



**ÇOKLU ZAMAN PENCERELİ VE ÇOK KULLANIMLI
ARAÇ ROTALAMA PROBLEMİNİN ÇÖZÜMÜ
İÇİN İKİ AMAÇLI SEZGİSEL YAKLAŞIM**

(Yüksek Lisans Tezi)

Gölnur EŞGÜNOĞLU

Kütahya – 2024

T. C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
Endüstri Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

**ÇOKLU ZAMAN PENCERELİ VE ÇOK KULLANIMLI ARAÇ
ROTALAMA PROBLEMİNİN ÇÖZÜMÜ İÇİN İKİ AMAÇLI
SEZGİSEL YAKLAŞIM**

Danışman:
Doç. Dr. Derya DELİKTAŞ

Hazırlayan:
Gülnur EŞGÜNOĞLU

Kütahya – 2024

Kabul ve Onay

KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Endüstri Mühendisliği Ana bilim dalında, 202185241001 öğrenci numaralı, Gülnur EŞGÜNOĞLU'nun hazırlamış olduğu “ÇOKLU ZAMAN PENCERELİ VE ÇOK KULLANIMLI ARAÇ ROTALAMA PROBLEMİNİN ÇÖZÜMÜ İÇİN İKİ AMAÇLI SEZGİSEL YAKLAŞIM” başlıklı yüksek lisans tez çalışması ile ilgili tez savunma sınavı jüri tarafından yapılmış ve adayın tezinin OY BİRLİĞİ ile kabul edilmesine karar verilmiştir.

10/09/2024

Tez Jürisi	İmza	
	Kabul	Ret
Doç. Dr. Derya DELİKTAŞ (Danışman)		
Prof. Dr. Özden ÜSTÜN		
Doç. Dr. İhsan EROZAN		

Doç. Dr. Eray ACAR

Enstitü Müdürü

Bilimsel Etik Bildirimi

Yüksek Lisans tezi olarak hazırladığım “*Çoklu Zaman Pencere*li ve *Çok Kullanımlı Araç Rotalama Probleminin Çözümü için İki Amaçlı Sezgisel Yaklaşım*” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

30/10/2024

Gülnur EŞGÜNOĞLU

Özgeçmiş

İlk, orta ve lise eğitimini İstanbul'da tamamladı. 2021 yılında Kütahya Dumlupınar Üniversitesi Endüstri Mühendisliği bölümünden bölüm ve fakülte birincisi olarak mezun oldu. 2018 yılında İSTAÇ A.Ş. firmasında bilgisayar stajını, 2019 yılında Özkoç Hidrolik Makine Sanayi ve Ticaret A.Ş. fabrikasında üretim stajını ve 2021 yılında General Mobile firmasında yönetim stajını tamamladı. 2021 yılında Kütahya Dumlupınar Üniversitesi Endüstri Mühendisliği bölümünde yüksek lisans eğitimine başladı. Aynı yıl İstanbul'da medikal sektörde faaliyet gösteren bir firmada üretim mühendisi olarak çalışmaya başladı ve halen aynı firmada görev almaktadır.



ÖZET

ÇOKLU ZAMAN PENCERELİ VE ÇOK KULLANIMLI ARAÇ ROTALAMA PROBLEMİNİN ÇÖZÜMÜ İÇİN İKİ AMAÇLI SEZGİSEL YAKLAŞIM

EŞGÜNOĞLU, Gülnur

Yüksek Lisans Tezi, Endüstri Mühendisliği Ana Bilim Dalı

Tez Danışmanı: Doç. Dr. Derya DELİKTAŞ

Kasım, 2024, 72 sayfa

Rekabetin hızla arttığı günümüz dünyasında firmaların devamlılığını sürdürebilmesi için üretimdeki israfları minimize etmesi en önemli faktörlerden biridir. Üretimdeki israfların büyük bir kısmını da gereksiz yere olan duruşlar oluşturmaktadır. Bu çalışmada, medikal sektörde faaliyet gösteren ve laboratuvar sarf malzemeleri üreten bir firmanın duruşları incelenmiştir. İnceleme sonucunda duruşların, istasyonlara hammadde ve yarı mamul tedariki esnasında meydana geldiği saptanmıştır. İstasyonlara yarı mamul ve hammadde tedarik edecek bir su örümceği ile mevcut duruşların engellenebileceği öngörülmüştür. Duruşların engellenmesi için gerekli olan su örümceğinin rota probleminin zaman pencereli araç rotalama problemi (ZPARP) olduğu belirlenmiştir. Su örümceğinin istasyon zaman pencerelerini dikkate alacak şekilde toplam seyahat süresini ve toplam gecikme süresini enküçükleme amacıyla karma tamsayılı doğrusal matematiksel model kullanılmıştır. Ele alınan gerçek hayat probleminin çözümü için iki amaçlı genetik algoritma önerilmiştir. Her iki amaç fonksiyonu ağırlıklı-toplam skalerleştirme yöntemi ile birleştirilmiştir. Genetik algoritmanın parametreleri sonuç üzerinde etkisi olduğundan deney tasarımı ve **irace** yöntemi ile parametre kalibresi yapılmıştır. Elde edilen sonuçlar, deney tasarımı ile önerilen parametre değerlerinin irace yönteminden daha iyi sonuç verdiğini göstermiştir. Ağırlıklı-toplam skalerleştirme yöntemine ek olarak, iki amaçlı problem konik ve chebyshev skalerleştirme yöntemleri kullanılarak da birleştirilmiştir. Elde edilen sonuçlar, chebyshev ile bütünleştirilen genetik algoritmanın diğer skalerleştirme yöntemlerden daha iyi performans gösterdiğini kanıtlamıştır. Sonuçlar, modelin uygulanabilir olduğunu ve firmanın önerilen rotada kullanabileceği bir su örümceği ile duruşları engelleyerek kullanılabilirlik oranını arttıracaklarını göstermektedir.

Anahtar Kelimeler: Araç Rotalama, Çoklu Kullanım, Çoklu Zaman Penceresi, Karma Tamsayılı Doğrusal Programlama, Su Örümceği

ABSTRACT**THE BI-OBJECTIVE HEURISTIC APPROACH FOR SOLVING THE
MULTIPLE TIME WINDOW AND MULTIPLE USE VEHICLE ROUTING
PROBLEM****EŞGÜNOĞLU, Gülnur****Master's Thesis, Department of Industrial Engineering****Supervisor: Assoc. Prof. Dr. Derya DELİKTAŞ****November, 2024, 72 pages**

In today's world, where competition is rapidly increasing, minimizing production waste has become a key factor for firms to maintain their continuity. A significant portion of production waste results from unnecessary downtimes. This study examines the downtimes encountered by a company in the medical sector that produces laboratory consumables. The analysis reveals that these downtimes primarily occur during the supply of raw materials and semi-finished products to workstations. To mitigate these downtimes, it is proposed that a "water spider" could be utilized to supply the necessary materials to the workstations. The routing issue of the water spider is identified as a Time-Window Vehicle Routing Problem (TWVRP), given its objective of minimizing the total travel and tardiness times while adhering to the time windows for each workstation. A bi-objective genetic algorithm is proposed to solve the real-life problem. Both objective functions are combined using the weighted-sum scalarization method. Since the parameters of the genetic algorithm impact the results, experimental design and parameter calibration with the **irace** method were performed. The results indicate that the parameter values proposed by the experimental design yielded better outcomes than those obtained through the **irace** method. Additionally, the bi-objective problem was combined using other scalarization methods, namely conic and chebyshev scalarizations. The findings demonstrate that the genetic algorithm integrated with chebyshev scalarization outperforms other scalarization methods. The results confirm that the model is feasible and that implementing the proposed route with a water spider will reduce downtimes, thereby increasing the firm's utilization rate.

Keywords: Vehicle Routing, Multiple Use, Multiple Time Window, Mixed Integer Linear Programming, Water Spider

ÖNSÖZ

Yüksek lisans tez çalışmam boyunca desteğini hiçbir zaman esirgemeyen, yürüdüğüm bu yolda asla yalnız olmadığımı hissettiren ve her zaman yanımda olan, lisans ve yüksek lisans eğitimim boyunca bana her gün yeni şeyler öğreten, ne olursa olsun her zaman bir umut olduğunu ve pes etmeden yola devam etmenin her zaman kazandıracağını gösteren, öğrencisi olmaktan her zaman gurur duyacağım çok değerli danışman hocam Doç. Dr. Derya DELİKTAŞ'a teşekkürü bir borç bilirim.

Tez jürime katılma nezaketini gösteren değerli hocalarım Prof. Dr. Özden ÜSTÜN'e ve Doç. Dr. İhsan EROZAN'a teşekkürlerimi sunarım.

Bugünlere gelmemi sağlayan, aldığım her kararda arkamda duran ve beni cesaretlendiren, hayatımın her anında yanımda olup sonsuz sevgilerini hissettiren, yüksek lisans sürecim boyunca hoşgörü ve desteklerini esirgemeyen canım annem Rabia EŞGÜNOĞLU'na, canım babam Mehmet EŞGÜNOĞLU'na ve biricik kardeşim Tayfur EŞGÜNOĞLU'na sonsuz teşekkürlerimi sunarım.

Ayrıca bu süreç boyunca beni her zaman motive eden ve desteklerini esirgemeyen çok değerli iş arkadaşlarıma ve yöneticilerime teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET	v
ABSTRACT	vi
TABLolar LİSTESİ	x
ŞEKİLLER LİSTESİ	xi
KISALTMALAR VE SEMBOLLER	xiii
GİRİŞ	1

BİRİNCİ BÖLÜM

LİTERATÜR ARAŞTIRMASI

1.1. LİTERATÜR ARAŞTIRMASI	4
----------------------------------	---

İKİNCİ BÖLÜM

ARAÇ ROTALAMA PROBLEMİ

2.1. ARAÇ ROTALAMA PROBLEMİ	13
2.1.1. Araç Rotalama Problem Çeşitleri	14
2.1.2. Zaman Pencerele Araç Rotalama Problemi	17
2.2. MİLK-RUN VE SU ÖRÜMCEĞİ	18

ÜÇÜNCÜ BÖLÜM

GENETİK ALGORİTMA

3.1. GENETİK ALGORİTMA	22
3.1.1. Genetik Algoritma Çalışma Prensipleri	23
3.2. PARAMETRE BELİRLEME	25
3.2.1. Otomatik Parametre Ayarlama (Iterated F-Race Algorithm)	26
3.2.2. Deney Tasarımı	29

DÖRDÜNCÜ BÖLÜM

ÇOKLU ZAMAN PENCERELİ VE ÇOKLU KULLANIMLI ARAÇ ROTA LAMA PROBLEMİNİN ÇÖZÜMÜ İÇİN İKİ AMAÇLI SEZGİSEL YAKLAŞIM

4.1. PROBLEMİN TANITIMI	34
4.2. MEVCUT DURUM ANALİZİ	36

4.3. ELE ALINAN GERÇEK HAYAT PROBLEMİ İÇİN ÖNERİLEN MODEL	38
4.3.1. İki Amaçlı Matematiksel Model	39
4.4. İKİ AMAÇLI GENETİK ALGORİTMA ÇÖZÜMÜ	44
4.4.1. Kromozom Gösterimi	45
4.4.2. Uygunluk Fonksiyonu	48
4.4.3. Seçim	48
4.4.4. Genetik Operatörler	49
4.4.5. Genetik Algoritmalarda Parametre Seçimi	51
4.5. PERFORMANS METRİKLERİ	56
SONUÇLAR VE ÖNERİLER	62
KAYNAKÇA	64
DİZİN	72

TABLÖLAR LİSTESİ

	<u>Sayfa</u>
Tablo 1.1: Literatür Taraması Özeti	9
Tablo 4.1: HM ve YM Tedariğinden Kaynaklı Duruş Sürelerinin Hesaplanması	36
Tablo 4.2: Önerilen Algoritmanın Sözde (Pseudo) Kodu	45
Tablo 4.3: Uzaklık Matrisi (7x7)	47
Tablo 4.4: Faktörler ve Faktörlerin Düzeyi	52
Tablo 4.5: irace’de Kullanılan Parametre Aralıkları	53
Tablo 4.6: irace ve DoE Sonucu Elde Edilen Parametre Değerleri	54
Tablo 4.7: KKV Kombinasyonlarına Göre Referans Noktaları	57
Tablo 4.8: Skalerleştirme Yöntemlerinin Amaç Fonksiyonlarına Etkisi	58

ŞEKİLLER LİSTESİ

	<u>Sayfa</u>
Şekil 2.1: Fabrika İçi Milk-run Sistemi	20
Şekil 3.1: Genetik Algoritma Akış Diyagramı	22
Şekil 3.2: Gen, Kromozom ve Popülasyon Yapısı	23
Şekil 3.3: Mutasyon Örnekleri	25
Şekil 3.4: DoE Adımları	30
Şekil 4.1: Mevcut Durum Üretim Alanı	34
Şekil 4.2: Mevcut Durum Akış Diyagramı	35
Şekil 4.3: Mevcut Durum Makina Duruş Sürelerinin Kullanılabilirlik Oranına Etkisi	37
Şekil 4.4: Su Örümceği Kullanımı ile Oluşturulan Üretim Alanı	38
Şekil 4.5: Örnek Kromozom Yapısı	46
Şekil 4.6: Örnek OOX Çaprazlama Yöntemi	49
Şekil 4.7: Örnek POX Çaprazlama Yöntemi	50
Şekil 4.8: Genetik Algoritmanın 31 Çalışmasından DoE Sonucu Elde Edilen Ana ve Birleşik Etkiler Grafikleri (Diğer Ağırlıklar için)	52
Şekil 4.9: Sabit Değerin Belirlenmesi için Kullanılan Yakınsama Grafiği	53
Şekil 4.10: irace'e Karşı DoE Parametre Ayarlama Yönteminden Elde Edilen Sonuçların Pareto Grafiği	54
Şekil 4.11: 9 Farklı Amaç Ağırlığı Kombinasyonu için irace ve DoE Yöntemlerinden Elde Edilen Normalize Edilmiş Uyum Değerlerinin Kutu-Bıyık Grafikleri	55
Şekil 4.12: KKV Kombinasyonları ve Amaç Ağırlıkları için Frekans Grafikleri	57
Şekil 4.13: ATY, KSY ve CSY'den Elde Edilen Amaç Değerleri Sonucuna Dayanan Pareto Etkin Çözümler	59
Şekil 4.14: Üç Farklı Konveks Kombinasyon Vektörüne Sahip Üç Kıyaslama Örneği için Skalerizasyon Yöntemlerine Dayalı Çok Amaçlı Genetik Algoritmanın 31 Çalışmasından Elde Edilen İdeal Noktalara Olan Ağırlıklı Mesafenin Kutu-Bıyık Grafikleri	60

Şekil 4.15: Üç Farklı Konveks Kombinasyon Vektörüne Sahip Üç Kıyaslama Örneği için Skalerizasyon Yöntemlerine Dayalı Çok Amaçlı Genetik Algoritmanın 31 Çalışmasından Elde Edilen Referans Noktalara Olan Ağırlıklı Mesafenin Kutu-Bıyık Grafikleri	61
---	----



KISALTMALAR VE SEMBOLLER

ARP	Araç Rotalama Problemi
ZPARP	Zaman Pencereci Araç Rotalama Problemi
ÇZPARP	Çoklu Zaman Pencereci Araç Rotalama Problemi
ÇKZPARP	Çok Kullanımlı Zaman Pencereci Araç Rotalama Problemi
UBGA	Uygunluk Birleştirilmiş Genetik Algoritması
HM	Hammadde
YM	Yarı Mamul
DoE	Deney Tasarımı
HGA-SES	Hibrit Genetik Algoritma-Solomon Eklemeli Sezgiseli
ÖTARP	Ödül Toplamalı Araç Rotalama Problemi
ATY	Ağırlıklı-Toplam Yöntemi
KSY	Konik Skalerleştirme Yöntemi
CSY	Chebyshev Skalerleştirme Yöntemi
OOX	Order-Based One-Point Crossover
POX	Precedence Preserving Order-Based Crossover
KKV	Konveks Kombinasyon Vektörü
D_{IP}	İdeal Noktalara Ağırlıklı Uzaklık
D_{RIP}	Referans Noktalara Ağırlıklı Uzaklık



TEZ METNİ

GİRİŞ

Kombinatoriyal optimizasyon problemi olan araç rotalama problemi (ARP) ilk kez 1959 yılında ortaya çıkmıştır. ARP 'de amaç, bütün müşterilerin ihtiyacını karşılayan en küçük maliyetli rotaların belirlenmesidir. Çok kullanımlı zaman pencereli araç rotalama problemi (ÇKZPARP) ise, ARP'nin bir çeşididir. ÇKZPARP araçların birden fazla rotada kullanılmasına izin vermektedir (Koç ve Karaođlan, 2012). Zaman pencereli araç rotalama problemi (ZPARP), araçlara kapasite kısıtlamalarının yanı sıra müşterilere zaman penceresi kısıtlamaları uygulamaktadır. Bir ZPARP'de, her müşteri belirlenen zaman penceresi içinde hizmet almalı ve rotadaki müşterilerin toplam talebi aracın kapasitesini aşmamalıdır (Cai, vd., 2022). ZPARP'ni çözmek için 1980'lerden bugüne kadar çeşitli kesin, buluşsal ve meta sezgisel yöntemler ve bunların beraber kullanıldığı yöntemler önerilmiştir. Kesin yöntemler küçük boyutlu problemler için kullanılabilirken ZPARP'nin büyük boyutlu gerçek hayat problemlerinde kesin yöntemlerin kullanılması çok sayıda varyasyondan dolayı imkânsız hale gelmektedir (Expósito, vd., 2019). Bu nedenle bu tip problemlerin çözümünde sezgisel ve meta sezgisel teknikler kullanılmaktadır. Çözüm yöntemlerinden en yaygın olan genetik algoritma ise ilk olarak 1975 yılında John Holland ve çalışma arkadaşları tarafından tanıtılmıştır (Su, vd., 2023).

Bu tez çalışmasında medikal sektörde faaliyet gösteren ve laboratuvar sarf malzemeleri üreten bir firmada gerçek hayat problemi üzerine çalışılmıştır. Çalışma yapılan işletmede istasyonların hammadde (HM) ve yarı mamul (YM) tedariklerinden kaynaklı duruşların engellenebilmesi amaçlanmıştır. Problem, müşteri olarak varsayılan istasyonlarda gün içerisinde birden fazla zaman penceresinin olması sebebiyle çoklu zaman pencereli araç rotalama problemi (ÇZPARP) olarak belirlenmiştir. Alan darlığından dolayı, kullanılan tek aracın birden fazla sefer yapması gerektiği saptanmıştır. Mevcut durumda istasyonların ihtiyaç duyduğu HM ve YM'ler istasyonda çalışan operatör tarafından temin edilmektedir. Temin süresi boyunca ise makina kapatılmakta ve malzeme tedariklerinden kaynaklı duruşlar oluşmaktadır. Malzeme tedariki boyunca makinanın duruş yaşaması üretilen ürün adedinde azalmaya sebep olmakta ve makinaların kullanılabilirlik oranlarını düşürmektedir. Bu duruşların önlenmesi istasyonlara HM ve YM tedariki yapacak bir su örümceğine ihtiyaç duyulduğu tespit edilmiştir. Yalın üretimde malzeme veya ürünlerin belirli bir döngüde taşınması ve dağıtılması *milk-run* olarak adlandırılmaktadır (Sipahiođlu ve Altın, 2019). Milk-run sistemi fabrika içinde ve dışında olmak üzere iki çeşittir. Fabrika içinde milk-run sağlayan

yalın aracı *su örümceği* olarak isimlendirilmektedir. Su örümceği, üretim ortamında malzemelerin ihtiyaç duyulduğu yer ve zamanda temin edilmesinde görevli kişidir (Klenk, Galka ve Günthner, 2015). Belirli bir rota boyunca istasyonları ziyaret ederek kısıtlı araç kapasitesi ile malzeme tedarikinde bulunmaktadır.

Literatürde zaman pencereci, çok amaçlı, çok kullanımlı araç rotalama problemleri yer alsa da tek bir müşteriye ait birden fazla zaman penceresi bulunması ve her zaman penceresinde müşteriye uğranması gerekliliğine rastlanmamıştır. Çalışma, problemde her istasyonun kendine ait birden fazla zaman penceresi olması, her rotaya aynı aracın hizmet vermesi ve aracın gün içerisinde çok sefer yapıyor olması nedeniyle literatürdeki diğer çalışmalardan ayrılmaktadır.

Su örümceğinin istasyon zaman pencerelerini dikkate alacak şekilde toplam seyahat süresini ve toplam gecikme süresini enküçükleme amacıyla karma tamsayı doğrusal matematiksel model kullanılmıştır. Önerilen matematiksel model küçük bir örnek üzerinde Lingo 11.0 programında çalıştırılarak test edilmiştir. Problemin Np-Zor (Cömert vd., 2017) olması sebebiyle çözümü için iki amaçlı genetik algoritma önerilmiştir. Her iki amaç fonksiyonu ağırlıklı-toplam skalerleştirme yöntemi ile birleştirilmiştir. Genetik algoritmanın parametreleri sonuç üzerinde etkisi olduğundan deney tasarımı ve **irace** yöntemi ile parametre kalibresi yapılmıştır. Elde edilen sonuçlar, deney tasarımı ile önerilen parametre değerlerinin irace yönteminden daha iyi sonuç verdiğini göstermiştir. Ağırlıklı-toplam skalerleştirme yöntemine ek olarak, iki amaçlı problem konik ve chebyshev skalerleştirme yöntemleri kullanılarak da birleştirilmiştir. Elde edilen sonuçlar, chebyshev ile bütünleştirilen genetik algoritmanın diğer skalerleştirme yöntemlerden daha iyi performans gösterdiğini kanıtlamıştır. Sonuçlar, modelin uygulanabilir olduğunu ve firmanın önerilen rotada kullanabileceği bir su örümceği ile duruşları engelleyerek kullanılabilirlik oranını arttıracakını göstermektedir.

Çalışmanın devamında, birinci bölümde literatür araştırması sunulmuştur. İkinci bölümde, tezin konusu olan araç rotalama problemi açıklanmıştır. Üçüncü bölümde tezde kullanılan genetik algoritma ve parametre belirleme yöntemleri ayrıntılı olarak açıklanmıştır. Çalışmanın dördüncü bölümünde ele alınan gerçek hayat problemi tanıtılmış ve önerilen matematiksel model sunulmuş, genetik algoritma yöntemi detaylandırılmış ve algoritma performans metreklerine göre değerlendirilmiştir. Çalışmanın son bölümünde de sonuçlar ve öneriler kısmı yer almaktadır.



BİRİNCİ BÖLÜM
LİTERATÜR ARAŞTIRMASI

1.1. LİTERATÜR ARAŞTIRMASI

Bu tez çalışmasında ele alınan konu için literatür taraması yapılmıştır. Öncelikle literatürdeki ARP üzerine tarama yapılmış sonrasında ZPARP analiz edilmiştir. Literatürdeki çalışmalarda müşterilere ait tek bir zaman penceresi vardır. Çalışılan gerçek hayat probleminde ise her müşterinin gün içerisinde hizmet verilmesi gereken birden fazla zaman penceresi bulunmaktadır. Alan darlığından dolayı, kullanılan tek araç gün içerisinde birden fazla sefer yapmaktadır. Bu nedenle literatürdeki tek araç rotalama problemleri de incelenmiştir. Bu tarama, Tablo 1.1’de özetlenmiştir.

Koç ve Karaoğlan (2012), çalışmalarında ÇKZPARP için toplam taşıma maliyetini enküçükleyecek bir karma tamsayılı doğrusal programlama modeli önermişlerdir. Önerilen modeli literatürdeki Solomon’un 25 müşterilik test kümesinden türetilen 224 farklı test problemi üzerinde deneyerek ve GAMS paket programında kodlayarak en iyi çözüme ulaşma zamanı açısından kıyaslama yapmışlardır. ÇKZPARP, ARP’den farklı olarak araçların birden fazla rotada kullanılmasına izin vermektedir. ÇKZPARP kavramı ilk defa 1987 yılında Salhi’nin doktora tezinde ortaya atılmıştır. ZPARP’nin ARP’den farkı ise her müşteriye belli bir zaman aralığında hizmet verme zorunluluğunun olmasıdır.

Angel-Bello, Martinez-Salazar ve Alvarez (2013), çalışmalarında tek aracın birden fazla kullanımı ile müşterilerin hizmet almak için bekleyerek geçirdikleri sürelerinin toplamını en aza indirmeyi amaçlamışlardır. Bu amaç doğrultusunda iki karma tamsayılı model önererek karşılaştırmışlardır. Önerdikleri iki model arasındaki fark kapasite kısıtlamalarını ifade etmek için farklı yolların kullanılmış olmasıdır. Problem için kıyaslama verisini gerçek koordinatlara sahip rastgele noktalardan oluşturmuşlardır.

Kumar vd. (2014), çalışmalarında toplam kat edilen mesafenin, toplam araç sayısının ve rota uzunlukları arasındaki farkın en aza indirilmesini amaçlayan ZPARP’ni ele almışlardır. Problemin çözümü için uygunluk birleştirilmiş genetik algoritmasını (UBGA) tanıtmışlardır. Tanıtılan algoritma öncelikle ilk iki amaç için Solomon kıyaslama örnekleri ile test edilmiştir. Daha sonra üçüncü amaç olan rota uzunlukları arasındaki farkın minimize edilmesi modele dahil edilmiş ve UBGA’nın toplam mesafeyi ve toplam araç sayısını etkilemeden dengeli rotalar oluşturduğunu gözlemlemişlerdir.

Melián-Batista vd. (2014), çalışmalarında İspanya’nın Tenerife şehrindeki birden çok aracın ve zaman pencerelerinin bulunduğu, sürücülerin iş yükünü dengelerken

aynı zamanda toplam mesafenin en aza indirilmesi istenen gerçek bir sorunu ele almışlardır. Problemi iki amaçlı karma tamsayılı doğrusal model ile açıklayıp, dağınık arama meta sezgisel yöntemine dayanan bir çözüm önermişlerdir. Çözüm yöntemini test etmek için Solomon ZPARP kıyaslama problem örnekleri ve şirket tarafından sağlanan gerçek veriler kullanılmıştır. Sonuçlar önerilen yöntemin şirketin güncel kullandığı araçlardan daha iyi sonuçlar verdiğini göstermiştir.

Küçüköğlü ve Öztürk (2015), makalelerinde ana taşıyıcılı zaman pencereli araç rotalama probleminin çözümü için tavlama benzetimi ve tabu aramasından oluşan hibrit bir meta sezgisel algoritma önermektedirler. Bu problem hat ve ana taşıyıcılar olmak üzere müşterileri iki kümeye ayırmaktadır. Hat taşıma müşterisine depodan çıkan mallar teslim edildikten sonra ana taşıyıcı müşterilerinden mallar iade alınmaktadır. Önerilen yöntem Solomon kıyaslama setlerinden elde edilen problem üzerinde test edilmiş ve mevcut çalışmalarda daha iyi sonuçlar verdiği gözlemlenmiştir.

Yan vd. (2015), çalışmalarında bölünmüş teslimatlı zaman pencereli araç rotalama problemi için araçların çoklu kullanımına izin verildiği problem üzerinde durmuşlardır. Problemi formüle etmek için zaman-uzay ağı tekniğini kullanmışlardır. Formülasyon yan kısıtlamaları olan bir tamsayılı çok ürünlü ağ akış problemi olarak ifade edilebilmektedir. Problemin çözümü için iki aşamalı sezgisel bir algoritma önermişlerdir. Algoritmanın test edilebilmesi için gerçek dünya verileri kullanılmıştır. Sonuçlar, önerilen algoritmanın ilgilenen problemin çözümündeki yüksek potansiyelini göstermiştir.

Wang, Li ve Hu (2015), makalelerinde uyumsuz yükleme kısıtlamalı zaman pencereli heterojen araç rotalama problemini incelemişlerdir. Uyumsuz yükleme kısıtlaması, ürünlerin farklı koşullarda taşınma zorunluluğundan dolayı soğutmalı veya soğutmasız araçlar gibi heterojen araçların kullanılması gerekliliğinden kaynaklanmaktadır. Problemi rota maliyetlerinin en aza indirilmesi amacına uygun bir matematiksel model ile ifade etmişlerdir. Clarke ve wright tasarruf tabanlı yöntem ile başlangıç çözümleri oluşturmuşlardır. Problemin çözümü için yıkma ve yeniden yaratma sezgisel algoritması ve bir eşik tabu arama yöntemi olmak üzere iki farklı yöntem önermişlerdir. Solomon örnekleri ve gerçek bir durum ile önerilen iki yöntemi test etmişlerdir. Sonuçlar problemin çözümü için iki yöntemin de uygulanabilirliğini göstermektedir.

Nalepa ve Blocho (2015), makalelerinde araç sayısının ve toplam kat edilen mesafenin enküçülenmesini amaçlayan ZPARP için uyarlanabilir memetik algoritma önermişlerdir. Algoritmanın parametrelerinin çözüm sırasında dinamik olarak ayarlanması dezavantajını önlemek için yeni bir uyarlanabilir seçim şeması önermişlerdir. Önerilen algoritma Solomon ve Gehring-Homberger kıyaslama örnekleri ile test edilmiştir.

Kırcı (2016), makalesinde Google haritalardan aldığı verilerle gerçek dünyada karşılaşılabileceğimiz farklı senaryolarda ZPARP üzerinde durmuştur. Problemin çözümü için önerilen tabu arama yöntemini hopfield sinir ağı ve tavlama benzetimi ile karşılaştırmıştır.

Anggodo vd. (2016), çalışmalarında gerçek hayat çok yolculuklu zaman pencereli araç rotalama problemini ele almışlardır. Çalışmada Endonezya'nın başkentinde bulunan turistik yerler ve otellerin koordinatları Google Maps uygulamasından alınarak gelen turistler için seyahat rotası oluşturmak hedeflenmiştir. Makalelerinde ARP'ni gezgin problemini çözmek için kullanmışlardır. Ziyaret edilecek yerlerin ve turistlerin kendilerine ait zaman pencereleri vardır. Rota başlangıç-bitiş noktası turistlerin seçeceği otele bağlı olduğundan değişkendir.

Victoria, Afsar ve Prins (2016), makalelerinde doğal afet durumları gibi durumlarda karşımıza çıkan zamana bağlı talepli kapasiteli araç rotalama problemini ele almışlardır. Bu tip durumlarda amaç karşılanan talepleri maksimuma çıkarmaktır. Çözüm için dinamik programlama kullanarak sütun oluşturma tabanlı buluşsal yöntem önermişlerdir. Yöntem Solomon örnekleri ile test edilmiş ve çözüm süresini önemli ölçüde iyileştirdiği gözlemlenmiştir.

Rivera, Afsar ve Prins (2016), çalışmalarında önerdikleri algoritma ile büyük ölçekli kümülatif kapasiteli çok yolculuklu tek araç rotalama problemini kaynak kısıtlı en kısa yol problemine çevirmektedirler. Ortaya çıkan problemi ise bellman-ford algoritması ile çözmüşlerdir. 20 düğümlü küçük örnekler için ise karma tamsayılı doğrusal model, akış tabanlı model ve küme bölümlenme modeli önermektedirler. Sonuçlar önerilen yöntemin 40 düğüme kadar olan örnekleri çözebildiğini göstermiştir.

Kumar ve Panneerselvam (2017), çalışmalarında trafik sıkışıklığı nedeniyle araçların hızlarının zamana bağlı olarak sabit olmadığı ZPARP için genetik algoritma tabanlı bir algoritma geliştirmişlerdir. Hedefleri toplam kat edilen mesafeyi ve kullanılan

araç sayısını en aza indirirken trafik sıkışıklığından kaçınacak yeni rotalar oluşturmaktır. Önerilen algoritma, Solomon kıyaslama örnekleri ile başka bir yerel arama tabanlı algoritma ile karşılaştırılmıştır. Ayrıca küçük boyutlu problem için algoritmanın bulduğu sonuç ile matematiksel modelin Lingo programındaki çözümü ile bulunan sonuç arasında anlamlı bir fark olmadığını gözlemlemiştir.

Zhang vd. (2017), çalışmalarında palet ve zaman penceresi kısıtlamaları ile gerçekçi bir ARP'ni ele almışlardır. Problemin çözümü için tabu araması ve yapay arı kolonisi algoritmasını birleştirerek yeni hibrit bir yöntem kullanmışlardır. Araştırma sonucunda kullanılan araç sayısının fazla olmasının her zaman maliyeti arttırmadığını, uzun sürede daha fazla mesafe kat etmenin bazı durumlarda daha maliyetli olacağını göstermektedir.

Alvarez ve Munari (2017), çalışmalarında verilen hizmet sürelerinin rotaya atanan teslimatçı sayısına bağlı olarak değişen çoklu teslimatçılı zaman pencereli araç rotalama problemi üzerinde durmuşlardır. Yoğun bölgelerde hizmet süreleri seyahat sürelerinden uzun olabilmektedir. Rotadaki teslimatçı sayısı arttıkça hizmet süreleri kısalmaktadır. Makalede, problemin çözümü için dal-fiyat-kes (branch-price-and-cut) algoritmasını iki meta sezgisel yöntem (yinelenen yerel arama ve büyük mahalle araması) ile birleştirerek hibrit bir yöntem kullanmışlardır. Kullanılan iki meta sezgisel algoritma ile kesin yöntemin sonuca daha hızlı ulaşması sağlanmaktadır.

Expósito vd. (2019), çalışmalarında ZPARP'nde müşteri hizmet süresinin azaltılarak hizmet kalitesinin artırılmasını hedeflemişlerdir. Problemin çözümü için ağgözlü rastgele uyarlanabilir arama prosedürünün ve değişken komşuluk aramasının özelliklerini birleştiren hibrit bir meta sezgisel kullanmışlardır. Hizmet kalitesini ölçmek için toplam hizmet süresini ve ağırlıklı bekleme sürelerini minimize eden, müşterilerin zaman pencereleri boyunca hizmet dışı boş zamanlarını maksimize eden amaç fonksiyonları kullanmışlardır. Makalede kullandıkları kıyaslama örneklerini literatürdeki başka bir çalışmadan almışlardır.

Eydi ve Ghasemi-Nezhad (2021), makalelerinde ZPARP için gerçek dünyada müşteri talep değişikliklerini dikkate alarak seyahat sürelerinin en aza indirilmesini ve müşteri memnuniyetinin en üst düzeye çıkarılmasını hedefleyen bir matematiksel model önermişlerdir. Çözüm için baskın olmayan sıralama genetik algoritmasındaki mutasyon operatörü için uygulanan farklı modifikasyonlar ile iki yaklaşım önermişlerdir. Algoritma Solomon test örnekleri ile test edilmiştir.

Ke-Wei, San-Yang ve Xiao-Jun (2022), çalışmalarında amaçladıkları yumuşak zaman pencereli ve zamana bağlı araç rotalama probleminin toplam dağıtım maliyetini en aza indirmektir. Problemin çözümü için arama algoritması ve geliştirilmiş parçacık sürü optimizasyonundan oluşan hibrit bir algoritma önermişlerdir. Literatürdeki standart araç rotalama test problemleri ile parçacık sürü optimizasyonunun etkinliği kıyaslamışlardır. Son olarak, önerilen hibrit algoritmanın uygulanabilirliğini Solomon kıyaslama testleri ile kanıtlamışlardır.

Yesodha ve Amudha (2022), çalışmalarında zaman pencereli çok depolu araç rotalama problemini çözmek için müşteri-depo atama aşamasında atesböceği algoritmasını ve müşteri çizelgeleme aşamasında clarke ve wright algoritmasını kullanarak yeni bir teknik önermişlerdir. Önerilen tekniği Cordeau standart kıyaslama örnekleriyle değerlendirmiş ve tekniğin uygulanabilirliğini kanıtlamışlardır.

Cai vd. (2022), çalışmalarında çok sayıda çok amaçlı ZPARP'nin aralarındaki benzerlikten yararlanarak hepsinin aynı anda optimize edilmesini sağlayan hibrit bir evrimsel çoklu görev algoritması önermişlerdir. Gerçek hayatta bir dağıtım şirketinden aldıkları veriler ile farklı çoklu görev ortamları altında 45 tane kıyaslama testi oluşturmuşlardır. Bu kıyaslamaları; müşteri sayısı, zaman pencereleri, müşteri talebi ve hizmet süresi özelliklerine göre oluşturmuşlardır. Amaç kullanılan araç sayısını, toplam seyahat süresini, en uzun rotanın seyahat süresini, erken varıştan ve geç varıştan kaynaklı bekleme ve gecikme sürelerinin en aza indirilmesidir. İleri sürülen algoritma literatürde son teknoloji olan başka bir algoritma ile karşılaştırılmıştır. Sonuçlar, önerilen algoritmanın yalnızca çok amaçlı ARP'ni aynı anda etkili bir şekilde çözmekle kalmayıp, aynı zamanda tek görevli algoritmalara göre daha iyi performanslar elde ettiğini göstermektedir.

Wang vd. (2023), çalışmalarında zaman pencereli üç boyutlu yükleme kısıtlamalı ve çok noktalı araç probleminde kaynak dağılımları arasında denge sağlamak ve hesaplama karmaşıklığını azaltmak için üç boyutlu k -harmonik ortalama kümeleme algoritmasını önermişlerdir. Problemi formüle edebilmek için araçların ortalama yükleme oranını maksimuma çıkarırken toplam işletme maliyetini en aza indiren iki amaçlı bir karma tamsayı programlama modelini kullanmışlardır. Modelin çözümü için üç boyutlu k -harmonik ortalama kümeleme algoritması ve genişletilmiş baskın olmayan sıralama genetik algoritmasını birleştiren hibrit bir algoritma önermişlerdir. Algoritmanın uygulanabilirliğini test etmek için ise gerçek dünyadan bir vaka çalışması yapmışlardır.

Wu ve Gao (2023), makalelerinde eş zamanlı toplama-teslim ve ZPARP'nin çözümünü için ilk defa karınca koloni optimizasyonun kullanılmasını önermektedirler. Önerilen algoritma Solomon ve Gehring-Homberge test örnekleri üzerinde farklı algoritmalarla karşılaştırılmıştır. Sonuçlar bu problemin çözümü için önerilen algoritmanın kullanılabilir olduğunu kanıtlamıştır.

Su vd. (2023), çalışmalarında personel yeteneklerine göre insan gücü tahsisi ve ZPARP'ni ele almaktadırlar. Problemin çözümü için sundukları dal-fiyat-kes algoritmasını değerlendirmek için yapılan deneylerdeki gereken veri setlerini, Solomon örneklerini değiştirerek ve Hong Kong'daki acil olmayan ambulans taşıma hizmeti senaryosundan üretmişlerdir. İstatistiksel sonuçlar, önerilen algoritmanın önemli ölçüde daha iyi performans ile çalıştığını göstermiştir.

Kumari vd. (2023), çalışmalarında ARP için hibrit bir meta sezgisel çözüm yaklaşımı öne sürmüşlerdir. Öne sürülen çözüm yöntemi genetik algoritmanın ürettiği çözümleri yıkıp yeniden oluşturarak daha da geliştirmiştir. Algoritma literatürdeki kıyaslama örnekleri kullanılarak diğer çözüm yöntemleri ile karşılaştırılmıştır. Sonuç olarak yıkma ve yeniden oluşturma aşaması genetik algoritma tarafından üretilen çözümlerin optimizasyonunu önemli ölçüde arttırmıştır.

Tablo 1.1: Literatür Taraması Özeti

Yazar(lar)	Zaman Penceresi	Amaç Fonksiyonu	Çözüm Metodu	Uygulama Alanı
Koç ve Karaođlan (2012)	✓	Taşıma maliyeti (Enk)	Karma Tamsayılı Doğrusal Programlama	Test problemi
Angel-Bello, Martinez-Salazar ve Alvarez (2013)		Hizmet bekleme süreleri (Enk)	Karma Tamsayılı Doğrusal Programlama	Gerçek hayat problemi
Kumar vd. (2014)	✓	Araç sayısı ve seyahat süreleri (Enk) Rotalar arası dengenin sağlanması	Uygunluk Birleştirilmiş Genetik Algoritma	Test problemi
Melián-Batista vd. (2014)	✓	Toplam mesafe (Enk) Sürücü iş yükü dengelenmesi	Karma Tamsayılı Doğrusal Programlama, Dađınık Arama	Gerçek hayat problemi
Küçükođlu ve Öztürk (2015)	✓	Rota mesafeleri (Enk)	Tavlama Benzetimi Tabu Arama	Test problemi
Yan vd. (2015)	✓	Toplam Maliyet (Enk)	Tamsayılı Doğrusal Programlama, Sezgisel Algoritma	Gerçek hayat problemi

Tablo 1.1: (Devam) Literatür Taraması Özeti

Yazar(lar)	Zaman Penceresi	Amaç Fonksiyonu	Çözüm Metodu	Uygulama Alanı
Wang, Li ve Hu (2015)	✓	Rota maliyetleri (Enk)	Karma Tamsayı Doğrusal Programlama, Tabu Arama, Clark-Wright, Yıkma-Yeniden Yaratma Sezgiseli	Test problemi
Nalepa ve Blocho (2015)	✓	Rota mesafesi (Enk) Araç sayısı (Enk)	Memetik Algoritma	Test problemi
Kırııcı (2016)	✓	Toplam maliyet (Enk)	Tabu Arama, Tavlama Benzetimi	Gerçek hayat problemi
Anggodo vd. (2016)	✓	Toplam mesafe (Enk)	Genetik Algoritma	Gerçek hayat problemi
Victoria, Afsar ve Prins (2016)		Karşılana talepler (Enb)	Dinamik Programlama	Test problemi
Riveraa, Afsar ve Prins (2016)		Varış süreleri (Enk)	Karma Tamsayı Doğrusal Programlama, Bellman-Ford Algoritması	Test problemi
Kumar ve Panneerselvam (2017)	✓	Toplam maliyet (Enk) Araç sayısı (Enk)	Genetik Algoritma	Test problemi
Zhang vd. (2017)	✓	Toplam maliyet (Enk)	Tabu Arama, Yapay Arı Kolonisi Algoritması	Test problemi
Alvarez ve Munari (2017)	✓	Toplam maliyet (Enk)	Dal-Fiyat-Kes Algoritması, Yinelene Yere Arama Ve Büyük Mahalle Araması	Test problemi
Expósito vd. (2019)	✓	Toplam hizmet süresi ve ağırlıklı bekleme süreleri (Enk) Müşterilerin boşa kalma süreleri (Enb)	Karma Tamsayı Doğrusal Programlama, Açgözlü Rastgele Uyarlanabilir Arama, Değişken Komşuluk Araması	Test problemi
Eydi ve Ghasemi-Nezhad (2021)	✓	Seyahat süreleri (Enk)	Baskın Olmayan Sıralama Genetik Algoritma-II	Test problemi
Ke-Wei, San-Yang ve Xiao-Jun (2022)	✓	Toplam maliyet (Enk)	Arama Algoritması ve Parçacık Sürü Optimizasyonu	Test problemi
Yesodha ve Amudha (2022)	✓	Seyahat süreleri (Enk)	Clark-Wright ve Ateş Böceği Algoritması	Test problemi
Cai vd. (2022)	✓	Araç sayısı (Enk) Seyahat süreleri (Enk) Bekleme süreleri (Enk) Gecikme süreleri (Enk)	Evrimsel Çoklu Görev Algoritması	Gerçek hayat problemi

Tablo 1.1: (Devam) Literatür Taraması Özeti

Yazar(lar)	Zaman Penceresi	Amaç Fonksiyonu	Çözüm Metodu	Uygulama Alanı
Wang vd. (2023)	✓	Ortalama yükleme oranı (Enb) Toplam maliyet (Enk)	Genişletilmiş Baskın Olmayan Sıralama Genetik Algoritması-II, Üç Boyutlu K-Harmonik Ortalama Kümeleme Algoritması	Gerçek hayat problemi
Wu ve Gao (2023)	✓	Planlama ve seyahat maliyeti (Enk)	Karınca Koloni Optimizasyonu	Test problemi
Su vd. (2023)	✓	Toplam maliyet (Enk)	Dal-Fiyat-Kes Algoritması	Gerçek hayat problemi
Kumari vd. (2023)		Toplam rotalama maliyeti (Enk)	Genetik Algoritma, Yıkıp Yeniden Oluşturma	Test problemi
Mevcut Çalışma	✓	Toplam mesafe (Enk) Toplam gecikme süresi (Enk)	Genetik Algoritma	Gerçek hayat problemi

Tablo 1.1’de görüldüğü üzere, literatürde zaman pencereci, çok amaçlı, çok kullanımlı araç rotalama problemleri yer alsa da tek bir müşteriye ait birden fazla zaman penceresi bulunması ve her zaman penceresinde müşteriye uğranması gerekliliğine rastlanmamıştır. Ele alınan çalışmalardan sadece 8 tanesi gerçek hayat problemini geri kalan 16 tanesi ise test problemi üzerinde durmuştur. Bu çalışma gerçek hayat problemine uygulanması yönüyle de literatürdeki bu boşluğu dolduracaktır. İncelenen çalışmalardan 2012 yılında Koç ve Karaođlan, 2013 yılında Angel-Bello, Martinez-Salazar ve Alvarez, 2016 yılında ise Riveraa, Afsar ve Prins tek bir aracın çok kullanımını ele almışlardır. Bu çalışma, problemde her müşterinin kendine ait birden fazla zaman penceresi olması, her zaman penceresinde müşteriye uğranılması, her rotaya aynı aracın hizmet vermesi ve aracın gün içerisinde çok sefer yapıyor olması nedeniyle literatürdeki diğer çalışmalardan ayrılmaktadır. ARP ve ZPARP, Np-zor olması sebebiyle çözümü için literatürde en çok genetik algoritma, tabu arama gibi çeşitli sezgisel veya meta sezgisel yöntemler kullanılmıştır (Cömert vd., 2017). Çalışmada ele alınan problem de ARP’nin bir çeşidi olması ve iki amaçlı olması nedeniyle iki amaçlı sezgisel yöntem ile çözülmüştür.



İKİNCİ BÖLÜM
ARAÇ ROTALAMA PROBLEMİ

2.1. ARAÇ ROTALAMA PROBLEMİ

ARP, bir araç kümesi tarafından belirli bir müşteri grubuna hizmet vermek için en uygun rotaların belirlenmesi problemidir. İlk olarak 1959 yılında Dantzig ve Ramser tarafından literatüre tanıtılan bu problem üzerinde en çok çalışılan kombinatoryal optimizasyon problemlerinden biridir. Çalışmalarında benzin istasyonlarına benzin dağıtımına yönelik gerçek dünya uygulamasını ele alarak ilk matematiksel modeli kurmuşlardır. 1964 yılında Clarke ve Wright bu probleme sezgisel bir çözüm önermişlerdir. Bu iki çalışmadan günümüze kadar ARP'ye olan ilgi giderek artmıştır (Toth ve Vigo, 2002). Bununla beraber ARP'nin literatürde en çok uygulandığı alanlar aşağıdaki gibidir:

- Malzemelerin belirli bir veya daha fazla merkezden müşterilere dağıtımında (ulaşım ve lojistik sektöründeki uygulamalar),
- Dağıtım ve toplama problemlerinde,
- Okul taşıt güzergahlarının, uçak rotalarının belirlenmesinde,
- Devriye araçlarının rotalamasında,
- Atıkların toplanmasında,
- Stok alanlarındaki malzemelerin toplanmasında.

ARP genel olarak araçların merkezi bir depodan başlayarak tüm müşterilerin taleplerini karşılayacak toplam uzunluğu en kısa rotaların oluşturulması problemidir. Her müşteriye mutlaka hizmet verilmelidir. Rotalar belirlenirken araç ve müşteriden kaynaklı veya diğer kısıtlarda göz önüne alınmaktadır. Çözüme geçmeden önce müşterilerin birbirlerine ve depoya olan uzaklıkları, müşteri talepleri, araçların kapasitesi ve müşteri hizmet süreleri gibi çeşitli parametreler belirlenmelidir. Bu parametreler doğrultusunda ARP'de çoğunlukla kullanılan amaç çeşitleri;

- Toplam seyahat süresini enküçükleme,
- Toplam rotalama maliyetini enküçükleme,
- Araç yükleme oranının enbüyükleme,
- Toplam işletme maliyetinin enküçükleme,
- Toplam rota mesafelerinin enküçükleme,

- Kullanılan araç sayısının enküçükleme,
- Karşılanan talepleri enbüyükme,
- Toplam karı enbüyükme,
- Rotalar arası dengeyi sağlamak,
- Toplam karbon salınımını enküçükleme,

şeklinde olmaktadır. Problem yapısına göre amaçlar çeşitlenmekte ve birden fazla amacın aynı anda eniyilenmesi durumları oluşabilmektedir.

ARP'ye ilişkin çeşitli kısıtlar aşağıdaki şekilde olabilmektedir:

- Araç kapasitesi,
- Toplam rota tamamlanma zamanı,
- Araç sayısı veya tipi,
- Talebin bölünüp bölünemeyeceği,
- Zaman penceresi kısıtı,
- Tek veya çok depo bulunması,
- Aynı aracın gün içerisinde birden fazla tur yapıyor olması,
- Dağıtım ve/veya toplama işleminin gerçekleştirilmesi,
- Zamana bağlı olarak yolculuk sürelerinin değişmesi.

Bu kısıtlar literatürde genellikle karşılaşılan kısıtlar olup, problemin tipine göre ilave kısıtlar modele eklenebilmektedir. Araç rotalama problemi, Np-zor bir problem olarak bilinmektedir (Cömert vd., 2017).

2.1.1. Araç Rotalama Problem Çeşitleri

Araç rotalama problemleri gerçek hayatta karşılaşılan bazı özel durum ve kısıtlardan dolayı çeşitli dallara ayrılmıştır. Çıkış zamanlarına göre literatürde bilinen bazı ARP çeşitleri izleyen kısımda verilmiştir.

Kapasite Kısıtlı ARP (Capacitated VRP): Araçlara kapasite kısıtlamalarının uygulandığı problem türüdür. Problemden kullanılacak araçlar tek tip (homojen araç filosu) veya farklı tip (heterojen araç filosu) kapasiteye sahip olabilirler. Ayrıca kapasite

kısıtlı ARP uzaklık matrisindeki özelliklere göre simetrik veya asimetrik olarak ikiye ayrılabilir (Oosthuizen, 2018). Literatürde yer alan problemlerin çoğuna araç kapasite kısıtı uygulanmakta, hatta ARP'den bahsedildiğinde ayrıca bir araç kapasite kısıtı uygulandığına dair ibare beklenmemektedir.

Zamana Bağlı ARP (Time-dependent VRP): ARP'de müşteriler arası veya müşteri ve depo arası Öklid uzaklığı sabit ve belirlidir. Ancak gerçek hayatta bazı durumlarda bu uzaklıklar araçların seyahat ettiği yollarda oluşan doğal afet ve trafik sıkışıklığı gibi nedenlerden dolayı zamana bağlı olarak değişmektedir. Dğümler arası uzaklıkların zamana bağlı olarak farklılık gösterdiği ARP çeşidi zamana bağlı ARP olarak isimlendirilmektedir. Bu problem tipine ilk olarak 1966'da Cooke ve Halsey'nin çalışmalarında rastlanmıştır. Cooke ve Halsey dğümler arası belirli süreye sahip en kısa yol problemini dğümler arası değişken süreyi dikkate alacak şekilde genişletmişlerdir (Sung vd., 2000).

Toplamalı ve Dağıtım ARP (Pickup and Delivery VRP): İlk olarak 1967'de Wilson ve Weissberg'in çalışmaları ile başladığı kabul edilmektedir. Bu ARP türünde müşterilere dağıtım yapmakla beraber müşterilerden toplamada yapılmaktadır. Bu problem türünün alt dallarından olan "Geri Toplamalı Dağıtım Probleminde (VRP with Backhauls)" tüm teslimatlar depoda yüklenmeli ve tüm alımlar depoya taşınmalıdır. Bu problem tipi toplama ve dağıtım yapma şekline göre 4 sınıfa ayrılmaktadır. Birinci sınıfta ilk önce dağıtım yapılacak müşteriler (Linehaul) sonra toplama yapılacak müşteriler (Backhaul) ziyaret edilmektedir. İkinci sınıf, dağıtım veya toplama yapılacak müşterilerin aynı rotada ziyaret edilmesine izin vermektedir. Üçüncü sınıfta hem dağıtım hem toplama yapılacak müşteriler iki kez ziyaret edilebilir. Dördüncü sınıfta, her iki hizmeti de talep eden müşteriler sadece bir kez ziyaret edilebilmektedir (Koç ve Laporte, 2018). Üçüncü ve dördüncü sınıfta aynı müşteriye hem malzeme teslimi hem de bazı malzemelerin geri alınması söz konusudur.

Çok Depolu ARP (Multi Depot VRP): İlk olarak 1969 yılında Tillman tarafından incelenen bu problem birden fazla depo içermektedir (Lin vd., 2014). Her müşteri bu depolardan birine atanmış araç tarafından ziyaret edilmekte ve araçlar başlangıç depolarına geri dönmektedir.

Stokastik ARP (Stochastic VRP): Müşteri talepleri, seyahat süreleri ve hizmet süreleri gibi bazı parametrelerin rassal olduğu bu ARP çeşidi ilk olarak 1969 yılında tanımlanmıştır (Gutierrez vd., 2018).

Dinamik ARP (Dynamic VRP): Geleneksel ARP’de tüm parametreler rotalama süreci başlamadan bilinmekte ve rotalama süreci boyunca sabit kalmaktadır. Dinamik ARP, gerçek hayat problemlerinde araç arızaları, trafik kontrolleri ve sürekli gelen müşteri talepleri gibi bazı durumlarda parametrelerin değişkenlik göstermesi ve bu değişikliğe göre rotalama işleminin yeniden düzenlenmesidir. İlk olarak 1976 yılında tanımlanmıştır (Lin vd., 2014).

Zaman Pencereci ARP (VRP with Time Windows): Zaman pencereci araç rotalama problemi, araçlara kapasite kısıtlamalarının yanı sıra müşterilere zaman penceresi kısıtlamaları uygulamaktadır. Ayrıca araçların müşterilere verdiği hizmet süreside dikkate alınmaktadır. İlk olarak 1977 yılında Russel tarafından tanımlanmış ve literatürde üzerinde durulmuş bir problemdir. Tez kapsamında bu problem türü ile ilgili Bölüm 2.1.2’de daha detaylı bilgi verilmiştir.

Çok Kullanımlı ARP (Multiple Use VRP): İlk kez 1987 yılında ortaya atılan çok kullanımlı ARP araçların birden fazla rotada kullanılmasına izin vermektedir. Araç sayısının arttırılamayacağı ve tüm müşterilere tek bir rotada hizmet vermenin olanaksız olduğu durumlarda araç başına düşen sefer sayısı arttırılmaktadır. Bununla beraber araç aynı anda iki farklı rotada kullanılmasına izin vermeyen kısıt eklenmektedir (Koç ve Karaoğlan, 2012).

Bölünmüş Teslimat ARP (Split Delivery VRP): ARP’de çoğu zaman her müşteriye bir kez uğrama kısıtı uygulanmaktadır. Fakat müşterinin malzeme talebi aracın bir kez uğramasındaki taşıyacağı malzeme miktarından fazla olduğu durumlarda bu kısıt gevşetilmelidir. Ele alınan bu genişletme bölünmüş teslimat ARP olarak isimlendirilmektedir. Böylelikle müşteri birden fazla araç tarafından hizmet alabilmektedir. İlk olarak 1989 yılında Dror ve Trudeau tarafından literatüre tanıtılmıştır.

Açık ARP (Open VRP): İlk olarak 2000 yılında Sariklis ve Powell tarafından tanıtılan ARP’nin bu tipinde araçların müşterilere hizmet verdikten sonra depoya geri dönme zorunluluğu yoktur (Tarantilis vd., 2004). Genellikle gazete veya posta dağıtım hizmetlerinde bu problemle karşılaşılmaktadır. Bu problem malzemelerin teslimatı esnasında üçüncü bir teslimat sağlayıcı dış kaynak kullanımlarında ortaya çıkmaktadır. Dış kaynakların depoya dönme zorunluluğu yoktur.

Yeşil ARP (Green VRP): Geleneksel ARP çeşitlerinde rotalama işlemi esnasında çevre ve enerji tüketimine ilişkin kriterler göz ardı edilmektedir. Yeşil ARP’de bu

kriterler diğer kriterlere göre daha önceliklidir. Teknolojik gelişmelerle birlikte bilim insanları, yeni enerji kaynaklarını arařtırmalarının temeli olarak küresel emisyonları düşürmeye yoğunlaşmaktadırlar. Bu şekilde, klasik ARP'nin bir uzantısı olarak yeşil ARP 2012 yılında Erdoğan ve Miller-Hooks tarafından ortaya atılmıştır. Yeni bir araştırma alanı olan yeşil ARP rotalama sonucunda enerji tüketimlerini azaltmayı hedeflemektedir. Bu nedenle mevcuttaki içten yanmalı araçlarla beraber veya yalnızca alternatif yakıtlı araçların kullanılmasını benimsemektedir (Asghari ve Al-e, 2021).

Bu araç rotalama problemlerinin dışında literatürde aşağıda verilen araç rotalama problemleri de mevcuttur (Lin vd., 2014):

- Yer Belirleme ve Rotalama Problemi (Location Routing Problem)
- Periyodik ARP (Periodic VRP)
- Envanter Rotalama Problemi (Inventory Routing Problem)
- Filo Boyutu ve Filo Kombinasyonu Belirlemeli ARP (Fleet Size and Mix VRP)
- Genelleştirilmiş ARP (Generalized VRP)
- Çok Kısımlı ARP (Multi-compartment VRP)
- Küme Bağımlı ARP (Site-dependent VRP)
- Bulanık ARP (Fuzzy VRP)
- Yükleme Kısıtlı ARP (VRP with Loading Constraints)
- Çok Aşamalı ARP (Multi-echelon VRP)

2.1.2. Zaman Pencerele Araç Rotalama Problemi

Araçlara kapasite kısıtlamalarının yanı sıra müşterilere zaman penceresi kısıtları uygulayan ZPARP, literatüre ilk kez 1977 yılında Russel tarafından tanımlanmıştır. ZPARP literatürde yaygın olarak ele alınan araç rotalama problemidir. Bu problemde herhangi bir müşteriye hizmet, belirli bir zaman aralığı içinde başlar ve bu zaman aralığına zaman penceresi denir (Kallehauge vd., 2005). Genel olarak literatürde yaygın olarak incelenen iki tür zaman penceresi vardır:

- *Sıkı Zaman Penceresi (Hard Time Windows)*: Araçın müşterinin zaman penceresi aralığından önce veya tam zamanında hizmet vermeye hazır olmasını kapsayan durumdur. Araç, müşteriye erken varması durumunda ilgili müşterinin

zaman penceresi başlangıç zamanına kadar beklemek zorundadır. Aracın geç kalınmasına izin verilmemektedir.

- *Esnek Zaman Penceresi (Soft Time Window)*: Müşteri zaman pencerelerinin belirli bir ceza maliyeti karşılığında ihlal edilebildiği durumdur. Araç, müşteriye zaman penceresi dışında hizmet verdiği takdirde ceza maliyetine katlanılmak durumundadır. Esnek zaman penceresinin bir çeşidi olan yarı esnek zaman penceresi durumunda ceza maliyeti sadece geç varışlarda uygulanmakta ve aracın müşteriye erken gitme durumunda herhangi bir ceza maliyeti uygulanmamaktadır. Bu çalışmada ele alınan problem yarı esnek ZPARP'dir.

Gerçek hayat problemlerinde tüm zaman penceresi kısıtlarının sağlanması gerektiğinde sıkı zaman penceresi kısıtı ile uygulanabilir bir çözüm elde edilemeyebilir. Bu sıkı zaman penceresi kısıtının gevşetilmesi toplam mesafe ya da kullanılan araç sayısı açısından daha iyi çözüm elde edilmesini sağlayabilmektedir. Gerçek hayatta müşterinin belirlediği zaman diliminden küçük bir sapma, müşteri memnuniyetini önemli ölçüde etkilemeden maliyeti düşürebilmektedir.

Şirketlerin tedarik zinciri boyunca verimlilik ve zamanlamaya odaklanmaları ile tedarik zinciri tasarımı ve operasyonların çeşitli yönlerini modellemede ZPARP önemli bir araç haline gelmiştir. ZPARP'nin önemli uygulamaları arasında süpermarket teslimatları, banka ve posta teslimatları, endüstriyel atık toplama, okul otobüsü rotalaması, güvenlik devriye hizmetleri ve kentsel gazete dağıtımı yer almaktadır (Kallehauge vd., 2005).

2.2. MİLK-RUN VE SU ÖRÜMCEĞİ

Çalışmada ele alınan problemde kullanılan araç, su örümceğidir. Su örümceği, yalın üretim sisteminin bir parçası olan milk-run taşıma sisteminde taşımayı gerçekleştiren bir araçtır. Yalın üretim; israf, memnuniyetsizlik, hata ve fire unsurlarının en aza indirildiği üretim sistemi olarak tanımlanabilmektedir (Şeker, 2016). Yalın üretim sisteminde üretimdeki her bir faaliyet, katma değerli, zorunlu fakat katma değersiz ve katma değersiz faaliyetler olarak sınıflandırılmaktadır (Ömürgönülşen ve Çatman, 2018). Katma değersiz olan faaliyetler müşterinin bedel ödemeyeceği ve israf olarak nitelendirilen faaliyetlerdir. Bu israflar; hatalı üretim, gereksiz işlemler, aşırı üretim, her türlü fazla stok, gereksiz bekleme, hareket ve taşıma olmak üzere yedi tanedir. Son zamanlarda çalışanların beyinlerinden yararlanmamakta yalın üretimde bir israf

olarak görülmektedir. Ayrıca yalın üretim değer, değer akışı, akış, çekme, mükemmellik olmak üzere beş temel ilke üzerine kurulmuştur (Ersöz T., Sarız ve Ersöz F., 2020).

Değer, ürün ya da hizmet olarak müşteriye doğru zamanda, uygun fiyattan ulaştırılan ve müşteri tarafından tanımlanan faaliyetlerdir.

Değer akışı, üretim sürecindeki HM'den son ürüne kadar ürünün geçtiği tüm adımlardır. Bu adımları incelemek için kullanılan metot değer akış haritalamadır. Haritalandırma sonucunda ürüne değer katmayan faaliyetler tespit edilerek bu faaliyetlerin minimize edilmektedir.

Sürekli akış, ürünün planlama aşamasında müşteriye ulaşmasına kadar geçtiği aşamalarda kesintiye uğramamasıdır. Sürekli akışın sağlanabilmesi tek parça akışına bağlıdır. Tek parça akış parti büyüklüğü bir olacak şekilde, ürünlerin birbirini beklemeden akmasıdır.

Çekme sistemi, üretimin müşteri siparişine göre başlamasını sağlamaktadır. Çekme sisteminde üretim, müşteriden geriye doğru ürünün üretilmesi için gereken tüm istasyonlardan geçerek sipariş noktasına gelene kadar ters yönlü olarak tetiklenmektedir.

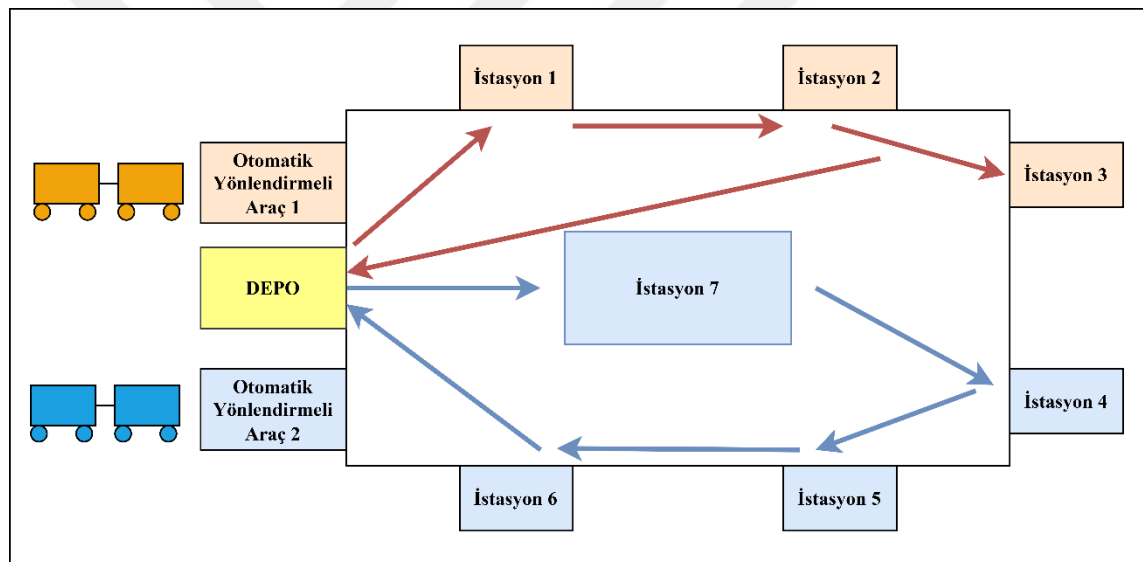
Mükemmellik, yalın üretim için sonu gelmeyen bir yolculuktur. Diğer yalın üretim ilkeleri uygulanarak sistem iyileşmeye başlamaktadır. Fakat sistemin her parçası daha iyisi için çabalamaya devam etmelidir.

Yalın üretim, yukarıda bahsedilen ilkeleri gerçekleştirebilmek ve israfları ortadan kaldırmak için çeşitli araçlar kullanmaktadır. Bekleme, taşıma, fazla ve hatalı üretim gibi israfların ortadan kaldırılıp sürekli akışın sağlanması için ürünlerin küçük partiler halinde üretilip o şekilde üretim alanına taşınması gerekmektedir (Nemoto, Hayashi ve Hashimoto, 2010). Yalın üretimde malzeme veya ürünlerin belirli bir döngüde taşınması ve dağıtılması veya toplanması *milk-run* olarak adlandırılmaktadır (Sipahioğlu ve Altın, 2019) Milk-run, üretim alanı ve depo arasında ise *tesis içi milk-run*, fabrika ve tedarikçi veya müşteri arasında ise *tesis dışı milk-run* olarak tanımlanmaktadır. Tesis içi milk-run, malzemelerin depolama alanlarından kısa aralıklarla ve küçük partiler halinde kullanım alanlarına temin edilmesidir (Klenk, Galka ve Günthner, 2015). Tesis içi milk-run sağlayan yalın aracı su örümceği olarak isimlendirilmektedir. Su örümceği terimi, Japonca'da dönen böcek anlamına gelen "*Mizusumashi*" kelimesinden esinlenmiştir (Yalın'da su örümceği nedir?, 2020). Mizusumashi su ve hava arasında asılı duruyor gibi görünmekte, bir yerden diğerine hızla hareket ederken aynı anda her iki ortamı da görüş

alanında tutmaktadır. Benzer şekilde bir üretim ortamındaki su örümceği, üretim alanında hızla hareket eden ve tüm üretim adımlarını aynı anda izleyen bir stok yöneticisi olup endüstriyel üretimde ürünlerin taşınması ve dağıtılması için kullanılan yalın üretim aracıdır.

Su örümceği, belirli zaman aralıklarında dağıtacağı malzemeleri depodan taşıma aracına yükleyerek önceden belirlenmiş rota üzerindeki istasyonlara dağıtarak tekrar depoya dönmektedir (Klenk, Galka ve Günthner, 2015). Su örümceği için çoğunlukla basit kullanımı olan araçlar tercih edilmektedir. Fakat büyük hacimli malzemeler için daha büyük araçlar tercih edilebilmektedir. Milk-run sistemindeki bu araç bir su örümceği tarafından kullanılabilir gibi uzaktan kontrollü ve otomatik yönlendirmeli olarak da kullanılabilir.

Şekil 2.1: Fabrika İçi Milk-run Sistemi



Kaynak: (Sipahioğlu ve Altın, 2019).

Şekil 2.1 fabrika içinde 2 farklı su örümceğinin bulunduğu durum için örnektir. Bu örnekte milk-run otomatik yönlendirilmiş araçlar tarafından gerçekleştirilmektedir. Üretim alanlarında operatörlerin yeni HM veya YM tedariki için istasyonlarını terk etmeleri gerekebilmektedir. Bu durum süreçte bekleme ve gereksiz hareketlere yol açmaktadır. Su örümceği, istasyonların gerekli malzemelerinin bulunduğu küçük bir araçla malzeme taşıyarak bu sorunun çözülmesine yardımcı olmaktadır.

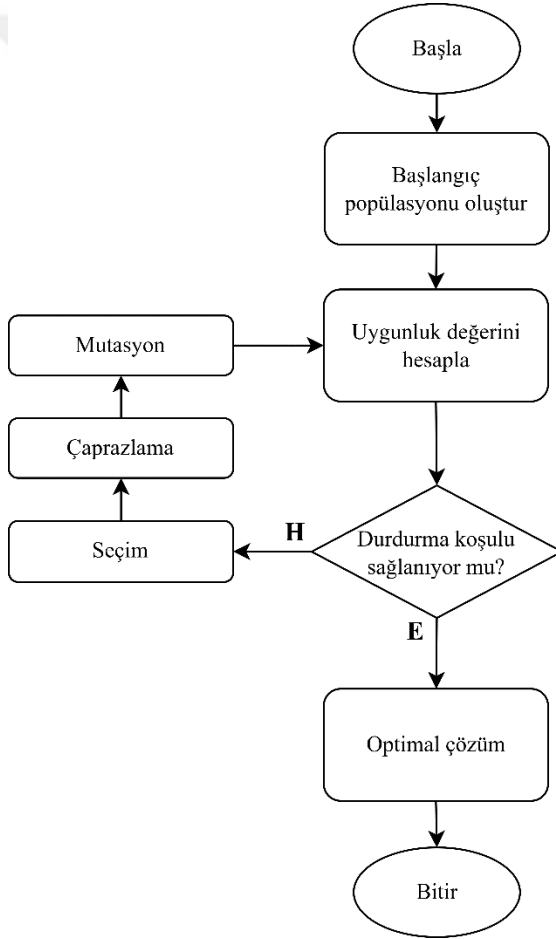


ÜÇÜNCÜ BÖLÜM
GENETİK ALGORİTMA

3.1. GENETİK ALGORİTMA

Günümüzde büyük boyutlu problemlerin çözümü için kısa sürede çözüm üreten pratik çözüm metotlarına ihtiyaç vardır. Bu çözüm metotları en iyi çözümü garantilememekle birlikte uygun veya yaklaşık çözümler üretmektedirler. Yaklaşık algoritma örneği olan genetik algoritma, ilk kez mühendis ve biyolog olan John H. Hollond tarafından 1975 yılında Michigan Üniversitesi'nde ortaya atılmıştır (Katoch, Chauhan ve Kumar, 2021). Algoritma doğadaki en uygunların hayatta kalma teorisi olan Darwin'in teorisini taklit etmektedir. Temel bileşenleri, kromozom temsili, uygunluk seçimi ve biyolojiye dayanan operatörlerdir (A.g.e).

Şekil 3.1: Genetik Algoritma Akış Diyagramı



Kaynak: (Elen, 2014).

Şekil 3.1 genetik algoritma için akış diyagramını göstermektedir. Genetik algoritma karmaşık optimizasyon problemleri için çeşitli çözümler üretmekte ve bu çözümleri uygunluk değerine göre seçmektedir. Algoritma her seferinde birden fazla çözüm üreterek en iyiler arasından seçim yapmaktadır. Seçilen çözümlere bir diğer adıyla

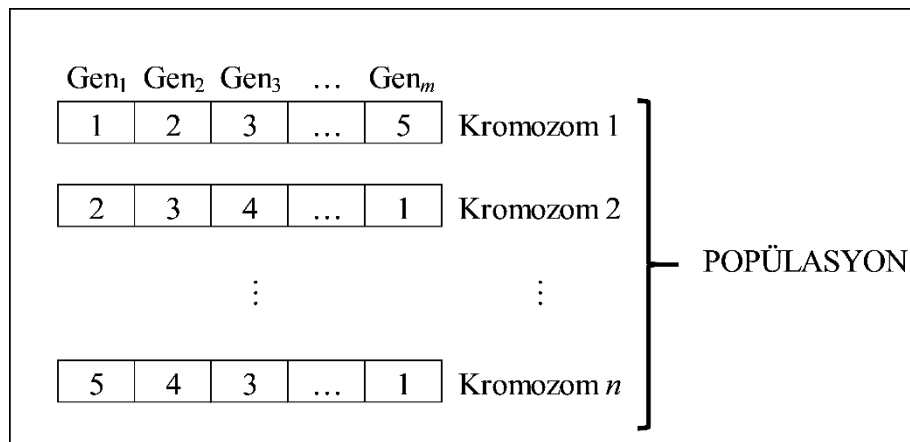
kromozomlara mutasyon ve çaprazlama operatörleri uygulayarak popülasyon çeşitliliğini korumakta ve çözümlerin yerel optimuma takılmasını önlenmektedir. Durdurma koşulu sağlanıncaya kadar döngü devam etmektedir. Bu çalışmada problemin çözülebilmesi için genetik algorithmadan yararlanılmıştır.

3.1.1. Genetik Algoritma Çalışma Prensipleri

Parametre Belirleme: Genetik algoritma parametrelerden çok parametrelerin kodlanması ile çalışmaktadır (Sastry, Goldberg ve Kendall,2005). Algoritma başlatılmadan önce popülasyon büyüklüğü, mutasyon oranı, oluşturulacak kromozom sayısı gibi parametreler belirlenir. Kromozomlar belirlenen parametreler doğrultusunda oluşturulmaktadır.

Başlangıç Popülasyonunun Oluşturulması: Başlangıç popülasyonu arama uzayında rastgele veya sezgisel yöntemlerle oluşturulan kromozom çözümleridir. Popülasyon büyüklüğü arttıkça algoritmanın çözüm süresi uzamakla beraber sonuç verme olasılığı da artmaktadır. Popülasyondaki her kromozom çözüm için bir adaydır. Sezgisel yöntemlerle oluşturulan başlangıç kromozom çözümlerinin rastgele oluşturulan başlangıç kromozom çözümlerinden daha iyi olması beklenmektedir. Kromozom gösterimi probleme göre ikili, sıralı gösterim gibi değişiklik göstermektedir (Çolak, 2010). Kromozomların her biri genlerden oluşmaktadır. Genler kromozom içinde bilgi taşıyan en küçük yapılardır. Genetik algorithmadaki en önemli adım gen ve kromozom yapılarını iyi kodlamaktır. Şekil 3.2 gen, kromozom ve popülasyon yapısını açıklamaktadır.

Şekil 3.2: Gen, Kromozom ve Popülasyon Yapısı



Kaynak: (Tunç, 2019).

Uygunluk Değerinin Hesaplanması: Genetik algoritmada oluşturulan çözümlerin değerlendirilebilmesi için matematiksel bir fonksiyon gereklidir. Başlangıç popülasyonu oluşturulduktan sonra belirlenen uygunluk değeri fonksiyonuna göre başlangıç kromozomlarının uygunluğu hesaplanmaktadır. Başlangıç popülasyonundan sonra oluşturulan her yeni popülasyondaki kromozomların da uygunluk değeri hesaplanmaktadır.

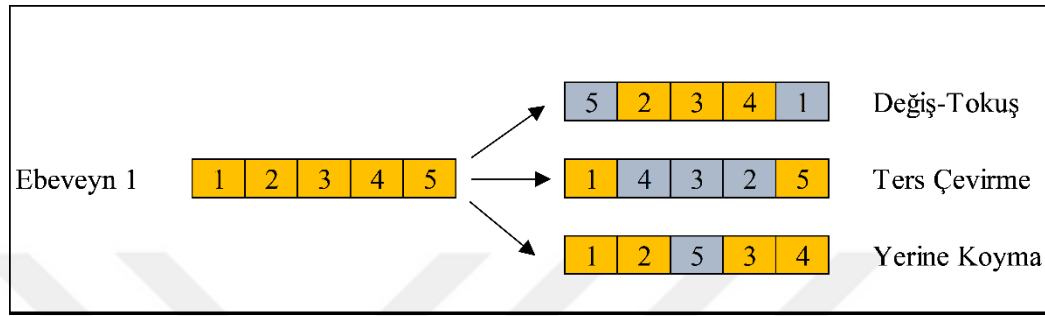
Seçim: Yeni kromozomların oluşturulabilmesi için popülasyondan iki kromozom seçilerek bu kromozomların çaprazlanması gerekmektedir. Seçimin ana amacı iyi olan çözüm kromozomlarının kötü olanlara tercih edilmesidir (Yadav ve Sohal, 2017). Ebeveyn kromozomların seçilmesi için rulet tekeri, sıralama ve turnuva gibi çeşitli yöntemler kullanılmaktadır. Rulet tekeri seçiminde kromozomun seçilme olasılığı kendi uygunluk fonksiyonu değerinin toplam uygunluk fonksiyonu değerine oranı kadardır. Sıralama yönteminde her kromozom uygunluk değerine göre sıralanır ve sıralı kromozomlardan ebeveynler rulet çemberi ile seçilmektedir. Turnuva yönteminde ise rastgele kromozomlar alınarak bunlardan en iyi uygunluk değerine sahip olan kromozom 1.ebeveyn olarak seçilmektedir. Aynı yöntem 2.ebeveyn seçilirken de kullanılmaktadır (Anand, Afreen ve Yazdani, 2015). Seçim sırasında oluşturulan en iyi kromozomların gelecek popülasyona aktarılabilmesi için elitizm operatörü uygulanmaktadır. Algoritmanın hızlı yakınsama yapmaması için seçim en önemli adımdır.

Çaprazlama: Algoritma daha iyi sonuçlar bulabilmek için seçilen iki veya daha fazla ebeveyni çaprazlayarak yeni yavrular oluşturmaktadır. Çaprazlama ebeveynler arasında gen değişimi ile yapılmaktadır. Çaprazlama operatörü oluşturulan kromozom yapısına göre seçilmektedir. Literatürde tek nokta çaprazlama, iki nokta çaprazlama, sıraya dayalı çaprazlama gibi birçok çaprazlama yöntemi bulunmaktadır (Pavai ve Geetha, 2016). Çaprazlama olasılığı genelde kullanılan çaprazlama yöntemine de bağlı olarak 0.6 ile 1.0 arasında belirlenmektedir. Çaprazlama olasılığının artırılması genetik çeşitliliği artırırken mevcut olan iyi kromozom yapılarının bozulmasına da neden olabilmektedir. Çaprazlama sonucu oluşan yavrular ebeveynleri ile aynı olmamakla beraber ebeveyne ait özellikleri birleştirmektedir (Sastry, Goldberg ve Kendall, 2005).

Mutasyon: Çaprazlama operatörü birden fazla kromozom üzerinde çalışırken mutasyon operatörü çeşitli kromozomlarda tek çalışmaktadır (Pavai ve Geetha, 2016). Mutasyon operatörü çaprazlama sonucu oluşan yavruların kromozom üzerindeki genlerinden bir veya birkaçını rastgele değiştirmektedir. Mutasyon operatörü belirlenen

mutasyon oranına göre uygulanmaktadır. Oluşan bütün yavru kromozomlara uygulanmamaktadır. Mutasyon oranının 0,01 gibi düşük tutulması önerilmektedir (Çolak, 2010). Literatürde, deęiş-tokuş (swap), ters çevirme, yerine koyma gibi çeşitli mutasyon yöntemleri bulunmaktadır. Bu mutasyon yöntemlerinin uygulanaşı Şekil 3.3'te verilmiştir.

Şekil 3.3: Mutasyon Örnekleri



3.2. PARAMETRE BELİRLEME

Literatürde yapılan çalışmalar algoritmaların performansının kullanılan parametrelere baęlı olduklarını göstermektedir. Doğru parametre seçimi algoritmaların performanslarını arttırmaktadır (Yavuz, 2022). Parametre belirleme yani algoritmanın parametre uzayında en iyi sonucu vereceęi parametrelerin kullanılması problemi iki duruma ayrılmaktadır. Bunlardan biri parametre ayarlama deęeri ise parametre kontrolüdür. Parametre ayarlama yönteminde algoritma çalıştırılmadan en uygun parametre deęeri evrensel bir yöntemle göre ayarlanmakta ve bu deęerler algoritma çalışırken deęişmemektedir. Parametre ayarlama evrensel bir yöntem kullanıldığından dolayı bir parametre ayarlama yöntemi deęer bir algoritmada da kullanılabilir. Dięer bir parametre belirleme yöntemi olan parametre kontrolü ise algoritma çalışırken parametrelerin deęeri deęiştirilerek en uygun parametre deęerlerinin belirlenmesidir. Bu yöntem için parametrelerin nasıl deęiştirileceęi konusunda bilgi sahibi olmak gerekmektedir (Huang, Li ve Yao, 2020). Bu çalışmada parametre ayarlama yöntemi olarak **irace** ve tam faktöriyel deney tasarımı (DoE) kullanılmıştır. Elde edilen parametreler ile ele alınan problem çözülerek sonuçlar kıyaslanmış ve DoE sonucu üretilen parametreler kullanılarak elde edilen sonuçların **irace**'den daha iyi olduęu gözlemlenmiştir. Bu parametre ayarları ile kısa bir bilgilendirme ilerleyen alt başlıklarda verilmiştir.

3.2.1. Otomatik Parametre Ayarlama (Iterated F-Race Algorithm)

Parametre belirleme yöntemlerinden biri olan geleneksel yöntemde parametrelerin belirlenebilmesi için tek tek deneme tercih edilse de bu yöntem hem zaman almakta hem de kişiye bağlı olmaktadır. Bu nedenle otomatik parametre ayarlama yöntemleri kullanılmaktadır. Otomatik parametre ayarlama yöntemlerinden bazıları şu şekildedir:

- Parametre Yinelemeli Yerel Arama
- Cinsiyete Dayalı Genetik Algoritma
- Sıralı Model Tabanlı Algoritma yapılandırması (SMAC)
- Yinelenen F-Yarış Algoritması (**irace**)

Bu çalışmada parametre belirlemek için otomatik parametre ayarlama yöntemlerinden biri **irace**'tir.

irace: F-Race yöntemi aday algoritma yapılandırmalarının arasında optimal yapılandırmanın seçilmesi için kullanılan bir yöntemdir. Aday algoritma yapılandırmalarını bir örnek üzerinde ayrı ayrı kademeli olarak değerlendirerek kötü sonuç veren yapılandırmaları eşleştirmektedir. Erkenden eşleştirilen yapılandırmalar sayesinde yöntem daha iyi yapılandırmalar üzerine yoğunlaşabilmektedir. **irace**, optimal parametre değerlerini bulurken yinelemeli bir prosedür kullanmaktadır. Aday yapılandırmaları her yinelemede bir olasılık modeline göre oluşturmaktadır. Sonrasında bu aday yapılandırmalara F-Race yöntemini uygulamaktadır. Hayatta kalan aday yapılandırmalar olasılık modelini güncellemek için kullanılmaktadır. Böylelikle aday yapılandırmaların daha optimal bölgelere yoğunlaşması sağlanarak arama prosedürünün verimliliğinin artırılması hedeflenmektedir. Bu döngü yapılandırma bütçesi gibi bir durdurma koşulu sağlanıncaya kadar devam etmektedir (Huang, Li ve Yao, 2020).

irace yönteminde yapılandırılabilir her parametrenin kısıtlamalar ve koşullar dışında kendine ait örnekleme dağılımı vardır ve bu diğer parametrelerin örnekleme dağılımlarından bağımsızdır. Bu her parametrenin bağımsız bir şekilde değerlendirilmesine olanak tanımaktadır (López-Ibáñez, vd., 2016). **irace** yöntemi ile alakalı literatürde yapılan çalışmalar çalışmanın devamında verilmiştir.

Massen vd. (2013), karınca kolonisi optimizasyonu ve karma tamsayılı matematiksel modeli birleştiren feromon tabanlı sezgisel sütun üretimi ARP'leri kara

kutu uygulanabilirliği ile ele alan hibrit bir algoritma kullanmışlardır. Algoritmanın deneysel analizi çok sayıda parametre ve her algoritma çalışması için gereken zaman nedeniyle tam faktöriyel istatistiksel analiz ile gerçekleştirilemez. Bu nedenle algoritmanın parametrelerini otomatik parametre belirleme yöntemi **irace** ile belirlemişlerdir. Parametreler belirlendikten sonra her bir parametre için de hassasiyet analizi yapılmıştır.

Kalayci (2016), çalışmasında elemeli yarış prosedürünü uygulayan F-race'in bir uzantısı olan **irace** ile parametre değerleri ayarlanmış reaktif arama optimizasyonunu, GENOPT fonksiyonlarını optimize etmek için kullanmışlardır. Sonuçlar çoğunlukla daha yüksek atama puanlarına ulaştıklarını gösterse bile gönderimlerde hedefe atış puanlarının düşük olduğunu göstermektedir.

López-Ibáñez vd. (2016) çalışmalarında yinelemeli yarış prosedürlerinin altında yatan mantığı açıklayarak bir dizi otomatik yapılandırma prosedürünü uygulayan bir yazılım paketi olan **irace**'in özel ve genel olarak otomatik algoritma yapılandırmasındaki kullanımını ve potansiyelini göstermişlerdir. Otomatik algoritma yapılandırmasını iki örnek üzerinde anlatmışlardır. **irace**'in mevcut sürümünde hızlı yakınsamayı önlemek için yeniden başlatma mekanizmasını ve benzer yapılandırma oluşturmaması için parametreler arası etkileşimi hesaba katarak kesik örnekleme dağılımlarının kullanılmasını sağlayan araçlar önermişlerdir.

Souza ve Ritt (2016), kısıtsız ikinci dereceden programlama probleminin çözümünde otomatik yöntemleri ve manuel ayarlama sürecini karşılaştırmışlardır. Çalışmalarında sekiz sezgisel algoritma üretebilen ve bu yöntemlerin parametrelerini de içeren bir metot önermişlerdir. Önerilen metodun en iyi algoritma ve parametre değerlerini bulmak için yinelenen F-race yöntemini kullanmışlardır. Otomatik yöntem için 2000 bütçe ile bir yinelemeli F-race gerçekleştirmişlerdir ve deneyler için GNU R paketinde bulunan **irace**'i kullanmışlardır. Sonuçlar hem otomatik hem de manuel yaklaşımların en iyi algoritma adayını bulabildiğini fakat otomatik yöntemin daha az zaman ve çaba gerektirdiğini göstermektedir.

Cáceres vd. (2017), çalışmalarında ParamILS'in kullandığına benzer bir uyarlanabilir sınırlama (capping) mekanizmasını **irace**'e entegre ederek, **irace**'i algoritma çalışma süresini en aza indirme amacı olan optimizasyon senaryolarında daha etkili hale getirmişlerdir. **irace** belirli bir çalışma süresi içinde ulaşılan çözüm kalitesinin optimizasyonu için tasarlanmıştır. Ancak hedef algoritmanın çalışma süresini en aza

indirmeyi amaçlayan yapılandırma senaryolarına uygulandığında zayıf yapılandırmaları denemek için çok fazla zaman harcadığından genel amaçlı yapılandırma yaklaşımlarının performansına ulaşamamaktadır. Sonuçlar entegre edilen mekanizmanın, çalışma süresini en aza indirme senaryolarında son teknoloji algoritma yapılandırıcılarıyla rekabetçi performans seviyelerine ulaştığını göstermektedir.

Colmenar, Martí ve Duarte (2018), makalelerinde çeşitliliği maksimize etme problemi için önerilen maksimum-toplam çeşitlilik ve maksimum-minimum çeşitlilik olmak üzere iki farklı modeli incelemişlerdir. Her iki modelinde aynı anda optimize edildiği iki amaçlı çeşitlilik problemini çözmüşlerdir. Çözüm için 6 farklı sezgisel algoritma önerilmiş ve performansları incelenmiştir. Algoritmaların parametre değerlerini belirlemek için **irace**'den faydalanmışlardır.

Cintrano, Chicano ve Alba (2020) Malaga'da paylaşımlı bisiklet toplama ve bırakma alanları belirleme problemini ele almışlardır. Optimizasyonda var olan p-medyan sorunu; bir dizi tesisi, bir dizi müşteri ve tesisler arasındaki mesafeyi en aza indirecek şekilde yerleştirmeyi amaçlamaktadır. Problemin çözümü ve karşılaştırma için genetik algoritma, yinelemeli yerel arama, parçacık sürü optimizasyonu, simüle edilmiş tavlama ve değişken mahalle aramasını kullanmışlardır. Kullanılan her algoritmanın parametreleri belirlenirken **irace**'i kullanmışlardır.

Pan, Zhang ve Lim (2021), çalışmalarında gün boyunca farklı zaman aralıklarında seyahat sürelerinin değiştiği zamana bağlı zaman pencereci toplam seyahat süresini en aza indirmeyi hedefleyen araç rotalama problemini incelemişlerdir. Problemi karma tamsayı programlama ile modelleyerek çözüm için tabu arama ve uyarlanabilir büyük mahalle araması içeren hibrit bir algoritma önermişlerdir. Algoritma parametrelerini en iyi yapılandırma kümesini bulmak için **irace** ile belirlemişlerdir. Sonuçlar otomatik yapılandırma ile parametreleri belirlenen algoritmanın literatürdeki kıyaslama örnekleri için bilinen en iyi çözümleri iyileştirdiğini göstermektedir.

Souza, Ritt ve López-Ibáñez (2022) otomatik algoritma yapılandırmalarında kötü yapılandırmaları erken elimine etmek için yeni sınırlama yöntemleri önermişlerdir. Farklı örnekler üzerinde farklı yapılandırmaların değerlendirilmesi gerektiğinden bu süreç pahalıdır. Algoritmanın çalışma süresinin en aza indirmek amacı olan karar problemlerinde kötü yapılandırmaları erken elemek için sınırlama yöntemleri uygulanmaktadır. Bu yöntemler doğrudan optimizasyon problemlerine uygulanabilir değildir. Çalışmalarında önerdikleri yeni sınırlama yöntemlerini **irace** yapılandırıcısına

entegre ederek farklı optimizasyon senaryolarında değerlendirmişlerdir. Sonuçlar önerilen yöntemlerin yapılandırılmalarının kalitesini korurken otomatik algoritma yapılandırmasının hesaplama çabasını azalttığını göstermektedir. Ayrıca önerilen sınırlamaların yapılandırma zaman sınırı olan senaryolarda mevcut bütçeyi daha iyi kullanmaya yardımcı olduğunu kanıtlamışlardır.

Deliktaş ve Aydın (2023), Tip-1 basit montaj hattı dengeleme probleminin çözümün için üst sezgisel tabanlı yapay arı algoritması önermişler ve literatürdeki 61 tane kıyaslama problemini çözmüşlerdir. Algoritmanın parametre ayarları için, **irace** kullanmışlardır. Kurada ve Deliktaş (2024), Tip-2 montaj hattı dengeleme ve işgücü atama probleminin çözümü için çok-amaçlı genetik algoritma önermişler ve algoritmanın parametre ayarları için **irace** ve deney tasarımı yöntemini önermişlerdir.

Erişilebilen literatür çalışmaları incelendiğinde zamana bağlı zaman pencereli araç rotalama probleminde kullanılmış olsa da çoklu zaman pencereli ve çoklu kullanımlı araç rotalama probleminde otomatik parametre belirleme metodu olan **irace** ile ilgili çalışmalara rastlanmamıştır.

3.2.2. Deney Tasarımı

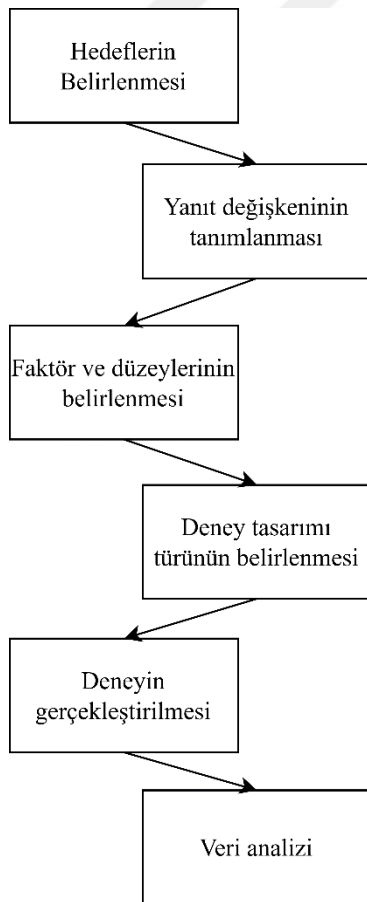
DoE, deney için uygun verilerin toplanıp istatistiksel yöntemlerle analiz edileceği şekilde planlaması sürecini ifade etmektedir (Montgomery, 2008). Deneysel tasarım kavramı, 1920'lerde Ronald A. Fisher'in deney tasarımı metodolojisi üzerine bulgularını kitaplarında yayınlamasına dayanmaktadır. Fisher, deneylerin tasarımı ve analizi için sistematik olarak istatistiksel ilkeler ortaya koymuştur (Beg vd., 2019). Verilerden geçerli ve nesnel sonuç çıkarılması gerektiği durumlarda deneysel tasarıma yönelik istatistiksel bir yaklaşım gerekmektedir. DoE, bir sistemdeki kontrol edilebilir ve kontrol edilemez girdi faktörlerinin sistemin çıktısı üzerindeki etkisini belirlemeye yardımcı olmaktadır. Yaygın olarak kullanılan tasarım türleri aşağıdaki gibidir (Durakovic, 2017).

- *Karşılaştırma*: En iyi seçeneği seçmek için bir faktör arasında yapılan çoklu karşılaştırmalardır.
- *Değişken Tarama*: Çıktıyı etkileyen birçok faktör (değişken) arasından önemli faktörleri seçmek amacıyla iki seviyeli faktöriyel tasarımlar kullanılmaktadır.
- *Transfer Fonksiyonu Tanımlama*: Girdi değişkenleri ile çıktı değişkeni arasındaki ilişki transfer fonksiyonu kullanılarak incelenmektedir.

- *Sistem Optimizasyonu*: Deneyi deęişkenlerin optimum ayarlarına hareket ettirerek optimizasyon yapıla bilinmektedir.
- *Saęlam Tasarım*: Sistem, süreç veya üründeki varyasyonu nedenlerini ortadan kaldırmadan azaltmakla ilgilenmektedir.

DoE’de ilk olarak problemin net bir şekilde belirlenmesi gerekmektedir. Ele alınan amacın ölçülebilir olması gerekmektedir. İkinci adımda tanımlanan hedeflere dayalı ölçülebilir deney sonucu belirlenmelidir. Yanıt deęişkeni, deęişken veya nitelik olarak tanımlanabilir. Yanıt deęişkeninde deęişiklik yaratan faktörlerin de seçilmesi gerekmektedir. Bu adımda balık kılçığı diyagramı kullanılabilir. Faktörler belirlendikten sonra düzeyler belirlenmelidir. DoE türü, kaç kere tekrarlanacağı yanıt deęişkenini doğrudan etkilemektedir. Kullanılacak deney türü belirlendikten sonra veri toplama adımına geçilmektedir. Elde edilen verilerin analiz edilebilmesi için veriye uygun istatistiksel bir yöntem seçilerek sonuçlar elde edilmektedir. Deneyi yapan kişi tarafından sonuçlar yorumlanmaktadır. Şekil 3.4 DoE adımlarını ifade etmektedir.

Şekil 3.4: DoE Adımları



DoE yöntemi ile alakalı literatürde yapılan çalışmalar çalışmanın devamında verilmiştir.

Wang ve Lu (2009), çalışmalarında kapasite kısıtlı ARP'nin çözümü için pratik olarak kullanılabilir, yenilikçi hibrit bir genetik algoritma önermişleridir. Üç aşamadan oluşan bu algoritmanın ilk aşamasında en yakın ekleme yöntemi depo ile dağıtım noktaları arasındaki aksel ve yarıçap ilişkilerini aynı anda dikkate alan tarama algoritmasına dahil edilmiştir. İkinci adımda DoE yoluyla çaprazlama ve mutasyon olasılığını optimize etmek için yanıt yüzeyi yöntemi kullanmışlardır. Son olarak, genişletilmiş tarama algoritması genetik algoritmaya entegre edilmiştir. Sonuçlar önerilen hibrit algoritmanın kapasite kısıtlı ARP'yi optimize etme konusundaki uygulanabilirliğini göstermektedir.

Ayadi vd. (2014), çalışmalarında araçların çalışma günü boyunca birden fazla rotada kullanılmasına izin verilen araç rotalama probleminin bir çeşidi olan çok seferli yeşil araç rotalama problemi için bir matematiksel model önermişlerdir. Bunu çözmek için genetik algoritma ile yerel arama prosedürünü birleştirerek bir evrimsel algoritma önermişlerdir. Algoritmanın parametrelerinin seçimini optimize etmek için, birkaç değişkenin (parametrenin) yanıt değişkeni (amaç fonksiyonu) üzerindeki etkilerini analiz eden deneysel bir tasarım olan Taguchi yöntemini uygulamışlardır. Geliştirilen algoritmanın etkinliği, bir dizi kıyaslama (benchmark) üzerinde test edilmiştir. Önerilen algoritma mevcut algoritmalarla karşılaştırıldığında, rekabetçi performans göstermekte ve birçok yeni en iyi çözüm sunmaktadır.

Rabbani vd. (2017), çalışmalarında taşımacılığın neden olduğu çevresel sorulara çözüm olması amacıyla kirlilik rotalama problemi olarak adlandırılan araç rotalama probleminin bir uzantısı tanıtılmışlardır. Kirlilik rotalama problemi, sadece kat edilen mesafeyi değil, aynı zamanda bir araç filosu tarafından yayılan sera gazı miktarını da hesaplamaktadır. Çalışmada sunulan çok bölmeli yeşil araç rotalama problemi olarak adlandırılan yeni bir yeşil araç rotalama probleminin amacı, araçların yaydığı kirliliği dikkate alarak yük, hız ve sürücü ödemelerindeki değişikliklerin maliyetini en aza indirmektir. Sunulan problemi çözmek için genetik algoritma ve benzetim tavlama dahil olmak üzere iki meta-sezgisel algoritma ile bir hibrit meta-sezgisel algoritma seçmişlerdir. Genetik algoritmada kullanılan mutasyon ve çaprazlama oranı, popülasyon büyüklüğü ve maksimum iterasyon sayısı parametrelerini ve benzetim tavlama da kullanılan başlangıç sıcaklığı, maksimum iterasyon sayısı, soğuma oranı ve son sıcaklık parametrelerini

optimize etmek için DoE kullanmışlardır. Son olarak, bu yöntemlerle elde edilen sonuçlar birbirleriyle karşılaştırmışlardır.

Long vd. (2019), çalışmalarında araç rotalama probleminin yeni bir varyantı olan ödül toplamalı araç rotalama problemini (ÖTARP) incelemektedir. ÖTARP'de araç atama ve ziyaret sıralamasının yanı sıra mevcut araçlar yetersiz olduğu için müşteri seçimi de belirlenmelidir. Bu nedenle ÖTARP'de birden fazla optimizasyon amacı söz konusu olmaktadır. Ele aldıkları problemin çözümü için genetik algoritma ve yerel arama stratejisi içeren Pareto tabanlı bir evrimsel algoritma önermişlerdir. Önerilen algoritmanın, popülasyon büyüklüğü, çaprazlama ve mutasyon olasılığı, maksimum iterasyon sayısı, değiştirme olasılığı, yeniden başlatma eşiği, keşfedilecek maksimum paralel yol sayısı parametreleri için 32 adet parametre yapılandırması kullanılarak optimum parametre değerlerini elde etmişlerdir. Sonuçlar, kıyaslama problemleri üzerinde önerilen algoritmanın üstünlüğünü göstermektedir.

Mosayebi ve Sodhi (2020), çalışmalarında DoE teorisini kullanarak gezgin satıcı problemini çözmek için bir genetik algoritmanın operatörlerinin ve parametrelerinin ayarlanmasının çözüm üzerindeki etkisini incelemişlerdir. Küçük ölçekli problem için popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı ve elitizm gibi parametreler deneyler ile belirlenmiştir. Deneyler sonucu elde edilen en iyi parametre değerlerini büyük ölçekli problemleri çözmek için kullanmışlardır. Sonuçlar, deneyler ile seçilen parametrelerin, ayarlanmamış parametre ayarlarıyla karşılaştırıldığında hem elde edilen sonuçların kalitesinde hem de yakınsama oranında gelişmiş performansla sonuçlandığını göstermektedir.

Maroof, Ayvaz ve Naeem (2024), çalışmalarında NP-zor bir problem olan ZPARP'yi çözmek için Solomon eklemeli yapıcı sezgi ile desteklenen yenilikçi bir hibrit genetik algoritma-solomon eklemeli sezgiseli (HGA-SES) çözümünü sunmaktadır. HGA-SIH yaklaşımının, popülasyon büyüklüğü, çaprazlama ve mutasyon oranları ve nesil sayısı gibi genetik algoritmaya ilişkin parametreleri algoritmanın birçok kez çalıştırılması ve DoE yöntemi kullanılarak optimize edilmiştir. Önerilen HGA-SES'in performansı, Solomon'un ZPARP kıyaslama örnekleri ile doğrulanmıştır. Deneysel bulgular, HGA-SES'nin çeşitli çözüm kalitesi ölçütlerine göre değerlendirildiğinde, diğer gelişmiş algoritmalarla eşit veya üstün sonuçlar sunduğunu doğrulamaktadır.

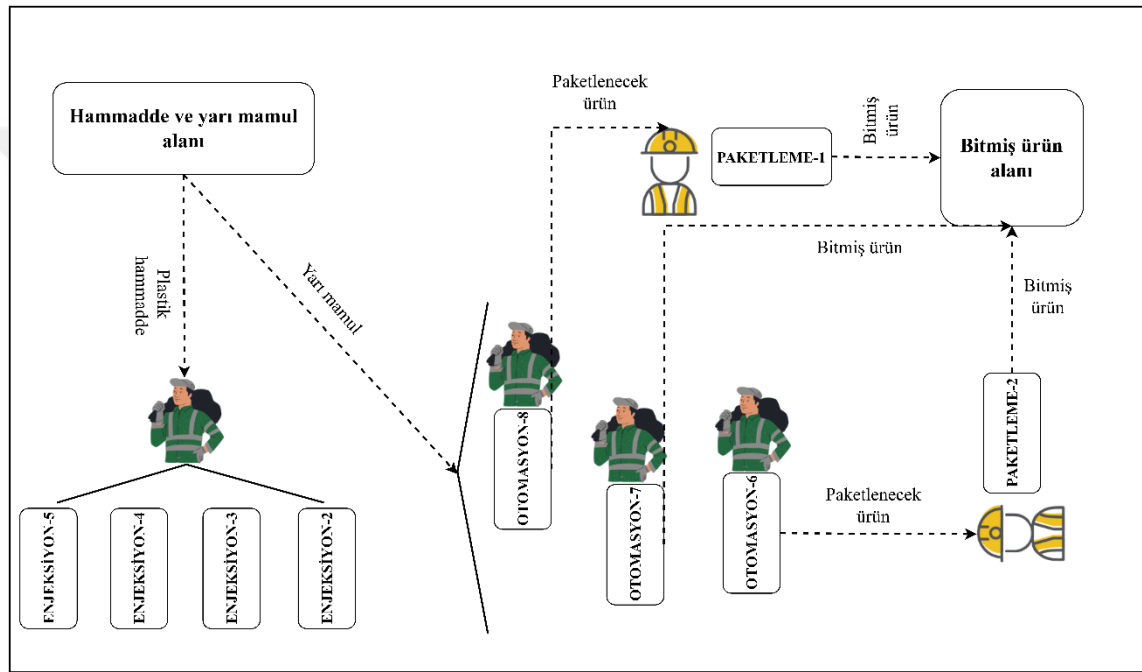


DÖRDÜNCÜ BÖLÜM
ÇOKLU ZAMAN PENCERELİ VE ÇOKLU KULLANIMLI ARAÇ
ROTALAMA PROBLEMİNİN ÇÖZÜMÜ İÇİN İKİ AMAÇLI SEZGİSEL
YAKLAŞIM

4.1. PROBLEMİN TANITIMI

Bu çalışmada laboratuvar sarf malzemeleri üreten bir firmanın otomasyon ve plastik enjeksiyon bölümü ele alınmış ve bu sistemdeki istasyonların kendilerine hammadde (HM) ve yarı mamul (YM) tedarik etme sürecindeki duruşların önlenmesi amaçlanmıştır. Bu firmada enjeksiyon bölümünde basılan çeşitli plastik ürünler otomasyon bölümünde montajlanmaktadır. Şekil 4.1, üretim mevcut üretim alanını göstermektedir.

Şekil 4.1: Mevcut Durum Üretim Alanı

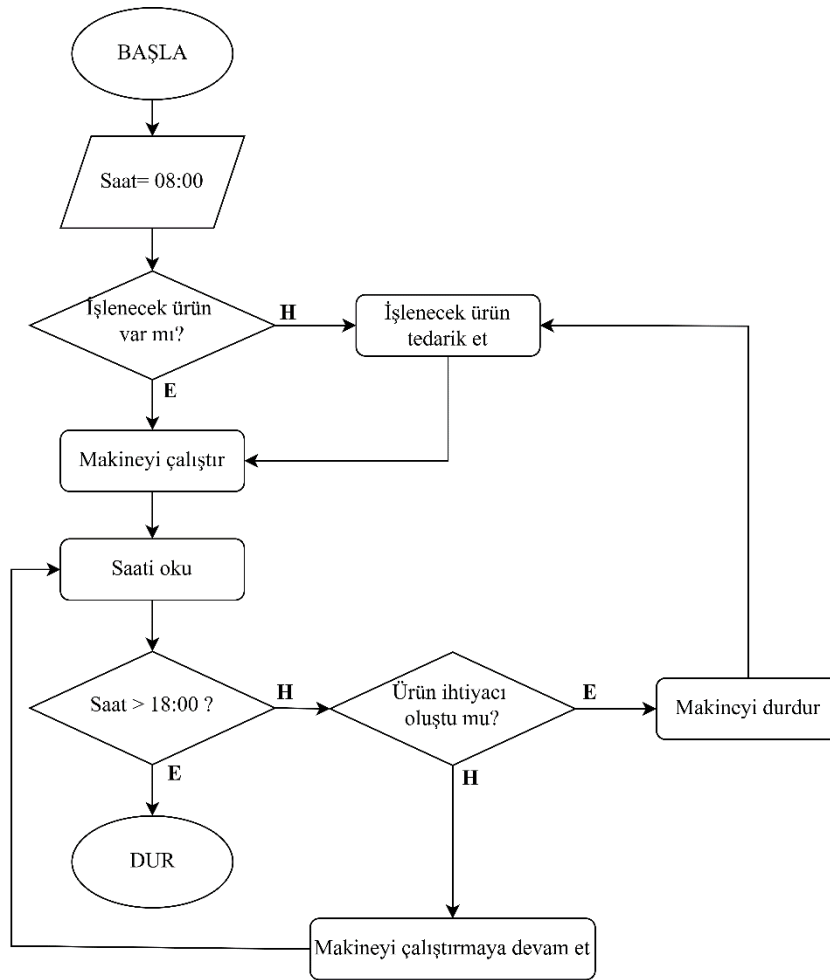


Montajı tamamlanan bazı ürünler paketlemeye, bazı ürünler ise direk bitmiş ürün alanına gitmektedir. Paketlenecek ürünlerin paketleme alanına alınması paketlemedeki operatör tarafından yapılmaktadır. Bu ürün tedariki sırasında operatör makinaları kapatmak durumunda kalmaktadır, fakat paketleme makinalarının çevrim süreleri otomasyon makinalarının çevrim sürelerinden daha küçük olduğundan dolayı bu üretim adedinde duruştan kaynaklı azalmaya sebep olmamaktadır. Bu nedenle bu çalışmaya firmada bulunan paketleme bölümü eklenmemiştir. Paketli veya paketsiz bitmiş ürünlerin bitmiş ürün alanına ulaşımı ise depo personelleri tarafından yapılmaktadır. Böylelikle istasyon operatörü bitmiş ürünü taşımak için makinaları kapatmak durumunda kalmamaktadır.

Plastik enjeksiyon ve otomasyon alanlarında HM ve YM tedariki istasyon operatörleri tarafından yapılmaktadır. Mesai başlangıcından bitimine kadar belirli zaman aralıklarında istasyonların HM ve YM ihtiyaçları oluşmaktadır. Plastik enjeksiyon

bölümündeki makinalardan tek bir operatör sorumludur. Bu operatör sabah mesai başlangıcında tüm makinalara HM tedarik etmekte ve gün içerisinde de makinalardaki HM azaldıkça yerine yenisini getirmektedir. Mevcut durumda operatör mesai başlangıcında hem makinaları açıp hem de HM tedarik etmeye çalıştığı için üretime başlama süresi gecikmekte bu da üretim adedinde duruştan kaynaklı azalmaya sebep olmaktadır. Otomasyon bölümündeki makinaların her birine bir operatör bakmaktadır ve gün içerisinde belirli periyotlarda YM tedarigi için makinayı kapatmak durumunda kalmaktadır. Bu durum duruşlardan kaynaklı olarak üretim adedini azaltmaktadır.

Şekil 4.2: Mevcut Durum Akış Diyagramı



Mevcut durumda mesai başlangıcından bitimine kadar plastik enjeksiyon ve otomasyon alanlarına ait akış diyagramı Şekil 4.2’de gösterilmektedir. Mevcut durumda mesai başlangıcında her bir istasyon için işlenecek ürün olup olmadığı kontrol edilmektedir. Yeterli işlenecek ürün varsa üretim başlamaktadır. Yeterli işlenecek ürün olmadığı durumda ilk önce ürün tedarigi yapılmaktadır. Mesai bitimine kadar ürün ihtiyacı oluştuğu durumlarda operatör ürün tedarik edene kadar makineyi durdurmaktadır.

4.2. MEVCUT DURUM ANALİZİ

Plastik enjeksiyon bölümünde mevcut durumda oluşan HM tedariklerinden kaynaklı duruş süreleri, operatörün mesai başlangıcında her makinaya ürün tedariki için HM alanına giderken harcadığı süre, makinaya yükleme yapmak için harcadığı süre ve 4 makinaya bir operatörün bakmasından kaynaklı mesai başlangıcında oluşan bekleme sürelerinin toplamıdır. Otomasyon alanında YM tedariklerinden kaynaklı duruş süreleri, operatörlerin gün içerisindeki malzeme ihtiyaç frekansı ile bu malzeme tedariki için firma içinde harcadığı sürenin çarpımıdır. Ayrıca bu süreye tedarik ettiği YM'yi makinaya yüklemek için geçirdiği sürede eklenmiştir. Otomasyon bölümünde her makinaya bir operatör baktığından dolayı otomasyon makinalarının mesai başlangıcında herhangi bir operatör bekleme süresi bulunmamaktadır. Tablo 4.1'de makinalara ait duruş sürelerinin hesaplanması verilmektedir.

Tablo 4.1: HM ve YM Tedariğinden Kaynaklı Duruş Sürelerinin Hesaplanması

Makina Adı	Yolda Geçen Süre (dk)	Makine Yükleme Süresi (dk)	Bekleme Süresi (dk)	Toplam Makine Duruş Süresi (dk)
Enjeksiyon 2	4	2	0	6
Enjeksiyon 3	4	2	6	12
Enjeksiyon 4	4	3	12	24
Enjeksiyon 5	4	2	24	30
	Yolda Geçen Süre*Frekans (dk)	Makine Yükleme Süresi (dk)	Bekleme Süresi (dk)	Toplam Makine Duruş Süresi (dk)
Otomasyon 6	32	20	-	52
Otomasyon 7	24	28	-	52
Otomasyon 8	16	20	-	36

Üretimde yaşanan yemek molası, çay molası gibi zorunlu duruşlar haricindeki plansız duruşlar makinaların kullanılabilirlik oranını düşürmektedir. HM ve YM tedariklerinden kaynaklı duruşlar da makinaların kullanılabilirlik oranını düşüren plansız duruşlardır. Kullanılabilirlik oranının hesaplanabilmesi için öncelikle kullanılabilir süre hesaplanmalıdır. Kullanılabilir süre, toplam çalışma süresinden zorunlu duruş sürelerinin çıkarılması ile elde edilmektedir ve Eşitlik (4.1)'de verilmiştir. Kullanılabilirlik oranı ise elde edilen kullanılabilir süreden çıkarılan plansız duruş sürelerinin kullanılabilir süreye oranlanması ile bulunmaktadır ve Eşitlik (4.2)'de verilmiştir.

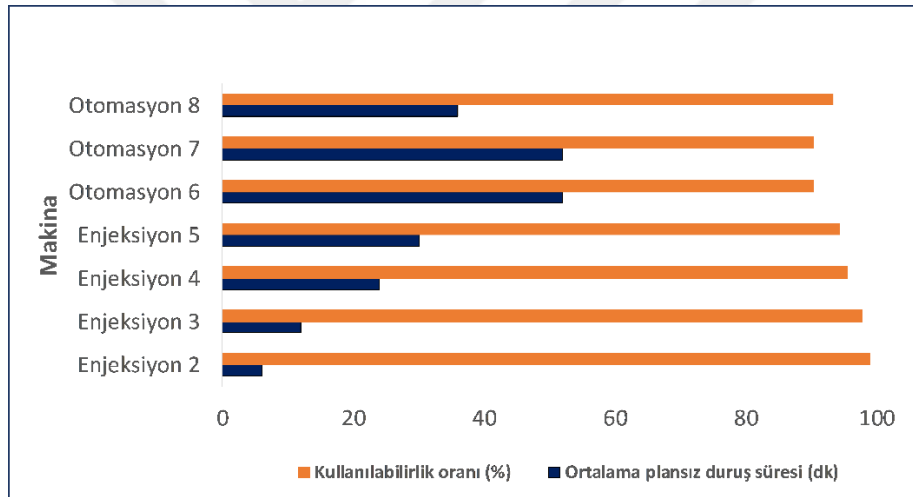
$$\text{Kullanılabilir Süre} = \text{Toplam Çalışma Süresi} - \text{Zorunlu Duruş Süreleri} \quad (4.1)$$

$$\text{Kullanılabilirlik Oranı \%} = \frac{\text{Kullanılabilir Süre} - \text{Plansız Duruş Süreleri}}{\text{Kullanılabilir Süre}} \quad (4.2)$$

Kaynak: Yükselen.

Bu çalışmada, ele alınan firmada günde 10 saat çalışılmaktadır. Çalışma süresi boyunca 30 dakika yemek molası, günde 2 kez 15 dakika da çay molası yapılmaktadır. Ayrıca her gün temizlik için 5 dakika zorunlu duruş yaşanmaktadır. Bu firmada toplam çalışma süresi olan 600 dakikadan mola ve zorunlu duruş çıkarıldığı zaman kalan 535 dakika kullanılabilir süredir. Enjeksiyon bölümünde makinaların üretime başlayabilmesi için gerekli olan ısınma süresi 30 dakikadır. Bu süre 535 dakikadan çıkarıldığında enjeksiyon bölümü için kullanılabilir süre 505 dakika iken otomasyon bölümü için 535 dakikadır. Mevcut durumda her makinaya ait HM ve YM tedarigiinden kaynaklı olan duruş süreleri ölçülmüş ve kullanılabilirlik oranları hesaplanmıştır. Şekil 4.3'te makinalara ait HM ve YM tedarigiinden kaynaklı duruş sürelerinin kullanılabilirlik oranı üzerindeki etkisi verilmiştir. Duruş süresi arttıkça makinaların kullanılabilirlik oranı düşmektedir.

Şekil 4.3: Mevcut Durum Makina Duruş Sürelerinin Kullanılabilirlik Oranına Etkisi



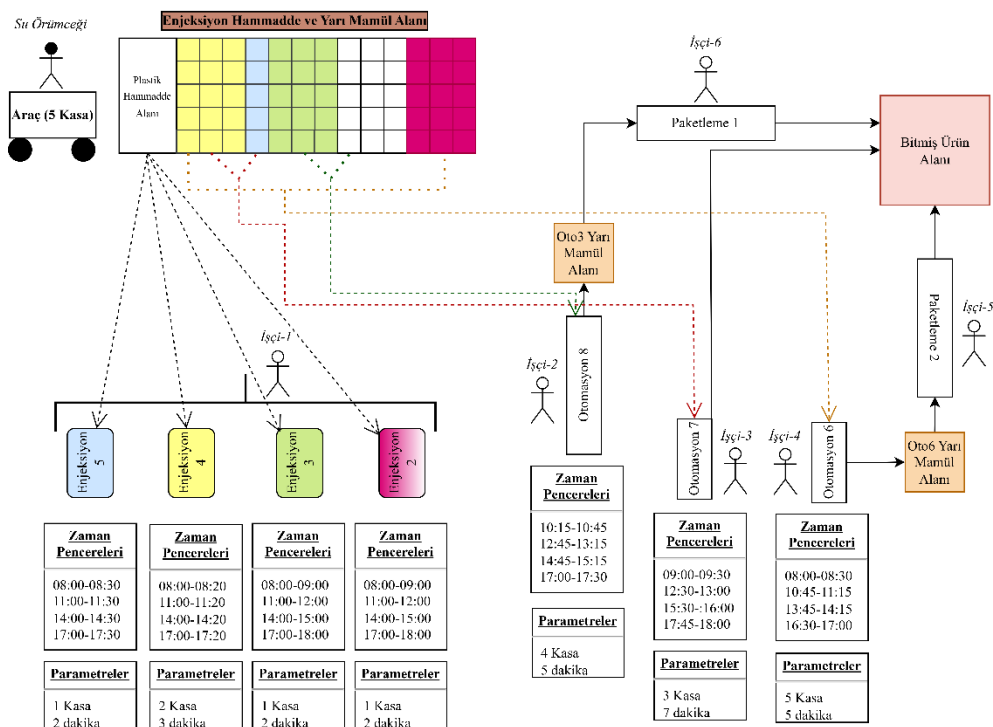
Problemin ele alındığı firmada enjeksiyon bölümünde makina başı robotları kullanılmaktadır. Bu nedenle bölümde bulunan 4 makina tek bir operatör tarafından kullanılabilir. Mesai başlangıcında bir operatörün 4 makinayı açabilmesi ve aynı zamanda makinaların ısınması için gereken süre de zorunlu duruş olarak nitelendirilmektedir. Bu nedenle enjeksiyon bölümünde kullanılabilirlik oranının %100 olması beklenmektedir. Fakat HM tedarigiinden kaynaklı duruşlar sebebiyle bu orana ulaşamamaktadır. Otomasyon bölümünde ise gün içerisinde çalışma şekliinden kaynaklı olarak çeşitli duruşlar yaşanmaktadır. Bu problem için kullanılabilirlik oranı hesabında plansız duruş olarak sadece YM tedarigiinden kaynaklı duruşlar ele alınmaktadır. Bunun sebebi gelecek durumda bu duruş ortadan kaldırıldığında iyileşme oranını görebilmektir.

4.3. ELE ALINAN GERÇEK HAYAT PROBLEMİ İÇİN ÖNERİLEN MODEL

İncelenen gerçek hayat problemindeki durum gözlemlendiğinde enjeksiyon ve otomasyon alanlarına HM ve YM tedarikliğini gerçekleştirebilecek bir su örümceği ihtiyacının olduğu tespit edilmiştir. Önerilen su örümceği ile HM ve YM tedarikinden kaynaklı duruşların önlenerek kullanılabilirlik oranının artırılması hedeflenmektedir. Ele alınan projenin incelendiği departmanda alan darlığından dolayı *tek bir* su örümceği kullanılmasının yeterli olacağı öngörülmüştür ki istasyonlar arası mesafe en fazla 6 dakikadır. Ayrıca su örümceğinin taşıyacağı ürünler küçük olduğundan ve malzeme indirme ve bindirme süreleri göz ardı edilecek kadar küçük olduğundan bu çalışmada dikkate alınmamıştır.

Su örümceği sadece fabrika içinde dolaşarak istasyonların gün içerisinde oluşan HM veya YM ihtiyacını karşılayacaktır. İstasyonlarda gün içerisinde HM ve YM ihtiyacı belirli periyotlar halinde ve belirli zaman aralıklarında oluşmaktadır. Su örümceğinin belirli periyot ve zaman aralıklarında dengeli ve israfsız bir dağıtım yapabilmesi için problem zaman pencereci tek araç rotalama problemi olarak ele alınmıştır. Şekil 4.4, incelenen gerçek hayat problemindeki gün boyunca istasyonların zaman pencerelerini, taleplerini ve hizmet sürelerini göstermektedir. Ayrıca su örümceği kullanımıyla akışın nasıl olacağına dair bir örnek sunmaktadır.

Şekil 4.4: Su Örümceği Kullanımı ile Oluşturulan Üretim Alanı



Şekil 4.4'te gösterildiği üzere her istasyonun gün içerisinde farklı zaman pencereleri vardır. Su örümceğine bu zaman pencerelerinde istasyonların taleplerini karşılayacağı ve mevcut durumda da operatörlerin kullandığı en fazla 5 kasa yükleme yapılabilen bir araç tahsis edilmiştir. Su örümceği, rota üzerinde bulunan istasyonların taleplerini araca yükleyerek dağıtım yapacaktır. Talep miktarları tüm günün talebi değil her zaman penceresinde oluşan talep miktarıdır. Araçta ürün bitmesi durumunda tekrar depoya dönerek aracı dolduracaktır. Her istasyonda teslimat için gereken hizmet süreleri boyunca o istasyona hizmet verecektir.

Paketleme makinalarının çevrim süreleri otomasyon makinalarının çevrim sürelerinden daha küçük olduğundan dolayı bu üretim adedinde duruştan kaynaklı azalmaya sebep olmamaktadır. Bu nedenle su örümceği sadece otomasyon ve paketleme alanlarına hizmet verecektir. Mesai bitimine kadar tüm istasyonların her zaman penceresinde hizmet alması sağlanacaktır.

4.3.1. İki Amaçlı Matematiksel Model

Matematiksel modelleme birçok uygulamada strateji ve hassasiyet için önemlidir. Ayrıca modellenen sistemi sistematik hale getirerek anlaşılmasını sağlamaktadır. Kısıtlar altında amaçların eniyilenmeye çalışıldığı matematiksel modeli oluşturmak problemin genel çerçevesini oluşturmak için önemli bir adımdır. Bu çalışmada ele alınan problem tek bir aracın, n sayıda müşteriye belirli birden fazla zaman penceresi içerisinde hizmet verebilmesi için en iyi rotanın oluşturulması problemidir.

Önerilen çoklu kullanım ve çoklu zaman pencereli araç rotalama probleminin matematiksel modeli, Koç ve Karaoğlan (2012) çalışmasındaki modele gecikme süresi kısıtı eklenerek ve amaca da gecikme süresinin enküçüklenmesi eklenerek güncellenmiştir. Çoklu kullanım ve çoklu zaman pencereli araç rotalama problemi için önerilen matematiksel modelin varsayımları aşağıdaki gibidir:

- Her istasyon (müşteri) istisnasız ziyaret edilmelidir.
- Her istasyona belirli zaman pencerelerinde hizmet verilmelidir.
- İlgili zaman penceresinde her istasyona sadece bir kez uğranmalıdır.
- Her rota depoda başlamalı ve depoda bitmelidir.
- Araç tek bir rota üzerinde birden fazla istasyonu ziyaret edebilir.
- Bir güzergahtaki istasyonların toplam talepleri araç kapasitesini aşmamalıdır.

- Gerekirse araç malzemeleri yenilemek için depoya dönebilir.
- İstasyon talepleri bölünemez.
- Malzeme yenilenirken indirme bindirme süreleri göz ardı edilecek kadar küçük olduğundan dikkatte alınmayacaktır.
- Her rota izin verilen maksimum süre içerisinde tamamlanmalıdır.

İndisler:

I, J : İstasyon (Müşteri) Kümesi, $i \in I = \{0, 1, \dots, N\}$

R : Rota Kümesi $r \in R = \{1, 2, \dots, R\}$

Parametreler:

N : Depo ve İstasyon sayısı ($N = '0' \cup N_c$)

Q : Araç kapasitesi

t_{ij} : i . istasyondan j . istasyona seyahat süresi ($\forall i, j \in N$)

d_i : i . istasyonun talebi ($\forall i \in N_c$)

p_i : i . istasyona hizmet süresi ($\forall i \in N_c$)

A_i : i . istasyon zaman penceresi başlangıç zamanı ($\forall i \in N_c$)

B_i : i . istasyon zaman penceresi bitiş zamanı ($\forall i \in N_c$)

$Max_{Süre}$: Her rotanın tamamlanması için izin verilen maksimum süre

Karar Değişkenleri:

X_{ijr} : 1, eğer i - j ayrıtı rotada ise; 0, diğer durumda

T_i : i . istasyona gecikme zamanı

W_r : 1, eğer araç r rotasını tamamladıktan sonra $r+1$ rotasına başlarsa; 0, diğer durumda

u_i : i . istasyondan önce araçta bulunan ürün miktarı

S_{ir} : 1, eğer i . istasyon r . rotada ise i . istasyon için hizmet başlangıç zamanı; 0, diğer durumda

Str_r : r . rota hizmet başlangıç zamanı

Fns_r : r . rotanın depoya dönüş zamanı

İki Amaçlı Matematiksel Model:

$$Enk f_1(x) = \sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} \quad (4.3)$$

$$Enk f_2(x) = \sum_{i \in N_c} T_i \quad (4.4)$$

$$\sum_{i \in N} \sum_{r \in R} X_{ijr} = 1 \quad (\forall j \in N_c) \quad (i \neq j) \quad (4.5)$$

$$\sum_{j \in N} X_{ijr} = \sum_{j \in N} X_{jir} \quad (\forall i \in N, \forall r \in R) \quad (4.6)$$

$$\sum_{j \in N_c} X_{0jr} \leq 1 \quad (\forall r \in R) \quad (4.7)$$

$$\sum_{j \in N_c} X_{0jr} \geq \sum_{j \in N_c} X_{0j,r+1} \quad (\forall r \in R : r < |R|) \quad (4.8)$$

$$u_j - u_i + Q \sum_{r \in R} X_{ijr} + (Q - d_i - d_j) \sum_{r \in R} X_{jir} \leq Q - d_i \quad (4.9)$$

($\forall i, j \in N_c, i \neq j$)

$$u_i \geq d_i + \sum_{j \in N_c, j \neq i} \sum_{r \in R} d_j X_{ijr} \quad (\forall i \in N_c) \quad (4.10)$$

$$u_i \leq Q - (Q - d_i) \sum_{r \in R} X_{i0r} \quad (\forall i \in N_c) \quad (4.11)$$

$$S_{ir} - S_{jr} + (M + p_i + t_{ij}) X_{ijr} \leq M \quad (\forall i, j \in N_c, \forall r \in R, i \neq j) \quad (4.12)$$

$$T_j \geq S_{jr} - B_j \sum_{i \in N} X_{ijr} \quad (\forall j \in N_c, \forall r \in R, i \neq j) \quad (4.13)$$

$$S_{jr} \geq A_j \sum_{i \in N} X_{ijr} \quad (\forall j \in N_c, \forall r \in R, i \neq j) \quad (4.14)$$

$$S_{jr} \geq Str_r + (t_{oj} + M) X_{0jr} - M \quad (\forall j \in N_c, \forall r \in R) \quad (4.15)$$

$$w_r \leq \sum_{i \in N_c} X_{i0r} \quad (\forall r \in R) \quad (4.16)$$

$$w_r \geq \sum_{j \in N_c} X_{0j,r+1} \quad (\forall r \in R : r < |R|) \quad (4.17)$$

$$Str_{r+1} \geq Fns_r - M(1 - w_r) \quad (\forall r \in R : r < |R|) \quad (4.18)$$

$$Fns_r \geq S_{ir} + p_i + t_{i0} - M(1 - X_{i0r}) \quad (\forall i \in N_c, \forall r \in R) \quad (4.19)$$

$$Fns_r - Str_r \leq Max_{Süre} \quad (\forall r \in R) \quad (4.20)$$

$$X_{ijr} \in \{0,1\} (\forall_{i,j} \in N, \forall_r \in R) \quad (4.21)$$

$$w_r \in \{0,1\} (\forall_r \in R) \quad (4.22)$$

$$u_i, T_i \geq 0 (\forall_i \in N_c) \quad (4.23)$$

$$S_{ir} \geq 0 (\forall_i \in N_c, \forall_r \in R) \quad (4.24)$$

$$Str_r, Fns_r \geq 0 (\forall_r \in R) \quad (4.25)$$

Bu modelde Eşitlik (4.3) – (4.4) sırasıyla, toplam seyahat süresini ve toplam gecikme süresini hesaplayan amaç fonksiyonlarıdır. Eşitlik (4.3) toplam seyahat süresini en aza indirmeyi hedeflerken Eşitlik (4.4) gecikme sürelerini en aza indirmeyi hedeflemektedir. Eşitlik (4.5) her istasyonun ziyaret edilmesini sağlamaktadır. Eşitlik (4.6) istasyonlardaki giriş çıkış dengesini korumaktadır. Eşitlik (4.7) ile araç her rotada sadece bir istasyona çıkış yapmaktadır. Eşitlik (4.8) r rotası tamamlanmadan r+1 rotasının başlatılamayacağını belirtmektedir. Eşitlik (4.9) – (4.11) alt tur eleme ve araç kapasite kısıtlarıdır. Eşitlik (4.12) her istasyon için hizmet başlangıç zamanı kısıtıdır. Eşitlik (4.13) her istasyon için gecikme süresi kısıtıdır. Eşitlik (4.14) zaman penceresi başlangıç zamanı kısıtıdır. Eşitlik (4.15) her rotadaki ilk istasyon için hizmet başlangıç zamanı kısıtıdır. M yeterince büyük bir sayıdır. Eşitlik (4.16) – (4.20) rota başlangıç kısıtlarıdır ve aracın aynı anda birden fazla rotada kullanılmasını engellemektedir. Eşitlik (4.21) – (4.25) karar değişkeni kısıtlarıdır.

Çok amaçlı matematiksel modeller aynı anda birden fazla ve çoğunlukla birbirleriyle çelişen amaçları göz önünde bulundurmaktadır. Bu durumlarda her amacı ayrı ayrı değerlendirmek tercih edilmemektedir. Çok amaçlı optimizasyon problemlerinde genellikle tek bir en iyi çözüm olmamakla beraber her amacı aynı anda makul düzeyde sağlayan çözümler üretmek daha önemli olmaktadır. Literatürde çok amaçlı çeşitli yöntemlerle birleştirip tek bir amaç haline getirilmesini sağlayan skalerleştirme yöntemleri bulunmaktadır. Bunlardan bazıları; ağırlıklı-toplam (ATY), konik skalerleştirme (KSY), benson skalerleştirme, pascoletti-serafini skalerleştirme ve chebyshev skalerleştirme (CSY) yöntemleridir (Kasimbeyli vd., 2019). Bu çalışmada ele alınan problem ATY, KSY ve CSY ile skalerleştirilerek elde edilen sonuçlar kıyaslanmıştır.

ATY, hedef fonksiyonları arasındaki ağırlıkları sistematik olarak değiştirerek Pareto optimal çözümleri tek tek arayan yöntemdir (Yang, 2014). Pareto optimal çözümleri elde etme de kullanılan en yaygın yöntemdir. Amaç ağırlıkları amaç değerleri ile çapılıp toplanarak tek bir amaca dönüştürülmektedir. Bu toplam problemin türüne göre enküçüklenmesi veya enbüyüklenmesi sağlanacak tek bir amaç fonksiyonunu verir. Eşitlik (4.26)'da ATY'nin matematiksel modeli verilen probleme uyarlanması verilmektedir.

$$enk\ ATY(x) = w_1 * \sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} + w_2 * \sum_{i \in N_c} T_i \quad (4.26)$$

Burada w_1 ve w_2 ele alınan problemdeki amaçların önem ağırlıklarını temsil etmektedir. $w_1 + w_2 = 1$ ve $w_1, w_2 \geq 0$ olmalıdır ve matematiksel modeldeki tüm kısıtlar sağlanmalıdır.

KSY, Pareto etkin değerleri bulmak için hiper-düzlemler yerine destek konilerinin kullanıldığı Gasimov tarafından birden fazla amaçlı optimizasyon problemleri için geliştirilmiş skalerleştirme yöntemidir. Çok geniş problem çeşitlerine uygulanabilir olması yöntemin üstünlüğü olarak değerlendirilmektedir. Bu yöntem literatürde birçok dışbükey olmayan optimizasyon problemlerine uygulanmış ve etkin sonuçlar alınmıştır (Yaşar ve Özyön, 2018). Eşitlik (4.27)'de KSY'nin matematiksel modeli verilen probleme uyarlanması verilmektedir.

$$enk\ KSY(x) = [w_1(\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1) + w_2(\sum_{i \in N_c} T_i - RfP_2) + \alpha(|\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1| + |\sum_{i \in N_c} T_i - RfP_2|)] \quad (4.27)$$

RfP_1 ve RfP_2 , olası desteklenmeyen ya da gizli etkin çözümleri aramak için karar verici tarafından her amaç için belirli bir aralıkta rastgele belirlenen sabit referans noktalarıdır. Koninin tepe açısı olan α , $w_1 + w_2 = 1$ ve $0 \leq \alpha \leq enk\ \{w_1, w_2\}$ olacak şekilde belirlenmeli ve matematiksel modeldeki tüm kısıtlar sağlanmalıdır. Bu çalışmada, $\alpha = enk\ \{w_1, w_2\} - 0,01$ olarak hesaplanmıştır.

Çok amaçlı optimizasyon problemlerinde skalerleştirme yapılırken belirlenen ağırlıklar çözüm kalitesini etkilemektedir. Uygun olmayan bir ağırlık Pareto optimal kümesinden sapabilir. CSY, çözüm alanı dışbükey olması fark etmeksizin tüm çözüm yüzeylerinde kullanılan doğrusal olmayan bir skalerizasyon yöntemidir (Giagkiozis ve Fleming, 2015). Çözümleri daha geniş ve etkili şekilde arayarak Pareto çözümleri elde etmektedir. Eşitlik (4.28)'de CSY'nin matematiksel modeli verilen probleme uyarlanması

verilmektedir.

$$enk\ CSY(x) = enb \left[w_1 \left(\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1 \right), w_2 \left(\sum_{i \in N_c} T_i - RfP_2 \right) \right] + \left[\rho \left(w_1 \left(\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1 \right) + w_2 \left(\sum_{i \in N_c} T_i - RfP_2 \right) \right) \right] \quad (4.28)$$

RfP_1 ve RfP_2 , karar verici tarafından her amaç için belirli bir aralıkta rastgele belirlenen sabit referans noktalarını, $\rho > 0$ ise küçük bir sayıyı temsil etmekte ve matematiksel modeldeki tüm kısıtlar sağlanmalıdır. Bu çalışmada ρ değeri, 0,01 olarak ele alınmıştır.

Önerilen çok amaçlı model, küçük boyutlu bir örnek üzerinde (3 istasyonlu, tek zaman pencereli) denenmiş ve Lingo 11.0 paket programında kodlanarak çözülmüştür. Modelin, ele alınan problemi karşıladığı doğrulanmıştır. Ele alınan gerçek hayat probleminde ise 7 farklı istasyon ve her istasyonun 4 farklı zaman penceresi bulunmaktadır. Bu problemin matematiksel modeli ile çözümü için Lingo 11.0, 72 saat boyunca çalıştırılmasına rağmen uygun bir çözüme ulaşamamıştır. Problemin literatürde Np-zor (Cömert vd., 2017) olması sebebiyle çözüm için iki amaçlı genetik algoritma önerilmiştir.

4.4. İKİ AMAÇLI GENETİK ALGORİTMA ÇÖZÜMÜ

ZPARP'i için 1980'lerden bugüne kadar çeşitli kesin, buluşsal ve meta sezgisel yöntemler ve bunların beraber kullanıldığı yöntemler önerilmiştir. Kesin yöntemler küçük boyutlu problemler için kullanılabilirken ZPARP'nin büyük boyutlu gerçek hayat problemlerinde kesin yöntemlerin kullanılması çok sayıda varyasyondan dolayı imkânsız hale gelmektedir (Expósito, vd., 2019). Bu nedenle bu tip problemlerin çözümünde sezgisel ve meta sezgisel teknikler kullanılmaktadır. Ele alınan problemde 7 istasyon, her istasyona ait 4 farklı zaman penceresi ve 1 adet depo bulunmaktadır. Çözümün amacı kullanılacak olan su örümceğinin toplam kat ettiği mesafeyi ve istasyonlara gecikme süresini enküçüklenmesidir. Bu nedenle ele alınan problemin çözümü için iki amaçlı genetik algoritma önerilmiştir. Önerilen algoritmaya ait sözde (pseudo) kod, Tablo 4.2'de verilmiştir. Başlangıçta parametre belirleme yöntemleri ile elde edilen popülasyon büyüklüğü ($P_{Büyüklük}$) kadar başlangıç popülasyonu ($P_{Başlangıç}$) rassal olarak oluşturulur. Başlangıç popülasyonundaki her bir kromozomun uygunluk değeri Eşitlik (4.31)-(4.33)'den biri kullanılarak hesaplanır. Algoritmanın durdurma koşulu, en büyük uyum değeri hesaplama olarak dikkate alınmıştır.

Tablo 4.2: Önerilen Algoritmanın Sözde (Pseudo) Kodu

Algoritma 1: Önerilen algoritmanın sözde (pseudo) kodu

Input: $N, Q, t_{ij}, d_i, p_i, A_i, B_i, w_1, w_2, RfP_1, RfP_2, \alpha, \rho, \text{MaxSüre}, P_{\text{Büyüklik}}, CrOran, MOran, T_{\text{Büyüklik}}$

Output: En iyi rota C_{Eniyi} , En iyi uygunluk değeri $f(C_{\text{Eniyi}})$

1. **begin**
2. $P_{\text{Başlangıç}} \leftarrow$ Başlangıç popülasyonunu rassal oluştur ($P_{\text{Büyüklik}}$)
3. **foreach** (s in $P_{\text{Başlangıç}}$)
4. $f(s) \leftarrow$ Uygunluk fonksiyonu hesabı (s) (Bkz, Eşitlik (4.31)-(4.33))
5. **end foreach**
6. $P_{\text{Mevcut}} \leftarrow P_{\text{Başlangıç}}$
7. **while** durdurma koşulu sağlanana kadar **do**
8. $P' \leftarrow$ Turnuva Seçim ($P_{\text{Başlangıç}}, T_{\text{Büyüklik}}$)
9. $G \leftarrow$ Çaprazlama ($P', CrOran$)
10. $G' \leftarrow$ Değiş-tokuş Mutasyonu ($G, MOran$)
11. **foreach** (s in G')
12. $f(s) \leftarrow$ Uygunluk fonksiyonu hesabı (s) (Bkz, Eşitlik (4.31)-(4.33))
13. **end foreach**
14. $P_{\text{Mevcut}} \leftarrow$ Elitizm (P_{Mevcut}, G')
15. **end while**
16. **return** P_{Mevcut} içindeki C_{Eniyi} ve $f(C_{\text{Eniyi}})$
17. **end**

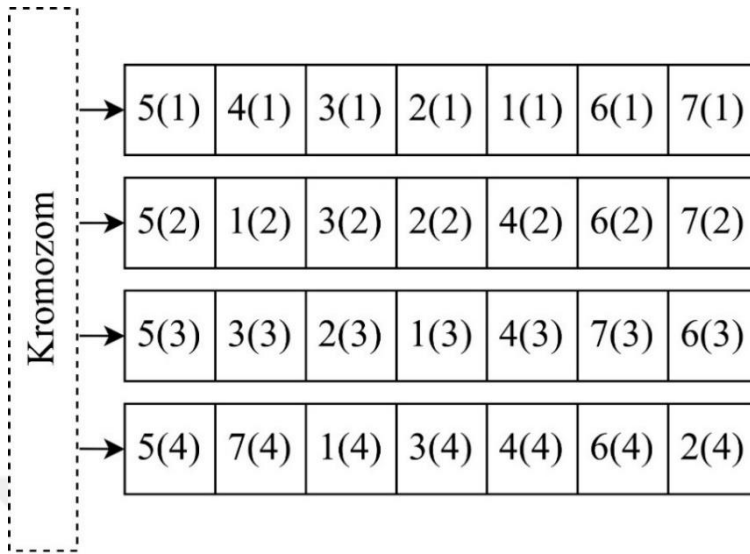
Her bir döngüde, turnuva seçim operatörü kullanılarak çaprazlanacak ebeveynler (P') için bir eşleşme havuzu oluşturulur. Seçilen ebeveynler parametre belirleme yöntemi ile elde edilen çaprazlama operatörlerinden biri ile çaprazlanarak yavru havuzu (G) oluşturulur. Oluşan yavru havuzunda, her bir yavrunun geni mutasyon oranı (M_{Oran}) dikkate alınarak değiş-tokuş mutasyon işlemine maruz kalır. Turnuva büyüklüğü ($T_{\text{Büyüklik}}$), çaprazlama operatörü, çaprazlama oranı ($CrOran$), mutasyon oranı (M_{Oran}) için parametre belirleme yöntemleri ile belirlenen değerler kullanılmıştır. Oluşan yeni popülasyondaki (G') kromozomlar için uygunluk değeri hesaplanır. Buradaki bir elit birey diğer nesle aktarılır. Durdurma koşulu sağlanana kadar bu işlem tekrar eder ve bu işlem sonundaki en iyi birey ve o bireye ait amaç değerleri algoritmanın çıktısını oluşturur.

4.4.1. Kromozom Gösterimi

Genetik algortmada her kromozomun uygun bir aday çözüm olmasından kaynaklı bu yapının başlangıçta iyi tanımlanması gerekmektedir. İncelenen problemde 7 istasyon olmasına karşın her istasyonun 4 farklı zaman penceresi bulunmaktadır. Bu nedenle oluşturulan kromozom yapısı bir dizi halinde olmakla beraber zaman pencerelerine göre ayrılmıştır. Kromozom yapısı ile araç kapasitesini aşmayacak rotalar oluşturulmaktadır. Ayrıca oluşturulan rotaların maksimum rota tamamlanma süresi

boyunca tamamlanması gerekmektedir.

Şekil 4.5: Örnek Kromozom Yapısı



Şekil 4.5, oluşturulan örnek kromozom yapısıdır. Her gen bir istasyon ve o istasyona ait bir zaman penceresini ifade etmektedir. Örneğin; 5(4) geni 5.istasyonun 4.zaman penceresi aralığını, 4(3) geni 4.istasyonun 3.zaman penceresi aralığını vb. ifade etmektedir. Kromozom yapısı oluşturulurken istasyonların zaman pencerelerinin önceliği dikkate alınmıştır. Önerilen su örümceği, ilgili istasyona 1.zaman penceresinde hizmet vermeden 2.zaman penceresinde hizmet verememektedir. Su örümceğinin izleyeceği rotalar, kromozom içinde genlerin sırası dikkate alınarak araç kapasitesini aşmayacak şekilde oluşturulmalıdır. Su örümceği, rotasını tamamlanmadan diğer bir rotaya başlayamaz ve her rota depoda başlayıp depoda bitmelidir. Her istasyona o istasyona ait ilgili zaman penceresinde bir kez uğranmalıdır. Rotalama süreci tamamlandığında su örümceği, her istasyona o istasyonun zaman penceresi sayısı kadar uğramış olmaktadır. Bir rotanın bitiş süresi kendisini takip eden diğer rotanın başlangıç süresine eşittir. Bir istasyondan diğer bir istasyona geçerken geçilen istasyondaki hizmete başlangıç süresi, yol süresine ilk istasyonun hizmet süresi eklenerek bulunmaktadır. İlk rota tamamlandıktan sonra ikinci rotaya geçilir ve tüm istasyonlara o istasyonların bütün zaman pencerelerinde hizmet verene kadar bu işleme devam edilmektedir. Tablo 4.3'te ele alınan problemdeki istasyonlar arasındaki ve istasyon ve depo arasındaki uzaklıkların dakika cinsinden değerleri verilmektedir.

Tablo 4.3: Uzaklık Matrisi (7x7)

0*	1(1)	2(1)	3(1)	4(1)	5(1)	6(1)	7(1)	1(2)	2(2)	3(2)	4(2)	5(2)	6(2)	7(2)	1(3)	2(3)	3(3)	4(3)	5(3)	6(3)	7(3)	1(4)	2(4)	3(4)	4(4)	5(4)	6(4)	7(4)	
0*	0	2	2	2	2	4	3	2	2	2	2	2	4	3	2	2	2	2	2	4	3	2	2	2	2	2	4	3	2
1(1)	2	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1
2(1)	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2
3(1)	2	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3
4(1)	2	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4
5(1)	4	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2
6(1)	3	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1
7(1)	2	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0
1(2)	2	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1
2(2)	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2
3(2)	2	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3
4(2)	2	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4
5(2)	4	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2
6(2)	3	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1
7(2)	2	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0
1(3)	2	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1
2(3)	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2
3(3)	2	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3
4(3)	2	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4
5(3)	4	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2
6(3)	3	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1
7(3)	2	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0
1(4)	2	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1	0	1	2	3	3	2	1
2(4)	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2	1	0	1	2	4	3	2
3(4)	2	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3	2	1	0	1	5	4	3
4(4)	2	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4	3	2	1	0	6	5	4
5(4)	4	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2
6(4)	3	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1	2	3	4	5	1	0	1
7(4)	2	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0	1	2	3	4	2	1	0

*Hammadde deposu, 0 ile gösterilmiştir.

4.4.2. Uygunluk Fonksiyonu

Ele alınan problemin çözümü için kullanılan genetik algorithmada en iyi sonuca ulaşmak için kromozomların matematiksel bir fonksiyon ile uygunluğunun değerlendirilmesi gerekmektedir. Uygunluk değeri düşük olan bireylere enküçükleme probleminde, uygunluk değeri büyük olan bireylere enbüyükme probleminde daha fazla hayatta kalma şansı verilmektedir. Böylelikle her iterasyon sonucu oluşturulan yeni popülasyonun bir önceki popülasyondan daha iyi olduğuna emin olunur. Genetik algorithmada problem türüne uygun kromozom yapısının oluşturulması uygunluk değerinin hesaplanmasında kolaylık sağlamaktadır.

Bu çalışmada ele alınan problemde su örümceğinin toplam kat ettiği mesafeyi ve istasyonlara olan gecikme süresini enküçükleme amaçlanmıştır. Algoritma ilk rotayı tamamlamadan ikinci rotayı oluşturmayacağı için ilk rotanın bitiş süresi ikinci rotanın başlangıç süresine eşittir. İlk rotanın tamamlanma süresi aracın depodan istasyona olan uzaklığının, istasyon hizmet süresinin ve depoya geri dönüş süresinin toplanmasıyla elde edilmektedir. Eğer rotada birden fazla istasyon varsa rota tamamlanma süresi bulunurken istasyonlar arası uzaklık ve her istasyonun hizmet süresi de dikkate alınmaktadır.

$$\text{Enk } f1(x) = \sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} \quad (4.29)$$

$$\text{Enk } f2(x) = \sum_{i \in N_c} T_i \quad (4.30)$$

Eşitlik (4.29) rota tamamlanma süreleri en küçük olan kromozomların seçilmesi için gerekli olan uyum fonksiyonudur. Süreler hesaplanırken mesai başlangıç zamanı olan 08:00, başlangıç zamanı olarak ele alınmıştır. Eşitlik (4.30) toplam gecikme sürelerini enküçükleyerek istasyonların duruş yaşamasının önüne geçecek çözümlerin seçilmesini sağlayan uyum fonksiyonudur. Bu fonksiyon ile zaman penceresi dışında hizmet veren ve gecikmeye neden olan çözümler elimine edilmektedir. Önerilen genetik algoritma için kromozomların uygunluk değerleri ATY, KSY ve CSY olmak üzere üç farklı skalerleştirme yöntemi kullanılarak hesaplanmıştır. Her bir yöntemden elde edilen sonuçlar kıyaslanmıştır.

4.4.3. Seçim

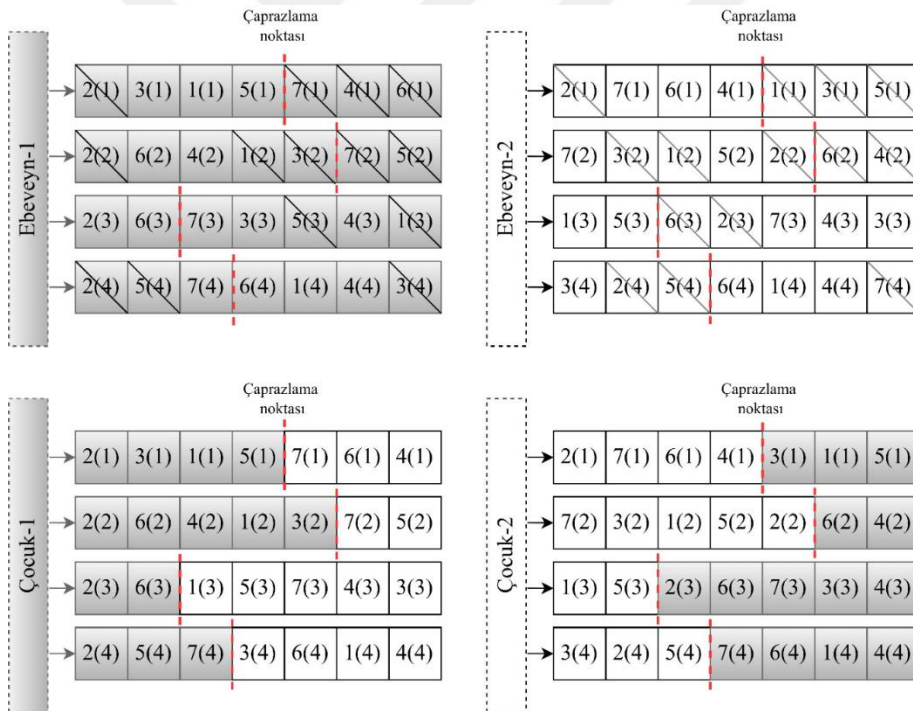
Önerilen genetik algorithmada seçim yöntemi olarak turnuva seçim yöntemi kullanılmıştır. Turnuva seçimi, tüm popülasyondaki bireylere seçilme şansı verirken

uygunluk değerleri iyi olan kromozomlara daha yüksek seçilme şansı vermektedir, bu nedenle hızlı sonuçlar üretir ve uygulaması kolaydır. Bu yöntemde kötü uygunluk değerine sahip bir birey de çaprazlamaya gidebilir fakat bu her zaman kötü sonuçlar elde edeceği anlamına gelmemektedir. Ayrıca bu durum genetik algoritmanın yerel optimuma takılmasının önüne geçmektedir.

4.4.4. Genetik Operatörler

Kromozomlar seçildikten sonra genetik çeşitliliğin korunması ve algoritmanın yerel optimuma takılmaması için çaprazlama ve mutasyon operatörleri uygulanmaktadır. Kullanılan genetik algorithmada *sıraya dayalı tek noktalı çaprazlama (order-based one-point crossover/OOX)* ve *önceliği koruyan sıra tabanlı çaprazlama (precedence preserving order-based crossover/POX)* olarak 2 farklı çaprazlama operatörü önerilmiştir.

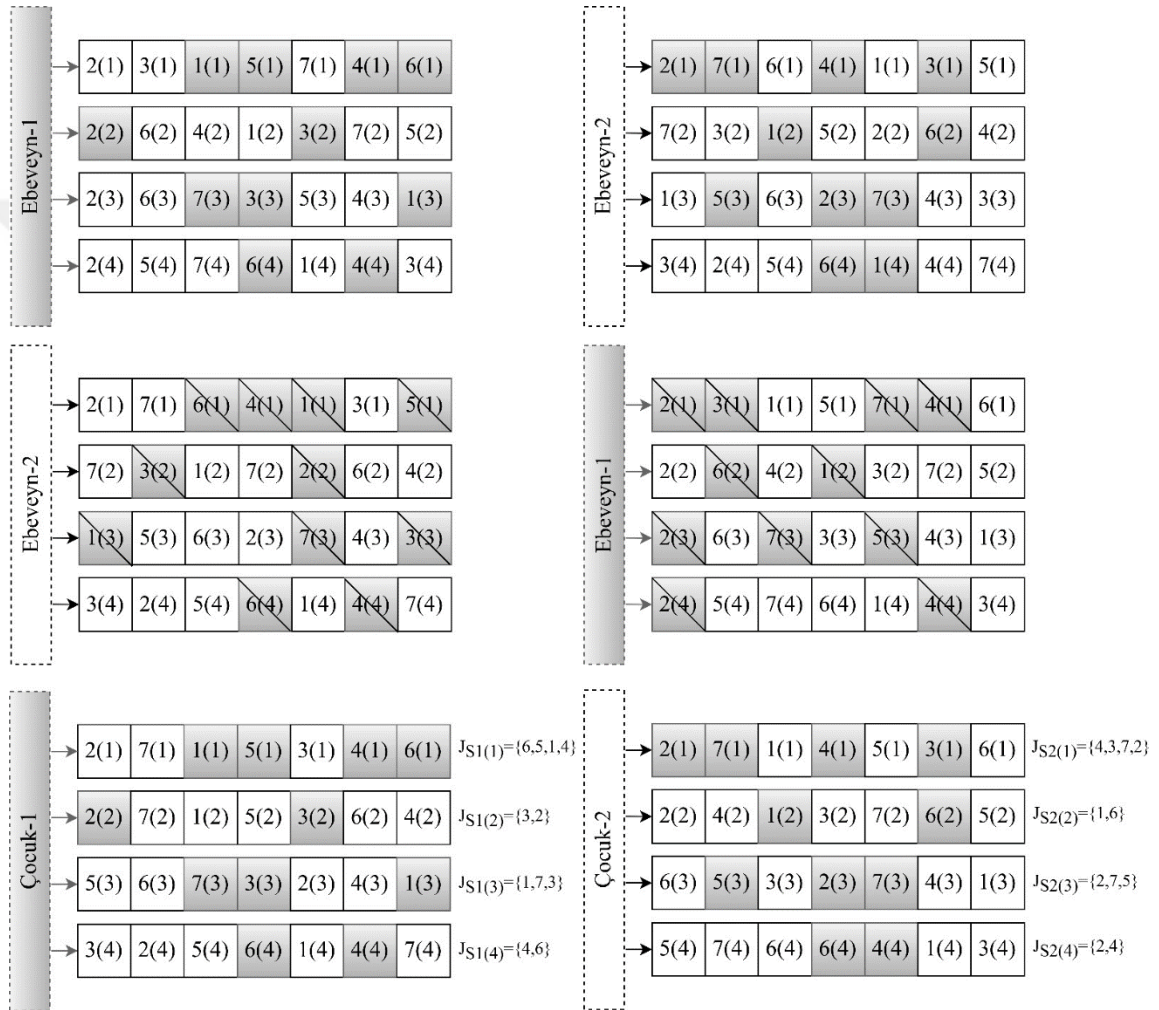
Şekil 4.6: Örnek OOX Çaprazlama Yöntemi



OOX, Deliktaş vd. (2021) tarafından geliştirilmiştir. Geliştirilen çaprazlama yönteminde tek noktalı çaprazlama ve sıralı çaprazlama yöntemleri birleştirilmiştir. Kullanılan yöntemde ebeveynlerin her katmanı için rastgele çaprazlama noktaları seçilmektedir. Bu çalışmada ele alınan problemde kromozom zaman pencerelerine göre katmanlara ayrılmıştır. İlk yavru için çaprazlama noktasına kadar olan kısım birinci ebeveynden birinci yavruya direk aktarılmaktadır. Daha sonrasında yavruya aktarılan

genler ikinci ebeveynden silinmekte ve ikinci ebeveynde kalan genler öncelik sırasına göre birinci yavruya aktarılmaktadır. İkinci yavru için ise ikinci ebeveynin çaprazlama noktasına kadar olan kısım ikinci yavruya direk aktarılmaktadır. Sonrasında yavruya aktarılan genler birinci ebeveynden silinmekte ve birinci ebeveynde kalan genler öncelik sırasına göre ikinci yavruya aktarılmaktadır. Kromozomun tüm katmanları için bu işlem tekrarlanmaktadır. Şekil 4.6’da OOX çaprazlama yöntemine ait örnek verilmiştir.

Şekil 4.7: Örnek POX Çaprazlama Yöntemi



Kullanılan genetik algorithmada önerilen bir diğer çaprazlama yöntemi POX ise, ilk olarak Lee vd. (1998) tarafından kullanılan yöntemdir. POX’de yavrular oluşturulurken ebeveynlerin her katmanından rastgele genler seçilir ve seçilen genlerden alt kümeler oluşturulur (Deng vd., 2017). Bu alt küme uzunluğu katman uzunluğu ile aynı olduğunda hiç çaprazlama olmamaktadır. Sonrasında birinci ebeveynden rastgele seçilen alt küme genleri yerleri değiştirilmeden birinci yavruya aktarılmaktadır. Yavruya aktarılan genler ikinci ebeveynden silinmekte ve ikinci ebeveynde kalan genler öncelik

sırasına göre birinci yavruya aktarılmaktadır. İkinci yavru için ise ikinci ebeveynden rastgele seçilen alt küme genleri yerleri değiştirilmeden ikinci yavruya aktarılmaktadır. Yavruya aktarılan genler birinci ebeveynden silinmekte ve birinci ebeveynde kalan genler öncelik sırasına göre ikinci yavruya aktarılmaktadır. Kromozomun tüm katmanları için bu işlem tekrarlanmaktadır. Böylelikle uğranılmayan ya da tekrar eden düğümler olmamaktadır. Şekil 4.7’de POX çaprazlama yöntemine ait örnek verilmiştir.

Önerilen genetik algorithmada çeşitliliği sağlamak için değiş-tokuş mutasyon operatörü uygulanmıştır. Uygulanan mutasyon operatöründe rastgele seçilen birey üzerinden iki farklı nokta seçilir. Seçilen 2 noktada bulunan genler için birbiri ile takas işlemi uygulanır. Mutasyon işlemi gerçekleştirilirken kromozomun katman yapısı dikkate alınmıştır. Seçilen iki farklı genin aynı katmanda olması gerekmektedir. Böylece kromozom yapısının bozulmasının önüne geçilerek tamir işlemine gerek kalmayacaktır.

4.4.5. Genetik Algoritmalarda Parametre Seçimi

Genetik algoritmaların en iyi çözüme ulaşmasında en önemli etkilerden biri parametre seçimidir. Parametre seçimi algoritmanın performansını etkilemektedir. Algoritmanın başında probleme göre parametre belirlenmektedir. Bu çalışmada kullanılan genetik algoritma için belirlenmesi gereken parametreler; popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı, turnuva büyüklüğü, çaprazlama operatörüdür.

Bu çalışmada önerilen iki amaçlı genetik algoritma yönteminin parametreleri, **irace** ve tam faktöriyel deney tasarımı yönteminin (DoE) kıyaslanması ile belirlenmiş ve elde edilen uygun parametreler kullanılmıştır. Parametre belirleme yöntemleri karşılaştırılırken uygunluk değerinin hesaplanabilmesi için amaç fonksiyonlarının normalize edilerek birimlerinden arındırılması gerekmektedir. Farklı amaç fonksiyonu değerleri, farklı büyüklüklere sahip olduğu için amaç değerleri ilgili amacın nadir değerine bölünerek normalize edilmektedir (Marler ve Arora, 2005). Nadir değer, ilgili amacın o ana kadar elde edilen sonuçları arasında aldığı en kötü değerdir. Enküçükleme problemi için bu değer o ana kadar elde edilen sonuçlar arasındaki en büyük değerdir (Deliktaş ve Ustun, 2023). Bu çalışmada, toplam seyahat süresi amaç fonksiyonunun nadir değerini elde etmek için amaç ağırlıkları $w_1 = 1$ ($w_2 = 0$) olacak şekilde algoritma 31 kez çalıştırılırken, toplam gecikme süresi amaç fonksiyonunun nadir değerini bulmak için ise amaç ağırlıkları $w_1 = 0$ ($w_2 = 1$) olacak şekilde algoritma 31 kez çalıştırılmıştır.

Algoritmanın 31 kez çalıştırılması sonucu her bir amaç için belirlenen en büyük değer ilgili amacın nadir değerini oluşturmaktadır. N_i , i . amaç fonksiyonunun nadir noktasını temsil etmektedir. ATY tarafından skalerleştirilen uyum fonksiyonu Eşitlik (4.31)'de verildiği gibi normalleştirilmiştir.

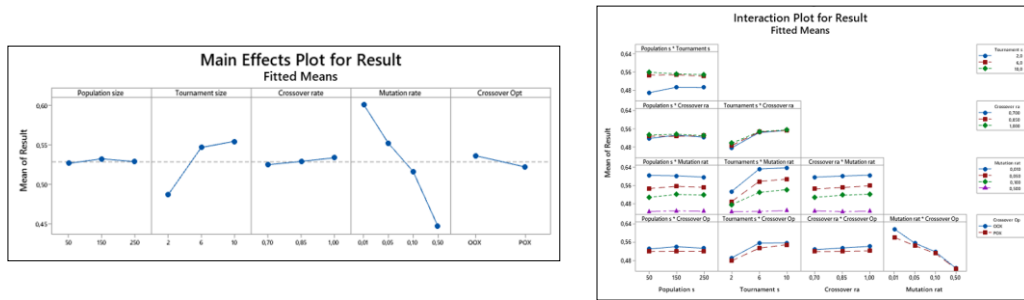
$$enk\ ATY^{Norm}(x) = \left[w_1 \frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr}}{N_1} + w_2 \frac{\sum_{i \in N_c} T_i}{N_2} \right] \quad (4.31)$$

Algoritmada kullanılacak olan uygun parametre değerlerinin belirlenebilmesi amacıyla DoE için belirlenen parametre faktörleri ve onların düzeyleri Tablo 4.4'te verilmiştir.

Tablo 4.4: Faktörler ve Faktörlerin Düzeyi

FAKTÖRLER	DÜZEYLER			
	Düzye 1	Düzye 2	Düzye 3	Düzye 4
Popülasyon Büyüklüğü	50	150	250	
Çaprazlama Oranı	0,70	0,85	1,0	
Mutasyon Oranı	0,01	0,05	0,10	0,50
Turnuva Büyüklüğü	2	6	10	
Çaprazlama Operatörleri	OOX	POX		

Şekil 4.8: Genetik Algoritmanın 31 Çalışmasından DoE Sonucu Elde Edilen Ana ve Birleşik Etkiler Grafikleri (Diğer Ağırlıklar için)



a) Ortalama sonuçlar için ana etkiler grafiği

b) Ortalama sonuçlar için bileşik etki grafiği

Belirlenen her faktör düzeyi için önerilen iki amaçlı genetik algoritma yaklaşımı Minitab 21 yazılımı kullanılarak analiz edilmiştir. Ele alınan deney modelinde 5 faktör ve her faktörün farklı düzeyleri bulunmaktadır. Bu durumda farklı amaç ağırlıkları için ele alınan tam faktöriyel deney tasarımında her ağırlık için $3 \times 3 \times 4 \times 3 \times 2 = 216$ farklı kombinasyonlarda deney yapılması gerekmektedir. Her bir deney, 11 farklı ağırlık ($w_1 = \{0, 0.1, 0.2, \dots, 1\}$ ve $w_2 = 1 - w_1$) için 31 kez çalıştırıldığından toplamda $216 \times 31 \times 11 = 73656$ deney yapılmıştır. $w_1 = 1$ ($w_2 = 0$) ve $w_1 = 0$ ($w_2 = 1$) amaç

ağırlıkları dışındaki diğer ağırlıklar için elde edilen ana etkiler grafiği ve bileşik etkiler grafiği Şekil 4.8. (a)-(b)'de görülmektedir.

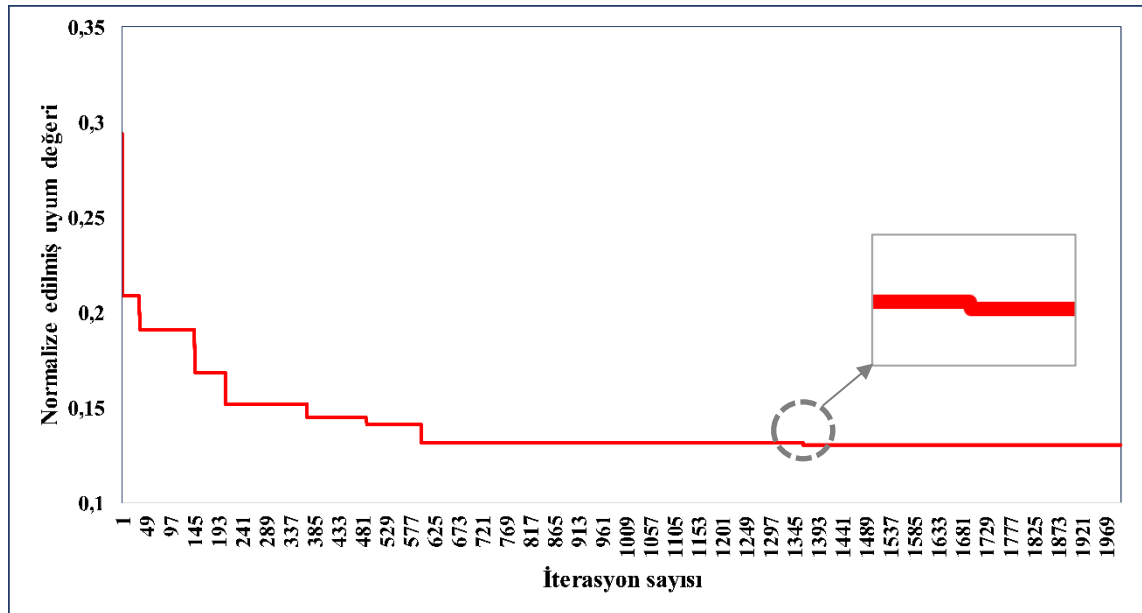
Önerilen genetik algortmada kullanılacak olan parametrelerin belirlenebilmesi için kullanılan diğer bir yöntem **irace**'dir. **irace**'te her bir parametre için kullanılan parametre aralıkları Tablo 4.5'te verilmiştir. Bu parametre aralıkları kullanılarak algoritma çalıştırılmış ve program normalleştirilmiş uyum değerlerini dikkate alarak farklı amaç ağırlıkları için en iyi algoritma konfigürasyonlarını sunmuştur.

Tablo 4.5: irace'de Kullanılan Parametre Aralıkları

Parametreler	Parametre Aralıkları
Popülasyon Büyüklüğü	U ~ [50; 250]
Çaprazlama Oranı	U ~ [0,7; 1]
Mutasyon Oranı	U ~ [0,01; 0,5]
Turnuva Büyüklüğü	U ~ [2; 10]
Çaprazlama Operatörleri	U ~ [OOX; POX]

Algoritmanın durdurma koşulu olarak, en büyük uyum değerini hesaplama olarak belirlenmiştir. Bu değer, müşteri (istasyon) sayısı ile belirlenen bir sabit çarpılması ile belirlenmiştir. Bu sabit değer, Şekil 4.9'daki yakınsama grafiği dikkate alınarak 20000 olarak belirlenmiştir.

Şekil 4.9: Sabit Değerin Belirlenmesi için Kullanılan Yakınsama Grafiği



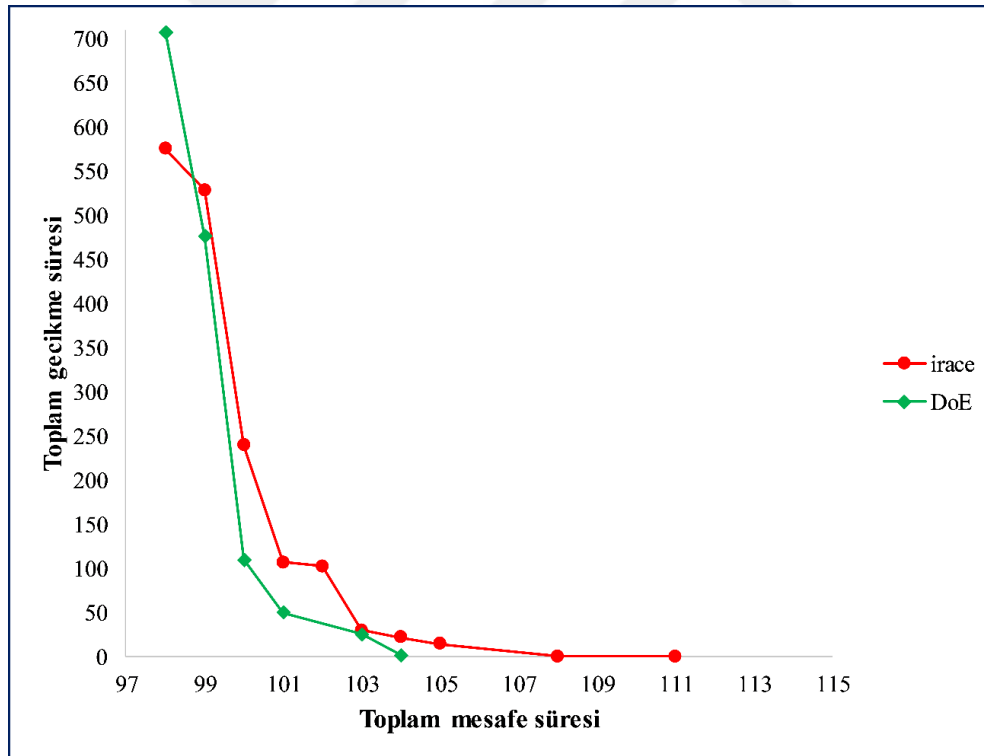
Tablo 4.6'da ise farklı amaç ağırlıkları için irace ve DoE'den elde edilen parametre değerleri verilmektedir.

Tablo 4.6: irace ve DoE Sonucu Elde Edilen Parametre Değerleri

	w_1	w_2	Popülasyon Büyüküğü	Çaprazlama Oranı	Mutasyon Oranı	Turnuva Büyüküğü	Çaprazlama Operatörü
irace	1	0	226	0,96	0,42	4	POX
	0	1	192	0,95	0,44	10	POX
	Diğer ağırlıklar		163	0,95	0,43	5	OOX
DoE	1	0	50	0,7	0,1	2	POX
	0	1	50	0,7	0,05	2	POX
	Diğer ağırlıklar		250	1	0,5	6	POX

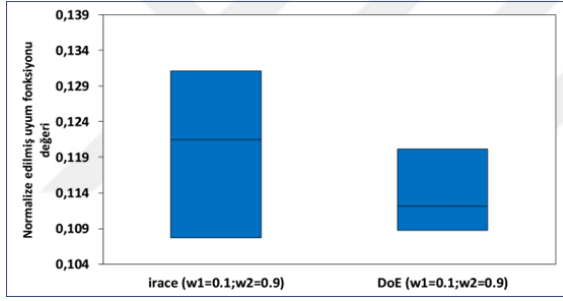
9 farklı amaç ağırlıkları için Tablo 4.6’da verilen tam faktöriyel DoE’den ve **irace**’ten elde edilen uygun parametre değerleri kullanılarak önerilen algoritmanın 31 kez tekrarlanmasından sonra bulunan ortalama normalize edilmiş uygunluk değerleri karşılaştırılmıştır. Önerilen algoritma, belirlenen parametre değerleri ile 31 kez çalıştırılmıştır.

Şekil 4.10: irace’e Karşı DoE Parametre Ayarlama Yönteminden Elde Edilen Sonuçların Pareto Grafiği

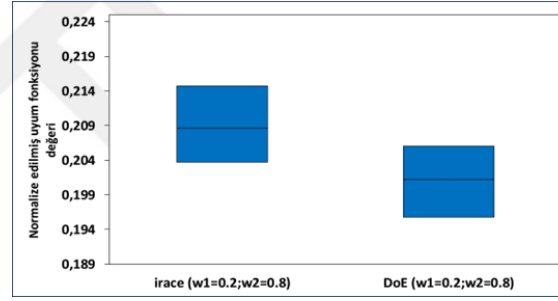


irace ve DoE’den elde edilen sonuçlara ait Pareto grafiği Şekil 4.10’da verilmektedir. Ele alınan problemde amaçların toplam seyahat ve gecikme süresinin enküçüklenmesi olması sebebiyle Pareto grafiğinde de görüldüğü üzere bu problem özelinde DoE, **irace**’e karşı daha etkin Pareto çözümler elde ettiğinden daha iyi performansa sahiptir. .

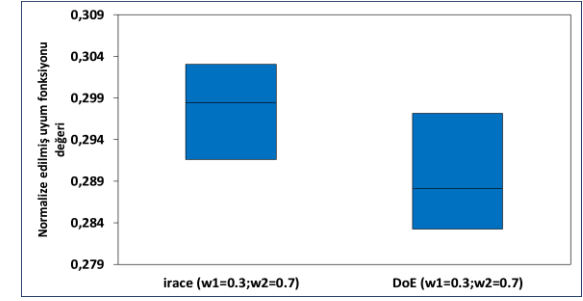
Şekil 4.11: 9 Farklı Amaç Ağırlığı Kombinasyonu için **irace** ve DoE Yöntemlerinden Elde Edilen Normalize Edilmiş Uyum Değerlerinin Kutu-Bıyık Grafikleri



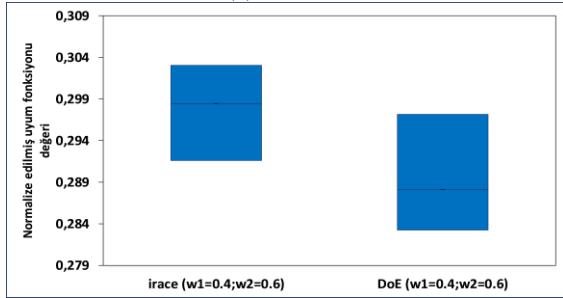
(a) Örnek#1



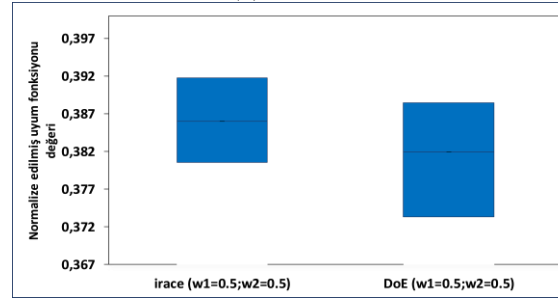
(b) Örnek#2



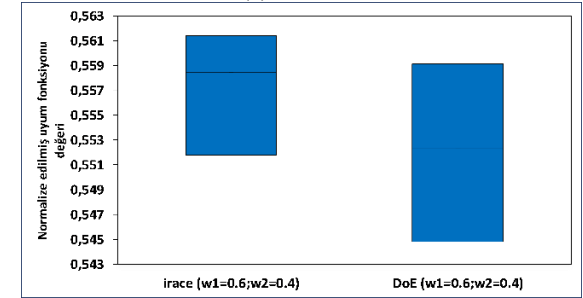
(c) Örnek#3



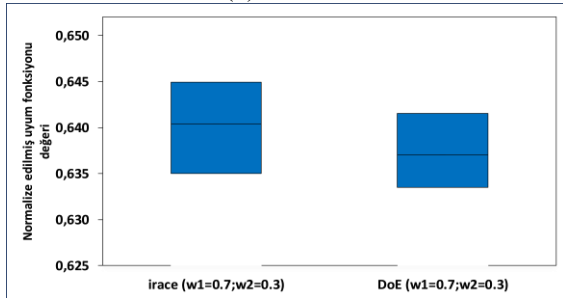
(d) Örnek#4



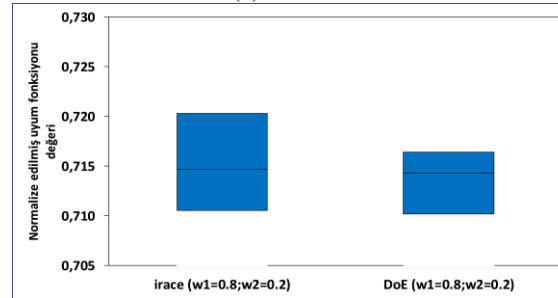
(e) Örnek#5



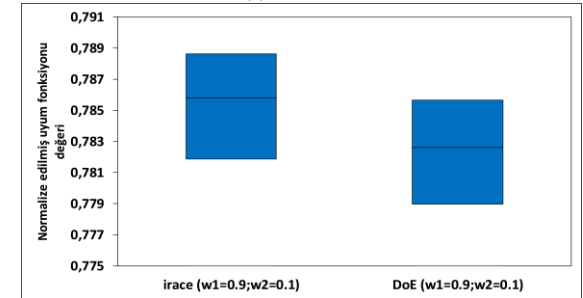
(f) Örnek#6



(g) Örnek#7



(h) Örnek#8



(i) Örnek#9

irace ve DoE'den elde edilen sonuçlara ait kutu-bıyık grafiği Şekil 4.11'de verilmektedir. Kutu-bıyık grafiğinden de anlaşılacağı üzere 9 farklı amaç ağırlıkları için, bu problem özelinde DoE'den elde edilen normalize edilmiş uygunluk değerleri **irace**'e göre çoğunlukla daha etkin sonuçlar vermiştir. Bu nedenle çalışmanın devamında tam faktöriyel DoE'den elde edilen parametre değerleri kullanılmaktadır

4.5. PERFORMANS METRİKLERİ

Bu bölümde, hem ATY ile bütünleştirilen genetik algorithmada farklı parametre ayarlama yöntemlerinin performansını karşılaştırmada, hem de konik ve chebyshev skalerleştirme yöntemleri ile birleştirilen genetik algoritma yaklaşımlarının performanslarını karşılaştırmada kullanılan performans metriklerine yer verilmiştir. Ele alınan algoritmaların performans değerlendirme metrikleri olarak, ideal ve referans noktalarına olan ağırlıklı uzaklıkları önerilmiştir. Birden fazla amaçlı eniyileme problemlerinde, ideal bir nokta, *enküçükleme problemi* için o ana kadar incelenen tüm çözümler arasından *en küçük* amaç fonksiyonu değeridir. Amaç uzayındaki bir referans noktası ise karar vericinin istediği bölgeyi belirlemesi için daha odaklı bir arama yapmasına izin veren değerdir. Bu çalışmada, toplam seyahat süresi amaç fonksiyonun ideal değerini elde etmek için amaç ağırlıkları $w_1 = 1$ ($w_2 = 0$) olacak şekilde algoritma 31 kez çalıştırılırken, toplam gecikme süresi amaç fonksiyonun ideal değerini bulmak için ise amaç ağırlıkları $w_1 = 0$ ($w_2 = 1$) olacak şekilde algoritma 31 kez çalıştırılmıştır. Algoritmanın 31 kez çalıştırılması sonucu amaçların enküçükleme olması sebebiyle her bir amaç için belirlenen en küçük değer ilgili amacın ideal değerini oluşturmaktadır. Ek olarak nadir değer, ilgili amacın o ana kadar elde edilen sonuçları arasında aldığı en kötü değerdir ve enküçükleme problemi için aynı amaç ağırlıkları ile elde edilen sonuçlar arasından her bir amacın en büyük değeri olarak hesaplanmaktadır.

Skalerleştirme yöntemlerinde (KSY ve CSY için) kullanılan referans noktaları, her bir amacın ideal ve nadir noktaları kullanılarak hesaplanmıştır. Referans noktalarının hesabında konveks kombinasyon tekniği kullanılmıştır (Kim J. ve Kim S.K., 2006).

$x^k = \operatorname{argmin} \{f_k(x) \mid x \in S\}, \forall k, k.$ amacın optimum ya da optimuma yakın çözümünü olsun. Burada $f_1(x)$ ve $f_2(x)$ sırasıyla amaç fonksiyonlarını temsil ederken, k , amaç fonksiyonu sayısını ve S , uygun çözüm kümesini ifade etmektedir. $V_k = [f_1(x^k), f_2(x^k)]$, $\forall k, x^k$ 'ya karşılık gelen amaç fonksiyonu vektörüdür. $\lambda = \lambda_1 +$

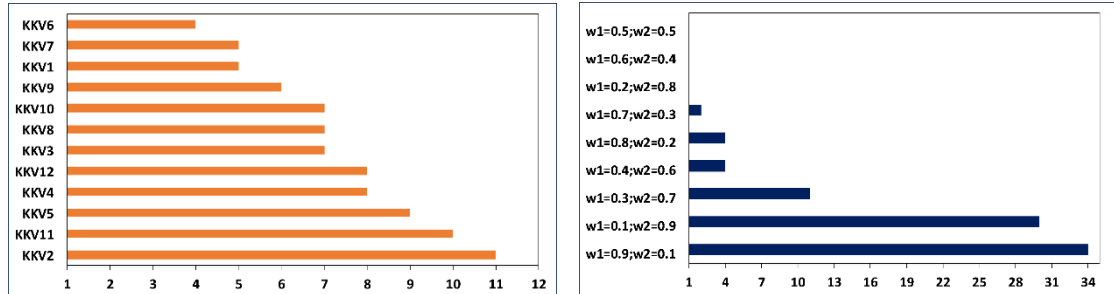
$\lambda_2 = 1$; $\lambda_1, \lambda_2 \geq 0$, konveks kombinasyon vektörünü (KKV) temsil etmektedir. Daha sonrasında, referans noktası vektörleri kümesi, $RfP^\lambda = \lambda_1 V_1 + \lambda_2 V_2$ formülasyonu ile elde edilmiştir. Bu problem için 12 farklı KKV'den elde edilen referans noktası Tablo 4.7'de sunulmuştur. Ütopik değerler, ideal değerlerin sırasıyla %10 ve %20 değerlerin altında değerlerden elde edilmiştir.

Tablo 4.7: KKV Kombinasyonlarına Göre Referans Noktaları

	λ_1	λ_2	RfP_1	RfP_2
KKV ₁	0,5	0,5	100	150,5
KKV ₂	0,1	0,9	104	30,1
KKV ₃	0,9	0,1	96	270,9
KKV ₄	İdeal Noktalar		95	0
KKV ₅	0,2	0,8	103	60,2
KKV ₆	0,3	0,7	102	90,3
KKV ₇	0,4	0,6	102	90,3
KKV ₈	0,6	0,4	99	180,6
KKV ₉	0,7	0,3	98	210,7
KKV ₁₀	0,8	0,2	97	240,8
KKV ₁₁	Ütopik Nokta (%10)		85,5	0
KKV ₁₂	Ütopik Nokta (%20)		76	0

KKV kombinasyonları ve amaç ağırlıklarının Pareto çözüm elde etme frekanslarına ait grafikler Şekil 4.12. (a)-(b)'de verilmiştir. Grafiklerden de anlaşılacağı üzere KKV₂, KKV₅ ve KKV₁₁ kombinasyonlarının ve $w_1 = 0,9$ ($w_2 = 0,1$), $w_1 = 0,1$ ($w_2 = 0,9$) ve $w_1 = 0,3$ ($w_2 = 0,7$) amaç ağırlıkları değerlerinin Pareto çözüm elde etme frekansı en yüksektir. Bu nedenle skalerleştirme yaklaşımlarına dayanan genetik algoritmanın performanslarını karşılaştırılırken Pareto çözüm elde etme frekansı en yüksek olan KKV kombinasyonları ve amaç ağırlıkları dikkate alınmıştır. Şekil 4.12, KKV kombinasyonları ve amaç ağırlıkları için frekans grafiklerini göstermektedir.

Şekil 4.12: KKV Kombinasyonları ve Amaç Ağırlıkları için Frekans Grafikleri



a) KKV Kombinasyonları için Frekans Grafikleri

b) Amaç Ağırlıkları için Frekans Grafikleri

Farklı büyüklüklere sahip amaç fonksiyonları ilgili amacın nadir değerine bölünerek normalize edilip birimlerinden arındırılmaktadır. Bu çalışmada, skalerizasyon

yöntemleri ile önerilen iki amaçlı genetik algoritmada uygunluk değeri olarak normalize edilmiş amaç fonksiyonu değerleri kullanılmıştır. Bu eşitliklerde $N_i, i.$ amaç fonksiyonunun nadir noktasını temsil etmektedir. KSY ve CSY tarafından skalerize edilen uygunluk değerleri, Eşitlik (4.32) -(4.33)'de verildiği gibi hesaplanmıştır.

$$enk\ KSY^{Norm}(x) = \left[w_1 \left(\frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1}{N_1} \right) + w_2 \left(\frac{\sum_{i \in N_c} T_i - RfP_2}{N_2} \right) + \alpha \left(\left| \frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1}{N_1} \right| + \left| \frac{\sum_{i \in N_c} T_i - RfP_2}{N_2} \right| \right) \right] \quad (4.32)$$

$$enk\ CSY^{Norm}(x) = enb \left[w_1 \left(\frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1}{N_1} \right), w_2 \left(\frac{\sum_{i \in N_c} T_i - RfP_2}{N_2} \right) \right] + \left[\rho \left(w_1 \left(\frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1}{N_1} \right) + w_2 \left(\frac{\sum_{i \in N_c} T_i - RfP_2}{N_2} \right) \right) \right] \quad (4.33)$$

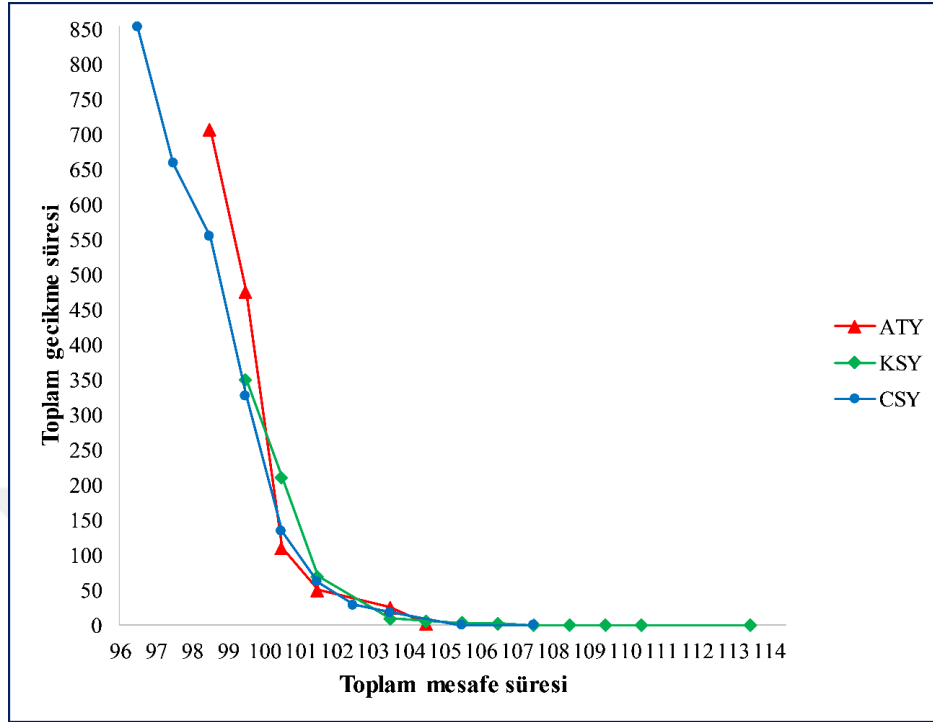
Farklı amaç ağırlıkları ve farklı referans noktaları için skalerleştirme yöntemlerinin amaç fonksiyonlarına etkisi Tablo 4.8'de verilmiştir.

Tablo 4.8: Skalerleştirme Yöntemlerinin Amaç Fonksiyonlarına Etkisi

No	ATY				KSY						CSY					
	w1	w2	TD	TT	w1	w2	RP1	RP2	TD	TT	w1	w2	RP1	RP2	TD	TT
1	0,9	0,1	98	707	0,7	0,3	96	270,9	99	350	0,9	0,1	85,5	0	96	854
2	0,9	0,1	99	476	0,9	0,1	96	270,9	100	211	0,8	0,2	85,5	0	97	658
3	0,8	0,2	100	110	0,9	0,1	96	270,9	101	69	0,9	0,1	96	270,9	98	554
4	0,9	0,1	101	50	0,3	0,7	104	30,1	103	9	0,8	0,2	85,5	0	99	327
5	0,2	0,8	103	25	0,3	0,7	100	150,5	104	6	0,9	0,1	98	210,7	100	134
6	0,8	0,2	104	2	0,2	0,8	104	30,1	105	3	0,9	0,1	99	180,6	101	61
7					0,4	0,6	104	30,1	106	2	0,8	0,2	99	180,6	102	29
8					0,1	0,9	100	150,5	107	0	0,9	0,1	104	30,1	103	18
9					0,1	0,9	104	30,1	108	0	0,1	0,9	104	30,1	105	0
10					0,3	0,7	104	30,1	109	0	0,4	0,6	104	30,1	107	0
11					0,1	0,9	104	30,1	110	0						
12					0,1	0,9	100	150,5	113	0						

Şekil 4.13, skalerleştirme yöntemlerinden elde edilen amaç değerleri sonucuna dayanan Pareto etkin çözümlerin grafiğidir. Ele alınan problemde amaçların toplam seyahat ve gecikme süresinin enküçüklenmesi olması sebebiyle Pareto grafiğinde de görüldüğü üzere bu problem özelinde CSY kullanılan diğer skalerleştirme yöntemlerine (ATY ve KSY) göre daha iyi sonuçlar vermiştir. İki amacın da en küçükleme olması sebebiyle daha dışta kalan Pareto yüzeyde sonuçlar bulan skalerleştirme yöntemi diğer yöntemlere tercih edilmektedir.

Şekil 4.13: ATY, KSY ve CSY'den Elde Edilen Amaç Değerleri Sonucuna Dayanan Pareto Etkin Çözümler



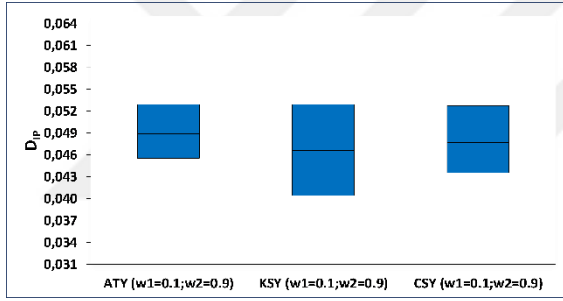
İdeal noktalara ağırlıklı uzaklık (D_{IP}) değeri, normalize edilmiş amaç fonksiyonu değerinden ideal noktaya öklid mesafesi ile amaç değeri önem ağırlığının çarpımı ile elde edilmiştir. Referans noktalara ağırlıklı uzaklık (D_{RfP}) değeri ise, referans noktaya öklid mesafesi ile amaç değerinin önem ağırlığının çarpımı ile elde edilmiştir. D_{IP} ve D_{RfP} değerleri, Eşitlik (4.34) -(4.35) kullanılarak hesaplanmıştır. Bu eşitliklerde I_i , i . amaç fonksiyonunun ideal noktasını temsil etmektedir.

$$D_{IP} = \sqrt{w_1 \left(\frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - I_1}{N_1} \right)^2 + w_2 \left(\frac{\sum_{i \in N_c} T_i - I_2}{N_2} \right)^2} \quad (4.34)$$

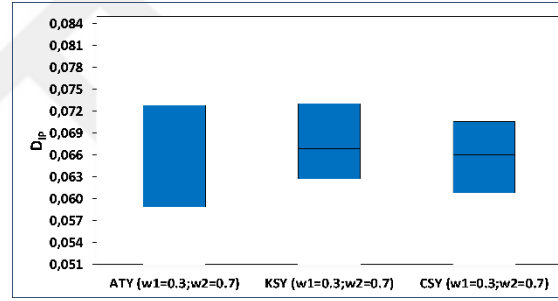
$$D_{RfP} = \sqrt{w_1 \left(\frac{\sum_{i,j \in N} \sum_{r \in R} t_{ij} X_{ijr} - RfP_1}{N_1} \right)^2 + w_2 \left(\frac{\sum_{i \in N_c} T_i - RfP_2}{N_2} \right)^2} \quad (4.35)$$

Şekil 4.14 skalerleştirme yöntemlerinin ideal noktalara olan ağırlıklı mesafesinin ait kutu-bıyık grafiğini, Şekil 4.15 skalerleştirme yöntemlerinin referans noktalara olan ağırlıklı mesafesinin ait kutu-bıyık grafiğini verilmektedir. Kutu-bıyık grafiğinden de anlaşılacağı üzere frekansı en yüksek olan KKV kombinasyonları ve amaç ağırlıkları için, bu problem özelinde CSY kullanılan diğer skalerleştirme yöntemlerine (ATY ve KSY) göre daha iyi sonuçlar vermiştir

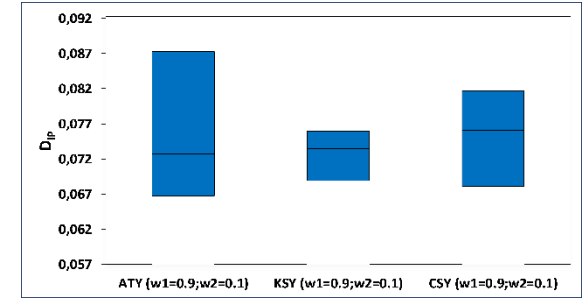
Şekil 4.14: Üç Farklı Konveks Kombinasyon Vektörüne Sahip Üç Kıyaslama Örneği için Skalerizasyon Yöntemlerine Dayalı Çok Amaçlı Genetik Algoritmanın 31 Çalışmasından Elde Edilen İdeal Noktalara Olan Ağırlıklı Mesafenin Kutu-Bıyık Grafikleri



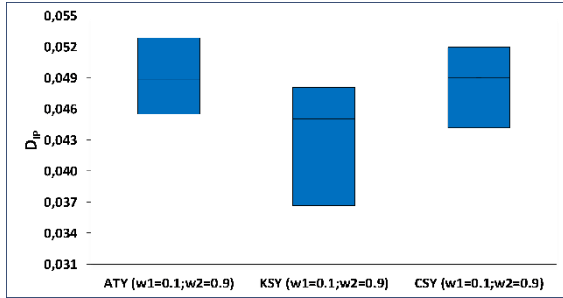
(a) KKV₂ için Örnek#1



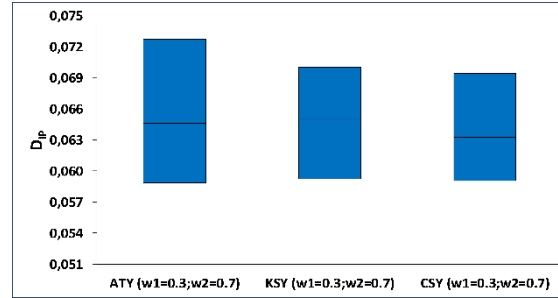
(b) KKV₂ için Örnek#3



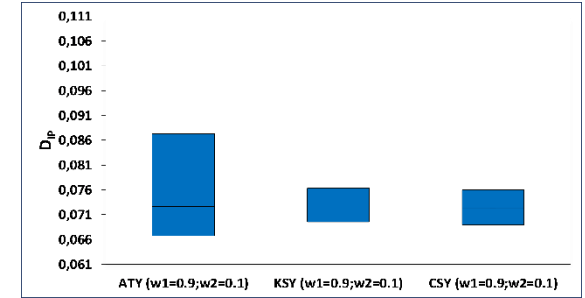
(c) KKV₂ için Örnek#9



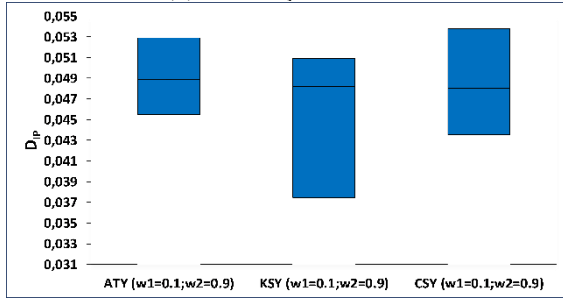
(d) KKV₅ için Örnek#1



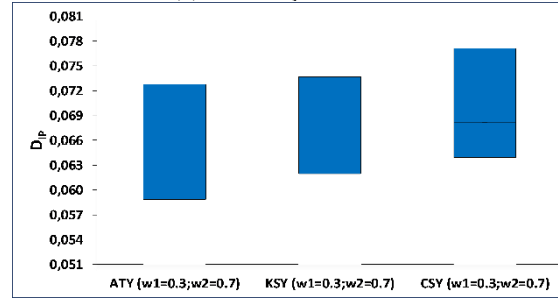
(e) KKV₅ için Örnek#3



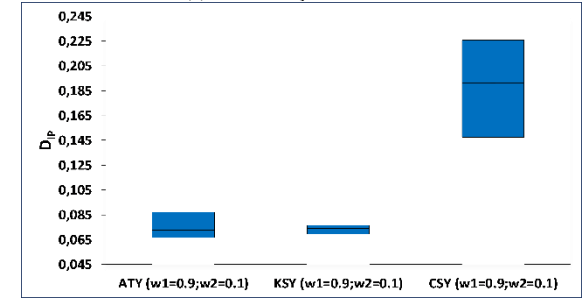
(f) KKV₅ için Örnek#9



(g) KKV₁₁ için Örnek#1

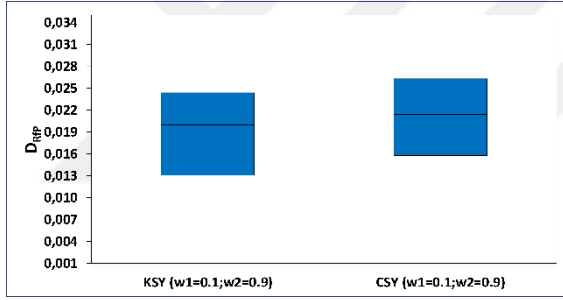


(h) KKV₁₁ için Örnek#3

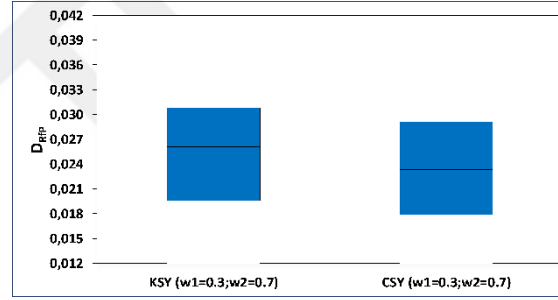


(i) KKV₁₁ için Örnek#9

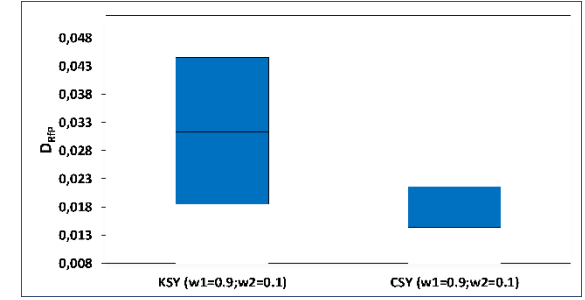
Şekil 4.15: Üç Farklı Konveks Kombinasyon Vektörüne Sahip Üç Kıyaslama Örneği için Skalerizasyon Yöntemlerine Dayalı Çok Amaçlı Genetik Algoritmanın 31 Çalışmasından Elde Edilen Referans Noktalara Olan Ağırlıklı Mesafenin Kutu-Bıyık Grafikleri



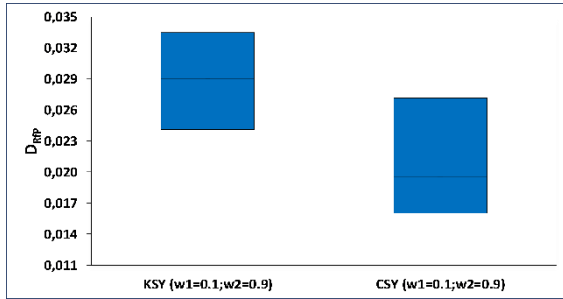
(a) KKV_2 için Örnek#1



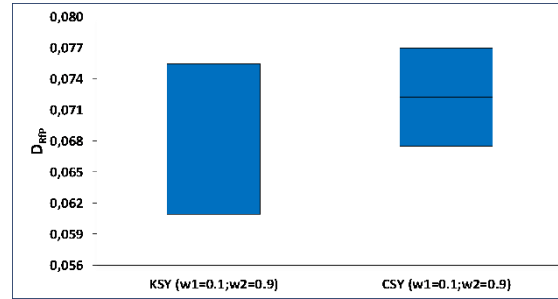
(b) KKV_2 için Örnek#3



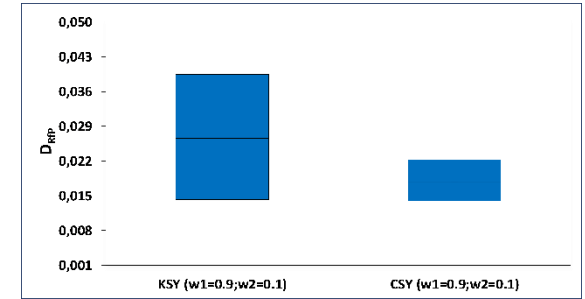
(c) KKV_2 için Örnek#9



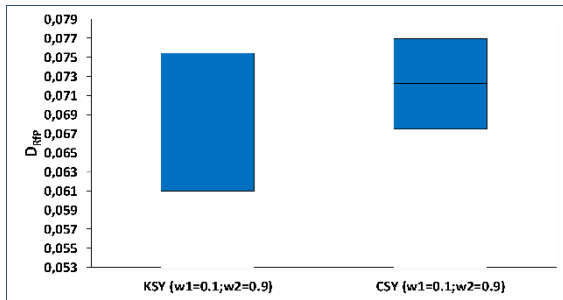
(d) KKV_5 için Örnek#1



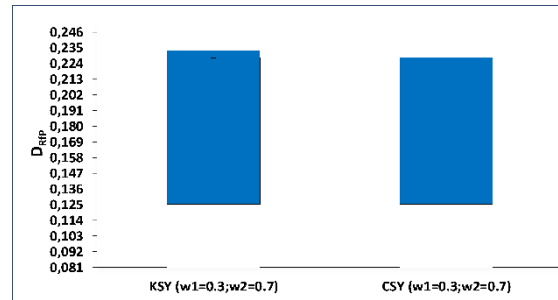
(e) KKV_5 için Örnek#3



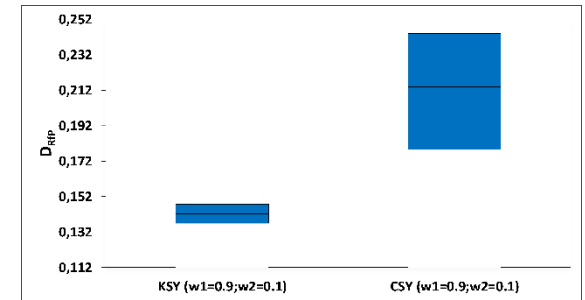
(f) KKV_5 için Örnek#9



(g) KKV_{11} için Örnek#1



(h) KKV_{11} için Örnek#3



(i) KKV_{11} için Örnek#9

SONUÇLAR VE ÖNERİLER

Rekabetin hızla arttığı günümüz dünyasında firmaların devamlılığını sürdürebilmesi için üretimdeki israfları enküçüklemesi en önemli faktörlerden biridir. Üretimdeki israfların büyük bir kısmını da çeşitli duruşlar oluşturmaktadır. Bu çalışmada İstanbul'da faaliyet gösteren laboratuvar sarf malzemeleri üreten bir firmanın duruşları incelenmiş ve duruşların engellenmesi için gerekli olan su örümceğinin rota probleminin ZPARP olduğu belirlenmiştir. Amaç, su örümceğinin istasyon zaman pencerelerini dikkate alacak şekilde rotalar oluşturup toplam kat ettiği mesafe süresini ve makine duruşlarını enküçükleyecek hatta mümkünse sıfırlayacak şekilde istasyonlara gecikme süresini azaltmaktır. Öncelikle problemin matematiksel modeli oluşturulmuştur ve oluşturulan model Lingo 11.0 programı kullanılarak küçük bir örnek üzerinde test edilmiş ve doğrulanmıştır. Örnek boyutu arttığında optimizasyon programının problemin çözümü için makul sürelerde çözüm üretememesi ve problemin literatürde NP-zor sınıfına girmesi sebebiyle ele alınan problemin çözümü için iki amaçlı genetik algoritma önerilmiştir.

Önerilen genetik algoritmanın parametrelerinin ayarlanabilmesi için **irace** ve DoE kullanılmıştır. Elde edilen parametreler ile ele alınan problem çözülerek sonuçlar kıyaslanmış ve DoE sonucu üretilen parametreler kullanılarak elde edilen sonuçların **irace**'den daha iyi sonuç ürettiği gözlemlenmiştir. Ayrıca önerilen iki amaçlı genetik algortmada amaçlar ATY, KSY ve CSY kullanılarak 3 farklı şekilde hesaplanmıştır. Her bir yöntem için algoritma 31 kez çalıştırılmıştır. Kullanılan performans metriklerine göre bu problem özelinde CSY, diğer sklerleştirme yöntemlerine göre daha iyi sonuç vermiştir. Dolayısıyla bu yöntemin en fazla Pareto çözüm elde ettiği KKV₂ kombinasyonu ve $w_1 = 0,9$; $w_2 = 0,1$ ağırlıkları sonucu elde edilen sonuçla mevcut durum kıyaslanmıştır. Mevcut durumda her operatör kendi YM veya HM'sini tedarik ettiğinden dolayı oluşan karmaşa su örümceğinin toplam 105 dakika hareket süresi ile ortadan kaldırılmıştır. Ayrıca mevcut durumda her operatör kendi YM veya HM'sini tedarik ettiğinden dolayı oluşan duruş süreleri su örümceğinin izleyeceği, gecikmesi 0 olan rota ile önlenmiştir.

Sonuçlar, modelin uygulanabilir olduğunu ve firmanın önerilen rotada kullanabileceği bir su örümceği ile HM ve YM tedarikinden kaynaklı duruşlarını engelleyerek kullanılabilirlik oranını arttıracığını göstermektedir.

Bu çalışmadan elde edilen veriler ve kullanılan varsayımlar altında çok kullanımlı zaman pencereli araç rotalama problemi ile ilgili gelecek çalışmalarda, ele

alınan problemde rotalama alan darlığından dolayı bir su örümceği üzerinden yapılmıştır, birden fazla su örümceği için önerilen model denenebilir. Yeni bir amaç olarak su örümceğinin istasyonlara erken gitmesi durumunda oluşan bekleme zamanının enküçüklenmesi modele eklenebilir. Çalışmada ele alınan firmada üretilen ürünlerin boyutsal olarak küçük olmasından dolayı indirme ve bindirme süreleri dikkate alınmamıştır, gelecek çalışmalarda indirme ve bindirme süreleri dikkate alınabilir. Farklı parametre ayarlama yöntemleri ile önerilen algoritmanın parametreleri belirlenip sonuçlar kıyaslanabilir. Çözüm için benzetim tavlama, memetik algoritma, tabu arama gibi farklı algoritmalar kullanılabilir.



KAYNAKÇA

- Alvarez, A., & Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers and Operations Research*, 83, 1-12.
- Anand, S., Afreen, N., & Yazdani, S. (2015). A novel and efficient selection method in genetic algorithm. *International Journal of Computer Applications*, 129(15), 7-12.
- Angel-Bello, F., Martinez-Salazar, I., & Alvarez, A. (2013). Minimizing waiting times in a route design problem with multiple use of a single vehicle. *Electronic Notes in Discrete Mathematics*, 41, 269-276.
- Anggodo, Y. P., vd. (2016). Optimization of multi-trip vehicle routing problem with time windows using genetic algorithm. *Journal of Environmental Engineering & Sustainable Technology*, 3(2), 92-97.
- Asghari, M., & Al-e, S. M. J. M. (2021). Green vehicle routing problem: A state-of-the-art review. *International Journal of Production Economics*, 231, 1-66.
- Ayadi, R., vd. (2014, 10 June). *Evolutionary algorithm for a green vehicle routing problem with multiple trips*. Retrieved from: https://www.researchgate.net/publication/271544656_Evolutionary_algorithm_for_a_Green_vehicle_routing_problem_with_multiple_trips.
- Beg, S., vd. (2019). *Application of design of experiments (DoE) in pharmaceutical product and process optimization* (1st ed.). London: Academic Press.
- Cáceres, L. P., vd. (2017, 20 July). *An experimental study of adaptive capping in irace*. Retrieved from: <https://ada.liacs.nl/papers/CacEtA117.pdf>.
- Cai, Y., vd. (2022). A hybrid evolutionary multitask algorithm for the multiobjective vehicle routing problem with time windows. *Information Sciences*, 612, 168-187.
- Cintrano, C., Chicano, F., & Alba, E. (2020). Using metaheuristics for the location of bicycle stations. *Expert Systems with Applications*, 161, 113-154.
- Colmenar, J. M., Martí, R., & Duarte A. (2018). Heuristics for the bi-objective diversity

- problem. *Expert Systems With Applications*, 108, 193-205.
- Cömert, S. E., vd. (2017). A new approach for solution of vehicle routing problem with hard time window: An application in a supermarket chain. *Sādhanā*, 42, 2067-2080.
- Çanakçıoğlu, M. (2019). Yalın düşünce felsefesinde israfla mücadele araçları. *Sosyal Bilimler Araştırma Dergisi*, 8(3), 270-282.
- Çolak, S. (2010). Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama. *Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi*, 19(3), 423-438.
- Deliktaş, D., & Aydın, D. (2023). An artificial bee colony based-hyper heuristic algorithm with local search for the assembly line balancing problems. *Engineering Computations*, 40(9/10), 2453-2482.
- Deliktaş, D., & Ustun, O. (2023). Multi-objective genetic algorithm based on the fuzzy MULTIMOORA method for solving the cardinality constrained portfolio optimization. *Applied Intelligence*, 53(12), 14717-14743.
- Deliktaş, D., vd. (2021). Evolutionary algorithms for multi-objective flexible job shop cell scheduling. *Applied Soft Computing*, 113, 107-120.
- Deng, Q., vd. (2017). A bee evolutionary guiding nondominated sorting genetic algorithm II for multiobjective flexible job-shop scheduling. *Computational Intelligence and Neuroscience*, 1, 523-518.
- Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL)*, 37(3), 383-402.
- Durakovic, B. (2017). Design of experiments application, concepts, examples: State of the art. *Periodicals of Engineering and Natural Sciences*, 5(3), 421-439.
- Elen, A. (2014, 13 April). *Genetik algoritmanın genel akış şeması*. Erişim Adresi: https://tr.wikipedia.org/wiki/Genetik_algoritma#/media/Dosya:GAelen.jpg.
- Erozan, İ. (2017). *Geleneksel ve modern üretim sistemleri* (1. Basım). Kütahya: Monopol Kırtasiye Kitabevi.
- Ersöz, T., Sarız, K. ve Ersöz, F. (2020). Demir-çelik üretim hattında yalın üretim. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8, 801-826.

- Expósito, A., vd. (2019). Quality of service objectives for vehicle routing problem with time windows. *Applied Soft Computing*, 84, 105-127.
- Eydi, A., & Ghasemi-Nezhad, S. A. (2021). A bi-objective vehicle routing problem with time windows and multiple demands. *Ain Shams Engineering Journal*, 12, 2617-2630.
- Giagkiozis, I., & Fleming, P. J. (2015). Methods for multi-objective optimization: An analysis. *Information Sciences*, 293, 338-350.
- Gutierrez, A., vd. (2018). A multi-population algorithm to solve the VRP with stochastic service and travel times. *Computers & Industrial Engineering*, 125, 144-156.
- Huang, C., Li, Y., & Yao, X. (2020). A survey of automatic parameter tuning methods for metaheuristic. *IEEE Transactions on Evolutionary Computation*, 24(2), 201–216.
- Kalayci, T.E. (2016, 20 June). *Solving GENOPT functions with the reactive affine shaker using irace for parameter tuning*. Retrieved from: <https://tekrei.gitlab.io/presentations/2016-GenOpt-LION-Sunum.pdf>.
- Kallehauge, B., Larsen, J., & Madsen, O. B. (2005). Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 33(5), 1464-1487.
- Kasimbeyli, R., vd. (2019). Comparison of some scalarization methods in multiobjective optimization: comparison of scalarization methods. *Bulletin of the Malaysian Mathematical Sciences Society*, 42, 1875-1905.
- Katoch, S., Chauhan, S.S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80, 8091–8126.
- Ke-Wei, J., San-Yang, L., & Xiao-Jun, S. (2022). A hybrid algorithm for time-dependent vehicle routing problem with soft time windows and stochastic factors. *Engineering Applications of Artificial Intelligence*, 109, 104-126.
- Kırcı, P. (2016). An optimization algorithm for a capacitated vehicle routing problem with time windows. *Sadhana*, 41(5), 519–529.
- Kim J., & Kim S. K. (2006). A chim-based interactive tchebycheff procedure for multiple objective decision making. *Computers & Operations Research*, 33(6), 1557–

1574.

- Klenk E., Galka, S., & Günthner W. A. (2015). Operating strategies for in-plant milk-run system. *IFAC-PapersOnLine*, 48(3), 1882–1887.
- Koç, Ç., & Karaoğlan, İ. (2012). Çok kullanımlı ve zaman pencereli araç rotalama problemi için bir matematiksel model. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 27(3), 569-576.
- Koç, Ç., & Laporte, G. (2018). Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*, 91, 79-91.
- Kumar, S. N., & Panneerselvam, R. (2017). Development of an efficient genetic algorithm for the time dependent vehicle routing problem with time windows. *American Journal of Operations Research*, 7, 1-25.
- Kumar, V. S., vd. (2014). Solving multi-objective vehicle routing problem with time windows by FAGA. *Procedia Engineering*, 97, 2176-2185.
- Kumari, M., vd. (2023). Utilizing a hybrid metaheuristic algorithm to solve capacitated vehicle routing problem. *Results in Control and Optimization*, 13, 100-122.
- Kurada, G., & Deliktaş, D. (2024). Multi-objective genetic algorithm for the assembly line worker assignment and balancing problem: A case study in the automotive supply industry. *Journal of Scientific Reports-A*, 56, 3-22.
- Küçükoğlu, İ., & Öztürk, N. (2015). An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows. *Computers & Industrial Engineering*, 86, 60-68.
- Lee, K. M., Yamakawa, T., & Lee, K. M. (1998, 13 May). *A genetic algorithm for general machine scheduling problems*. Retrieved from: <https://www.scribd.com/document/495546052/KES-1998-725893>.
- Lin, C., vd. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems With Applications*, 41(4), 1118-1138.
- Long, J., vd. (2019). A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. *Information Sciences*, 478, 40-61.
- López-Ibáñez, M., vd. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43-58.

- Marler, R. T., & Arora, J. S. (2005). Function-transformation methods for multi-objective optimization. *Engineering Optimization*, 37(6), 551-570
- Maroof, A., Ayvaz, B., & Naeem, K. (2024, 20 April). *Logistics optimization using hybrid genetic algorithm (HGA): A solution to the vehicle routing problem with time windows (VRPTW)*. Retrieved from: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10459172>.
- Massen, F., vd. (2013). Experimental analysis of pheromone-based heuristic column generation using irace. In *International workshop on hybrid metaheuristics* (pp. 92-106). Berlin: Springer Heidelberg.
- Melián-Batista, B., vd. (2014). A bi-objective vehicle routing problem with time windows: A real case in Tenerife. *Applied Soft Computing*, 17, 140-152.
- Montgomery, D.C. (2008). *Design and analysis of experiments* (8th ed.). Hoboken: John Wiley & Sons.
- Mosayebi, M., & Sodhi, M. (2020, 7 June). *Tuning genetic algorithm parameters using design of experiments*. Retrieved from: https://www.researchgate.net/publication/342790193_Tuning_genetic_algorithm_parameters_using_design_of_experiments.
- Nalepa, J., & Blocho, M. (2015). Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. *Soft Computing*, 20, 2309–2327.
- Nemoto, T., Hayashi, K., & Hashimoto M. (2010). Milk-Run logistics by Japanese automobile manufacturers in Thailand. *Procedia Social and Behavioral Sciences*, 2, 5980–5989.
- Oosthuizen, N. M. (2018). *A decision support framework towards a simulation model for the risk-constrained vehicle routing problem* (Unpublished Doctoral Thesis). Stellenbosch University, South Africa.
- Ömürgönülşen, M., & Çatman R. (2018). Bir kamu kurumunda değer akış haritalama ve simülasyon yöntemiyle hizmet sürelerinin değerlendirilmesi. *Finans Politik & Ekonomik Yorumlar Dergisi*, 55(636), 47-70.
- Pan, B., Zhang, Z., & Lim, A. (2021). A hybrid algorithm for time-dependent vehicle routing problem with time windows. *Computers and Operations Research*, 125, 105-143.

- Pavai, G., & Geetha, T. V. (2016). A survey on crossover operators. *ACM Computing Surveys (CSUR)*, 49(4), 1-43.
- Rabbani, M., vd. (2017, 3 December). *Using meta-heuristic algorithms and hybrid of them to solve multi compartment Vehicle Routing Problem*. Retrieved from: https://www.researchgate.net/publication/323135715_Using_metaheuristic_algorithms_and_hybrid_of_them_to_solve_multi_compartment_Vehicle_Routing_Problem.
- Rivera, J. C., Afsar, H. M., & Prins, C. (2016). Mathematical formulations and exact algorithm for the multitrip cumulative capacitated single-vehicle routing problem. *European Journal of Operational Research*, 249, 93–104.
- Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic algorithms. *Search methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, 3, 97-125.
- Sipahioğlu A., & Altın, İ. (2019). A mathematical model for in-plant milk-run routing. *Pamukkale Üniversitesi Mühendislik Bilim Dergisi*, 25(9), 1050-1055.
- Souza, M. D., Ritt, M., & López-Ibáñez, M. (2022). Capping methods for the automatic configuration of optimization algorithms. *Computers and Operations Research*, 139, 105-115.
- Souza, M., & Ritt, M. (2016, 3 January). *A study of the automatic design of heuristics for binary quadratic programming*. Retrieved from: <http://www.din.uem.br/~ademir/sbpo/sbpo2016/pdf/156632.pdf>.
- Su, X., vd. (2023). A branch-and-price-and-cut for the manpower allocation and vehicle routing problem with staff qualifications and time windows. *Advanced Engineering Informatics*, 57, 102-143.
- Sung, K., vd. (2000). Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research*, 121(1), 32-39.
- Şeker, A. (2016). Yalın üretim sisteminde kanban, tek parça akışı ve U tipi yerleştirme sistemleri. *Akademik Sosyal Bilimler Dergisi*, 50, 449-470.
- Tarantilis, C. D., vd. (2004). A threshold accepting approach to the open vehicle routing problem. *RAIRO-Operations Research*, 38(4), 345-360.

- Toth, P., & Vigo, D. (2002, 4 May). *The vehicle routing problem*. Retrieved from: <https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/002-The-Vehicle-Routing-Problem-Monograf-on-discrete-mathematics-and-applications-Paolo-Toth-Daniele-Vigo-Edisi-1-2002.pdf>.
- Tunç, M. (2019, 13 Şubat). *Genetik algoritma (örnek uygulama ve kodları)*. Erişim Adresi: <https://mlhtnc.github.io/genetic-algorithm.html>.
- Victoria, J. F., Afsar, H. M., & Prins C. (2016). Column generation based heuristic for the vehicle routing problem with time-dependent demand. *FAC-PapersOnLine*, 49(12), 526-531.
- Wang, C. H., & Lu, J. Z. (2009). A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert Systems with Applications*, 36(2), 2921-2936.
- Wang, Y., vd. (2023). A clustering-based extended genetic algorithm for the multidepot vehicle routing problem with time windows and three-dimensional loading constraints. *Applied Soft Computing*, 133, 109-122.
- Wang, Z., Li, Y., & Hu X. (2015). A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint. *Computers & Industrial Engineering*, 89, 162-176.
- Wu, H., & Gao, Y. (2023). An ant colony optimization based on local search for the vehicle routing problem with simultaneous pickup–delivery and time window. *Applied Soft Computing*, 139, 1-16.
- Yadav, S. L., & Sohal, A. (2017). Comparative study of different selection techniques in genetic algorithm. *International Journal of Engineering, Science and Mathematics*, 6(3), 174-180.
- Yalın'da Su Örümceği Nedir?. (2020, 7 Temmuz). Erişim Adresi: <https://kanbantool.com/kanban-guide/water-spider>.
- Yan, S., vd. (2015). A planning model and solution algorithm for multi-trip split-delivery vehicle routing and scheduling problems with time windows. *Computers & Industrial Engineering*, 87, 383-393.
- Yang, X. S. (2020). *Nature-inspired optimization algorithms* (1st ed.). London: Academic Press.

- Yaşar, C., & Özyön, S. (2018). A new approach in the solution of the environmental economic power dispatch problems with piecewise quadratic fuel cost function: Conic scalarization method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33(2), 541-556.
- Yavuz, G. (2022). L-shade algoritmasının otomatik parametre yapılandırma yöntemi ile iyileştirilmesi. *Bilişim Teknolojileri Dergisi*, 15(2), 189-197.
- Yesodha, R., & Amudha, T. (2022). A bio-inspired approach: Firefly algorithm for multi-depot vehicle routing problem with time windows. *Computer Communications*, 190, 48-56.
- Yükselen, C. (2024, 7 Nisan). *OEE nedir? Nasıl uygulanır ve hesaplanır?*. Erişim Adresi: <https://www.yalindanisman.com/oe-e-nedir/>
- Zhang, D., vd. (2017). A hybrid algorithm for a vehicle routing problem with realistic constraints. *Information Sciences*, 394, 167-182.

DİZİN

-A-

ARP, xiii, 1, 4, 6, 7, 8, 9, 11, 13, 14, 15,
16, 17, 26, 31

-Ç-

Çaprazlama, xi, 24, 45, 49, 50, 52, 53,
54
ÇKZPARP, xiii, 1, 4

-D-

Duruş süresi, 37

-G-

Genetik algoritma, v, 2, 22, 23, 24, 31,
45, 48, 51, 65, 70
Gerçek hayat problemi, 9, 10, 11

-İ-

İdeal nokta, 59
irace, v, vi, x, xi, 2, 25, 26, 27, 28, 29,
51, 53, 54, 55, 56, 62, 64, 66, 68
İstasyon, 40

-K-

Kromozom, ix, xi, 23, 45, 46
Kullanılabilirlik, xi, 36, 37

-M-

Matematiksel model, 39

-N-

Nadir değer, 51

-P-

Parametre, viii, ix, x, xi, 23, 25, 26, 51,
53, 54
Parametre ayarlama, 25

-R-

Referans nokta, 56, 59

-S-

Seçim, ix, 24, 45, 48
Skalerleştirme, x, xiii, 42, 56, 58
Su örümceği, v, 2, 18, 19, 20, 38, 39, 46

-T-

Turnuva büyüklüğü, 45

-U-

Uygunluk değeri, 48

-Z-

Zaman penceresi, 14
ZPARP, v, xiii, 1, 4, 5, 6, 7, 8, 9, 11, 17,
18, 32, 44, 62