması

Çözüm oluşturma aşamasında yapay karıncalar her adımda bir çözüm bileşeni ekleyerek bir çözüm oluşturur. Eklenecek çözüm bileşeninin seçimi, yollardaki feromon izini ve sezgisel bilgiyi kullanan stokastik bir karar verme sürecine dayanmaktadır. Bir karıncanın çözüm oluştururken kullandığı geçiş olasılığı olarak adlandırılan olasılık denklem 5.1'de verilmiştir.

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in j_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta} & \text{eğer } j \in j_k(i) \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.1)$$

Denklem 5.1'deki  $p_k(i, j)$  geçiş olasılığı yani yolların seçilme olasılığıdır.  $j_k(i)$ ,  $i$  noktasından gidilebilecek ve henüz  $k$  karıncası tarafından ziyaret edilmemiş noktalar kümesidir.  $\tau(i, j)$ ,  $i$  ve  $j$  noktaları arasındaki feromon miktarını,  $\eta(i, j)$  ise  $i$  ve  $j$  noktaları arasındaki mesafeyle ters orantılı olan seçilebilirlik parametresini göstermektedir.  $\alpha$  ve  $\beta$  ise sırasıyla feromon miktarının önemini belirleyen ve sezgisel bilginin önemini belirleyen sabitlerdir (Dorigo ve ark, 1996).

### 5.1.2. Feromon Güncelleme Aşaması

Feromon miktarının güncellenmesi çözüm uzayının aranması amacıyla yapılmaktadır. Feromon güncelleme kuralı iki kısımdan oluşmaktadır. İlk kısımda karıncalar turlarını tamamladıktan sonra önceki yollardaki tüm feromon miktarlarını

azaltan feromon buharlaşması işlemi yapılmaktadır. Bu buharlaşma işlemi arama uzayında yeni alanları keşfetmeye yardımcı olan kullanışlı bir unutma şeklidir. İkinci kısımda ise, karıncanın kullandıkları yolun uzunluğuyla ters orantılı olarak geçtikleri yollardaki feromon miktarlarının arttırılmasıdır (Dorigo ve Blum, 2005).

Genel olarak iki farklı feromon güncellemesi yapılmaktadır. Bunlar lokal ve global feromon güncellemesidir (Dorigo ve ark, 1996).

### 5.1.2.1. Lokal feromon güncellemesi

Lokal feromon güncellemesinin amacı; karıncaların turlarının dinamik olarak değiştirilmesi ve bütün karıncaların bir önceki turdan kalan en iyi tura bağlı kalmadan yeni turlar aramasına olanak sağlamaktır. Lokal feromon güncellemesi işlemi denklem 5.2'deki gibi yapılmaktadır.

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t+1) \quad (5.2)$$

Denklem 5.2'deki  $\tau_{ij}(t)$  t. iterasyondaki i ve j noktaları arasındaki feromon miktarını ve  $\rho$  parametresi ise buharlaşma oranını göstermektedir.  $\Delta\tau_{ij}^k(t+1)$ , denklem 5.3'teki gibi hesaplanmaktadır.

$$\Delta\tau_{ij}^k(t+1) = \begin{cases} \frac{1}{L^k(t+1)} & \text{eğer } k \text{ karıncası } i - j \text{ yolunu kullanıyorsa} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.3)$$

Denklem 5.3'teki  $L^k(t+1)$ , k karıncasının mevcut iterasyondaki toplam tur uzunluğudur.

### 5.1.2.2. Global feromon güncellemesi

Global feromon güncellemesi, her iterasyonda en iyi sonucu bulan karıncanın yolundaki feromon miktarının arttırılmasıyla gerçekleştirilir. Böylelikle her iterasyonda bulunan en iyi sonuç bir sonraki iterasyona aktarılmaktadır. Global feromon güncellemesi işlemi denklem 5.4'teki gibi yapılmaktadır.

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^{\text{best}}(t+1) \quad (5.4)$$

$\Delta\tau_{ij}^{\text{best}}(t + 1)$ , denklem 5.5'teki gibi hesaplanmaktadır.

$$\Delta\tau_{ij}^{\text{best}}(t + 1) = \begin{cases} \frac{1}{L_{\text{best}}(t + 1)} & \text{eğer } i - j \text{ yolu en iyi tura aitse} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.5)$$

$L_{\text{best}}(t + 1)$ , iterasyondaki en iyi turun uzunluğudur.

## 5.2. Yapay Arı Kolonisi Algoritması (YAKA)

YAKA, Karaboğa (2005) tarafından bal arısı kolonilerinin besin kaynağı arama davranışlarından esinlenilerek geliştirilen metasezgisel bir algoritmadır. Bu algoritmanın iki arı grubu ve besin kaynakları olmak üzere üç temel bileşeni vardır. Arı grupları, görevli (işçi arılar) ve görevsiz arılardan (gözcü ve kaşif arılar) oluşmaktadır. Daha önce ziyaret ettiği besin kaynağına giden ve besin kaynağının uzaklığı, yönü ve kalitesi (besin kaynağındaki nektar miktarı) hakkında kovandaki dans alanına dönüp gözcü arılara bilgi veren arılara işçi arı denir. Rastgele besin kaynağı arayışında olan arılara kaşif arı ve dans alanında bekleyen arıya gözcü arı denmektedir. Bu algoritmada, arı grupları kovana yakınlıklarına göre besin kaynakları arar. Algoritmanın iterasyonları sayesinde, yüksek kaliteli besin kaynaklarıyla ilgili olarak arıların sayısında bir artış olmaktadır. Bu durum, arıların davranışlarının daha da iyileştirilmesini ve arılar tarafından zayıf besin kaynaklarının azaltılmasını teşvik etmektedir (Karaboğa, 2005).

YAKA'da besin kaynaklarının sayısı işçi veya gözcü arı sayısına eşittir. Bir besin kaynağının konumu, optimizasyon problemi için potansiyel bir çözüm görevi görmektedir. En iyi besin kaynağının konumu ise, bir optimizasyon probleminin umut verici bir çözümünü göstermekte ve elde edilen çözümün kalitesini değerlendirmek için bir uygunluk fonksiyonu kullanılmaktadır (Karaboga ve Akay, 2009).

YAKA'nın dört temel aşaması vardır. Bunlar başlangıç aşaması, işçi arı, gözcü arı ve kâşif arı aşamalarıdır (Akay, 2009).

### 5.2.1. Başlangıç aşaması

YAKA'nın başlangıç aşamasında, besin kaynaklarının ilk konumları denklem 5.6 yardımıyla düzgün rastgele dağılım olarak oluşturulmaktadır.

$$x_{ij} = x_j^{\min} + \alpha(x_j^{\max} - x_j^{\min}) \quad (5.6)$$

Denklem 5.6'daki  $x_{ij}$ , rastgele oluşturulan besin kaynağını ifade etmektedir.  $\alpha$ ,  $[0,1]$  arasında düzgün bir rassal sayıdır. SN besin kaynaklarının sayısını ( $i \in 1,2,3 \dots SN$ ), D optimize edilecek parametre sayısını ( $j \in 1,2,3 \dots D$ ),  $x_j^{\min}$  parametrelerin alt sınırını ve  $x_j^{\max}$  ise parametrelerin üst sınırını göstermektedir. Ayrıca bu aşamada her kaynağın geliştirilememe sayacı da ( $failure_i$ ) sıfırlanmaktadır.

### 5.2.2. İşçi arı aşaması

İşçi arılar bu aşamada yiyecek aramak için besin kaynaklarına rastgele olarak gönderilir. Buldukları besin kaynaklarındaki nektar miktarlarını değerlendirerek hafızalarına kaydederler. Daha sonra görevli olduğu  $x_{ij}$  konumunun komşuluğundaki besin kaynaklarını aramaya başlarlar. Yeni besin kaynağının konumu olan  $v_{ij}$  denklem 5.7'deki gibi hesaplanmaktadır.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (5.7)$$

Denklem 5.7'deki  $\varphi_{ij}$ ,  $[-1,1]$  arasında üretilen rastgele sayıdır.

Bu işlemin sonunda üretilen  $v_{ij}$  parametreye ait alt ve üst sınırları aşarsa bu sınırları aşan değer belirli olan alt ve üst sınır değerlerine eşitlenmektedir. Bu durum denklem 5.8'de verilmiştir.

$$v_{ij} = \begin{cases} x_j^{\min}, & v_{ij} < x_j^{\min} \\ v_{ij}, & x_j^{\min} \leq v_{ij} \leq x_j^{\max} \\ x_j^{\max}, & v_{ij} > x_j^{\max} \end{cases} \quad (5.8)$$

Bu yapılan kontrolden sonra yeni besin kaynağının uygunluk değeri ( $fitness_i$ ) denklem 5.9 kullanılarak hesaplanır.

$$fitness_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0 \\ 1 + \text{abs}(f_i), & f_i < 0 \end{cases} \quad (5.9)$$

Denklem 5.9'daki  $f_i$  değeri,  $v_i$  besin kaynağının amaç fonksiyonu değeridir ve negatif veya pozitif değer alması durumunda eşitliğe göre normalize edilmektedir.

Yeni besin kaynağı olan  $v_i$ 'nin uygunluk değeri mevcut besin kaynağı olan  $x_i$ 'den daha iyi ise, mevcut besin kaynağı yeni besin kaynağı ile değiştirilir ve geliştirilememe sayacı sıfırlanır. Aksi durumda işçi arı eski besin kaynağına geri döner ve sayacı bir tane artırır.

### 5.2.3. Gözcü arı aşaması

İşçi arılar besin kaynağı arama işlemini tamamladıktan sonra, bir besin kaynağına ilişkin hafızalarına kaydettikleri bilgileri gözcü arılar ile paylaşırlar, onlar da bu bilgi üzerine olasılık yoluyla kendi besin kaynaklarını seçerler. YAKA'da bu seçim işlemi, işçi arı tarafından gözcü arıya aktarılan bir besin kaynağının uygunluk değerine dayanmaktadır. Uygunluk değeri daha yüksek olan kaynakların gözcü arılar tarafından seçilme olasılığı da daha yüksek olacaktır. Uygunluk tabanlı bu seçimde rulet çarkı yöntemi kullanılmakta ve seçilme olasılığının ( $p_i$ ) hesaplanması denklem 5.10'da yer almaktadır.

$$p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i} \quad (5.10)$$

Seçilme olasılığı hesaplandıktan sonra her besin kaynağı için  $[0,1]$  arasında rastgele bir sayı üretilir. Hesaplanan  $p_i$  olasılık değeri üretilen rastgele sayıdan büyükse gözcü arı bu besin kaynağına atanır. Gözcü arılar, besin kaynağına atandıktan sonra işçi arılar gibi yeni bir besin kaynağı keşfeder. Keşfettiği yeni kaynağın kalitesi (nektar miktarı) mevcut besin kaynağından daha iyi ise gözcü arı eski kaynağı hafızadan siler ve geliştirilememe sayacını sıfırlar. Diğer durumda ise mevcut besin kaynağı kullanılır ve sayaç bir artırılır.

### 5.2.4. Kaşif arı aşaması

İşçi ve gözcü arı aşamalarından sonra besin kaynaklarının tükenip tükenmediği kontrol edilir. Besin kaynaklarının tükenip tükenmediği çözüm geliştirememe sayacı tarafından belirlenmektedir. Bir kaynak için çözüm geliştirememe sayacı "limit" parametresi olarak tanımlanan değer üstünde ise, artık bu kaynak tükenmiş demektir ve ilgili işçi arının bu kaynağı terk ederek kendisi için başka bir besin kaynağı bulması gerekmektedir. İşçi arı besin kaynağını bıraktıktan sonra besin

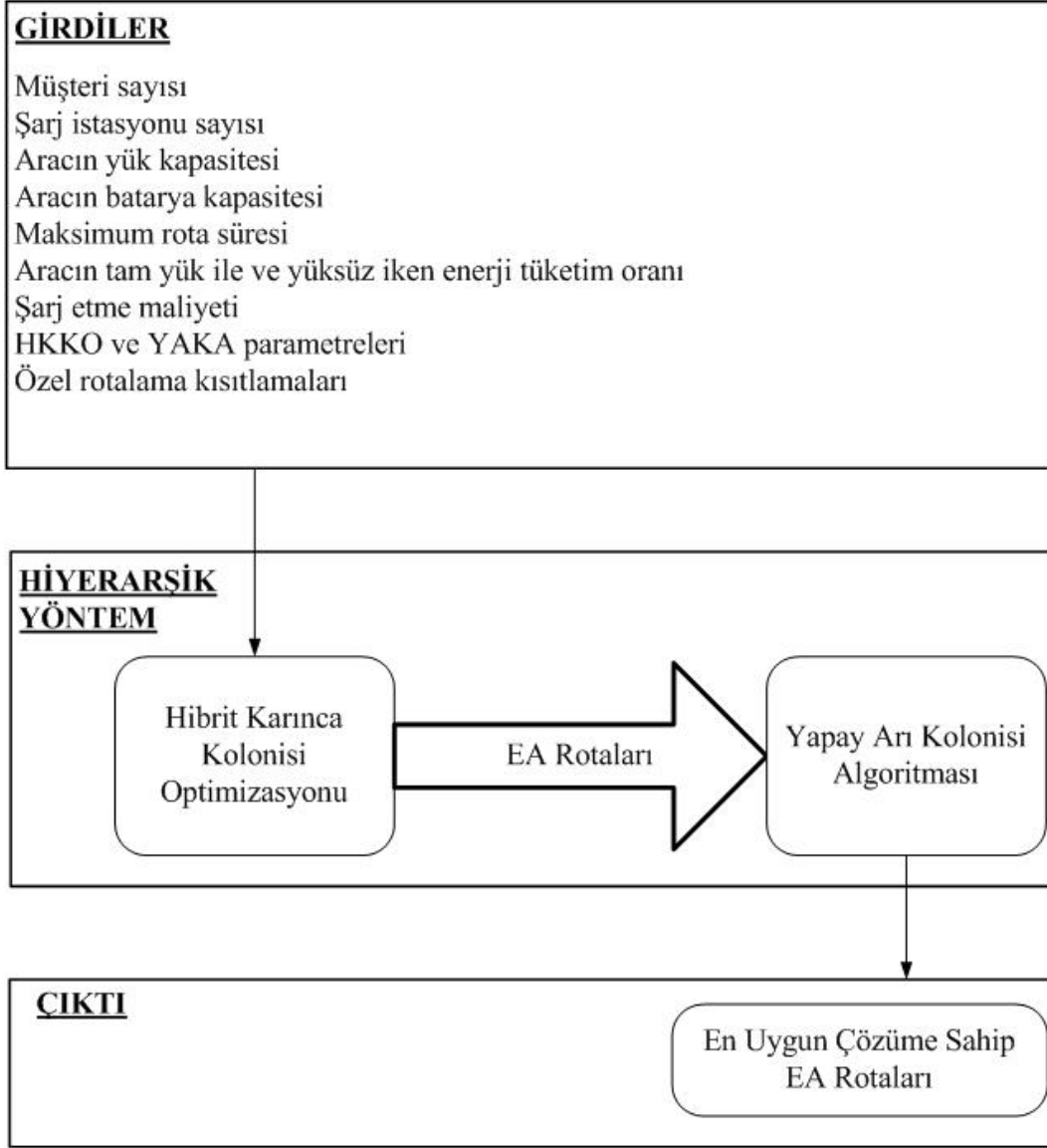
kaynađı arayacak olan kâşif arıya dönüşmekte ve rastgele besin kaynađı arama işlemleri başlamaktadır.

### **5.3. Geliştirilen Hiyerarşik Yöntem (HKKO-YAKA)**

Bu tez çalışmasında, önerilen ÇAEARP'yi çözmek için, iki aşamadan oluşan hiyerarşik bir yöntem (HKKO-YAKA) geliştirilmiştir. HKKO-YAKA, Pareto-optimal çözümleri bulmak için arama işlemi olarak kullanılmıştır.

İlk olarak, ÇAEARP'deki beş adet amaç fonksiyonunun birimleri farklılık gösterdiğinden, her bir amaç fonksiyonunu aynı ölçekte sunmak için normalizasyon işlemi uygulanmıştır. Normalleştirme işleminden sonra beş adet amaç fonksiyonunun Bölüm 2'de detayları verilen ATY ve KSY kullanılarak skalerleştirilmiş amaç fonksiyonları elde edilmiştir. Daha sonra da oluşturulan tek amaçlı problem geliştirilen HKKO-YAKA yöntemi ile çözümlenerek Pareto-optimal çözümlerin bulunması hedeflenmiştir.

Geliştirilen yöntemin çalışma prosedürü Şekil 5.1'de verilmiştir.



**Şekil 5.1.** Geliştirilen yöntemin çalışma prosedürü.

Geliştirilen hiyerarşik yöntemin çözüm aşamaları aşağıdaki gibidir.

### 5.3.1. 1. aşama: HKKO

#### 5.3.1.1. Çözüm oluşturma aşaması

Hizmet verilecek ilk müşteri her bir karınca için rastgele belirlenir. Daha sonraki hizmet verilecek müşterilerin sırası Bölüm 5.1’de anlatıldığı gibi geçiş olasılıkları bir başka ifade ile yolların seçilme olasılıkları ile belirlenmektedir. Denklem 5.1’deki  $\eta(i,j)$  ile ifade edilen seçilebilirlik parametresi problemin amaç fonksiyonu değeri ile ters orantılı olarak alınmaktadır. Bu çalışmada ele alınan problem için beş adet amaç fonksiyonu bulunmaktadır. Bunlar; toplam şarj etme maliyetini, gerekli olan EA



sayısını, toplam seyahat mesafesini, enerji tüketimini ve uğranması gereken şarj istasyonu sayısını en küçükmektir. Bu beş amaç fonksiyonunun birimleri farklılık gösterdiğinden, her bir amaç fonksiyonunu aynı ölçekte sunmak için normalizasyon işlemi uygulanmıştır. Normalleştirme işleminden sonra beş adet amaç fonksiyonunun Bölüm 2’de detayları verilen ATY ve KSY kullanılarak skalerleştirilmiş amaç fonksiyonları elde edilmiştir. Dolayısıyla  $\eta(i,j)$ ,  $1/(\text{skalerleştirilmiş amaç fonksiyonunun değeri})$  olarak ayarlanır.

Her karınca, hizmet verilecek müşterilerin sırasını temsil etmektedir. Araçların yük kapasitesi ve toplam rota süresi kısıtları nedeniyle rotaya müşteri eklenemediğinde araç rotası tamamlanır. Bir aracın rotası tamamlandığında kalan müşteriler arasından araçların yük kapasitesi ve toplam rota süresi kısıtları dikkate alınarak yeni bir araç rotası oluşturulur ve tüm müşteriler ziyaret edilene kadar bu işlem tekrarlanır. Ayrıca araçların rotaları oluşturulurken, araçların tükettiği enerji miktarı da hesaplanarak batarya durumları kontrol edilmektedir. Eğer aracın bataryasındaki şarj miktarı yeterli değilse araç bulunduğu müşteriye en yakın şarj istasyonuna uğrayarak rotasını güncellenmektedir.

Her bir karınca için araç rotaları oluşturulduktan sonra amaç fonksiyonu değerleri hesaplanır ve en iyi değer ile karşılaştırılır. Daha önce bulunan ve kaydedilen en iyi değerden daha iyi bir değer bulunduysa hafızaya kaydedilir.

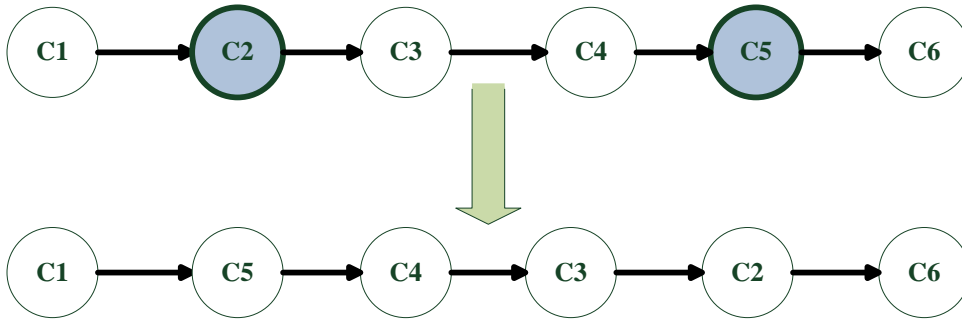
### **5.3.1.2. Yerel arama**

Yerel arama, optimizasyon problemlerini deneysel başarı ile birleştiren birkaç genel yaklaşımdan biridir (Johnson ve ark, 1988). Yerel aramanın arkasındaki temel fikir, adından da anlaşılacağı gibi, arama uzayında belirli bir nokta etrafında çözümler aramaktır. Yerel arama algoritmaları, aday bir çözümle başlar ve ardından yinelemeli olarak bu noktadan, optimize edilmekte olan amacın iyileştirildiği bir komşu çözüme geçer. Yerel arama yöntemlerinde komşuluk yapısı çözüm hızının ve kalitesinin ana belirleyicisidir. Yerel arama yöntemi, rotalar-arası ve rota-içi olmak üzere iki farklı prosedür olarak uygulanabilmektedir (Woodruff, 1998). EARP'nin doğası gereği, şarj istasyonlarına uğrama durumları, aracın yük kapasitesi kısıtı ve toplam rota süresi kısıtı gibi çeşitli kısıtlamaları bulunmaktadır ve bunların tümü yerel arama sürecinden sonra kontrol edilmelidir. Bu kontrol koşulu, sonuçları bulmak için

programı daha uzun süre çalıştırmamıza neden olacağından, her karıncadaki müşteri sırası üzerinde yerel arama yöntemini uyguluyoruz. Yerel arama yöntemi sonucunda oluşan yeni müşteri sırasını EARP kısıtları altında rotalara bölüp kabul kriterini uyguluyoruz.

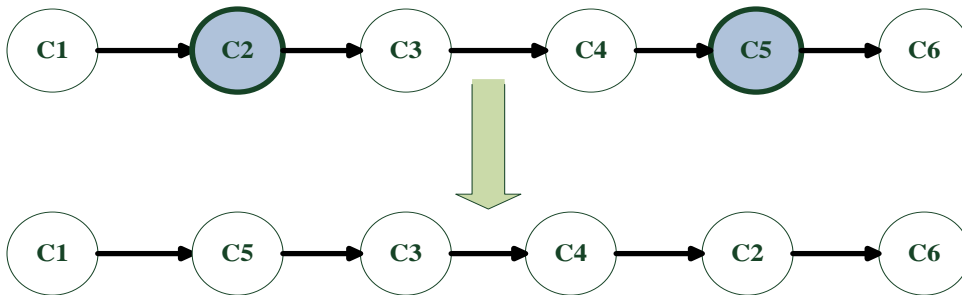
Komşuluk çözümlerini oluşturmak için kullanılan beş farklı yerel arama operatörleri ve bunların çalışma mantıkları aşağıda verildiği gibidir.

- Ters çevirme: Bir karınca içerisinde rastgele iki müşteri seçilir. Sonrasında bu iki müşteri ve arasında kalan müşterilerin sırası tersine çevrilir (Savelsberg, 1992). Şekil 5.2’de seçilen bir karıncadaki 2 (C2) ve 5 (C5) numaralı müşteriler arasında kalan parçanın tersine çevrilmesiyle elde edilen komşu çözüm gösterilmiştir.



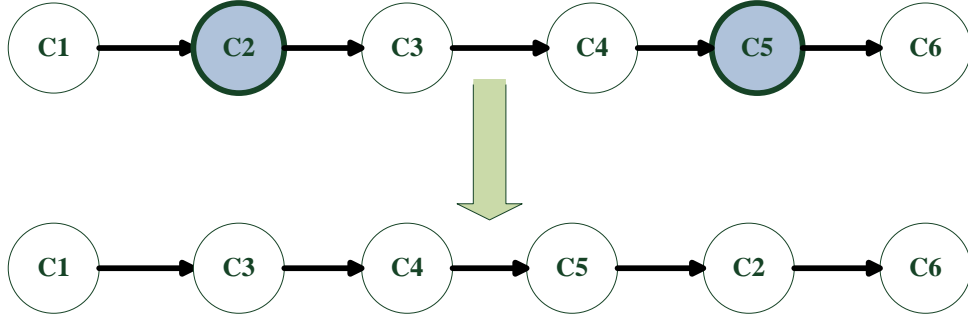
Şekil 5.2. Ters çevirme operatörünün işleyişinin gösterimi.

- Yer değiştirme: Bir karınca içerisinde rastgele iki müşteri seçilir ve bu iki müşterinin pozisyonları değiştirilir (Savelsberg, 1992). Şekil 5.3’te seçilen bir karıncadaki 2 (C2) ve 5 (C5) numaralı müşterilerin pozisyonlarının değiştirilmesi ile elde edilen komşu çözüm gösterilmiştir.



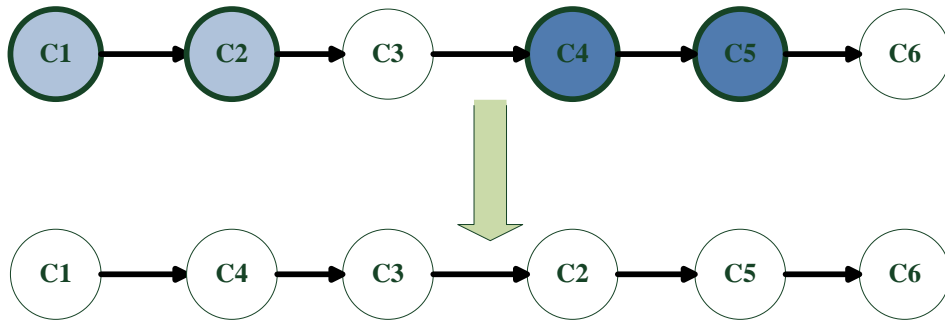
Şekil 5.3. Yer değiştirme operatörünün işleyişinin gösterimi.

- Ekleme: Bir karınca içerisinde rastgele iki müşteri seçilir ve birinci müşteri ikinci müşterinin arkasına eklenir (Savelsberg, 1992). Şekil 5.4'te seçilen bir karıncadaki 2 (C2) numaralı müşterinin 5 (C5) numaralı müşterinin arka pozisyonuna eklenmesi ile elde edilen komşu çözüm gösterilmiştir.



Şekil 5.4. Ekleme operatörünün işleyişinin gösterimi.

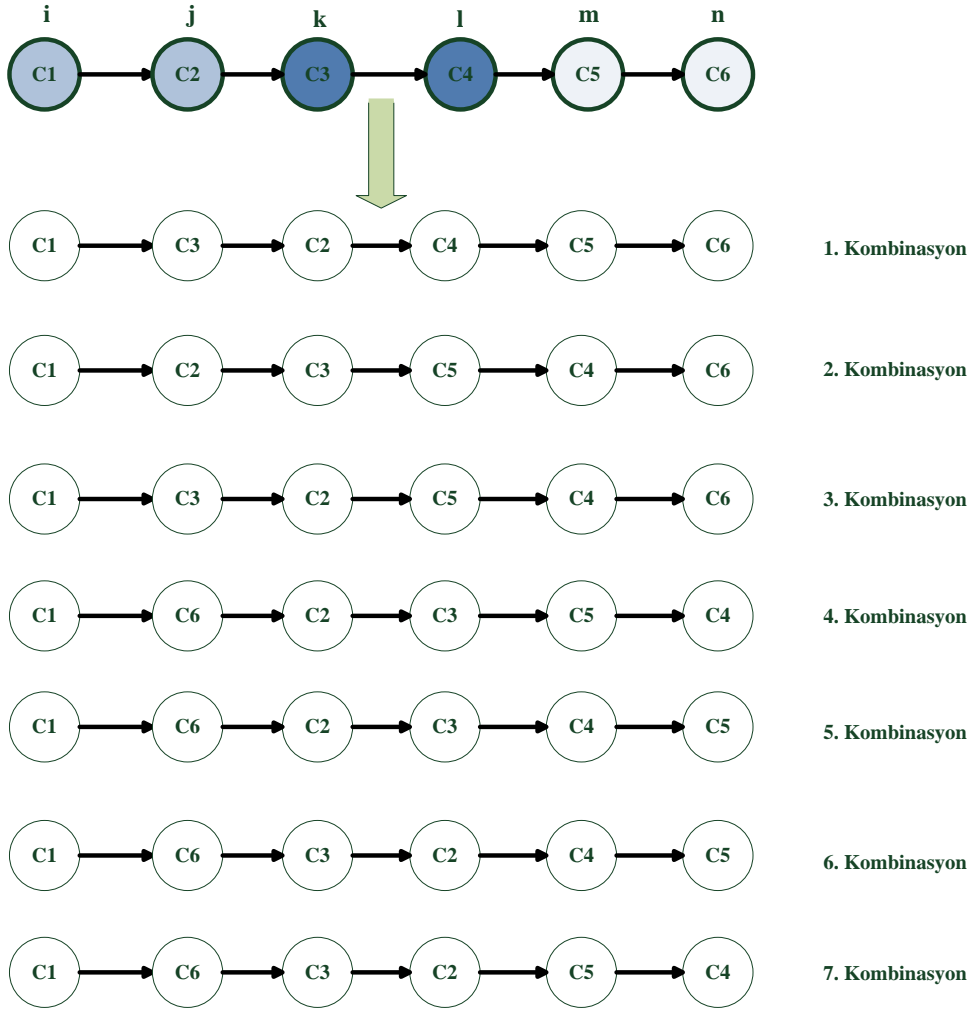
- 2-opt: İlk olarak Croes (1958) tarafından gezgin satıcı problemini çözmek için önerilmiştir. Bu algoritma, turdan iki yolu kaldırır ve oluşan iki yeni alt turu yeniden birleştirir. Bu bağlantı işlemi sadece yeni tur daha kısa ise tekrarlanır. Şekil 5.5'teki gibi [C1-C2] ve [C4-C5] müşterileri arasındaki mesafenin toplamının [C1-C4] ve [C2-C5] müşterileri arasındaki mesafenin toplamından büyük olduğunu varsayalım. C2 ve C4 müşterilerinin pozisyonlarını değiştirerek gerçekleştirilir.



Şekil 5.5. 2-opt algoritmasının işleyişinin gösterimi.

- 3-opt: İlk olarak Lin (1965) tarafından önerilen algoritma 2-opt'a benzer şekilde çalışmaktadır. 2-opt'tan farklı olarak, turdan üç kenarı kaldırır ve böylece üç alt tur elde edilmiş olur. Üç alt turu yeniden birleştirmek için yedi farklı kombinasyon bulunmaktadır. Tüm kombinasyonlardaki yeni turların uzunlukları

kontrol edilir. Kısa ise işlem 2-opt gibi tekrarlanır. Şekil 5.6’da 3-opt örneği ve tüm kombinasyonları gösterilmiştir.



Şekil 5.6. 3-opt algoritmasının işleyişinin gösterimi.

### 5.3.1.3. Kabul kriteri

Yerel arama yöntemi ile elde edilen çözüm mevcut çözümden daha iyi ise bir sonraki iterasyon için temel çözüm olarak kaydedilmektedir. Fakat sadece iyi olan çözümlerin seçimi iterasyonlar devam ettikçe algoritmanın yerel en iyi noktalara takılmasına neden olabilmektedir. Bu durumla karşılaşmamak için TB’ye dayalı kabul fonksiyonunu yerel arama yöntemine eklenmiştir. TB'deki kabul fonksiyonuna göre, elde edilen yeni çözüm mevcut çözümden daha kötü olsa dahi kaydedilebilir. Buradaki temel fikir, çözüm daha kötü bir amaç fonksiyonu değerine sahip olsa bile arama uzayı hakkındaki bilgisinden faydalanılması gerektiğidir.

TB'deki bu fonksiyon Boltzmann-Gibbs olasılık dağılımına dayanmaktadır. Bu olasılık, yeni ve mevcut çözümün amaç fonksiyonu değerleri ile önceden tanımlanmış bir soğutma programını izleyen sıcaklık parametresi arasındaki farka bağlıdır (Kirkpatrick ve ark, 1983). Kabul fonksiyonu denklem 5.11'de verilmiştir.

$$p(\text{kabul}) = e^{-\left(\frac{\Delta}{T}\right)} \quad (5.11)$$

$\Delta$ , yerel arama sonucunda elde edilen amaç fonksiyonu değerleri ile mevcut çözümün arasındaki farktır.  $T_0$  ve  $T_f$ , sırasıyla başlangıç ve son sıcaklığı temsil eder. Başlangıçta  $T$ ,  $T_0$ 'a eşittir. Daha sonra sıcaklık,  $\epsilon$  katsayısına ( $T=T*\epsilon$ ) bağlı olarak her iterasyon için güncellenir.  $\epsilon$  soğuma katsayısıdır ve 0 ile 1 aralığında bir değer alır.  $T$ ,  $T_f$  değerinin altında ise kabul kriteri durdurulur.

#### 5.3.1.4. Feromon güncelleme aşaması

Bölüm 5.1 'de detayları anlatılan denklem 5.2 ve denklem 5.4 yardımıyla sırasıyla lokal ve global feromon güncellemeleri yapılmaktadır. Denklem 5.2'deki  $\Delta\tau_{ij}^k(t+1)$ ,  $k$  karıncası tarafından yollara bırakılan feromon miktarının mevcut iterasyondaki değerini ifade etmektedir ve  $\frac{1}{L^k(t+1)}$  şeklinde hesaplanmaktadır.  $L^k(t+1)$  ise skalerleştirilmiş amaç fonksiyonunun mevcut iterasyondaki değeridir. Denklem 5.4'teki  $\Delta\tau_{ij}^{\text{best}}(t+1)$ , mevcut iterasyondaki en iyi feromon miktarını ifade etmektedir ve  $\frac{1}{L_{\text{best}}(t+1)}$  şeklinde hesaplanmaktadır.  $L_{\text{best}}(t+1)$  ise skalerleştirilmiş amaç fonksiyonunun mevcut iterasyondaki en iyi değeridir.

### 5.3.2. 2. aşama: YAKA

#### 5.3.2.1. Başlangıç aşaması

Geliştirilen hiyerarşik yöntemin ilk aşaması sonucunda EARP kısıtlarını sağlayan uygun çözümler elde edilir. Bu rotalar YAKA'nın başlangıç çözümü olarak kabul edilir. Başlangıç çözümü, HKKO sonucunda elde edilen çözümler olduğu için besin kaynağı sayısı karınca sayısına eşittir. Her karınca, hizmet verilecek müşterilerin sırasını yani probleme ait çözümü temsil ettiği için YAKA'da da besin kaynaklarında yer alan her bir besin satırı bir çözümü temsil etmektedir. Ayrıca bu aşamada, durdurma kriteri olarak maksimum iterasyon sayısı, çözümlerin tükenip tükenmediğinin kontrolü için "Limit" parametresinin değeri, çözümlere bağlı olarak

işçi arı sayısı, gözcü arı sayısı belirlenmekte ve tüm çözümlere ait geliştirilememe sayacı ( $failure_i$ ) da sıfırlanmaktadır.

### **5.3.2.2. Uygunluk değerlerinin hesaplanması**

Besin kaynağındaki her bir besinin hizmet verilecek müşterilerin sırasını temsil ettiğinden daha önce bahsedilmiştir. Araçların yük kapasitesi ve toplam rota süresi kısıtları dikkate alınarak her bir besindeki müşteri sırasına göre araç rotaları oluşturulur. Bu rota oluşturma işlemi tüm müşteriler ziyaret edilene kadar tekrarlanır. Ayrıca araçların rotaları oluşturulurken, araçların tükettiği enerji miktarı da hesaplanarak batarya durumları kontrol edilmektedir. Eğer aracın bataryasındaki şarj miktarı yeterli değilse araç bulunduğu müşteriye en yakın şarj istasyonuna uğrayarak rotasını güncellenmektedir. Besin kaynağındaki her bir besin için araç rotaları oluşturulduktan sonra amaç fonksiyonu değerleri hesaplanır. HKKO'da açıklandığı gibi ele alınan problem beş adet amaç fonksiyonuna sahiptir. Bu beş amaç fonksiyonunun birimleri farklılık gösterdiğinden, her bir amaç fonksiyonunu aynı ölçekte sunmak için önce normalizasyon işlemi uygulanmış, daha sonra ATY ve KSY kullanılarak skalerleştirilmiş amaç fonksiyonları elde edilmiştir. Bu nedenle uygunluk değeri skalerleştirilmiş amaç fonksiyonunun değeri olarak alınır.

### **5.3.2.3. İşçi arı aşaması**

Başlangıç çözümlerinin uygunluk değerleri hesaplandıktan sonra işçi arılar devreye girer. Her işçi arı rastgele bir çözüme atanır ve daha iyi komşuluk çözümleri için atanan çözümün komşuluk alanını araştırır. Bu çalışmada daha kaliteli komşuluk çözümleri üretmek için mevcut çözüm üzerinde üç yerel arama operatörü kullanılmıştır. Bu operatörler, Bölüm 5.3.1.2'de açıklanan ters çevirme, yer değiştirme ve ekleme operatörleridir. Oluşturulan yeni komşuluk çözümlerin uygunluk değeri Bölüm 5.3.2.2'de açıklandığı gibi hesaplanır. Çözümlerin uygunluk değeri hesaplandıktan sonra aralarında aç gözlü bir seçim işlemi uygulanır. Yeni çözümün uygunluk değerinin mevcut çözümden daha iyi (amaç fonksiyonu ile uyumlu) olduğunu varsayalım. Bu durumda işçi arılar mevcut çözümü hafızadan silerek geliştirilememe sayacı sıfırlanacaktır. Aksi takdirde, mevcut çözüm hafızada kalır ve sayaç bir artırılır.

#### **5.3.2.4. Gözcü arı aşaması**

Bu aşamada işçi arılar mevcut çözümlerin uygunluk değerleri ile ilgili bilgileri gözcü arılar ile paylaşırlar. Gözcü arılar için işçi arıların paylaştığı bu uygunluk değerlerine göre bir seçilme olasılığı ( $p_i$ ) hesaplanır. Bu seçilme olasılığı hesabı denklem 5.10'da verilmiştir. Seçilme olasılığı hesaplandıktan sonra her çözüm için  $[0,1]$  arasında rastgele bir sayı üretilir. Eğer seçilme olasılığının değeri ( $p_i$ ) rastgele üretilen değerden büyükse gözcü arı bu çözüme atanır. Gözcü arılar, çözüme atandıktan sonra işçi arılar haline gelirler ve işçi arı aşamasında açıklandığı gibi aynı süreçle yeni bir çözüm keşfederler. Yeni çözümün uygunluk değerinin mevcut çözümden daha iyi ise mevcut çözümü hafızadan silinerek geliştirilemeye sayacı sıfırlanacaktır. Aksi takdirde, mevcut çözüm hafızada kalır ve sayaç bir arttırılır.

#### **5.3.2.5. Kaşif arı aşaması**

İşçi ve gözcü arıların adımları tamamlandıktan sonra geliştirilemeye sayacı kontrol edilerek çözümlerin bitip bitmediği kontrol edilir. Sayaç “limit” parametresinin belirlenen değerinden fazla ise o çözüm alanında kaliteli çözümün kalmadığı ve bırakılması gerektiği sonucuna varılır. Bu durumda işçi arı, kâşif arıya dönüşür ve yeni bir çözüm arama işlemi başlar. Kaşif arılar yeni çözüm arama işlemi Bölüm 5.3.1.2'de açıklanan 2-opt ve 3-opt yerel arama operatörlerini kullanarak yürütmektedirler.

#### **5.3.3. HKKO-YAKA'nın parametreleri**

Metasezgisel algoritmalarda parametre değerlerinin etkin olarak belirlenmesi algoritmanın performansını ve başarısını önemli ölçüde etkilemektedir. Geliştirilen hiyerarşik yöntemin ilk aşaması olan HKKO için tanımlanan parametreler; karınca sayısı, feromon miktarının önemini belirleyen sabit ( $\alpha$ ), sezgisel bilginin önemini belirleyen sabit ( $\beta$ ), buharlaşma oranı ( $\rho$ ), başlangıç sıcaklığı ( $T_0$ ), bitiş sıcaklığı ( $T_f$ ), soğutma katsayısı ( $\epsilon$ ) ve iterasyon sayısıdır. İkinci aşaması olan YAKA için ise tanımlanan parametreler; besin kaynağı sayısı, işçi arı sayısı, gözcü arı sayısı, limit ve iterasyon sayısıdır.

Karınca sayısı, kolonide kaç adet karınca olacağını belirlemektedir. Karınca sayısının az olması feromon buharlaşması nedeniyle algoritmanın performansını azaltmaktadır. Karınca sayısının arttırılması çözümü iyileştirecek fakat hesaplama

işlemlerinin de fazla olmasına neden olacaktır. Hesaplama işlemlerinin artması problemin çözüm süresini ciddi oranda etkileyebileceğinden bu parametrenin amaca uygun bir şekilde belirlenmesi önemlidir.

$\alpha$  ve  $\beta$  parametresi birbirleriyle ilişkili olarak belirlenir.  $\alpha$  parametresi yollar arasındaki feromon miktarının önemini belirler.  $\alpha=0$  olduğunda en yakın şehirlerin karıncalar tarafından seçilmesi daha olası olmaktadır.  $\alpha$  değerinin yüksek olması ise feromon miktarının yüksek olduğu yolların seçilme olasılığını arttırarak rassallığı azaltacaktır.  $\beta$  parametresi ise sezgisel bilginin önemini belirler.  $\beta=0$  olduğunda sadece feromon miktarında yükselme meydana gelmektedir. Bu durumda karıncaların sezgisel bilgi olmadan sadece feromon izini dikkate alarak aynı turları seçmelerine sebebiyet verecektir.  $\beta$  değerinin artması ise rassallığı arttıracak böylece alternatif yolların araştırılması olasılığı artmış olacaktır.

$\rho$ , feromonların buharlaşma oranının belirleyen parametredir. Her iterasyonun sonunda kullanılmayan yolların seçilme olasılıklarının düşürmek için kullanılır. 0 ile 1 arasında değer almaktadır (Dorigo ve ark, 1991).

Yerel arama yöntemi aşamasında kabul kriteri olarak TB'ne dayalı kabul fonksiyonu kullandığımızdan bahsetmiştik. TB'de Başlangıç sıcaklığının ( $T_0$ ) çok yüksek bir değer seçilmesi gereksiz hesaplamalara neden olacağı için işlem süresini uzatacaktır. Başlangıç sıcaklığı düşük bir değer seçildiğinde ise yerel en iyi noktalara takılma olasılığını arttıracaktır. Bu yüzden başlangıç sıcaklığı seçimi oldukça önemlidir.

Son sıcaklık ( $T_f$ ), kabul kriteri için durdurma kriteri olarak kullanılan değerdir. Son sıcaklığın sıfırda ulaşmasına genellikle izin verilmektedir. Fakat bu işlem yöntemin çalışma süresini uzamasına sebep olabilmektedir. Ayrıca son sıcaklık sıfıra yaklaştıkça daha kötü sonuç veren çözümleri kabul etme ihtimali neredeyse sıfır olmaktadır. Bu nedenle son sıcaklığın sıfıra ulaşması gerekli değildir. Durma kriteri olarak uygun bir düşük sıcaklık değeri seçilmelidir.

$\epsilon$  ile ifade edilen soğutma katsayısı 1'den küçük ancak 1'e yakın pozitif bir değerdir. Deneyler soğutma katsayısının 0,8 ile 0,99 arasında seçildiğinde daha iyi sonuçlar verdiğini göstermektedir (Kendall, 2000).

İterasyon sayısı (HKKO), geliştirilen hiyerarşik yöntemin birinci aşaması olan HKKO'daki arama işleminin kaç adım gerçekleşeceğini belirler.



HKOO'nun sonucunda EARP kısıtlarını sağlayan uygun çözümler elde edilir. Bu çözümlerde YAKA'nın başlangıç çözümü olarak kabul edildiği için karınca sayısı kadar besin kaynağı sayısı mevcuttur. Besin kaynağı sayısı, arıların nektar elde etmek için gittikleri kaynakların sayısıdır.

YAKA'nın çalışma prensibi gereği her besin kaynağında sadece bir işçi arı bulunmaktadır. Bu nedenle de kolonideki işçi veya gözcü arı sayısı besin kaynağı sayısına eşit olarak belirlenir.

Besin kaynaklarının (çözümlerin) tükenip tükenmediğinin kontrolü için kullanılan Limit parametresi ise, Karaboğa ve Baştürk (2008)'un çalışmasında belirttiği gibi müşteri sayısı\*besin kaynağı sayısı olarak hesaplanır.

İterasyon sayısı (YAKA), geliştirilen hiyerarşik yöntemin ikinci aşaması olan YAKA'daki arama işleminin kaç adım gerçekleşeceğini belirler.

#### 5.3.4. HKKO-YAKA'nın sözde kodu

Bölüm 5.3'te detayları açıklanan HKKO-YAKA için sözde kod Tablo 5.1'de verilmiştir.

**Tablo 5.1.** HKKO-YAKA'nın sözde kodu.

---

---

1:	<b>Başlat</b>
	// <b>HKKO</b>
2:	<b>while</b> maksimum iterasyon sayısına ulaşılmadı <b>do</b>
3:	<b>for</b> karınca=1 to karınca sayısı
4:	Her karınca için rassal bir başlangıç müşterisi belirle
5:	Denklem 5.1'i kullanarak daha sonraki hizmet verilecek müşterilerin sırasını belirle
6:	Aracın yük kapasitesi ve toplam rota süresi kısıtlarını dikkate alarak müşterileri araçlara ata
7:	Her aracın enerji tüketimini hesapla
8:	<b>if</b> aracın bataryası yeterli şarj miktarına sahip değilse
9:	En yakın şarj istasyonuna uğra ve aracın rotasını güncelle
10:	<b>end if</b>
11:	Tüm araçlar için amaç fonksiyonu değerini hesapla
12:	Mevcut çözümle en iyi çözümü karşılaştır ve daha iyiyse kaydet

---

**Tablo 5.1. (Devamı) HKKO-YAKA'nın sözde kodu.**

---

```
13:     Yerel arama yöntemi uygula
14:     TB'ne dayalı kabul kriterini uygula
15:     Denklem 5.2 ve Denklem 5.4'ü kullanarak feromon güncellemesi yap
16:     end for
17:     end while

    // YAKA
18:     Başlat: HKKO'nun çözümünden elde edilen EA rotaları
19:     Çözüm geliştirilememe sayacını sıfırla ( failurei = 0)
20:     Başlangıç çözümlerinin uygunluk değerlerini Bölüm 5.3.2.2'de anlatıldığı gibi hesapla
21:     En iyi çözümü hafızaya al
22:     while maksimum iterasyon sayısına ulaşılmadı do
23:         for işçi arı=1 to işçi arı sayısı
24:             İşçi arıları çözümlere ata
25:             Ters çevirme, yer değiştirme ve ekleme operatörlerini uygula
26:             Yeni komşuluk çözümlerinin uygunluk değerini Bölüm 5.3.2.2'de anlatıldığı
                gibi hesapla
27:             if fitness(yeni çözüm) > fitness(mevcut çözüm)
28:                 Mevcut çözümü yeni çözümle değiştir
29:                 failurei=0
30:             else
31:                 failurei = failurei + 1
32:             end if
33:         end for
34:         for gözcü arı=1 to gözcü arı sayısı
35:             Gözcü arılar için çözümlerin seçilme olasılıklarını hesapla (Denklem 5.10)
36:             Gözcü arıları çözümlere gönder
37:             Ters çevirme, yer değiştirme ve ekleme operatörlerini uygula
38:             Yeni komşuluk çözümlerinin uygunluk değerini Bölüm 5.3.2.2'de anlatıldığı
                gibi hesapla
39:             if fitness(yeni çözüm) > fitness(mevcut çözüm)
40:                 Mevcut çözümü yeni çözümle değiştir
41:                 failurei=0
42:             else
```

---

**Tablo 5.1. (Devamı) HKKO-YAKA'nın sözde kodu.**

---

---

43:	$failure_i = failure_i + 1$
44:	<b>end if</b>
45:	<b>if</b> $\max(failure_i) > \text{limit}$
46:	Kaşif arıları yeni çözümler bulmaya yönlendirir, 2-opt ve 3-opt algoritmalarını uygula
47:	<b>end if</b>
48:	<b>end while</b>
49:	<b>Bitir</b>

---

### 5.3.5. Bir ÇAEARP örneğinin geliştirilen HKKO-YAKA ile çözümü

Geliştirilen HKKO-YAKA'nın işleyişi 9 müşteri 1 ana depo ve 2 şarj istasyonundan oluşan bir ÇAEARP örneği üzerinden gösterilmiştir. Daha önceden de bahsedildiği gibi problemin beş adet amaç fonksiyonu vardır. Bu amaç fonksiyonlarının ATY ile skalerleştirilmesi sonucunda oluşan amaç fonksiyonu değerinin en büyüklenmesi amaçlanmaktadır. Problemdaki şarj istasyonları tek tip şarj teknolojisine sahiptir ve şarj etme hızları orta dır. Araçların şarj istasyonuna uğradığında bataryalarını tam doldurdukları varsayılmaktadır. Müşterilerin talep miktarları ile ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki uzaklık matrisi Tablo 5.2 ve Tablo 5.3'te verilmiştir.

**Tablo 5.2. Müşterilerin talep miktarları (ton).**

---

---

C1	C2	C3	C4	C5	C6	C7	C8	C9
9	6	5	7	7	3	9	3	4

---

**Tablo 5.3. Ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki uzaklık miktarları (km).**

---

---

	C1	C2	C3	C4	C5	C6	C7	C8	C9	Depo	R1	R2
C1	0	51	48	29	32	37	60	176	103	94	50	56
C2	51	0	44	29	24	23	23	39	66	41	88	80
C3	48	44	0	56	55	50	100	106	43	82	76	67
C4	29	29	56	0	7	17	143	159	186	130	71	70
C5	32	24	55	7	0	12	138	164	181	126	55	96
C6	37	23	50	17	12	0	138	164	181	125	42	60

---

**Tablo 5.3. (Devamı)** Ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki uzaklık miktarları (km).

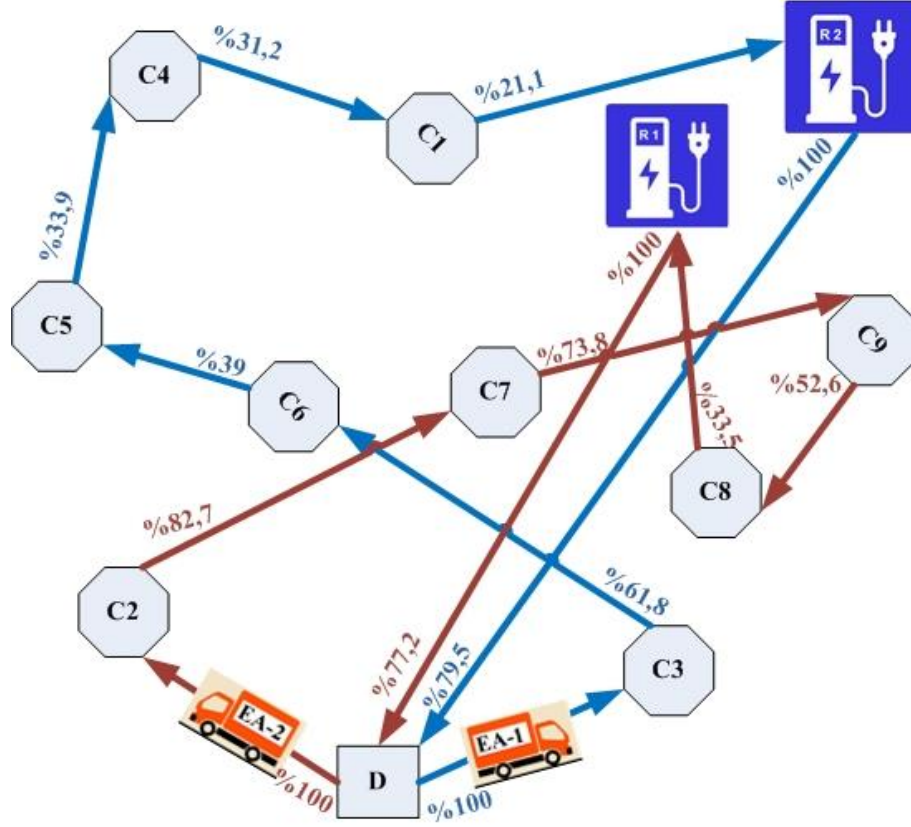
	C1	C2	C3	C4	C5	C6	C7	C8	C9	Depo	R1	R2
C7	60	23	100	143	138	138	0	92	63	130	36	67
C8	76	39	106	159	164	164	92	0	61	156	55	97
C9	83	66	43	186	181	181	63	61	0	193	70	77
Depo	94	41	82	130	126	125	130	156	193	0	77	69
R1	50	88	76	71	55	42	36	55	70	77	0	90
R2	56	80	67	70	96	60	67	97	77	69	90	0

Probleme ait diğer bilgiler ise aşağıdaki Tablo 5.4'te verilmiştir.

**Tablo 5.4.** Örnek probleme ait bilgiler.

Aracın yük kapasitesi	36 ton
Aracın batarya kapasitesi	435 kWsa
Şarj etmenin sabit maliyeti	19,38 TL/döngü
EA'ların Şarj istasyonlarında şarj edilme maliyeti	0,95 TL/kWsa'dir.
EA'ların depoda şarj edilme maliyeti	0,14 TL/kWsa
Aracın tam yüklü iken enerji tüketim oranı	2,18 kWsa/km
Aracın yüksüz iken enerji tüketim oranı	1,29 kWsa/km
Aracın rota süresi	40 sa
Müşterilerin servis süresi	30 dk
EA'ların ortalama hızı	90 km/s

Bu bilgiler dahilinde problem HKKO-YAKA ile çözülmüştür. 1. Aşaması olan HKKO sonucunda oluşan EA'ların rotası Şekil 5.7'de gösterilmiştir. Şekil 5.7'deki rotalardaki yüzde değerleri, araç bir müşteriye veya istasyona vardığında ve bataryasını doldurup şarj istasyonundan ayrıldığındaki bataryasının doluluk oranını göstermektedir.



Şekil 5.7. HKKO sonucunda oluşan EA'ların rotası.

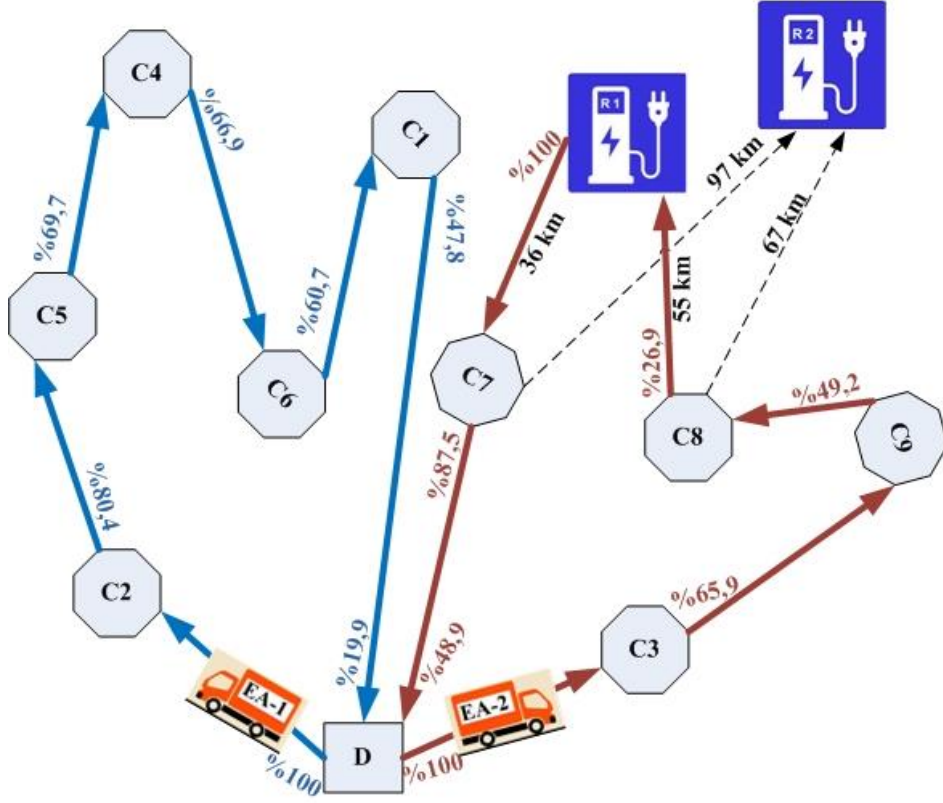
HKKO sonucunda oluşan EA'ların rotalarına ait amaç fonksiyonlarının değerleri ise Tablo 5.5'te verilmiştir.

Tablo 5.5. HKKO sonucunda elde edilen rotaya ait amaç fonksiyonu değerleri.

Amaç fonksiyonu	$\sum_{i=1}^5 w_i f'_i$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
Değeri	0,58	16,28 ₺	2 adet	625 km	964,10 kWsa	2 adet

$f_1$ : Toplam şarj etme maliyeti,  $f_2$ : Gerekli olan EA sayısı,  $f_3$ : Toplam seyahat mesafesi,  $f_4$ : Enerji tüketimi,  $f_5$ : Uğranması gereken şarj istasyonu sayısı

2. Aşaması olan YAKA sonucunda oluşan EA'ların rotası Şekil 5.8'de gösterilmiştir. Şekil 5.8'deki rotalardaki yüzde değerleri, araç bir müşteriye veya istasyona vardığında ve bataryasını doldurup şarj istasyonundan ayrıldığındaki bataryasının doluluk oranını göstermektedir.



Şekil 5.8. YAKA sonucunda oluşan EA'ların rotası.

EA'ların rotalarına ait amaç fonksiyonlarının değerleri ise Tablo 5.6'da verilmiştir.

Tablo 5.6. YAKA sonucunda elde edilen rotaya ait amaç fonksiyonu değerleri.

Amaç fonksiyonu	$\sum_{i=1}^5 w_i f_i'$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
Değeri	0,77	12,42 ₺	2 adet	613 km	951,96 kWsa	1 adet

$f_1$ :Toplam şarj etme maliyeti,  $f_2$ :Gerekli olan EA sayısı,  $f_3$ :Toplam seyahat mesafesi,  $f_4$ :Enerji tüketimi,  $f_5$ :Uğranması gereken şarj istasyonu sayısı

## 6. SAYISAL ANALİZLER

Bu bölümde, geliştirilen hiyerarşik yöntemin performansını ölçmek için yapılan hesaplama deneyleri açıklanacaktır. İlk olarak, hiyerarşik yöntemdeki algoritmaların parametre değerleri Taguchi yöntemi kullanılarak belirlenecek ve ardından geliştirilen hiyerarşik yöntemin test problemleri kullanılarak literatürde mevcut olan diğer çözüm yöntemleri ile karşılaştırmaları sunulacaktır.

### 6.1. Taguchi Yöntemi ile Parametre Tasarımı

Parametre değerlerinin belirlenmesi metasezgisel algoritmaların performansı veya etkinliği üzerinde önemli bir etkiye sahiptir, çünkü parametreler algoritmadaki buluşsal mekanizmaların davranışını kontrol etmektedirler. Bu nedenle, yüksek performans elde etmek için algoritma parametrelerinin uygun şekilde ayarlanması gerekir. Bu amaçla literatürde tam faktöriyel deney tasarımı ve Taguchi yöntemi gibi istatistiksel deneysel tasarım yöntemleri sıkça tercih edilmektedir (Hoss, 2012).

Tam faktöriyel deney tasarımında her bir faktörün her bir seviyesini içeren olası tüm kombinasyonlar yapılacak olan deneylerin sayısını belirlemektedir. Faktör ve seviyelerinin sayısı arttıkça hem zaman hemde maliyet artacaktır. Taguchi yönteminde ise geliştirilen ortogonal diziler sayesinde çok az deney ile çok iyi sonuçlar bulunabilmektedir.

Taguchi, çeşitliliği azaltmak ve performans kriteri olarak kullanılması amacıyla sinyal/gürültü (S/G) oranı olarak adlandırılan bir dizi istatistik geliştirmiştir. Taguchi, deneyin hedefinin türüne göre en küçük en iyi, hedef değer en iyi ve en büyük en iyi olarak üç çeşit S/G oranı tanımlamıştır. Tanımlanan S/G oranları Tablo 6.1'de gösterilmiştir (Şirvancı, 1997).

**Tablo 6.1.** S/G oranlarının hesaplanması.

<b>Karakteristik tipi</b>	<b>S/G Oranı</b>
En küçük en iyi	$-10\log\left(\frac{1}{n}\sum y_i^2\right)$
Hedef değer en iyi	$-10\log\left(\frac{\bar{y}}{s^2}\right)$
En büyük en iyi	$-10\log\left(\frac{1}{n}\sum \frac{1}{y_i^2}\right)$

Hangi çeşit S/G oranı kullanılırsa kullanılsın en büyük S/G oranı en iyi deney sonucunu göstermektedir. Ayrıca S/G oranına ek olarak varyans analizi (ANOVA) ile faktörlerin süreçler üzerindeki etkileride istatistiksel olarak belirlenmektedir (Yang ve Tarn, 1998).

Taguchi yönteminin adımları sırasıyla problemin belirlenmesi, performans karakteristiklerini etkileyen parametrelerin ve her parametre için seviyelerin belirlenmesi, uygun ortogonal dizinin seçilmesi, parametrelerin sütunlara atanması, deneyin uygulama sırasının belirlenmesi, deneylerin gerçekleştirilmesi ve sonuçların analiz edilmesidir (Şirvancı, 1997). Geliştirilen HKKO-YAKA için tanımlanan parametreler, karınca sayısı, iterasyon sayısı, feromon miktarının önemini belirleyen sabit ( $\alpha$ ), sezgisel bilginin önemini belirleyen sabit ( $\beta$ ), buharlaşma oranı ( $\rho$ ), başlangıç sıcaklığı ( $T_0$ ), bitiş sıcaklığı ( $T_f$ ) ve soğutma katsayısı ( $\epsilon$ ), besin kaynağı sayısı, işçi arı sayısı, gözcü arı sayısı, limit ve iterasyon sayısı (YAKA)'dır. Bu parametrelerden karınca sayısı için iki farklı seviye, iterasyon sayısı, feromon miktarının önemini belirleyen sabit ( $\alpha$ ), sezgisel bilginin önemini belirleyen sabit ( $\beta$ ), buharlaşma oranı ( $\rho$ ), başlangıç sıcaklığı ( $T_0$ ), bitiş sıcaklığı ( $T_f$ ) ve soğutma katsayısı ( $\epsilon$ ) için ise üç farklı seviye belirlenmiştir. Besin kaynağı sayısı, işçi arı sayısı, gözcü arı sayısı, limit ve iterasyon sayısı (YAKA) için sabit seviyeler belirlenmiştir. Bu doğrultuda deneysel çalışma için L18 ortogonal dizisi kullanılmıştır. Tablo 6.2'de belirlenen parametreler ve seviyeleri görülmektedir.



**Tablo 6.2.** Belirlenen parametreler ve seviyeleri.

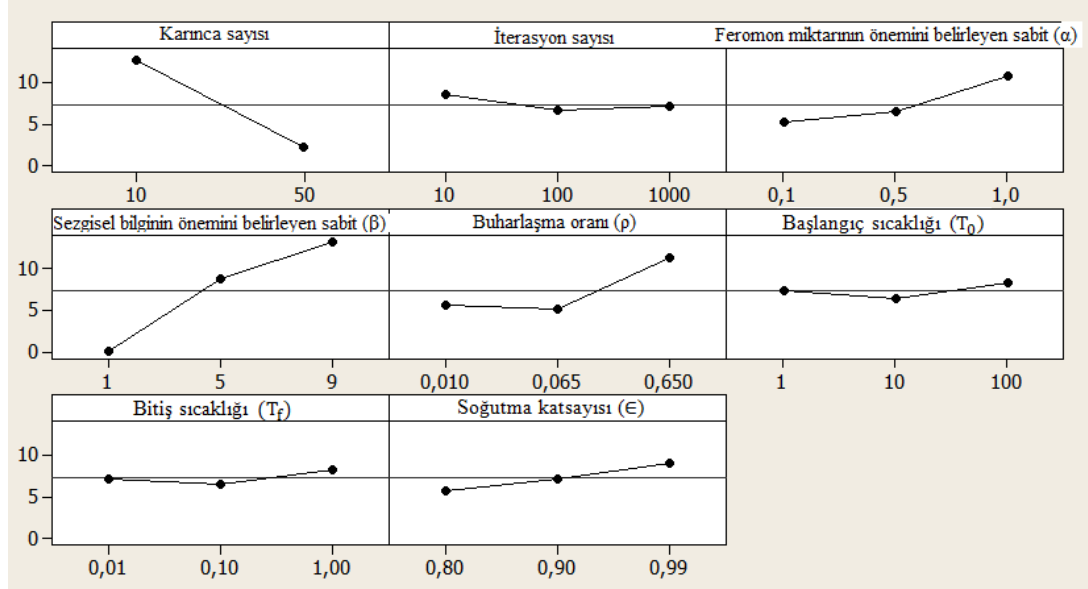
Parametreler	Seviye 1	Seviye 2	Seviye 3
Karınca sayısı	10	50	-
İterasyon sayısı (HKKO)	10	100	1000
$\alpha$	0,1	0,5	1
$\beta$	1	5	9
$\rho$	0,01	0,065	0,65
$T_0$	1	10	100
$T_f$	0,01	0,1	1
$\epsilon$	0,8	0,9	0,99
Besin kaynağı sayısı	Karınca sayısı		
İşçi arı sayısı	Karınca sayısı		
Gözcü arı sayısı	Karınca sayısı		
Limit	Müşteri sayısı* besin kaynağı sayısı		
İterasyon sayısı (YAKA)	10000		

Bu çalışmada kullanılacak FORT test problemleri 5 ve 9 adet şarj istasyonu içeren iki farklı veri setinden oluşmaktadır. Her veri setinde, rastgele dağıtılmış 100, 200 ve 400 müşteri ile 10 örnekten oluşan üç set bulunmaktadır. Toplamda, veri seti 60 örnek içermektedir. Her bir veri seti için farklı müşteri sayısına sahip birer problem olacak şekilde toplamda 6 adet problem tesadüfi olarak seçilmiştir. Seçilen problemler N100-16, N100-21, N200-16, N200-29, N400-12 ve N400-22'dir. Bu 6 problem için, HKKO-YAKA L18 dizisinin her bir parametre kombinasyonuna göre üç kez çalıştırıldı. Böylece toplam 324 deneme yapılmıştır. Test problemlerinin çözümü sonucunda amaç fonksiyonu olarak toplam maliyet hesabı yapılmaktadır. Fakat algoritmanın sonucunda bazı parametre kombinasyonları ile daha uzun sürede bazılarıyla ise daha kısa sürede en iyi sonuçlar elde edilmektedir. Bu nedenle geliştirilen yöntemin performansını ölçmek için amaç fonksiyonu değeri ile çözüm süresi birlikte değerlendirilmelidir. Bu doğrultuda amaç fonksiyonu değeri olan toplam maliyetin ve algoritmanın çözüm süresinin üç tekrar sonucu elde edilen ortalamaları normalleştirilip toplanmış ve normalleştirilmiş toplam değer (NTD) olarak Tablo 6.3'te sekiz parametrelili üç seviyeli ortogonal dizisi ile gösterilmiştir.

**Tablo 6.3.** L18 ortogonal dizisi ve deney sonuçları.

Karınca Sayısı	İterasyon Sayısı (HKKO)	$\alpha$	$\beta$	$\rho$	$T_0$	$T_f$	$\epsilon$	NTD (N100-16)	NTD (N100-21)	NTD (N200-16)	NTD (N200-29)	NTD (N400-12)	NTD (N400-22)
1	1	1	1	1	1	1	1	0,728	0,757	0,777	0,807	1,069	1,080
1	1	2	2	2	2	2	2	0,282	0,315	0,276	0,320	0,328	0,330
1	1	3	3	3	3	3	3	0,043	0,004	0,002	0,055	0,007	0,014
1	2	1	1	2	2	3	3	0,864	0,901	0,952	0,892	0,959	0,977
1	2	2	2	3	3	1	1	0,198	0,199	0,191	0,201	0,188	0,196
1	2	3	3	1	1	2	2	0,172	0,144	0,110	0,144	0,118	0,132
1	3	1	2	1	3	2	3	0,276	0,264	0,262	0,251	0,201	0,203
1	3	2	3	2	1	3	1	0,204	0,187	0,197	0,191	0,130	0,146
1	3	3	1	3	2	1	2	0,338	0,290	0,239	0,290	0,213	0,233
2	1	1	3	3	2	2	1	0,422	0,452	0,523	0,574	0,284	0,284
2	1	2	1	1	3	3	2	1,622	1,639	1,965	1,961	1,998	1,997
2	1	3	2	2	1	1	3	0,433	0,450	0,558	0,575	0,300	0,290
2	2	1	2	3	1	3	2	0,513	0,533	0,621	0,572	0,318	0,298
2	2	2	3	1	2	1	3	0,499	0,520	0,606	0,624	0,285	0,294
2	2	3	1	2	3	2	1	1,637	1,647	1,903	1,942	1,813	1,908
2	3	1	3	2	3	1	2	0,786	0,807	0,831	0,782	0,318	0,322
2	3	2	1	3	1	2	3	1,478	1,438	1,400	1,405	0,885	0,966
2	3	3	2	1	2	3	1	0,832	0,869	0,898	0,870	0,375	0,353

Bu çalışmanın amacı toplam maliyeti ve algoritmanın çözüm zamanını en aza indirmektir. Bu yaklaşımdan yola çıkarak 8 faktör ve 3 seviyeli L18 ortogonal dizisine göre her bir deneme için Minitab 21 programı kullanılarak “en küçük en iyi” analizi yapılmıştır. Parametrelerin her bir seviyesi için ortalama S/G oranı grafiği Şekil 6.1’de yer almaktadır.



Şekil 6.1. Parametrelerin seviyeleri için S/G oranları.

Şekil 6.1’deki S/G oranları incelendiğinde, her bir parametre için en yüksek orana sahip olan seviyeler en uygun seviye olarak seçilmiştir. Parametrelerin en uygun seviyeleri ve değerleri Tablo 6.4’te gösterilmektedir.

Tablo 6.4. Parametrelerin en uygun seviyeleri ve değerleri.

Parametreler	En uygun seviye	Değeri
Karınca sayısı	Seviye 1	10
İterasyon sayısı (HKKO)	Seviye 1	10
$\alpha$	Seviye 3	1
$\beta$	Seviye 3	9
$\rho$	Seviye 3	0,65
$T_0$	Seviye 3	100
$T_f$	Seviye 3	1
$\epsilon$	Seviye 3	0,99

Tablo 6.4 incelendiğinde Taguchi deney tasarımı yöntemi ile karınca sayısı (10) ve iterasyon sayısı (10) için birinci seviye, feromon miktarının önemini belirleyen sabit (1), sezgisel bilginin önemini belirleyen sabit (9), buharlaşma oranı (0,65), başlangıç sıcaklığı (100), bitiş sıcaklığı (1) ve soğutma katsayısı (0,99) için ise üçüncü seviye en uygun seviye olarak seçilmiştir.

Parametreler için hangi seviyelerin daha uygun olduğu bulunduğundan sonra varyans analizi yapılmış ve sonuçları Tablo 6.5'te gösterilmiştir. Parametreler içerisinde en önemli parametre, P değerinin sifıra en yakın olduğu (0,01) karınca sayısı olduğu sonucuna varılmıştır.

**Tablo 6.5.** S/G oranlarının varyans analizi.

Parametreler	Serbestlik Derecesi	Düzeltilmiş Kareler Toplamı	Düzeltilmiş Kareler Ortalaması	F-Değeri	p-Değeri
Karınca sayısı	1	479,221	479,221	100,82	0,010
İterasyon sayısı (HKKO)	2	11,639	5,820	1,22	0,450
$\alpha$	2	99,983	49,991	10,52	0,087
$\beta$	2	540,345	270,172	56,84	0,017
$\rho$	2	136,269	68,135	14,33	0,065
$T_0$	2	11,198	5,599	1,18	0,459
$T_f$	2	8,964	4,482	0,94	0,515
$\epsilon$	2	32,585	16,293	3,43	0,226
<b>Hata</b>	2	9,507	4,753		
<b>Toplam</b>	17				

## 6.2. Geliştirilen HKKO-YAKA'nın Sınanması

Bu bölümde, geliştirilen HKKO-YAKA'nın performansı literatürde FORT (Felipe ve ark, 2014) olarak bilinen test problemi kullanılmıştır. Araştırmalarımıza göre Felipe ve ark. (2014) EARP'yi birden fazla şarj teknolojisi ve kısmi şarj ile birlikte ele alarak toplam şarj etme maliyetini en küçükleyen bir amaç fonksiyonu kullanan literatürde tek çalışmadır. FORT test problemleri 5 ve 9 adet şarj istasyonu içeren iki farklı veri setinden oluşmaktadır. Her veri setinde, rastgele dağıtılmış 100, 200 ve 400 müşteri ile 10 örnekten oluşan üç set bulunmaktadır. Toplamda, veri seti 60 örnek içermektedir. Tüm veri seti <http://www.vrp-rep.org/datasets/item/2018->

0004.html adresinde mevcuttur. Geliştirilen HKKO-YAKA, MATLAB programa dilinde kodlanmıştır ve tüm örnekler Intel Core i7 CPU 4.5 GHz işlemci ve 16 GB RAM'e sahip bir bilgisayar üzerinde koşturulmuştur. Elde edilen sonuçlar Felipe ve ark. (2014) ve Keskin ve Çatay (2018)'in sonuçları ile karşılaştırılmıştır. Felipe ve ark. (2014) çalışmalarında 2-opt ve yeniden ekleme gibi yerel arama operatörler kullanarak tavlama benzetimi tabanlı bir algoritma geliştirmişlerdir. Keskin ve Çatay (2018) ise çalışmalarında kesin çözüm yöntemi ve uyarlanabilir geniş komşu arama yöntemini bir arada kullanarak bir mat-sezgisel algoritma önermişlerdir.

FORT örneklerini çözerek elde edilen ortalama sonuçlar Tablo 6.6'da ve ayrıntılı sonuçlar ise Tablo 6.7'de verilmiştir. Tablo 6.6 ve Tablo 6.7'deki FORT Felipe ve ark. (2014)'ün sonuçlarını, Mat-sezgisel Yöntem Keskin ve Çatay (2018)'ün sonuçlarını göstermektedir. “% İyileştirme” Felipe ve ark. (2014)'ün sonuçları arasındaki farkın yüzdesini göstermektedir. İyileştirme yüzdesi aşağıdaki gibi formüle edilir.

$$\%İyileştirme = \frac{((\text{Ortalama toplam maliyet (FORT)} - \text{Ortalama toplam maliyet}) / \text{Ortalama toplam maliyet (FORT)}) * 100}{1}$$

**Tablo 6.6.** Ortalama toplam maliyet ve hesaplama süresinin karşılaştırılması.

Şarj istasyonu sayısı	Müşteri Sayısı	FORT		Mat-sezgisel Yöntem			Geliştirilen Hiyerarşik Yöntem		
		Ortalama Toplam Maliyet	Ortalama Hesaplama süresi	Ortalama Toplam Maliyet	Ortalama Hesaplama süresi	% İyileştirme	Ortalama Toplam Maliyet	Ortalama Hesaplama süresi	% İyileştirme
9	100	70,34	274	63,91	181	9,13	62,56	261	11,03
5		71,59	268	65,12	180	9,02	63,55	259	11,24
<b>Ortalama</b>		70,97	271	64,52	181	<b>9,07</b>	63,06	260	<b>11,14</b>
9	200	110,38	533	98,75	798	11,19	97,57	656	12,67
5		114,36	522	101,27	770	11,44	100,55	650	12,03
<b>Ortalama</b>		112,37	528	100,01	784	<b>11,32</b>	99,06	653	<b>12,35</b>
9	400	195,75	1181	176,61	2936	9,84	177,94	2480	9,17
5		203,18	1101	181,61	3247	10,58	180,01	2300	11,35
<b>Ortalama</b>		199,66	1141	179,11	3092	<b>10,21</b>	178,98	2390	<b>10,26</b>

FORT: Felipe ve ark. (2014), Mat-sezgisel Yöntem: Keskin ve Çatay (2018)

**Tablo 6.7.** Detaylı toplam maliyetlerin karşılaştırılması.

Şarj istasyonu sayısı	Müşteri Sayısı	FORT	Mat-sezgisel Yöntem		Geliştirilen Hiyerarşik Yöntem	
		Toplam Maliyet	Toplam Maliyet	% iyileştirme	Toplam Maliyet	% iyileştirme
<b>9</b>	<b>100</b>	73,42	65,12	11,30	<b>60,89</b>	17,07
		65,97	60,49	8,31	<b>60,82</b>	7,81
		66,98	62,95	6,02	<b>61,84</b>	7,67
		70,96	65,41	7,82	<b>64,01</b>	9,79
		78,22	73,82	5,63	<b>72,21</b>	7,68
		72,16	65,72	8,92	<b>65,16</b>	9,70
		73,87	65,40	11,47	<b>65,25</b>	11,67
		62,70	57,21	8,76	<b>56,43</b>	10,00
		72,45	62,80	13,32	<b>61,22</b>	15,50
		66,70	60,21	9,73	<b>57,73</b>	13,45
<b>5</b>	<b>100</b>	73,79	65,29	11,52	<b>62,14</b>	15,79
		66,58	60,92	8,50	<b>60,22</b>	9,55
		66,92	63,12	5,68	<b>59,51</b>	11,07
		71,74	68,24	4,88	<b>67,70</b>	5,63
		82,02	76,28	7,00	<b>75,51</b>	7,94
		73,59	66,25	9,97	<b>64,43</b>	12,45
		74,00	67,65	8,58	<b>64,59</b>	12,72
		62,74	57,21	8,81	<b>55,64</b>	11,32
		73,28	65,37	10,79	<b>64,97</b>	11,34
		71,19	60,91	14,44	<b>60,82</b>	14,57
<b>9</b>	<b>200</b>	-	105,54	-	108,2	-
		108,24	96,4	10,94	<b>95,25</b>	12,00
		110,21	97,34	11,68	<b>94,6</b>	14,16
		108,69	95,78	11,88	<b>92,18</b>	15,19
		117,44	103,65	11,74	<b>102,77</b>	12,49
		111,26	98,17	11,77	<b>96,57</b>	13,20
		110,42	98,48	10,81	99,5	9,89
		102,37	93,66	8,51	<b>90,56</b>	11,54
		110,08	97,41	11,51	<b>95,89</b>	12,89
		114,71	101,06	11,90	<b>100,14</b>	12,70

**Tablo 6.7. (Devamı)** Detaylı toplam maliyetlerin karşılaştırılması.

Şarj istasyonu sayısı	Müşteri Sayısı	FORT	Mat-sezgisel Yöntem		Geliştirilen Hiyerarşik Yöntem	
		Toplam Maliyet	Toplam Maliyet	% iyileştirme	Toplam Maliyet	% iyileştirme
<b>5</b>	<b>200</b>	124,11	110,48	10,98	<b>109,12</b>	12,08
		110,15	98,49	10,59	<b>97,98</b>	11,05
		109,64	99,18	9,54	<b>97,95</b>	10,66
		112,65	99,03	12,09	100,1	11,14
		121,81	104,88	13,90	<b>101,13</b>	16,98
		115,05	102,23	11,14	<b>101,77</b>	11,54
		113,76	102,52	9,88	<b>99,75</b>	12,32
		106,7	92,42	13,38	95,01	10,96
		113,46	100,32	11,58	<b>99,56</b>	12,25
		116,23	103,12	11,28	<b>103,09</b>	11,31
<b>9</b>	<b>400</b>	198,48	179,56	9,53	181,20	8,71
		196,5	176,34	10,26	179,4	8,70
		195,72	177,34	9,39	<b>176,18</b>	9,98
		190,28	172	9,61	175,25	7,90
		192,67	176,13	8,58	180,9	6,11
		200,66	179,52	10,54	181,4	9,60
		194,4	176,17	9,38	177,3	8,80
		194,92	176,03	9,69	<b>174,72</b>	10,36
		198,16	175,12	11,63	<b>173,57</b>	12,41
		-	177,88	-	179,5	-
<b>5</b>	<b>400</b>	206,6	182,7	11,57	<b>178,13</b>	13,78
		201,98	180,38	10,69	<b>179,75</b>	11,01
		207,55	183,56	11,56	<b>179,75</b>	13,39
		192,96	177,07	8,23	179,02	7,22
		199,21	177,34	10,98	<b>175,93</b>	11,69
		206,75	184,41	10,81	<b>183,91</b>	11,05
		205,96	182,73	11,28	183,2	11,05
		195,06	181,1	7,16	<b>179,67</b>	7,89
		207,38	184,38	11,09	<b>180,54</b>	12,94
		208,33	182,41	12,44	<b>180,18</b>	13,51

FORT: Felipe ve ark. (2014), Mat-sezgisel Yöntem: Keskin ve Çatay (2018)



Tablo 6.6'daki sonuçlara göre, geliştirilen hiyerarşik yöntemin Felipe ve ark. (2014)'ün elde ettiği sonuçlara kıyasla müşteri sayısı 100, 200 ve 400 için sırasıyla %11,14, %12,35 ve %10,26 iyileştirme elde ettiği görülmektedir. Ayrıca Felipe ve ark. (2014) test problemlerinin çözüm sürelerini müşteri sayısı 100, 200 ve 400 için sırasıyla ortalama 271, 528, 1141 saniye, Keskin ve Çatay (2018) ise sırasıyla ortalama 181, 784 ve 3092 saniye olarak raporlamışlardır. Önerdiğimiz hiyerarşik yöntem ise ortalama 260, 653 ve 2390 saniyede sırasıyla müşteri sayısı 100, 200 ve 400 olan problemleri çözebilmiştir.

Tablo 6.7'deki sonuçlar ayrıca geliştirilen hiyerarşik yöntemin Felipe ve ark. (2014)'ün elde ettiği çözümlerden hepsini, Keskin ve Çatay (2018)'in elde ettiği çözümlerden ise kırk yedi örneğinin sonucunu iyileştirdiğini göstermektedir. İyileştirilmiş çözümler koyu fontta gösterilmiştir. Yöntemler farklı işlemciler ile kullanıldığından, hesaplama zamanını karşılaştırmak çok doğru olmayacaktır, fakat yine de, iyileştirme sonuçlarının geliştirdiğimiz yöntemin etkinliğini gösterdiğine inanmaktayız.



## 7. ÖRNEK OLAY ÇALIŞMASI

Bu bölümde Türkiye genelinde faaliyet gösteren bir süpermarket zinciri ele alınarak haftalık taleplerinin karşılanabilmesi için en uygun araç rotalarının belirlenmesi problemi ele alınmıştır. Süpermarket zincirinin 1 adet ana deposu (D) ve ürün dağıtımının yapılacağı 94 adet mağazası (M) bulunmaktadır. Dağıtımını yapan araç filosu homojen kapasiteye sahip elektrikli kamyonlardan oluşmaktadır. Dağıtım alanında araçların bataryalarını şarj edebilecekleri 5 adet şarj istasyonu (R) bulunmaktadır. Mağazaların önceki üç aylık taleplerinin ortalaması alınarak 20 haftalık talep verileri elde edilmiştir. Müşterilerin 1. hafta talep miktarları ile ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki uzaklık matrisinin bir kısmı Tablo 7.1 ve Tablo 7.2’de verilmiştir.

**Tablo 7.1.** Mağazaların 1. hafta talepleri (ton).

M1	M2	M3	M4	M5	M6	M7	M8
9	6	5	7	7	3	9	3

**Tablo 7.2.** Ana deponun, müşterilerin ve şarj istasyonlarının birbirleri arasındaki mesafeler (km).

	M1	M2	M3	M4	M5	D	R1	R2	R3	R4	R5
M1	0	51	48	29	32	136	50	56	63	80	65
M2	51	0	44	29	24	153	88	80	68	65	76
M3	48	44	0	56	55	132	76	67	26	40	51
M4	29	29	56	0	7	64	71	70	64	64	95
M5	32	24	55	7	0	153	55	96	65	97	44
D	136	153	132	64	153	0	40	87	69	28	55
R1	50	88	76	71	55	40	0	50	37	46	82
R2	56	80	67	70	96	87	50	0	77	74	75
R3	63	68	26	64	65	69	37	77	0	85	31
R4	80	65	40	64	97	28	46	74	85	0	95
R5	65	76	51	95	44	55	82	75	31	95	0

Gerçek hayat problemi ile ilgili diğer bilgiler aşağıdaki gibidir:

- 36 ton kapasiteli 20 adet elektrikli kamyon bulunmaktadır.
- Araç bataryalarının kapasitesi 435 kWsa'dır.
- Şarj istasyonlarında iki tür (hızlı ve orta) şarj teknolojisi vardır. Hızlı olanın şarj etme hızı 45 kWsa/sa'dır. Şarj etme maliyeti 1,33 TL/kWsa'dır. Orta olanın yeniden şarj olma hızı 22 kWsa/sa'dır. Şarj etme maliyeti 0,95 TL/kWsa'dır.
- Depodaki şarj teknolojisi yavaş şarjdır. Yavaş şarj, gece şarjıdır ve en ucuzudur. Yavaş şarj etme hızı 3,7 kWsa/sa'dır. Şarj etme maliyeti 0,14 TL/kWsa'dır.
- Şarj etmenin sabit maliyeti 19,38 TL/döngü'dür. (Elektriğin 2020 yılındaki ortalama birim maliyeti dikkate alınarak hesaplamalar yapılmıştır.).
- Araçlar homojen olduğu için her aracın tam yüklü ve yüksüz enerji tüketim oranları sırasıyla 2,18 ve 1,29 kWsa/km'ye eşittir.
- Bir tur için izin verilen maksimum süre 40 saattir.
- Her mağazanın servis süresi ortalama 30 dakikadır.
- Mağazalar arasında beklenen ortalama hız 90 km/sa'dır.

Senaryo 1, Senaryo 2 ve Senaryo 3, 20 haftalık talep miktarları ele alınarak geliştirilen hiyerarşik yöntem, ÇAPSO ve uyarlanmış ÇAPSO ile çözülmüştür.

### 7.1. Senaryoların HKKO-YAKA ile Çözümü

Geliştirilen HKKO-YAKA, Pareto-optimal çözümleri bulmak amacıyla ATY ve KSY için arama işlemi olarak kullanılmıştır.

İlk olarak Senaryo 1, 2 ve 3'ün 1. hafta talepleri dikkate alınarak her bir amaç fonksiyonu ( $f_i$ ) problemin kısıtları altında geliştirilen hiyerarşik yöntem ile ayrı ayrı çözülmüş ve pozitif ideal çözümler (en iyi çözüm) bulunmuştur. Daha sonrasında ise, her bir çözümün karar değişkenleri girdi olarak alınmış ve kalan amaç fonksiyonları dikkate alınarak model çözülmüştür. İşlem, tüm amaç fonksiyonları için model çözümlene kadar tekrarlanmıştır. Tüm çözümler arasında, her bir amaç fonksiyonu için negatif ideal çözüm (en kötü çözüm) bulunmuştur. Bulunan çözümler hiyerarşik yöntemin 50 kez koşturulması sonucunda alınan ortalama değerlerdir. Senaryo 1 için elde edilen ortalama değerler Tablo 7.3'te verilmiştir.

**Tablo 7.3.** Senaryo 1 için amaç fonksiyonu değerleri.

Amaç Fonksiyonu	Enk $f_1$	Enk $f_2$	Enk $f_3$	Enk $f_4$	Enk $f_5$
$f_1$	5640,8 (en iyi)	8089 (en kötü)	6548,7	6452,2	6164
$f_2$	16	15 (en iyi)	16	16	17 (en kötü)
$f_3$	5622	6187 (en kötü)	5150 (en iyi)	5155	5973
$f_4$	9381,2	10731,8 (en kötü)	8674,1	8592,9 (en iyi)	9944,1
$f_5$	16	21 (en kötü)	20	19	14 (en iyi)

$f_1$ :Toplam şarj etme maliyeti,  $f_2$ :Gerekli olan EA sayısı,  $f_3$ :Toplam seyahat mesafesi,  $f_4$ :Enerji tüketimi,  $f_5$ :Uğranması gereken şarj istasyonu sayısı

Senaryo 2 için ortalama değerler Tablo 7.4'te verilmiştir.

**Tablo 7.4.** Senaryo 2 için amaç fonksiyonu değerleri.

Amaç Fonksiyonu	Enk $f_1$	Enk $f_2$	Enk $f_3$	Enk $f_4$	Enk $f_5$
$f_1$	3494,7 (en iyi)	6038,1 (en kötü)	3535,3	3650	4278,1
$f_2$	17	15 (en iyi)	16	16	17 (en kötü)
$f_3$	5336	6669 (en kötü)	5130 (en iyi)	5158	5959
$f_4$	8831,9	10649,9 (en kötü)	8634,3	8626,1 (en iyi)	9872,4
$f_5$	18	23 (en kötü)	19	19	14 (en iyi)

$f_1$ :Toplam şarj etme maliyeti,  $f_2$ :Gerekli olan EA sayısı,  $f_3$ :Toplam seyahat mesafesi,  $f_4$ :Enerji tüketimi,  $f_5$ :Uğranması gereken şarj istasyonu sayısı

Senaryo 3 için elde edilen ortalama değerler ise Tablo 7.5'te verilmiştir.

**Tablo 7.5.** Senaryo 3 için amaç fonksiyonu değerleri.

Amaç Fonksiyonu	Enk f <sub>1</sub>	Enk f <sub>2</sub>	Enk f <sub>3</sub>	Enk f <sub>4</sub>	Enk f <sub>5</sub>
f <sub>1</sub>	3657 (en iyi)	6565,5 (en kötü)	4078,1	3892	4832,1
f <sub>2</sub>	17 (en kötü)	15 (en iyi)	16	16	16
f <sub>3</sub>	5339	6280 (en kötü)	5203 (en iyi)	5194	5999
f <sub>4</sub>	8863,1	10846,7 (en kötü)	8817,3	8685,8 (en iyi)	9963,5
f <sub>5</sub>	18	23 (en kötü)	20	18	14 (en iyi)

f<sub>1</sub>:Toplam şarj etme maliyeti, f<sub>2</sub>:Gerekli olan EA sayısı, f<sub>3</sub>:Toplam seyahat mesafesi, f<sub>4</sub>:Enerji tüketimi, f<sub>5</sub>:Uğranması gereken şarj istasyonu sayısı

Amaç fonksiyonlarının birimleri farklılık gösterdiğinden, her bir amaç fonksiyonunu aynı ölçekte sunmak için normalizasyon işlemi uygulanmıştır. Denklem 7.1, amaç fonksiyonu hem en küçükleme hem de en büyükleme olduğundaki normalizasyon sürecini göstermektedir.

$$f'_i = \begin{cases} \frac{WS_{f_i} - f_i}{WS_{f_i} - BS_{f_i}}, & \text{en küçük amaç fonksiyonu için} \\ \frac{f_i - WS_{f_i}}{BS_{f_i} - WS_{f_i}}, & \text{en büyük amaç fonksiyonu için} \end{cases} \quad (7.1)$$

Denklem 7.1'de  $f'_i$ , i. amaç fonksiyonunun normalleştirilmiş değerini,  $WS_{f_i}$   $f_i$  amaç fonksiyonunun en kötü çözümünü,  $BS_{f_i}$   $f_i$  amaç fonksiyonunun en iyi çözümünü ifade etmektedir.

Amaç fonksiyon değerleri Tablo 7.3, Tablo 7.4 ve Tablo 7.5'teki en iyi ve en kötü değerler dikkate alınarak normalize edildikten sonra skalerleştirme işlemi yapılmıştır. Beş adet amaç fonksiyonunun ATY ve KSY ile skalerleştirilmesi sonucunda sırasıyla Denklem 7.2 ve denklem 7.3'teki amaç fonksiyonları elde edilmiştir.

$$\text{Enb } w_1f'_1(x) + w_2f'_2(x) + w_3f'_3(x) + w_4f'_4(x) + w_5f'_5(x) \quad (7.2)$$

$$\begin{aligned} & \text{Enb } w_1(f'_1(x) - a_1) + w_2(f'_2(x) - a_2) + w_3(f'_3(x) - a_3) + \\ & w_4(f'_4(x) - a_4) + w_5(f'_5(x) - a_5) + \alpha(|(f'_1(x) - a_1)| + \\ & |(f'_2(x) - a_2)| + |(f'_3(x) - a_3)| + |(f'_4(x) - a_4)| + |(f'_5(x) - a_5)|) \end{aligned} \quad (7.3)$$

Skalerleştirilmiş amaç fonksiyonlarındaki ağırlık değerleri her değiştirildiğinde yeni bir tek amaçlı optimizasyon problemi oluşmaktadır. Farklı ağırlık değerleri de farklı Pareto çözümlerin elde edilmesini sağlar. Farklı Pareto-optimal çözümlerin bulunabilmesi için optimizasyon algoritmasının farklı ağırlıklar kullanılarak çalıştırılması gerekmektedir. Bu nedenle amaç fonksiyonundaki  $w$  ağırlık değerleri için rastgele belirlenen 10 farklı değer Tablo 7.6' da verilmiştir.

**Tablo 7.6.** Rastgele seçilen amaç fonksiyonu ağırlık değerleri.

No	$\alpha$	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$
1	0,01	0,36	0,02	0,12	0,23	0,27
2	0,06	0,25	0,30	0,07	0,20	0,18
3	0,03	0,04	0,22	0,30	0,18	0,26
4	0,07	0,08	0,10	0,18	0,43	0,21
5	0,12	0,15	0,21	0,14	0,13	0,37
6	0,02	0,36	0,14	0,25	0,22	0,03
7	0,10	0,47	0,06	0,19	0,17	0,11
8	0,12	0,13	0,15	0,32	0,21	0,19
9	0,06	0,23	0,07	0,12	0,48	0,10
10	0,19	0,20	0,20	0,20	0,20	0,20

Senaryo 1, Senaryo 2 ve Senaryo 3, Tablo 7.6'daki farklı ağırlık değerleri dikkate alınarak HKKO-YAKA ile ayrı ayrı çözülmüştür. HKKO-YAKA'da Bölüm 6'da belirlenen parametre değerleri kullanılmıştır. Senaryoların 1. Hafta talep verileri dikkate alınarak çözümü sonucunda farklı ağırlık değerleri için ATY ile elde edilen Pareto-optimal çözümler Tablo 7.7'de verilmiştir.

**Tablo 7.7.** Farklı ağırlık değerleri için ATY ile elde edilen Pareto-optimal çözümler.

	Enk $f_1$	Enk $f_2$	Enk $f_3$	Enk $f_4$	Enk $f_5$	$\sum f'_i(x)$	
<b>Senaryo 1</b>	1	5923,4	17	6060	9890,5	15	3,45
	2	5994,7	16	5592	9436,5	16	1,77
	3	5982,1	16	5634	9098,7	17	1,69
	4	5922,0	16	5776	9825,8	15	1,91
	5	5821,2	17	5636	9357,5	15	1,64
	6	5944,3	16	5443	9120,8	18	1,57
	7	5963,9	16	5900	9867,5	16	2,68
	8	6005,9	16	5684	9322,7	16	1,82
	9	6047,9	16	5481	9209,9	17	1,87
	10	5994,7	16	5531	9304,5	16	1,50
<b>Senaryo 2</b>	1	3952,2	16	5495	9259,9	17	3,06
	2	3897,6	16	5245	8914,8	19	1,48
	3	3899,0	16	5340	9001,3	17	1,63
	4	3915,8	16	5377	9059,4	17	1,99
	5	3892,0	16	5508	9213,2	15	2,28
	6	3851,4	16	5479	8785,3	20	2,04
	7	4003,9	16	5379	9122,3	18	2,82
	8	3915,8	16	5301	8959,2	18	1,69
	9	3928,4	16	5308	8980,2	19	2,03
	10	3824,9	16	5333	8958,9	17	1,10
<b>Senaryo 3</b>	1	4159,2	17	5625	9428,3	15	3,23
	2	4149,4	16	5392	9111,6	18	1,71
	3	4328,5	16	5359	8988,5	16	1,60
	4	4134,0	16	5408	9188,2	18	1,82
	5	3999,7	17	5765	9599	14	3,24
	6	3897,6	16	5470	9152	18	1,34
	7	4115,8	16	5258	9079,1	18	1,32
	8	4101,9	16	5333,5	9094,85	17	1,32
	9	4009,5	16	5255	8916,3	21	1,26
	10	4090,7	16	5330	8926,3	18	1,18

$f_1$ :Toplam şarj etme maliyeti,  $f_2$ :Gerekli olan EA sayısı,  $f_3$ :Toplam seyahat mesafesi,  $f_4$ :Enerji tüketimi,  $f_5$ :Uğranması gereken şarj istasyonu sayısı

Senaryoların 1. Hafta talep verileri dikkate alınarak çözümü sonucunda farklı ağırlık değerleri için KSY ile elde edilen Pareto-optimal çözümler Tablo 7.8’de verilmiştir.



**Tablo 7.8.** Farklı ağırlık değerleri için KSY ile elde edilen Pareto-optimal çözümler.

	Enk $f_1$	Enk $f_2$	Enk $f_3$	Enk $f_4$	Enk $f_5$	$\sum f'_i(x)$	
<b>Senaryo 1</b>	1	5905,2	17	5439	9448,2	17	3,05
	2	6115,0	16	5644	9396,1	16	2,69
	3	5927,5	16	5384	9121,8	17	1,71
	4	6060,5	16	5543	9391,8	17	2,68
	5	5824,0	16	5932	9893,9	14	2,05
	6	5902,4	16	5394	9339,8	16	1,53
	7	5808,6	17	5318	8915,2	17	2,00
	8	5958,3	16	5604	9426,9	15	1,81
	9	5814,2	16	5348	9034,9	17	1,19
	10	5835,2	16	5412	9132,9	16	1,13
<b>Senaryo 2</b>	1	3731,1	17	5678	9388,8	15	3,00
	2	3836,0	16	5370	9001,2	18	1,95
	3	3960,6	16	5481	9164,3	17	2,73
	4	3799,7	16	5406	9071,8	19	2,22
	5	3887,8	16	5512	9236,4	15	2,09
	6	3841,6	16	5379	9064,7	17	1,84
	7	3826,3	16	5323	8987,7	16	1,28
	8	3816,5	16	5215	8737,9	19	1,37
	9	3882,2	16	5290	8780,6	17	1,39
	10	3731,1	16	5301	8884,5	18	1,16
<b>Senaryo 3</b>	1	3998,3	17	5424	9138,9	16	3,11
	2	3975,9	16	5395	9040,9	16	1,50
	3	4108,9	16	5344	8972,1	16	1,61
	4	4061,3	16	5268	8853,8	20	1,55
	5	4118,6	16	5312	9016,5	17	1,89
	6	4184,4	16	5323	8955,8	18	2,30
	7	4090,7	17	5325	8960,9	20	3,38
	8	4178,8	16	5243	8896,9	18	1,62
	9	4110,3	16	5342	9032,4	18	2,32
	10	3980,2	16	5302	8931,9	17	0,87

$f_1$ :Toplam şarj etme maliyeti,  $f_2$ :Gerekli olan EA sayısı,  $f_3$ :Toplam seyahat mesafesi,  $f_4$ :Enerji tüketimi,  $f_5$ :Uğranması gereken şarj istasyonu sayısı

Amaç fonksiyonlarının birimleri farklı olduğu için, beş amaç fonksiyonunun her birinin sonuçları normalize edilerek toplanmış ve Tablo 7.7 ve Tablo 7.8'in sağ

sütununda gösterilmiştir. Bu sonuçlar, farklı ağırlık değerlerinin Pareto-optimal çözümlerin üzerindeki etkisini göstermektedir. Ayrıca en uygun Pareto-optimal çözümü ağırlıklar eşit seçildiğinde gözlemlenmiştir. Bu nedenle  $w$ , hem ATY hem de KSY için 0,20 olarak seçilerek işlemler yapılmıştır.

HKKO-YAKA ile 20 haftalık talep verileri kullanılarak elde edilen Pareto-optimal çözümler ise Senaryo 1, Senaryo 2 ve Senaryo 3 için sırasıyla Tablo 7.9, Tablo 7.10 ve Tablo 7.11'de verilmiştir. Tablolardaki 1. sütun haftayı, 2. ve 7. sütun toplam şarj maliyetini (TŞM), 3. ve 8. sütun gerekli toplam araç sayısını (TAS), 4. ve 9. sütun toplam seyahat mesafesini (TSM), 5. ve 10. sütun toplam enerji tüketimini (TET) ve 6. ve 11. sütun ise uğranması gereken toplam şarj istasyonu sayısını (TŞİS) göstermektedir.

**Tablo 7.9.** Senaryo 1'in HKKO-YAKA ile çözümünden elde edilen sonuçlar.

Hafta	ATY-HY					KSY-HY				
	TŞM	TAS	TSM	TET	TŞİS	TŞM	TAS	TSM	TET	TŞİS
1	5994,7	16	5531	9304,5	16	5835,2	16	5412	9132,9	16
2	6150,0	16	5526	9327,7	17	5868,8	16	5312	8924,6	16
3	5976,5	16	5426	9122,5	17	5899,6	16	5201	8898,6	16
4	6036,7	16	5520	9391,3	16	5556,8	16	5564	9196,3	14
5	6063,3	16	5649	9476,4	17	5989,1	16	5633	9626	15
6	6540,3	16	5868	9838,9	16	6161,2	16	5695	9717,1	17
7	5944,4	16	5607	9295,3	14	5590,4	16	5412	9074	14
8	6406,0	16	5465	9121,5	19	6138,8	16	5433	9193,1	17
9	5944,4	16	5536	9265	18	5716,3	16	5564	9756,4	16
10	6341,7	16	5487	9259,8	15	6264,7	16	5361	9164,5	15
11	6159,8	16	5468	9336,6	17	5653,4	16	5581	9313,4	16
12	5744,3	16	5360	8967,6	16	5520,5	16	5376	9108,1	16
13	6207,4	16	5289	9024,4	18	5983,5	16	5583	9433,4	16
14	5940,2	16	5667	9403,7	15	5663,2	16	5709	9471,8	15
15	6031,1	16	5667	9499	16	5905,2	16	5476	9298,2	15
16	6383,6	16	5529	9232	17	5958,3	16	5505	9080,5	16
17	6235,3	16	5626	9615,9	16	5976,5	16	5160	8683,2	16
18	5896,8	16	5281	8808,7	18	5972,3	16	5123	8873,9	16
19	6150,0	16	5722	9643,9	16	6070,3	16	5531	9192,9	15
20	6859,3	16	5483	9464,6	16	5881,4	16	5400	8994,7	15

**Tablo 7.10.** Senaryo 2'nin HKKO-YAKA ile çözümünden elde edilen sonuçlar.

Hafta	ATY-HY					KSY-HY				
	TŞM	TAS	TSM	TET	TŞİS	TŞM	TAS	TSM	TET	TŞİS
1	3824,9	16	5333	8958,9	17	3731,1	16	5301	8884,5	18
2	3764,7	16	5457	9086,5	17	3837,5	16	5458	9106,2	16
3	3584,2	16	5128	8702,7	17	3466,7	16	5078	8594,2	18
4	4108,9	16	5624	9398,5	16	4015,1	16	5426	9271,4	16
5	3781,5	16	5366	8958,1	17	3596,8	16	5329	8918,3	17
6	3995,5	16	5437	9156,7	18	3871,0	16	5429	9022,2	19
7	3897,6	16	5457	9174,6	18	3831,9	16	5306	9049,6	16
8	4050,1	16	5525	9273,2	17	3843,1	16	5404	8967,2	17
9	3942,4	16	5499	9154,7	17	3629,0	16	5253	8755,8	16
10	3721,3	16	5172	8836,8	17	3696,2	16	5162	8772,1	17
11	3767,5	16	5421	9039,7	16	3746,5	16	5208	8833,0	17
12	3844,5	16	5416	9107,8	18	3795,5	16	5296	8956,7	18
13	3922,8	16	5412	9158,2	19	3892,0	16	5516	9185,9	17
14	4185,8	16	5587	9272,8	16	3932,6	16	5545	9186,2	15
15	3924,2	16	5237	8814,9	19	3665,4	16	5205	8765,6	18
16	3911,6	16	5417	9022,2	18	3707,4	16	5334	8847,1	18
17	3927,0	16	5441	9172,7	18	3850,0	16	5414	9031,3	16
18	3647,2	16	5218	8729,2	18	3556,3	16	5205	8764,1	16
19	4050,1	16	5332	9079,6	19	4114,5	16	5443	9274,5	18
20	3733,9	16	5272	8892,4	18	3568,8	16	5206	8733,5	16

**Tablo 7.11.** Senaryo 3'ün HKKO-YAKA ile çözümünden elde edilen sonuçlar.

Hafta	ATY-HY					KSY-HY				
	TŞM	TAS	TSM	TET	TŞİS	TŞM	TAS	TSM	TET	TŞİS
1	4090,7	16	5330	8926,3	18	3980,2	16	5302	8931,9	17
2	4051,5	16	5439	9103,7	15	3775,9	16	5249	8716,0	15
3	3420,6	16	4948	8393,2	16	3560,5	16	5136	8611,2	17
4	3978,8	16	5417	9081,8	17	3816,5	16	5268	8820,5	15
5	3921,4	16	5231	8697,9	16	3907,4	16	5001	8926,3	15
6	3815,1	16	5263	8829,4	19	4057,1	16	5266	8879,0	16
7	3894,8	16	5248	8881,1	16	4008,1	16	5215	8852,0	18
8	4178,8	16	5349	9035,8	19	3974,6	16	5248	8702,5	18
9	4220,8	16	5264	8826,9	18	3882,2	16	5050	8522,7	18
10	3914,4	16	5110	8787,6	20	3899,0	16	5240	8896,0	17
11	3981,6	16	5248	8824,6	19	3869,6	16	5313	8770,4	16
12	3896,2	16	5221	8746,7	19	4041,7	16	5361	8959,7	17
13	4236,2	16	5420	9154,1	19	4086,5	16	5475	9008,9	17
14	4374,7	16	5407	9101,9	19	4269,7	16	5346	9080,5	17
15	3840,3	16	5113	8661,2	21	3519,9	16	5099	8534,9	18
16	4380,3	16	5459	9132,2	17	3985,8	16	5142	9028,3	17
17	4097,7	16	5292	8855,1	18	3831,9	16	5227	8821,4	16
18	4113,1	16	5208	8682,3	18	3739,5	16	5219	8664,5	16
19	4391,5	16	5593	9442,5	19	4279,5	16	5442	9212,7	18
20	4201,2	16	5335	9149,6	18	4005,3	16	5381	9013,7	18

## 7.2. Senaryoların ÇAPSO ile Çözümü

Bu bölümde Senaryo 1, Senaryo 2 ve Senaryo 3 için ÇAPSO ile çözülmüş ve sonuçları verilmiştir.

Çok amaçlı geleneksel yöntemler ile çok amaçlı probleme ait yalnızca tek bir çözüm elde edilebilmektedir. Fakat çok amaçlı evrimsel algoritmalarda çözümlerin aralarında üstünlük bulunmadığı bir çözümler kümesi olan Pareto-optimal küme elde edilmektedir. ÇAPSO'da bunlardan biridir. ÇAPSO'da çok amaçlı problemin çözümüne ait baskın çözümler (Pareto-optimal) depo ismi verilen bir yapıda depolanır. Bu çözümler arasından karar verici bir seçimde bulunmak zorundadır. Amaç fonksiyonu sayısı arttıkça bu seçim daha da zorlaşmaktadır. Bu nedenle bu aşamada Pareto-optimal çözüm kümesi içindeki her bir çözüme ait amaç fonksiyonu değerlerini aynı ölçekte sunmak için kendi içerisinde normalize edip topladık. Amaç fonksiyonlarımız en küçükleme olduğu için amaç fonksiyonu değerleri normalize edilip toplanmış çözümlerden değeri en küçük olanı en uygun Pareto-optimal çözüm olarak belirledik.

ÇAPSO'nun parametre ayarlaması yapılırken Ganji ve ark. (2020)'nin önerdiği parametreler referans alınmıştır. Buna göre hızlandırma sabitleri olan  $c_1$  ve  $c_2$  sırasıyla 1 ve 1,25,  $w$  atalet ağırlığı 0,6, mutasyon olasılığı 0,1, parçacık sayısı 10 ve durdurma kriteri ise 1000 iterasyon olarak belirlenmiştir.

20 haftalık talep verileri kullanılarak Senaryo 1, Senaryo 2 ve Senaryo 3 için ÇAPSO 50 kez koşturulmuş ve Pareto-optimal çözümler elde edilmiştir. Senaryo 1 için elde edilen Pareto-optimal çözümler Tablo 7.12'de verilmiştir.

**Tablo 7.12.** Senaryo 1'in ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
1	12167,1	16	9115	15350,4	37
2	11267,5	16	8308	13990	32
3	10986,3	16	8032	13688	32
4	11313,7	16	7929	13506	35
5	12160,1	16	8288	14241	35
6	11056,3	16	8716	14611	31

**Tablo 7.12. (Devamı)** Senaryo 1'in ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
7	9567,8	16	7654	12971	30
8	11098,3	16	8730	14562	31
9	12529,4	16	8742	14873	37
10	11257,8	16	8802	14957	31
11	10605,8	16	7962	13441	31
12	12287,4	16	9325	15723	34
13	10605,8	16	8595	14440,4	32
14	11936,3	16	8874	15122,4	32
15	11228,4	16	8485	14383,2	33
16	12300,0	17	8630	14723,6	35
17	11022,7	16	7921	13293,7	34
18	11018,5	16	8284	13928,2	31
19	11144,4	16	8601	14361	30
20	11042,3	17	8876	15004,5	31

Senaryo 2 için elde edilen Pareto-optimal çözümler Tablo 7.13'te verilmiştir.

**Tablo 7.13.** Senaryo 2'nin ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
1	9802,8	16	9108	15296,1	37
2	8510,1	16	8308	13990	32
3	8230,3	16	8032	13688	34
4	8338,0	16	8137	13764	34
5	8808,1	16	8373	14259	34
6	8918,6	16	8368	14319	37
7	7530,8	16	7654	12971	30
8	8873,9	16	8436	14333	34
9	9903,5	16	9004	15386	36
10	9286,6	16	8675	14810	32
11	7977,1	16	7962	13441	31
12	8988,6	17	8667	14701	35
13	9579,0	16	8874	15122	32
14	10822,7	16	9596	16362	36

**Tablo 7.13. (Devamı)** Senaryo 2'nin ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
15	8747,9	16	8415	14215	33
16	9008,2	17	8630	14724	35
17	7894,6	16	7921	13294	34
18	8434,6	16	8284	13928	31
19	8796,9	17	8506	14499	35
20	9195,6	17	8876	15004	31

Senaryo 3 için elde edilen Pareto-optimal çözümler ise Tablo 7.14'te verilmiştir.

**Tablo 7.14.** Senaryo 3'ün ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
1	11192,0	16	9197	15470	38
2	9899,3	16	8308	13990	32
3	9524,4	16	8027	13828	33
4	9786,0	16	8137	13764	34
5	10338,6	16	8288	14241	35
6	9678,3	16	8340	13992	32
7	8636,0	16	7654	12971	30
8	9890,9	16	8697	14582	32
9	11340,3	16	9004	15386	36
10	10751,3	16	8802	14957	31
11	8833,3	16	7962	13441	31
12	11159,8	16	9198	15603	33
13	11813,2	16	9596	16362	36
14	10561,1	16	8874	15122	32
15	9558,0	16	8415	14215	33
16	10608,6	17	8630	14724	35
17	9115,9	16	7921	13294	34
18	9609,7	16	8284	13928	31
19	9725,8	16	8601	14361	30
20	10880,0	16	8524	14544	37



### 7.3. Senaryoların Uyarlanmış ÇAPSO ile Çözümü

Bu bölümde Senaryo 1, Senaryo 2 ve Senaryo 3 için uyarlanmış ÇAPSO ile çözülmüş ve sonuçları verilmiştir.

ÇAPSO'da başlangıç çözüm rastgele oluşturulmaktadır. Buda çözüm kalitesini olumsuz etkilemektedir. Uyarlanmış ÇAPSO'da ise başlangıç çözümü geliştirilen hiyerarşik yöntemin ilk aşaması ile belirlenmekte ve daha sonra ÇAPSO ile devam edilip nihai çözüm bulunmaktadır. Pareto-optimal küme içerisinde uygun çözümün seçimi ise Bölüm 7.2'de açıklandığı şekilde yapılmıştır.

Hiyerarşik yöntem için parametre ayarlaması Bölüm 6'da, ÇAPSO'da kullanılan parametrelerin ayarlaması ise Bölüm 7.2'de verildiği gibi yapılmıştır.

20 haftalık talep verileri kullanılarak Senaryo 1, Senaryo 2 ve Senaryo 3 için uyarlanmış ÇAPSO 50 kez oluşturulmuş Pareto-optimal çözümler elde edilmiştir. Senaryo 1 için elde edilen Pareto-optimal çözümler Tablo 7.15'te verilmiştir.

**Tablo 7.15.** Senaryo 1'in uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
1	6988,0	16	5977	10057,4	18
2	6443,8	16	5995	10143	17
3	7644,1	16	5414	9298,8	21
4	6856,5	16	5915	9839,6	17
5	7466,5	16	6066	10290	18
6	7332,2	16	6298	10776	18
7	6695,6	16	5842	10062	17
8	7290,2	16	5794	9635,6	20
9	7237,0	17	6567	11019	19
10	7542,0	16	5989	10020	20
11	7605,0	16	5866	9967,9	21
12	7070,5	17	6344	10481	18
13	7437,1	16	6517	10929	21
14	6753,0	16	6042	10153	16
15	7319,6	17	6430	10707	19
16	7287,4	16	5650	9611,4	20

**Tablo 7.15. (Devamı)** Senaryo 1'in uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
17	6684,4	17	6438	10741,5	16
18	7553,2	16	5818	9959,6	20
19	7314,0	16	5854	9957,9	18
20	7139,1	16	6055	10121,1	18

Senaryo 2 için elde edilen Pareto-optimal çözümler Tablo 7.16'da verilmiştir.

**Tablo 7.16.** Senaryo 2'nin uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
1	4787,4	16	5989	10171,7	19
2	4244,6	16	5803	9532,2	18
3	4549,5	16	5866	10019	20
4	5144,1	17	6523	10821	18
5	4486,6	16	5773	9730	20
6	4962,3	16	6126	10358	23
7	4464,2	16	5781	9878,1	19
8	5061,6	17	6508	10899	20
9	4594,3	16	5967	10050	19
10	4812,6	16	6080	10300	19
11	4195,6	16	5609	9553,5	16
12	4657,3	16	5876	10034	22
13	4669,9	16	5908	10180	17
14	4885,3	17	6328	10704	20
15	3981,6	16	5537	9285,2	22
16	5086,8	16	6298	10586	18
17	4253,0	16	5740	9711,8	18
18	4662,9	16	6001	10119	21
19	4304,7	16	5568	9503	19
20	4184,4	16	5583	9570,5	23

Senaryo 3 için elde edilen Pareto-optimal çözümler ise Tablo 7.17'de verilmiştir.

**Tablo 7.17.** Senaryo 3'ün uyarlanmış ÇAPSO ile çözümünden elde edilen sonuçlar.

HAFTA	TŞM	TAS	TSM	TET	TŞİS
1	5229,5	16	6075	10251,4	19
2	5685,5	16	6203	10512	19
3	5915,0	16	6341	10800	17
4	5615,6	16	6131	10223	20
5	5222,5	16	6042	10285	21
6	5383,4	16	6310	10682	21
7	5920,6	16	6334	10693	19
8	5051,8	16	5919	10017	18
9	6054,9	17	6525	11044	19
10	5647,8	16	6031	10315	19
11	6358,5	16	6532	11139	19
12	4585,9	16	5747	9654,7	19
13	5526,1	16	6000	10077	20
14	5784,9	16	6311	10817	18
15	5155,3	16	5799	9893,2	20
16	5240,7	16	5879	10019	18
17	6060,5	16	6549	10993	19
18	5173,5	16	6233	10519	18
19	5095,2	16	5840	9940,5	22
20	5018,2	16	6077	10426	17

#### 7.4. Parametrik Analizler

Bu bölümde ilk olarak hiyerarşik yöntem, ÇAPSO ve uyarlanmış ÇAPSO ile senaryoların çözümünden elde edilen sonuçlar arasında istatistiksel olarak anlamlı bir fark olup olmadığı araştırılmış ve daha sonra üç farklı senaryonun birbirlerine olan üstünlükleri test edilmiştir.

##### 7.4.1. Yöntemlerin elde ettiği sonuçlar açısından karşılaştırılması

Senaryo 1, 2 ve 3'ün hiyerarşik yöntem (ATY-HY ve KSY-HY), ÇAPSO ve uyarlanmış ÇAPSO ile çözümünden elde edilen her beş amaç fonksiyonunun sonuçları normalize edilerek toplanmış ve bu sonuçlara ANOVA testi uygulanmıştır. SPSS sonuçları aşağıda verilmiştir. Bu çalışmada;

$H_0$  : ATY-HY, KSY-HY, ÇAPSO ve uyarlanmış ÇAPSO'nun performansları arasında %95 güven aralığı ile önemli bir fark yoktur.

$H_a$  : ATY-HY, KSY-HY, ÇAPSO ve uyarlanmış ÇAPSO'nun performansları arasında %95 güven aralığı ile önemli bir fark vardır.

olacak şekilde hipotezler tanımlanmıştır.

ANOVA testi uygulanmak istendiğinde ilk olarak varyansların homojen olup olmadığı incelenmesi sağlıklı olacaktır. Varyansların homojenliğini test etmek için ise Levene testi kullanılmıştır. Tablo 7.18, Levene testinin sonuçlarını göstermektedir.

**Tablo 7.18.** Dört yöntem için varyansların homojenliği testi sonuçları.

	Levene İstatistiği	df1	df2	p
<b>Senaryo 1</b>	13,273	3	76	,000
<b>Senaryo 2</b>	16,755	3	76	,000
<b>Senaryo 3</b>	9,912	3	76	,000

Tablo 18'deki p (sig.) değerleri  $0 < 0,05$  olduğundan dolayı grup varyanslarının homojen olmadığı sonucuna varılır. Grup varyansları homojen olmadığı için klasik ANOVA yerine ANOVA Welch testi uygulanmıştır. Welch testinin sonuçları Tablo 7.19'da sunulmuştur.

**Tablo 7.19.** Dört yöntem için ANOVA Welch testi sonuçları.

	İstatistik	df1	df2	p
<b>Senaryo 1</b>	247,035	3	38,396	,000
<b>Senaryo 2</b>	178,275	3	38,236	,000
<b>Senaryo 3</b>	274,849	3	38,491	,000

Tablo 7.19'daki p değerleri ( $p \leq 0,05$ )  $H_0$  hipotezinin reddedilip  $H_1$  hipotezi kabul edileceğini göstermektedir. Yani, her üç senaryo için de ATY-HY, KSY-HY, ÇAPSO ve uyarlanmış ÇAPSO'nun ürettikleri sonuçlar arasında %95 güven aralığı ile önemli bir fark vardır diyebiliriz. Daha sonra, dört yöntem kullanılarak elde

edilen sonuçlar arasındaki farklılıkları analiz etmek için ise Tamhane posthoc testleri yapılmış ve Tablo 7.20’de sonuçları gösterilmiştir.

**Tablo 7.20.** Dört yöntem için Tamhane post-hoc testi sonuçları.

Metot (I)	Metot (J)	Ortalama Fark (I-J)	Standart Hata	p
<b>Senaryo 1</b>				
ATY-HY	KSY-HY	,11369*	,02798	,001
	ÇAPSO	-2,96303*	,11365	,000
	Uyarlanmış ÇAPSO	-,68296*	,11318	,000
KSY-HY	ATY-HY	-,11369*	,02798	,001
	ÇAPSO	-3,07672*	,11395	,000
	Uyarlanmış ÇAPSO	-,79665*	,11348	,000
ÇAPSO	ATY-HY	2,96303*	,11365	,000
	KSY-HY	3,07672*	,11395	,000
	Uyarlanmış ÇAPSO	2,28007*	,15815	,000
Uyarlanmış ÇAPSO	ATY-HY	,68296*	,11318	,000
	KSY-HY	,79665*	,11348	,000
	ÇAPSO	-2,28007*	,15815	,000
<b>Senaryo 2</b>				
ATY-HY	KSY-HY	,06719*	,02412	,049
	ÇAPSO	-2,99070*	,13329	,000
	Uyarlanmış ÇAPSO	-,59290*	,11304	,000
KSY-HY	ATY-HY	-,06719*	,02412	,049
	ÇAPSO	-3,05789*	,13348	,000
	Uyarlanmış ÇAPSO	-,66009*	,11326	,000
ÇAPSO	ATY-HY	2,99070*	,13329	,000
	KSY-HY	3,05789*	,13348	,000
	Uyarlanmış ÇAPSO	2,39779*	,17324	,000
Uyarlanmış ÇAPSO	ATY-HY	,59290*	,11304	,000
	KSY-HY	,66009*	,11326	,000
	ÇAPSO	-2,39779*	,17324	,000
<b>Senaryo 3</b>				
ATY-HY	KSY-HY	,08767*	,03094	,044
	ÇAPSO	-2,82474*	,10583	,000
	Uyarlanmış ÇAPSO	-,63763*	,07366	,000
KSY-HY	ATY-HY	-,08767*	,03094	,044
	ÇAPSO	-2,91242*	,10470	,000
	Uyarlanmış ÇAPSO	-,72531*	,07202	,000

**Tablo 7.20. (Devamı)** Dört yöntem için Tamhane post-hoc testi sonuçları.

Metot (I)	Metot (J)	Ortalama Fark (I-J)	Standart Hata	p
<b>Senaryo 3</b>				
ÇAPSO	ATY-HY	2,82474*	,10583	,000
	KSY-HY	2,91242*	,10470	,000
	Uyarlanmış ÇAPSO	2,18711*	,12422	,000
Uyarlanmış ÇAPSO	ATY-HY	,63763*	,07366	,000
	KSY-HY	,72531*	,07202	,000
	ÇAPSO	-2,18711*	,12422	,000

Tamhane post-hoc testleri sonucunda her üç senaryo için de KSY-HY'nin performansının diğer üç yöntemden daha iyi olduğu, ÇAPSO'nun ise en düşük performansı ürettiği görülmüştür.

#### 7.4.2. Senaryolardan elde edilen sonuçların karşılaştırılması

Bir önceki bölümde KSY-HY in diğer yöntemlere göre her üç senaryoda da üstünlüğü ANOVA testi sonucunda belirlenmiştir. Bu alt bölümde ise üç senaryonun çözümünden elde edilen beş amaç fonksiyonu değerlerine göre üstünlükleri test edilecektir. Bu çalışmada;

$H_0$  : Senaryo 1, Senaryo 2 ve Senaryo 3'ün çözüm sonuçları arasında %95 güven aralığı ile anlamlı bir fark yoktur.

$H_a$  : Senaryo 1, Senaryo 2 ve Senaryo 3'ün çözüm sonuçları arasında %95 güven aralığı ile anlamlı bir fark vardır.

Olacak şekilde hipotezler tanımlanmıştır.

ANOVA testi uygulandığında ilk olarak varyansların homojen olup olmadığı incelenmektedir. Varyansların homojenliğini test etmek için ise Levene testini kullanılmıştır. Tablo 7.21, Levene testinin sonuçlarını göstermektedir.

**Tablo 7.21.** Senaryolar için varyansların homojenliği testi sonucu.

Levene İstatistiği	df1	df2	p
,409	2	57	,666

Tablo 7.21'deki p (sig.) değerleri  $0,666 > 0,05$  olduğundan dolayı grup varyanslarının homojen olduğu sonucuna varılır. ANOVA testinin sonuçları Tablo 7.22'de sunulmuştur.

**Tablo 7.22.** Senaryolar için ANOVA testi sonucu.

	Kareler Toplamı	Serbestlik Derecesi	Kareler Ortalaması	F	p
<b>Gruplar arası</b>	12,431	2	6,216	32,204	,000
<b>Grup içi</b>	11,002	57	,193		
<b>Toplam</b>	23,433	59			

Tablo 7.22'deki P değerleri ( $p \leq 0.05$ )  $H_0$  hipotezinin reddedilip  $H_1$  hipotezi kabul edileceğini göstermektedir. Senaryo 1, Senaryo 2 ve Senaryo 3'ün çözüm sonuçları arasında %95 güven aralığı ile anlamlı bir fark vardır diyebiliriz. Senaryo 1, Senaryo 2 ve Senaryo 3'ün çözüm sonuçları arasındaki farkları analiz etmek için Tamhane posthoc testi ile ikili karşılaştırmalar yapılmıştır. Tablo 7.23, Tamhane'nin posthoc testlerinin sonuçları gösterilmiştir.

**Tablo 7.23.** Senaryolar için Tamhane post-hoc testi sonucu.

Metot (I)	Metot (J)	Ortalama Fark (I-J)	Standart Hata	p
Senaryo 1	Senaryo 2	,87424*	,13893	,000
	Senaryo 3	1,03640*	,13893	,000
Senaryo 2	Senaryo 1	-,87424*	,13893	,000
	Senaryo 3	,16216	,13893	,510
Senaryo 3	Senaryo 1	-1,03640*	,13893	,000
	Senaryo 2	-,16216	,13893	,510

Tablo 7.23'te görüldüğü gibi Senaryo 1, Senaryo 2 ve 3'e göre yönlendirme kararlarını olumlu yönde etkilemeyecektir. Elektrikli araç rotalama probleminde Senaryo 2 ve 3'teki kısmi şarj politikasının ve çoklu şarj teknolojilerinin kullanılması toplam şarj maliyeti, gerekli araç sayısı, toplam seyahat mesafesi, toplam enerji tüketimi ve uğranması gereken toplam şarj istasyonu sayısı gibi gerçek hayattaki hedefleri en düşükleyerek uzun vadede işletmelere tavsiye edebiliriz.





## 8. SONUÇ VE DEĞERLENDİRME

Bu tez çalışmasında, ürünlerin müşterilere dağıtım işleminin EA'lar ile yapıldığı ARP'nin güncel bir çeşidi olan EARP ele alınmıştır. Bu kapsamda, EARP'nin çok amaçlı varyantı gerçekçi kısıtlar altında üç farklı senaryo halinde sunulmuştur. Üç farklı senaryo halinde sunulan ÇAEARP, EA'ların şarj politikaları (tam veya kısmi) ve şarj teknolojileri (yavaş, orta veya hızlı) açısından farklılık göstermektedir. ÇAEARP'nin birbiriyle çelişen beş adet amaç fonksiyonu mevcuttur. Bunlar; toplam şarj etme maliyetini en küçükmek, gerekli olan EA sayısını en küçükmek, toplam seyahat mesafesini en küçükmek, enerji tüketimini en küçükmek ve uğranması gereken şarj istasyonu sayısını en küçükmektir.

ÇAEARP'nin çözümü için iki aşamadan oluşan hiyerarşik bir yöntem (HKKO-YAKA) önerilmiştir. Geliştirilen yöntemin ilk aşamasında arama uzayını genişletmek için yerel arama yöntemleriyle desteklenmiş ve kabul kriteri olarak tavlama benzetimi kullanılarak hibritleştirilen karınca kolonisi optimizasyonu kullanılmıştır. İkinci aşamada ise birinci aşamadan elde edilen çözüm YAKA için başlangıç çözümü olarak ele alınıp problemin çözümü tamamlanmıştır. YAKA'da başlangıç çözümü rastgele üretildiği için problemin çözüm süresini olumsuz yönde etkileyebilmektedir. Fakat başlangıç çözümünün HKKO ile üretilmesinin daha kısa sürede daha iyi sonuçlar elde ettiği yaptığımız analizler sonucunda tespit edilmiştir.

Bu tezde geliştirilen HKKO-YAKA'nın performansının değerlendirilmesi için Felipe ve ark. (2014) tarafından geliştirilen 60 adet veri setinden oluşan test problemleri kullanılmıştır. Elde edilen sonuçlar Felipe ve ark. (2014)'nin önerdiği TB tabanlı bir algoritma ve Keskin ve Çatay (2018)'in önerdiği mat-sezgisel yöntem ile karşılaştırılmıştır. Karşılaştırmalar toplam şarj etme maliyetleri açısından yapılmıştır. Sonuç olarak tezde geliştirilen HKKO-YAKA'nın 60 problem içerisinde Felipe ve ark. (2014)'ün elde ettiği çözümlerin hepsinden, Keskin ve Çatay (2018)'in elde ettiği çözümlerden ise 47 tanesinden daha iyi sonuç elde ettiği görülmüştür.

Ayrıca, geliştirilen HKKO-YAKA ile bir gerçek hayat ÇAEARP olan süpermarket zincirinin haftalık taleplerinin karşılanabilmesi için en uygun araç rotalarının

belirlenmesi problemi çözülmüştür. HKKO-YAKA, Pareto-optimal çözümleri bulmak amacıyla ATY ve KSY için arama işlemi olarak kullanılmıştır. İlk olarak beş adet amaç fonksiyonu ATY ve KSY ile ayrı ayrı skalerleştirilmiş daha sonra ise Pareto-optimal çözümler HKKO-YAKA ile bulunmuştur. Geliştirilen HKKO-YAKA'nın gerçek hayat problemi üzerindeki performansını değerlendirmek için çok amaçlı evrimsel algoritmalarından olan ÇAPSO ve uyarlanmış ÇAPSO kullanılmıştır. ÇAPSO'da başlangıç çözüm rastgele oluşturulmaktadır. Buda çözüm kalitesini olumsuz etkilemektedir. Uyarlanmış ÇAPSO'da ise başlangıç çözümü geliştirilen hiyerarşik yöntem ile belirlenmekte ve daha sonra çözüme ÇAPSO ile devam edilip nihai çözüm bulunmaktadır. Elde edilen sonuçlar geliştirilen hiyerarşik yöntem ile bulunan sonuçların üç senaryo içinde ÇAPSO ve uyarlanmış ÇAPSO'dan daha iyi sonuçlar ürettiğini göstermiştir. Hiyerarşik yöntemi de kendi içinde değerlendirecek olursak KSY kullanıldığında ATY'ne göre daha iyi performans elde edilmiştir. Yöntemlerin karşılaştırılmasının dışında aynı zamanda ÇAEARP'nin kısmi şarj politikası ve çoklu şarj teknolojisi seçeneklerinin yönlendirme kararlarını önemli ölçüde iyileştirebileceği sonucuna varılabilmektedir.

Ek olarak, bu çalışmanın çevre ve sürdürülebilir kalkınma için önemli pratik çıkarımları bulunmaktadır. Hükümetler, işletmeler ve lojistik şirketleri, çevresel bozulmayı ve enerji kısıtlıklarını yönetmede önemli zorluklarla karşı karşıyadır. Şu anda Türkiye'nin elektrikli araç pazarı gelişmekte olan bir pazardır. Son yıllarda özel tüketim vergisi ve motorlu taşıt vergileri ile ilgili çeşitli teşvikler uygulanmıştır (ICCT, 2019). Ayrıca yerli elektrikli araçlar üretilmekte olup, 2023 yılında tamamen piyasaya sürülmesi hedeflenmektedir. Shura Enerji Dönüşüm Merkezi raporuna göre, 2030 yılında bataryalı ve plug-in hibrit elektrikli araç satışlarının toplam binek araç satışlarının %55'ine ulaşacağı öngörülmektedir. Elektrikli araçlar ise toplam araç stoğunun %10'unu temsil edecektir. Bu sayede Türkiye'de kullanılan toplam elektrikli araç sayısının 2030 yılına kadar 2,5 milyona çıkacağı tahmin edilmektedir (Shura, 2019). Tüm bu durumlar Türkiye'deki elektrikli araç kullanımının giderek artmakta olduğunu göstermektedir. Bu nedenle, bu çalışma ile lojistik sektöründe faaliyet gösteren firmalara elektrikli araçların kullanımına yönelik öncü bir çalışma olması hedeflenmiştir.

## KAYNAKLAR

- ABB. (2021). Electric vehicle charging infrastructure. <https://new.abb.com/ev-charging/tr/events/ea-sarj-dokumanlari/elektrikli-arac-sarj-altyapisi>. adresinden 21 Haziran 2022 tarihinde alınmıştır.
- Aharon, B. T. (1980). Characterization of pareto and lexicographic optimal solutions. In G. Fandel & T. Gal (Eds.), *Multiple criteria decision marketing theory and application* (pp. 1-11). Springer-Verlag.
- Akay, B. (2009). *Nümerik optimizasyon problemlerinde yapay arı kolonisi (Artificial Bee Colony) algoritmasının performans analizi* [Doktora tezi]. Erciyes Üniversitesi.
- Branke, J., Deb, K., Miettinen, K., & Slowinski, R. (Eds.). (2008). *Multiobjective optimization: Interactive and evolutionary approaches*. Springer.
- Burke, E. K., & Kendall G. (Eds.). (2014). *Search methodologies: Introductory tutorials in optimization and decision support techniques*. Springer.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568–581. <https://doi.org/10.1287/opre.12.4.568>
- Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3), 129–156. <https://doi.org/10.1007/BF03325101>
- Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 256-279. <https://doi.org/10.1109/TEVC.2004.826067>
- Conrad, R.G., & Figliozzi, M.A. (2011). The recharging vehicle routing problem. In T. Doolen & E. Van Aken (Eds.), *Proceedings of the 2011 Industrial Engineering Research Conference*.
- Cortés-Murcia, D. L., Prodhon, C., & Murat Afsar, H. (2019). The electric vehicle routing problem with time windows, partial recharges and satellite customers. *Transportation Research Part E: Logistics and Transportation Review*, 130, 184–206. <https://doi.org/10.1016/j.tre.2019.08.015>
- Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6(6), 791-812. <https://doi.org/10.1287/OPRE.6.6.791>
- Dantzig, G. B., ve Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80-91. <https://doi.org/10.1287/mnsc.6.1.80>
- Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6), 1388–1405. <https://doi.org/10.1287/opre.2016.1535>

- Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2), 342-354. <https://doi.org/10.1287/opre.40.2.342>
- Dinger, A., Martin, R., Mosquet, X., Rabl, M., Rizoulis, D., Russo, M., & Sticher, G. (2010). *Batteries for electric cars: challenges, opportunities, and the outlook to 2020*. Boston Consulting Group.
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1), 29-41. <https://doi.org/10.1109/3477.484436>
- Dorigo, M., Caro, G. D., & Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, 5(2), 137-172. <https://doi.org/10.1162/106454699568728>
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3), 243-278. <https://doi.org/10.1016/j.tcs.2005.05.020>
- Dorigo, M., Maniezzo, V., & Colorni, A. (1991). *Ant System: An Autocatalytic Optimizing Process*. Technical Report TR91-016, Politecnico di Milano.
- Dorigo, M., & Birattari, M. (2011). Ant colony optimization. In *Encyclopedia of machine learning* (pp. 36-39). Springer, Boston, MA.
- Ehrgott, M. (2005). *Multicriteria optimization* (2th ed.). Springer.
- Erdogan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48, 100–114. <https://doi.org/10.1016/j.tre.2011.08.001>
- Faryad Aghjeh Kand, V. (2014). *Çok amaçlı NSGA-II ve MOPSO optimizasyon algoritmaları ile kablosuz algılayıcı ağlarında optimum küme başı yeri seçimi ve kümelemesi* [Yüksek lisans tezi] Karadeniz Teknik Üniversitesi.
- FedEx, (2018). Environmental sustainability. <http://www.fedex.com/bt/about/sustainability/>. adresinden 20 Haziran 2022 tarihinde alınmıştır.
- Felipe, Á., Ortuño, M.T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71, 111–128. <https://doi.org/10.1016/j.tre.2014.09.003>
- Froger, A., Mendoza, J.E., Jabali, O., & Laporte, G. (2017). A matheuristic for the electric vehicle routing problem with capacitated charging stations. CIRRELT-Technical Report 2017-3.
- Froger, A., Mendoza, J.E., Jabali, O., & Laporte, G. (2019). Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Computers & Operations Research*, 104, 256–294. <https://doi.org/10.1016/j.cor.2018.12.013>

- Ganji, M., Kazemipoor, H., Molana, S. M. H., Sajadi, S. M. (2020). A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. *Journal of Cleaner Production*, 259, 120824. <https://doi.org/10.1016/j.jclepro.2020.120824>
- Gasimov, R. N. (2001). Characterization of the Benson proper efficiency and scalarization in nonconvex vector optimization. In M. Köksalan & S. Zionts (Eds.), *Multiple criteria decision making in the new millennium* (pp. 189-198). Springer.
- Gass, S., & Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2, 39–45. <https://doi.org/10.1002/nav.3800020106>
- Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1), 81–99. <https://doi.org/10.1016/j.ejor.2015.01.049>
- Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). *The vehicle routing problem: latest advances and new challenges*. Springer.
- Hwang, C. L., & Masud, A. S. M. (1979). *Multiple objective decision making-methods and applications: A state-of-the-art survey*. Springer.
- Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R.F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3), 995–1018. <http://dx.doi.org/10.1016/j.ejor.2016.01.038>
- Hiermann, G., Hartl, R. F., Puchinger, J., & Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1), 235–248. <https://doi.org/10.1016/J.EJOR.2018.06.025>
- Hoos, H. H. (2012). Automated algorithm configuration and parameter tuning. In Y. Hamadi, E. Monfroy, & F. Saubion (Eds.), *Autonomous Search* (pp. 37-71). Heidelberg, Germany: Springer.
- ICCT. (2019). Vehicle registration tax as a policy instrument to help reduce CO2 emissions and fuel consumption in Turkey. <https://theicct.org/publications/turkey-vehicle-tax-emissions-tk> adresinden 21 Haziran 2022 tarihinde alınmıştır.
- IEA (2019). Transport sector CO2 emission by mode in the sustainable development scenario, 2000-2030. <https://www.iea.org/data-and-statistics/charts/transport-sector-co2-emissions-by-mode-in-the-sustainable-development-scenario-2000-2030>. adresinden 20 Aralık 2022 tarihinde alınmıştır.
- Jabir, E., Panicker, V.V., & Sridharan, R. (2017). Design and development of a hybrid artificial-variable neighbourhood search algorithm for a multi-depot green vehicle routing problem. *Transportation Research Part D: Transport and Environment*, 57, 422–457. <https://doi.org/10.1016/j.trd.2017.09.003>
- Jemai, J., Zekri, M., & Mellouli, K. (2012). An NSGA-II algorithm for the green vehicle routing problem. *Proceedings of the 12th European conference on Evolutionary Computation in Combinatorial Optimization*, Málaga, Spain, 37-48. [https://doi.org/10.1007/978-3-642-29124-1\\_4](https://doi.org/10.1007/978-3-642-29124-1_4)

- Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (1998). How easy is local search?. *Journal of Computer and System Sciences*, 37(1), 79-100. [https://doi.org/10.1016/0022-0000\(88\)90046-3](https://doi.org/10.1016/0022-0000(88)90046-3)
- Kancharla, S.R., Ramadurai, G. (2020). Electric vehicle routing problem with non-linear charging and load-dependent discharging. *Expert Systems with Applications*, 160, 113714. <https://doi.org/10.1016/j.eswa.2020.113714>
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1), 687–697. <https://doi.org/10.1016/j.asoc.2007.05.007>
- Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1), 108-132. <https://doi.org/10.1016/j.amc.2009.03.090>
- Karakatič, S. (2021). Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. *Expert Systems with Applications*, 164, 114039. <https://doi.org/10.1016/j.eswa.2020.114039>
- Kendall, G. (2000), AI Methods, <http://www.cs.nott.ac.uk/~pszgk/aim/> adresinden 20 Aralık 2022 tarihinde alınmıştır.
- Kennedy, J., ve Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Australia, 4, 1942-1948. <http://dx.doi.org/10.1109/ICNN.1995.488968>
- Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65, 111–127. <https://doi.org/10.1016/j.trc.2016.01.013>
- Keskin, M., & Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers & Operations Research*, 100, 172-188. <https://doi.org/10.1016/j.cor.2018.06.019>
- Keskin, M., Laporte, G., & Çatay, B. (2019). Electric vehicle routing problem with time dependent waiting times at recharging stations. *Computers & Operations Research*, 107, 77–94. <https://doi.org/10.1016/j.cor.2019.02.014>
- Keskin, M., Çatay, B., & Laporte, G. (2021). A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. *Computers and Operations Research*, 125, 105060. <https://doi.org/10.1016/j.cor.2020.105060>
- Kirkpatrick, S., Gelatt C.D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680. [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671)
- Knowles, J. D., & Corne, D. (2000). Approximating the non-dominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8, 149–172. <https://doi.org/10.1162/106365600568167>

- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3), 345-358. [https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C)
- Lee, D.-Y., Thomas, V. M., & Brown, M. A. (2013). Electric urban delivery trucks: Energy use, greenhouse gas emissions, and costeffectiveness. *Environmental Science & Technology*, 47(14), 8022-8030. <https://doi.org/10.1021/es400179w>
- Lenstra, J. K., & Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2), 221–227. <https://doi.org/10.1002/net.3230110211>
- Li, Y., Soleimani, H., & Zohal M. (2019). An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *Journal of Cleaner Production*, 227, 1161-1172. <https://doi.org/10.1016/j.jclepro.2019.03.185>
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10), 2245-2269. <https://doi.org/10.1002/j.1538-7305.1965.tb04146.x>
- Lu, J., Chen, Y., Hao, J.-K., & Renjie, H. (2020). The time-dependent electric vehicle routing problem: model and solution. *Expert Systems with Applications*, 161, 113593. <https://doi.org/10.1016/j.eswa.2020.113593>
- Melián-Batista, B., De Santiago, A., AngelBello, F. & Alvarez, A. (2014). A bi-objective vehicle routing problem with time windows: A real case in tenerife. *Applied Soft Computing*, 17, 140-153. <https://doi.org/10.1016/j.asoc.2013.12.012>
- Miettinen, K. M. (1999). *Nonlinear multiobjective optimization*. Kluwer, Boston.
- Molina, J.C., Eguia, I., Racero, J., & Guerrero, F. (2014). Multi-objective vehicle routing problem with cost and emission functions. *Procedia-Social and Behavioral Sciences*, 160, 254-263. <https://doi.org/10.1016/j.sbspro.2014.12.137>
- Montoya, A., Guéret, C., Mendoza, J.E., & Villegas, J.G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103, 87–110. <https://doi.org/10.1016/j.trb.2017.02.004>
- Muller, S. (2022). *How do we get the adoption of electric vehicles into a higher gear in europe?* [Yüksek lisans tezi] Delft University of Technology.
- Norouzi, N., Sadegh-Amalnick, M., & Tavakkoli-Moghaddam, R. (2017). Modified particle swarm optimization in a time-dependent vehicle routing problem: minimizing fuel consumption. *Optimization Letters*, 11, 121–134. <https://doi.org/10.1007/s11590-015-0996-y>
- Palmer, A. (2007). *The development of an integrated routing and carbon dioxide emissions model for goods vehicles* [Doktora tezi]. Canfield University.
- Pelletier, S., Jabali, O., & Laporte, G. (2016). Goods distribution with electric vehicles: Review and research perspectives. *Transportation Science*, 50(1), 3-22. <https://doi.org/10.1287/trsc.2015.0646>

- Peng, G. (2021). *Evolutionary Multi-Objective Optimization for Computation Offloading in Collaborative Edge-Cloud Computing* [Doktora Tezi]. Freien Universität Berlin.
- Poonthalir, G., & Nadarajan, R. (2018). A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP). *Expert Systems with Applications*, 100, 131–144. <https://doi.org/10.1016/j.eswa.2018.01.052>
- Qu, B. Y., Zhu, Y. S., Jiao, Y. C., Wu, M. Y., Suganthan, P. N., & Liang, J. J. (2018). A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. *Swarm and Evolutionary Computation*, 38, 1–11. <https://doi.org/10.1016/j.swevo.2017.06.002>
- Rao, S. S. (1991). *Optimization theory and application*. New Delhi: Wiley Eastern Limited.
- Savelsberg, M. W. P. (1992). The vehicle routing problem with time windows: minimizing route duration. *ORSA Journal on Computing*, 4(2), 146-154. <https://doi.org/10.1287/ijoc.4.2.146>
- Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle routing problem with time windows and recharging stations. *Transportation Science*, 48(4), 500–520. <https://doi.org/10.1287/trsc.2013.0490>
- Shi, Y., ve Eberhart, R. (1999). Empirical study of particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation, USA*, 3, 1945-1950. <https://doi.org/10.1109/CEC.1999.785511>
- Shibi, A., ve Eglese, R. W. (2010). Combinatorial optimization and green logistics. *Annals of Operations Research*, 175, 159–175. <https://doi.org/10.1007/s10479-009-0651-z>
- Shura. (2019). Transport sector transformation: Integrating electric vehicles into turkey's distribution grids. <https://www.shura.org.tr/wp-content/uploads/2020/10/Transport-sector-transformation-Integrating-electric-vehicles-into-Turkeys-distribution-grids.pdf> adresinden 21 Haziran 2022 tarihinde alınmıştır.
- Sipahioğlu, A., & Saraç, T. (2010). Çok amaçlı sırt çantası probleminin çözümüne yeni bir yaklaşım: Konik skalerleştirme. *Endüstri Mühendisliği Dergisi*, 21 (4), 2-12.
- Siu W. S. H., Chan, C.-K., Chan, H. C. B. (2012). Green cargo routing using genetic algorithms. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, 1, 170-175.
- Şirvancı, M. (1997). *Kalite için deney tasarımı Taguchi Yaklaşımı*. Literatür yayınları. İstanbul.
- Toth, P., & Vigo, D. (Eds.). (2001). *The vehicle routing problem*. Siam.
- Üstün, Ö. (2007). *Çok amaçlı portföy optimizasyon problemi ve çözüm yaklaşımları* [Doktora tezi]. Eskişehir Osmangazi Üniversitesi.
- Woodruff, D. L. (Eds.). (1998). *Advances in computational and stochastic optimization, logic programming and heuristic search*. Springer.



- Yang, W. H., & Tarng Y. S. (1998). Design optimization of cutting parameters for turning operations based on the Taguchi method. *Journal of Materials Processing Technology*, 84, 122- 129. [https://doi.org/10.1016/S0924-0136\(98\)00079-X](https://doi.org/10.1016/S0924-0136(98)00079-X)
- Zhang, S., Gajpal, Y., & Appadoo, S. S. (2018a). A meta-heuristic for capacitated green vehicle routing problem. *Annals of Operations Research*, 269 (1), 753–771. <https://doi.org/10.1007/s10479-017-2567-3>
- Zhang, S., Gajpal, Y., Appadoo, S. S., & Abdulkader, M. M. S. (2018b). Electric vehicle routing problem with recharging stations for minimizing energy consumption. *International Journal of Production Economics*, 203, 404–413. <https://doi.org/10.1016/J.IJPE.2018.07.016>
- Zhang, S., Chen, M., Zhang, W., & Zhuang, X. (2020). Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations. *Expert Systems with Applications*, 145, 113-123. <https://doi.org/10.1016/j.eswa.2019.113123>.
- Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32-49. <https://doi.org/10.1016/j.swevo.2011.03.001>
- Zuo, X., Xiao, Y., You, M., Kaku, I., & Xu, Y. (2019). A new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function. *Journal of Cleaner Production*, 236, 117687. <https://doi.org/10.1016/J.JCLEPRO.2019.117687>



## EKLER

### EK A. Yerel Aramanın Algoritmaya Etkisinin Analizi

Yerel aramanın etkisini göstermek için FORT örnekleri kullanılmıştır. Geliştirilen hiyerarşik yöntemin ilk aşaması olan KKO'da yerel arama operatörleri kullanılmadı. Bununla birlikte, ikinci aşamada (YAKA), üç aşamanın her birinde (işçi arı, gözcü arı ve kaşif arı) üç yerel arama operatöründen (ters çevirme, yer değiştirme ve ekleme) yalnızca biri kullanıldı ve sonuçları (çözüm süresi ve toplam maliyet açısından) karşılaştırıldı. Ayrıntılı sonuçlar Tablo A.1'de verilmektedir. Sonuçlar, HKKO ve KKO'nun 10 iterasyon, YAKA, YAKA1, YAKA2 ve YAKA3'ün ise 10000 iterasyon koşturulması sonucunda elde edilmiştir. “% Ortalama fark” geliştirdiğimiz yöntemin (HKKO-YAKA) KKO-YAKA1, KKO-YAKA2 ve KKO-YAKA3'ün sonuçları arasındaki hesaplama süresi ve toplam maliyet açısından farkın ortalamasını yüzde olarak göstermektedir.

**Tablo A.1.** “Yerel aramalı” ve “yerel arama olmadan” elde edilen sonuçların karşılaştırılması

Şarj istasyonu sayısı	Müşteri Sayısı	HKKO-YAKA		KKO-YAKA1		KKO-YAKA2		KKO-YAKA3	
		Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi
9	100	60,89	275	68,57	254	69,18	274	69,35	256
		60,82	251	68,51	230	68,6	235	68,03	232
		61,84	259	69,62	238	69,24	238	69,22	240
		64,01	269	70,68	245	71,78	249	71,55	247
		72,21	268	85,34	244	86,89	267	80,33	246
		65,16	249	74,40	232	74,96	230	72,78	234
		65,25	249	73,50	229	74,06	227	72,87	231
		56,43	274	63,46	256	64,19	273	63,43	258
		61,22	264	67,96	242	68,54	243	68,56	244
		57,73	247	65,04	226	64,53	228	64,82	228

**Tablo A.1. (Devamı)** “Yerel aramalı” ve “yerel arama olmadan” elde edilen sonuçların karşılaştırılması

Şarj istasyonu sayısı	Müşteri Sayısı	HKKO-YAKA		KKO-YAKA1		KKO-YAKA2		KKO-YAKA3	
		Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi
5	100	62,14	280	66,52	265	66,62	279	67,59	267
		60,22	246	65,61	230	65,51	245	65,59	232
		59,51	263	63,79	245	63,75	243	64,85	247
		67,7	263	74,30	240	74,25	238	73,37	242
		75,51	243	80,40	223	80,37	220	81,48	225
		64,43	245	68,90	225	68,96	223	69,97	227
		64,59	253	70,01	235	70,03	235	70,13	237
		55,64	265	60,77	248	60,84	249	60,83	250
		64,97	285	69,46	260	69,48	260	70,53	262
		60,82	251	66,15	236	66,21	232	66,21	238
9	200	108,2	638	116,67	599	116,8	588	117,23	601
		95,25	627	104,83	585	104,18	594	103,56	587
		94,6	675	102,08	634	103,49	630	102,87	636
		92,18	664	101,99	626	101,95	635	100,32	628
		102,77	651	112,01	615	112,09	610	111,50	617
		96,57	659	105,56	621	106,65	635	104,95	623
		99,5	643	107,61	606	107,65	615	108,05	608
		90,56	663	99,30	625	99,24	610	98,61	627
		95,89	691	104,85	651	105,74	655	104,23	653
		100,14	650	110,27	612	110,32	625	108,72	614
5	200	109,12	612	120,57	575	120,75	570	120,80	575
		97,98	658	108,36	615	108,62	625	108,79	630
		97,95	608	109,33	564	109,59	565	108,76	566
		100,1	639	108,69	594	108,93	585	111,08	596
		101,13	615	112,81	578	112,05	570	112,19	575
		101,77	701	113,51	665	113,75	661	112,88	667
		99,75	675	108,30	632	108,55	621	110,70	634
		95,01	705	104,11	668	104,39	670	105,59	670
		99,56	665	110,09	625	110,34	632	110,50	635
		103,09	619	114,96	580	114,18	575	114,30	582

**Tablo A.1. (Devamı) “Yerel aramalı” ve “yerel arama olmadan” elde edilen sonuçların karşılaştırılması**

Şarj istasyonu sayısı	Müşteri Sayısı	HKKO-YAKA		KKO-YAKA1		KKO-YAKA2		KKO-YAKA3	
		Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	Hesaplama Süresi
9	400	181,2	2415	197,64	2325	195,64	2320	198,64	2327
		179,4	2628	194,70	2535	195,69	2541	196,70	2537
		176,18	2331	194,22	2236	194,21	2240	193,22	2238
		175,25	2992	191,21	2898	192,24	2900	192,21	2900
		180,9	2280	197,31	2185	197,32	2175	198,31	2187
		181,4	2410	198,85	2314	198,47	2310	198,85	2316
		177,3	2303	195,43	2212	194,41	2205	194,43	2214
		174,72	2510	193,64	2417	193,81	2410	191,64	2419
		173,57	2376	193,40	2282	193,25	2324	192,40	2284
		179,5	2556	195,80	2460	195,65	2480	196,80	2462
5	400	178,13	2341	195,07	2256	195,04	2250	195,07	2258
		179,75	2483	197,81	2393	197,94	2398	196,81	2395
		179,75	1885	198,81	1796	198,74	1805	196,81	1801
		179,02	2865	195,02	2779	195,16	2814	196,02	2781
		175,93	2208	193,69	2118	193,73	2185	192,69	2120
		183,91	2300	199,30	2212	198,32	2231	201,30	2214
		183,2	2203	199,73	2113	198,54	2126	200,53	2120
		179,67	2303	195,43	2212	195,72	2241	196,73	2233
		180,54	2273	196,66	2183	196,78	2198	197,66	2189
		180,18	2146	197,42	2053	197,49	2035	197,28	2041
<b>% Ortalama Fark</b>		-9,02	6,04	-9,14	6,08	-9,03	5,63		

KKO-YAKA1: Ters çevirme operatörlü YAKA, KKO-YAKA2: Yer değiştirme operatörlü YAKA, KKO-YAKA3: Ekleme operatörlü YAKA

Tablo A.1'deki sonuçları incelediğimizde, KKO'da yerel arama operatörleri kullanılmadan YAKA'da ise farklı yerel arama operatörleri uygulandığında (KKO-YAKA1, KKO-YAKA2 ve KKO-YAKA3) elde edilen sonuçların birbirine yakın olduğu görülmektedir. HKKO-YAKA sonuçlarına göre, KKO'da yerel arama operatörlerini kullanmak, toplam maliyette %9'luk bir azalma sağlamaktadır. Ancak çözüm süresini ortalama %6 arttırmaktadır. Çözüm süresindeki artış, toplam maliyetteki iyileşmeden daha az olduğu ve sonuçlar saniyeler içinde değerlendirildiği

için KKO'da yerel arama operatörleri kullanıldığında daha iyi sonuçlar alındığını söyleyebiliriz.

## EK B. Başlangıç Çözümün YAKA'ya Etkisinin Analizi

Başlangıç çözümün YAKA'ya etkisini göstermek amacıyla FORT örnekleri kullanılmıştır. Klasik YAKA ve HKKO-YAKA ile elde edilen sonuçların karşılaştırılması Tablo B.1'de rapor edilmiştir. Sonuçlar, klasik YAKA'nın 1000 iterasyon, HKKO-YAKA'da ise HKKO'nun 10 iterasyon ve YAKA'nın da 1000 iterasyon koşturulması sonucunda elde edilmiştir. “% İyileştirme” geliştirdiğimiz yöntemin (HKKO-YAKA) klasik YAKA'nın sonuçları arasındaki hesaplama süresi ve toplam maliyet açısından iyileştirme yüzdesini göstermektedir.

**Tablo B.1.** YAKA ve HKKO-YAKA ile elde edilen sonuçların karşılaştırılması

Şarj istasyonu sayısı	Müşteri Sayısı	YAKA		HKKO-YAKA			
		Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	% İyileştirme	Hesaplama Süresi	% İyileştirme
9	100	82,59	427	60,89	26,27	275	35,60
		79,92	420	60,82	23,90	251	40,24
		85,26	381	61,84	27,47	259	32,02
		83,77	448	64,01	23,59	269	39,96
		97,76	435	72,21	26,14	268	38,39
		87,31	412	65,16	25,37	249	39,56
		83,44	425	65,25	21,80	249	41,41
		77,62	384	56,43	27,30	274	28,65
		85,03	376	61,22	28,00	264	29,79
		72,36	387	57,73	20,22	247	36,18
5	100	78,01	397	62,14	20,34	280	29,47
		80,45	413	60,22	25,15	246	40,44
		78,84	435	59,51	24,52	263	39,54
		92,98	401	67,7	27,19	263	34,41
		99,42	354	75,51	24,05	243	31,36
		86,93	407	64,43	25,88	245	39,80
		82,49	452	64,59	21,70	253	44,03
		71,33	435	55,64	22,00	265	39,08
		84,44	453	64,97	23,06	285	37,09
		73,84	379	60,82	17,63	251	33,77

**Tablo B.1. (Devamı) YAKA ve HKKO-YAKA ile elde edilen sonuçların karşılaştırılması**

Şarj istasyonu sayısı	Müşteri Sayısı	YAKA		HKKO-YAKA			
		Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	% İyileştirme	Hesaplama Süresi	% İyileştirme
<b>9</b>	<b>200</b>	184,61	1122	108,2	41,39	638	43,14
		168,36	1053	95,25	43,42	627	40,46
		163,66	1149	94,6	42,20	675	41,25
		164,47	1150	92,18	43,95	664	42,26
		174,79	1162	102,77	41,20	651	43,98
		162,07	1119	96,57	40,41	659	41,11
		165,14	1095	99,5	39,75	643	41,28
		160,67	1127	90,56	43,64	663	41,17
		168,89	1178	95,89	43,22	691	41,34
		173,24	1103	100,14	42,20	650	41,07
<b>5</b>	<b>200</b>	193,63	1095	109,12	43,65	612	44,11
		163,69	1073	97,98	40,14	658	38,68
		166,52	1034	97,95	41,18	608	41,20
		174,17	1116	100,1	42,53	639	42,74
		170,92	1146	101,13	40,83	615	46,34
		176,01	1122	101,77	42,18	701	37,52
		165,58	1118	99,75	39,76	675	39,62
		164,52	1169	95,01	42,25	705	39,69
		168,25	1089	99,56	40,83	665	38,93
		172,25	1012	103,09	40,15	619	38,83
<b>9</b>	<b>400</b>	407,48	6503	181,2	55,53	2415	62,86
		402,89	7058	179,4	55,47	2628	62,77
		394,07	6127	176,18	55,29	2331	61,96
		382,92	8780	175,25	54,23	2992	65,92
		392,17	5984	180,9	53,87	2280	61,90
		385,27	6448	181,4	52,92	2410	62,62
		372,29	6123	177,3	52,38	2303	62,39
		360,89	7028	174,72	51,59	2510	64,29
		349,25	6453	173,57	50,30	2376	63,18
		341,31	7257	179,5	47,41	2556	64,78



**Tablo B.1. (Devamı) YAKA ve HKKO-YAKA ile elde edilen sonuçların karşılaştırılması**

Şarj istasyonu sayısı	Müşteri Sayısı	YAKA		HKKO-YAKA			
		Toplam Maliyet	Hesaplama Süresi	Toplam Maliyet	% İyileştirme	Hesaplama Süresi	% İyileştirme
5	400	366,05	6981	178,13	51,34	2341	66,47
		329,5	7249	179,75	45,45	2483	65,75
		319,5	5855	179,75	43,74	1885	67,81
		348,04	8390	179,02	48,56	2865	65,85
		345,86	6924	175,93	49,13	2208	68,11
		365,82	7143	183,91	49,73	2300	67,80
		356,4	6959	183,2	48,60	2203	68,34
		358,34	6709	179,67	49,86	2303	65,67
		351,08	6819	180,54	48,58	2273	66,67
		363,36	6738	180,18	50,41	2146	68,15
<b>% Ortalama İyileştirme</b>				38,85		47,65	

Tablo B.1, HKKO-YAKA'nın YAKA'dan önemli ölçüde daha iyi sonuçlar sağladığını göstermektedir. HKKO'yu YAKA'nın başlangıç çözümü olarak kullandığımızda, elde edilen nihai çözümlerde hesaplama süresi ve toplam maliyet açısından sırasıyla ortalama %47,65 ve %38,85 oranında iyileştirme sağlanmış oldu.



## ÖZGEÇMİŞ

Ad-Soyad :Serap ERCAN CÖMERT

### ÖĞRENİM DURUMU:

- **Lisans** : 2012, Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü
- **Yükseklisans** : 2014, Sakarya Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği Programı

### MESLEKİ DENEYİM:

- 2012 yılından beri Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalışmaktadır.

### DİĞER ESERLER:

- Cömert, S. E., Gökler, S. H., & Yazgan, H. R. (2016). Hücresel imalat sistemlerinin K-means algoritması ve genetik algoritma ile tasarlanması: Bir uygulama. *Akademik Platform Mühendislik ve Fen Bilimleri Dergisi*, 4(3), 41-47. <http://dx.doi.org/10.21541/apjes.06335>
- Cömert, S. E., & Yener, F. (2016). Bir gıda firması için bulanık analitik hiyerarşi prosesi ile depo yeri seçimi. *Uluslararası İktisadi ve İdari Bilimler Dergisi*, 2(2), 161-177.
- Comert, S. E., Yazgan, H. R., Sertvuran, İ., & Şengül, H. (2017). A new approach for solution of vehicle routing problem with hard time window: an application in a supermarket chain. *Sādhanā*, 42(12), 2067-2080. <https://doi.org/10.1007/s12046-017-0754-1>
- Comert, S. E., Yazgan, H. R., Kır, S., & Yener, F. (2018). A cluster first-route second approach for a capacitated vehicle routing problem: a case study. *International Journal of Procurement Management*, 11 (4), 399-419. <https://doi.org/10.1504/IJPM.2018.092766>

- Comert, S. E., Yazgan, H. R., Sertvuran, İ., & Şengül, H. (2018). Sıkı zaman pencereli araç rotalama probleminin çözümü için yeni bir yöntem önerisi ve bir süpermarket zincirinde uygulanması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22 (2), 685-694. <https://doi.org/10.19113/sdufbed.20568>
- Cömert, S. E., Yazgan, H. R., & Görgülü, N. (2019). Eş zamanlı topla dağıt araç rotalama problemi için iki aşamalı bir çözüm yöntemi önerisi. *International Journal of Advances in Engineering and Pure Sciences*, 31(2), 107-117. <https://dx.doi.org/10.7240/jeps.461528>
- Cömert, S., Yener, F., Kır, S., & Yazgan, H. (2019). Analysis of the Most Important Factors that Affect Tensile and Shear Strength of Dual-Phase Steels Using Taguchi Method. *Acta Physica Polonica A*, 135(4), 684-686.
- Cömert, S. E., Yazgan, H. R., Çakır, B., & Sarı, N. (2020). Esnek zaman pencereli araç rotalama probleminin çözümü için önce kümele-sonra rotala temelli bir yöntem önerisi; bir süpermarket örneği. *Konya Mühendislik Bilimleri Dergisi*, 8(1), 18-31. <https://dx.doi.org/10.36306/konjes.698326>
- Cömert, S. E., Yazgan, H. R., & Kılıç, E. N. (2020). Araç rotalama probleminin sezgisel algoritmalar ile çözülmesi: Bir boya fabrikasında uygulama. *Journal of Turkish Operations Management*, 4(2), 549-563.
- Comert, S. E., & Yazgan, H. R. (2021). Effective cluster-first route-second approaches using metaheuristic algorithms for the capacitated vehicle routing problem. *International Journal of Industrial Engineering: Theory, Applications, and Practice*, 28(1), 14-38. <https://dx.doi.org/10.23055/ijietap.2021.28.1.7223>
- Comert, S. E., Yazgan, H. R., & Turk, G. (2022). Hopfield neural network based on clustering algorithms for solving green vehicle routing problem. *International Journal of Industrial Engineering Computations*, 13(4), 573-586. <https://dx.doi.org/10.5267/j.ijiec.2022.6.002>
- Dörtköşe, S., Yazgan, H. R., & Cömert, S. E. (2022). Elektrikli araç şarj istasyon yerlerinin akış yakıt ikmal yer modeli kullanılarak belirlenmesi. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 38(2), 371-382.
- Kır, S., Cömert, S. E., Yener, F., Yazgan, H. R., & Candan, G. (2019). Hazardous Waste Recycling: End of Life Tires Case. *Acta Physica Polonica A*, 135(4), 681-683.
- Şen, T., Yazgan, H. R., & Cömert, S. E. (2015). Kapasite kısıtlı araç rotalama probleminin çözümü için yeni bir algoritma geliştirilmesi: bir süpermarket zincirinde uygulanması. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 19(1), 83-88.
- Yazgan, H. R., Ercan, S., & Arslan, C. (2014). Talep ve kapasite kısıtlı optimizasyon problemi için yeni bir melez algoritma. *Journal of Industrial Engineering (Turkish Chamber of Mechanical Engineers)*, 25(1/2), 16-28.
- Yener, F., Yazgan, H. R., Cömert, S. E., Kır, S., & Kaya, Y. (2016). Birliktelik kuralları ve genetik algoritma ile sipariş yığınlama probleminin çözümü: Bir ecza deposunda uygulama. *Journal of Transportation and Logistics*, 1(2), 129-142. <https://dx.doi.org/10.22532/jtl.267841>

- Yener, F., Kır, S., Cömert, S. E., & Yazgan, H. (2019). Class Based Storage with Mixed Integer Linear Programming Approach for Hazardous Materials Storage. *Acta Physica Polonica A*, 135(4), 678-680.