

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

Ümit YILDIRIM

**KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMİNİN
YABANI OT VE HİBRİT METASEZGİSEL
ALGORİTMALARLA ÇÖZÜMÜ**

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

ADANA, 2019

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMİNİN YABANI OT
VE HİBRİT METASEZGİSEL ALGORİTMALARLA ÇÖZÜMÜ**

Ümit YILDIRIM

YÜKSEK LİSANS TEZİ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Bu Tez 29.07.2019 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından
Oybirliği/Oyçokluğu ile Kabul Edilmiştir.

.....
Dr. Öğr. Üyesi Yusuf KUVVETLİ
DANIŞMAN

Prof. Dr. Rızvan EROL
ÜYE

Doç. Dr. Cihan ÇETİNKAYA
ÜYE

Bu Tez Enstitümüz Endüstri Mühendisliği Anabilim Dalında hazırlanmıştır.
Kod No:

Prof. Dr. Mustafa GÖK
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZ

YÜKSEK LİSANS TEZİ

KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMİNİN YABANI OT
VE HİBRİT METASEZGİSEL ALGORİTMALARLA ÇÖZÜMÜ

Ümit YILDIRIM

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Danışman :Dr. Öğretim Üyesi Yusuf KUVVETLİ
Yıl: 2019, Sayfa: 151
Jüri :Dr. Öğretim Üyesi Yusuf KUVVETLİ
:Prof. Dr. Rızvan EROL
:Doç. Dr. Cihan ÇETİNKAYA

Lojistik planlamada araç rotalama problemleri operasyonel bazda firmalar için önemli kararlardan birisini oluşturmaktadır. Bu tez çalışmasında; özellikle lojistik planlarının oluşturulmasında sıklıkla karşılaşılan araç rotalama problemi incelenmiştir. Araç rotalama problemi Np-Zor sınıfındaki popüler bir optimizasyon problemidir. Problemin karakteristiğinden dolayı büyük ölçekli veri setlerini kesin çözüm yöntemler ile çözmek çok zordur. Makul sürelerde optimale yakın çözümler için sezgisel ve metasezgisel algoritmalar yaygın olarak kullanılmaktadır. Bundan dolayı bu çalışmada araç rotalama problemi; tasarruf algoritması ve genetik algoritmayla çözümlenmesinin yanında yabancı ot algoritması ve bu yaklaşımlardan oluşturulan hibrit yöntemler ile çözümlenerek bu algoritmaların performansları karşılaştırılmıştır. Böylelikle, kapasite kısıtlı statik araç rotalama problemine yeni bir hibrit çözüm yöntemi önerilmiştir. Sonuçlar önerilen yaklaşımın kısa sürede optimale oldukça yakın sonuçlar verdiğini göstermektedir. Buna ek olarak problemin dinamik koşullar altında davranışını incelemek için tek depolu, çok araçlı, dinamik talepli, kapasite kısıtlı araç rotalama problemi ele alınmıştır. Statik problem için önerilen çözüm yaklaşımları dinamik koşullara adapte edilmiştir. Problemin uygulama problemlerine implementasyonu için çalışma sonucunda ortaya çıkan çözüm yaklaşımlarını temel alan bir arayüz programı tasarlanmıştır.

Anahtar Kelimeler: Dinamik araç rotalama problemi, tasarruf algoritması, genetik algoritma, yabancı ot algoritması, kapasite kısıtlı araç rotalama

ABSTRACT

MSc THESIS

SOLUTION OF CAPACITATED VEHICLE ROUTING PROBLEM WITH INVASIVE WEED AND METAHEURISTIC ALGORITHMS

Ümit YILDIRIM

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF INDUSTRIAL ENGINEERING

Supervisor :Asst. Prof. Dr. Yusuf KUVVETLİ
Year: 2019, Pages: 151
Jury :Asst. Prof. Dr. Yusuf KUVVETLİ
:Prof. Dr. Rızvan Erol
:Assoc. Prof. Dr. Cihan ÇETİNKAYA

Vehicle routing problems in logistics planning constitute one of the important operational decisions for companies. In this thesis, the vehicle routing problem that is frequently faced in the building of logistics plans has been examined. Vehicle routing problem is a popular problem in optimization which is known as NP-Hard class problem. Due to the characteristics of the problem, it is very difficult to solve large-scale data instances with exact solution methods. Heuristic and meta-heuristic algorithms are widely used for achieving near optimal solutions at reasonable time. Therefore, in this study, vehicle routing problem is solved by savings algorithm, genetic algorithm, invasive weed optimization algorithm and hybrid methods that developed by these approaches and their performances are compared. Thus, a new hybrid solution method has been proposed for the capacitated vehicle routing problem. The results show that the proposed approach is very close to optimal results in a short computational time. In addition, in order to investigate the behavior of the problem under dynamic conditions, a single-depot, multi-vehicle, dynamic demand, capacity-constrained vehicle routing problem is considered. The proposed solution approaches for classic problem are adapted to dynamic conditions. In order to implement the problem into the real-life application problems, a graphical user interface is designed based on the proposed solution approaches.

Keywords: Dynamic vehicle routing problem, savings algorithm, genetic algorithm, invasive weed optimization algorithm, capacitated vehicle routing

GENİŞLETİLMİŞ ÖZET

Firmalar genellikle dağıtım işlemini geçmiş tecrübelerinden yararlanarak yapmaktadır. Etkin olmayan dağıtım stratejileri firmalara ek maliyet oluşturmaktadır. Araç rotalama problemi ile optimum rotalar oluşturularak dağıtım maliyetleri düşürülmeye çalışılır. Aynı zamanda optimum rotaların oluşturulması müşteriye ulaşma süresini kısaltacağı için, müşteri memnuniyetini ve firmaların verimini arttıracaktır. Bu anlamda, araç rotalama problemi dağıtım ve benzeri işlemlerle yakın ilişkili firmalar için önemli bir konudur. Müşterilerine araç filosuyla hizmet veren her kuruluş ve firma araç rotalama problemi ile karşı karşıya kalmaktadır. Uygun rotalama stratejileri olan firmalar maliyetlerini düşürerek günümüzün rekabetçi ortamında avantaj sağlarlar.

Araç rotalama problemleri en uygun maliyetli taşıma planlarının oluşturulmasıyla ilgilidir. Buna ek olarak, araç kapasitelerinin ve rota uzunluklarının sınırlı olması, depoya uğrama gibi durumların göz önüne alınması bu problemi zorlaştırmaktadır. Klasik problemlere ek olarak dinamik versiyonun göz önüne alınması problemi daha da zorlaştırmaktadır.

Dinamik araç rotalama problemleri, araç rotalama problemlerinin gerçek hayat uygulamalarına daha yakın bir versiyonudur. Araç rotalama problemlerinde probleme ait bilgiler planlama yapılırken bilinmektedir ve planlama boyunca değişmemektedir. Dinamik araç rotalama probleminde ise planlama dönemi boyunca tüm bilgiler bilinmeyebilir veya süreç boyunca değişiklik oluşabilir. Bilgilerin değişmesi araç rotalarının güncellenmesi gerektiği anlamına gelir. Dinamikliği sağlayacak faktörler çok farklı olabileceği gibi trafiğin yoğun olması, yeni gelen siparişler, iptal edilen siparişler gibi dinamik durumlar önceden belirlenmiş olan rotanın güncellenmesini gerektirir.

Bu çalışmada ele alınan problem tek depolu, çok araçlı, kapasite kısıtlı araç rotalama problemidir. Karakteristikleri belirtilen klasik problemin çözümü için

tasarruf algoritması, genetik algoritma, yabancı ot algoritması kullanılmıştır ve bu algoritmalarla oluşturulan hibrit yaklaşım önerilmiştir. İlk olarak problemin çözümü için tasarruf algoritması kullanılmıştır. Tasarruf algoritmasının paralel versiyonu tercih edilmiştir. Sonrasında genetik algoritma ile probleme çözüm aranmıştır. Probleme özgü olarak kromozom yapısı, çaprazlama ve mutasyon belirlenmiş ve klasik genetik algoritma adımlarına göre çözüm algoritması oluşturulmuştur. Bu çalışma kapsamında yabancı ot algoritması kapasite kısıtlı araç rotalama problemine uyarlanmıştır. Yabancı ot algoritması için genetik algorithmada kullanılan kodlama yöntemine benzer bir yaklaşım kullanılmış, yavrular üretilirken kullanılacak uygun yöntem belirlenmiş ve algoritmanın adımları oluşturulmuştur. Algoritmanın başlangıç çözümleri iyileştikçe algoritmanın çözüm kalitesinin de iyileşeceği düşünülmüş olup tasarruf algoritması, genetik algoritma ve yabancı ot algoritması birbirini besleyecek şekilde hibrit algoritmalar kurulmuş ve çeşitli denemeler yapılmıştır. Bir algoritmanın ürettiği sonuç diğer algoritmanın başlangıç popülasyonuna eklenmiştir. Hibrit yaklaşımlar olarak; tasarruf-genetik algoritması, tasarruf-yabancı ot algoritması, tasarruf-genetik-yabancı ot algoritması geliştirilmiştir.

Çözüm yaklaşımlarının uygunluğunun test edilmesi için literatürde yer alan P grubu veri setine ait 24 adet ve CMT grubu veri setine ait 5 adet test problemi kullanılmıştır. Farklı ölçeklerdeki problemlerde algoritmaların davranışları incelenmiştir. Bu test problemlerinin sonuçlarına göre süre, maliyet ve standart sapma açısından en iyi algoritmanın hibrit yaklaşım olduğu ortaya konmuştur.

Dinamik problemde, klasik araç rotalama problemi karakteristiklerine ek olarak taleplerin dinamik olarak değişim gösterdiği koşullar ele alınmıştır. Bu koşullara göre dinamik olarak yeni talep gelebilir, mevcut talepler iptal olabilir ya da her iki durum aynı anda oluşabilmektedir. Bu senaryolar gerçek hayat problemlerine yakın senaryolardır.

Dinamik problemin çözümü için algoritmanın temel adımlarında değişiklik yapılmadan dinamik olan girdileri kabul edecek şekilde ayarlamalar yapılmıştır. Kurgulanan senaryolar, P grubu veri setindeki P-n50-k7 isimli tek depolu, 49 müşteri, 7 araçlı kapasite kısıtlı probleme uygulanarak problem çözülmüştür. Klasik problem için elde edilen en iyi hibrit çözüm yaklaşımları dinamik probleme uyarlanmış ve dinamik araç rotalama probleminin çözümü sağlanmıştır. Dinamik problem için uyarlanan algoritmalar çalıştırılmış bulunan en düşük maliyetli sonuç çözüm olarak kabul edilmiştir.

Bu çalışma sonucunda ortaya koyulan uygulama için kullanıcı arayüzü tasarlanmıştır. Kullanıcı arayüzü, web tabanlı olarak geliştirilerek farklı noktalardan erişim ve kullanılabilirlik sağlanmıştır. Kullanım kolaylığı ve rahatlığı sağlamak amacıyla harita altyapısı olarak Google Maps JavaScript API kullanılmıştır. Google Maps ile veri girişi işlemi kolaylaştırılmıştır. Bu arayüz ile kullanıcı harita üzerinden koordinat seçerek veya adres bilgisi girerek adres tanımlaması yapılabilmekte, tanımlaması yapılan adresler seçilerek uygulamaya girdi olarak verilmektedir. Uygulamanın oluşturmuş olduğu en uygun rota harita üzerinde gösterilerek kullanıcı için anlaşılır sonuçlar üretilmiştir. Gerçek hayatta kullanılabilecek bir uygulama ortaya konmuştur.

Bu çalışmanın ilk bölümünde araç rotalama problemi ve bununla ilgili kavramlara yer verilmiştir. Problemin tanımı yapılmış, varsayımları belirlenmiş ve çözüm yöntemleri anlatılmıştır. Çalışmanın amaçları ortaya konulmuş ve çalışmanın özgün katkısı ile bu çalışmanın önemi ele alınmıştır. Çalışmanın ikinci bölümünde ise, araç rotalama problemi ve dinamik araç rotalama problemi ile ilgili yapılmış önceki çalışmalar incelenmiştir. Üçüncü bölümde, algoritmalar, kullanılan veriler ve uygulamanın ekran görüntülerine yer verilmiştir. Dördüncü bölümde, kullanılan veri setlerin çözüm sonuçlarına, bulgulara ve analizlere yer verilmiştir. Son bölümde elde edilen sonuçlar değerlendirilmiş ve gelecekte yapılacak çalışmalar için önerilerde bulunulmuştur.



TEŞEKKÜR

Çalışmamın her aşamasında yanımda olan, motive eden, yardımlarını esirgemeyen, fikirleriyle yönlendiren, sonsuz sabırla her zaman çalışmaya teşvik eden değerli danışman hocam Dr. Öğr. Üyesi Yusuf Kuvvetli'ye sonsuz teşekkürlerimi sunarım.

Çalışmama fikirleriyle destek sağlayan Sayın Prof. Dr. Rızvan Erol'a ve Sayın Doç. Dr. Cihan Çetinkaya'ya teşekkürlerimi sunarım.

Hayatımın her anında yanımda olan, bana destek veren, motive eden değerli kardeşim Kerem Yıldırım'a sonsuz teşekkürler. Eğitim hayatım boyunca desteklerini esirgemeyen sevgili annem Fatma Yıldırım'a babam Memet Yıldırım'a sonsuz sevgi ve saygılarımı sunarım.

Bu süreçte yanımda olan değerli arkadaşlarım Umut Yıldırım'a, Alperen Öztürk'e ve Yasemin Aşıkoğlu'na, değerli çalışma arkadaşlarım Tuğba Kaleli'ye ve Yeliz Yılmaz'a teşekkürlerimi sunarım.

İÇİNDEKİLER

SAYFA

ÖZ	I
ABSTRACT.....	II
GENİŞLETİLMİŞ ÖZET	III
TEŞEKKÜR.....	VII
İÇİNDEKİLER	VIII
ÇİZELGELER DİZİNİ	XIV
ŞEKİLLER DİZİNİ	XVI
SİMGELER VE KISALTMALAR.....	XVIII
1. GİRİŞ.....	1
1.1. Araç Rotalama Problemi.....	2
1.1.1.Araç Rotalama Problemi Tanımı	2
1.1.2.Araç Rotalama Probleminin Temel Bileşenleri	4
1.1.3.Araç Rotalama Probleminin Kullanım Alanları.....	5
1.1.4. Araç Rotalama Problemi Türleri	6
1.1.4.1.Kapasite Kısıtlı ARP (KKARP)	6
1.1.4.2.Zaman Pencereci ARP (ZPARP)	7
1.1.4.3. Dağıtım Toplamalı ARP (DTARP).....	7
1.1.4.4.Mesafe Kısıtlı ARP (MKARP).....	8
1.1.4.5. Çok Depolu ARP (ÇDARP).....	8
1.1.5.Araç Rotalama Probleminin Matematiksel Modeli.....	9
1.1.6. Statik ve Dinamik Araç Rotalama Problemi.....	11
1.1.6.1.Statik Araç Rotalama Problemi	11
1.1.6.2.Dinamik Araç Rotalama Problemi	11
1.2. Problemin Tanımı	12
1.3. Problemin Çözüm Yöntemleri	12
1.3.1. Araç Rotalama Probleminin Çözüm Yöntemleri	14

1.3.1.1.Kesin Çözüm Yöntemleri.....	16
1.3.1.2.Sezgisel Yöntemler.....	16
1.3.1.3.Metasezgisel Yöntemler	17
1.4. Çalışmanın Amacı.....	18
1.5. Çalışmanın Kapsamı	19
1.6. Çalışmanın Özgün Katkısı	20
1.7. Çalışmanın Adımları ve Organizasyonu.....	21
1.7.1. Çalışmanın Adımları	21
1.7.2. Çalışmanın Organizasyonu	22
2. ÖNCEKİ ÇALIŞMALAR	23
2.1. Statik Araç Rotalama Problemi	23
2.2. Dinamik Araç Rotalama Problemi.....	29
2.3. Yabani Ot Algoritması İle İlgili Çalışmalar.....	30
2.4. Önceki Çalışmaların Değerlendirilmesi.....	32
3. MATERYAL VE METOT	35
3.1. Materyal.....	35
3.1.1. Problemden Kullanılan Veriler	35
3.2. Metot.....	37
3.2.1. Tasarruf Algoritması	37
3.2.1.1. Algortimaya Genel Bakış	38
3.2.1.2. Algoritma Adımları	39
3.2.2. Genetik Algoritma.....	42
3.2.2.1. Algortimaya Genel Bakış	42
3.2.2.2. Algoritma Parametreleri	44
3.2.2.3. Algoritma Adımları	45
3.2.2.4. Algoritmanın Probleme Uyarlanması.....	47
3.2.3. Yabani Ot Algoritması	55
3.2.3.1. Algortimaya Genel Bakış	55

3.2.3.1.(1).Başlatma (Initialization)	56
3.2.3.1.(2).Çoğalma Süreci (Reproduction)	56
3.2.3.1.(3).Rastgele Dağıtım (Spatial Dispersal)	57
3.2.3.1.(4).Rekabetçi Eleme (Competitive Exclusion).....	58
3.2.3.2.Algoritma Parametreleri	58
3.2.3.3.Algoritma Adımları	60
3.2.3.4.Algoritmanın Probleme Uyarlanması.....	63
3.2.4.Hibrit Yöntemler	66
3.2.5. Dinamik Araç Rotalama.....	67
3.2.6. Sistem Tasarımı.....	68
3.2.6.1. Kullanıcı Arayüzü Tasarımı	72
3.2.6.1.(1).Adres Kayıt Sayfası	72
3.2.6.1.(2).Planlama Sayfası	73
3.2.6.1.(3).Rotalama Sayfası	74
3.2.6.2.Veri Tabanı Tasarımı.....	76
4. BULGULAR VE TARTIŞMA	87
4.1. Kullanıcı Tanımlı Parametrelerin Belirlenmesi	88
4.1.1. Genetik Algoritma Parametrelerinin Belirlenmesi.....	88
4.1.1.1.Birey Sayısının Belirlenmesi.....	88
4.1.1.2.Mutasyon Oranının Belirlenmesi	89
4.1.1.3. Çaprazlama Yöntemi Oranlarının Belirlenmesi	90
4.1.1.4.Jenerasyon Sayısının Belirlenmesi	92
4.1.1.5.Parametre Test Sonuçları.....	93
4.1.2.Yabani Ot Algoritması	93
4.1.2.1.Başlangıç Yabani Ot Sayısının Belirlenmesi.....	93
4.1.2.2.Maksimum Yabani Ot Sayısının Belirlenmesi	94
4.1.2.3.Maksimum Tohum Sayısının Belirlenmesi	96
4.1.2.4.Varyans Azaltma Bileşeninin Belirlenmesi.....	97

4.1.2.5.Standart Sapma Son Deęerinin Belirlenmesi	98
4.1.2.6.Jenerasyon Sayısının Belirlenmesi	99
4.1.2.7.Parametre Test Sonuçları.....	100
4.2. Senaryo Analizi.....	100
4.2.1. P Grubu Veri Seti Sonuçları.....	101
4.2.1.1.Tasarruf Algoritması Çözüm Sonuçları.....	101
4.2.1.2.Genetik Algoritma Çözüm Sonuçları	103
4.2.1.3.Yabani Ot Algoritması Çözüm Sonuçları.....	104
4.2.1.4.Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları	105
4.2.1.5.Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları.....	107
4.2.1.6.Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları	109
4.2.2.CMT Grubu Veri Seti Sonuçları	110
4.2.2.1.Tasarruf Algoritması Çözüm Sonuçları.....	110
4.2.2.2.Genetik Algoritma Çözüm Sonuçları	110
4.2.2.3.Yabani Ot Algoritması Çözüm Sonuçları.....	111
4.2.2.4.Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları	112
4.2.2.5.Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları.....	112
4.2.2.6.Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları	113
4.3. Tartışma	113
4.3.1. P Grubu Veri Seti Karşılaştırması.....	113
4.3.1.1.Algoritmaların Ortalama Maliyet Karşılaştırması	114
4.3.1.2.Algoritmaların Standart Sapma Karşılaştırması	116
4.3.1.3.Algoritmaların Ortalama Hız Karşılaştırması.....	119
4.3.2.CMT Grubu Veri Seti Karşılaştırması	121

4.3.2.1.Algoritmaların Ortalama Maliyet Karşılaştırması.....	121
4.3.2.2.Algoritmaların Standart Sapma Karşılaştırması.....	123
4.3.2.3.Algoritmaların Ortalama Hız Karşılaştırması.....	125
4.3.3. Farklı Çalışmalar ile Kıyaslamalar.....	126
4.3.4.Genel Değerlendirme	131
4.4. Dinamik ARP Çözümü	132
4.4.1.Müşteri Talep İptalleri.....	134
4.4.2.Yeni Müşteri Talepleri	135
4.4.3.Yeni Müşteri Talepleri ve Talep İptalleri.....	136
5. SONUÇLAR VE ÖNERİLER	139
5.1. Çalışmanın Özeti.....	139
5.2. Sonuçlar	139
5.3. Öneriler	141
KAYNAKLAR	143
ÖZGEÇMİŞ	151



ÇİZELGELER DİZİNİ

SAYFA

Çizelge 3.1.	24 Adet P Grubu KKARP Problem Bilgileri (Augerat, 1995)	36
Çizelge 3.2.	5 adet CMT Grubu KKARP Problem Bilgileri (Christofides vd., 1979)	37
Çizelge 3.3.	"marker" Tablosu Alanları	78
Çizelge 3.4.	"distance" Tablosu Alanları	80
Çizelge 3.5.	"plan_header" Tablosu Alanları	82
Çizelge 3.6.	"plan_detail" Tablosu Alanları	84
Çizelge 4.1.	Genetik Algoritma Birey Sayısı Test Sonuçları	89
Çizelge 4.2.	Genetik Algoritma Mutasyon Oranı Test Sonuçları	90
Çizelge 4.3.	Genetik Algoritma Çaprazlama Oranı Test Sonuçları	91
Çizelge 4.4.	Genetik Algoritma Jenerasyon Sayısı Test Sonuçları	92
Çizelge 4.5.	YOA Başlangıç Yabani Ot Sayısı Testi	94
Çizelge 4.6.	YOA Maksimum Yabani Ot Sayısı Testi	95
Çizelge 4.7.	YOA Maksimum Tohum Sayısı (S_{max}) Değer Testi	96
Çizelge 4.8.	YOA Varyans Azaltma Bileşeni (n) Değer Testi	97
Çizelge 4.9.	YOA Standart Sapma Son Değer (σ_{final}) Testi	98
Çizelge 4.10.	YOA Maksimum Jenerasyon Sayısı	99
Çizelge 4.11.	Tasarruf Algoritması Çözüm Sonuçları	102
Çizelge 4.12.	Genetik Algoritma Çözüm Sonuçları	103
Çizelge 4.13.	Yabani Ot Algoritması Çözüm Sonuçları	104
Çizelge 4.14.	Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları	106
Çizelge 4.15.	Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları	108
Çizelge 4.16.	Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları	109
Çizelge 4.17.	Tasarruf Algoritması Çözüm Sonuçları	110
Çizelge 4.18.	Genetik Algoritma Çözüm Sonuçları	111

Çizelge 4.19. Yabani Ot Algoritması Çözüm Sonuçları	111
Çizelge 4.20. Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları	112
Çizelge 4.21. Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları	113
Çizelge 4.22. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları	113
Çizelge 4.23. Algoritmaların Ortalama Maliyet Karşılaştırması	115
Çizelge 4.24. Algoritmaların Standart Sapma Karşılaştırması	118
Çizelge 4.25. Algoritmaların Ortalama Hız Karşılaştırması	120
Çizelge 4.26. Algoritmaların Ortalama Maliyet Karşılaştırması	122
Çizelge 4.27. Algoritmaların Standart Sapma Karşılaştırması	124
Çizelge 4.28. Algoritmaların Ortalama Hız Karşılaştırması	126
Çizelge 4.29. P Grubu Veri Setlerinin Çözüm Sonuçlarının Farklı Çalışmalar ile Karşılaştırılması (Karagül ve arkadaşları, 2016; Ahmed ve Sun, 2018).....	129

ŞEKİLLER DİZİNİ

SAYFA

Şekil 1.1. Örnek Bir ARP	3
Şekil 1.2. Örnek Bir ARP Çözümü.....	4
Şekil 1.3. raç Rotalama Problemi Çözüm Yöntemleri.....	15
Şekil 3.1. Planlamadan Önce.....	38
Şekil 3.2. Planlamadan Sonra	39
Şekil 3.3. Tasarruf Algoritmasının Akış Diyagramı.....	41
Şekil 3.4. Genetik Algoritmanın Akış Diyagramı	47
Şekil 3.5. Gen ve Kromozom Gösterimi	48
Şekil 3.6. Genetik Algoritma Bileşenleri (Gen, Kromozom, Popülasyon ve Uygunluk Değeri).....	49
Şekil 3.7. Yer Değiştirme ile Çaprazlama	51
Şekil 3.8. Tersleme ile Çaprazlama	52
Şekil 3.9. Ekleme ile Çaprazlama.....	53
Şekil 3.10. Mutasyon İşlemi	54
Şekil 3.11. Uygunluk Değeri ve Tohum Sayısı Arasındaki İlişki	57
Şekil 3.12. Yabani Ot Algoritmasının Akış Diyagramı.....	62
Şekil 3.13. Yabani Ot Gösterimi	63
Şekil 3.14. Yabani Ot Algoritması Bileşenleri (Yabani Ot, Koloni, Uygunluk Değeri).....	64
Şekil 3.15. Çoğalma Aşaması.....	65
Şekil 3.16. Tasarruf ve Genetik Algoritma Hibrit Yaklaşımı.....	66
Şekil 3.17. Tasarruf ve Yabani Ot Algoritması Hibrit Yaklaşımı	67
Şekil 3.18. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Yaklaşımı.....	67
Şekil 3.19. Sistem Mimarisi	69
Şekil 3.20. Adres Kayıt Sayfası.....	73
Şekil 3.21. Planlama Sayfası	74

Şekil 3.22. Rotalama Sayfası.....	75
Şekil 3.23. Rota Güncelleme	75
Şekil 3.24. Veri tabanı Diyagramı	77
Şekil 3.25. "marker" Tablosu.....	79
Şekil 3.26. "distance " Tablosu.....	81
Şekil 3.27. "plan_header" Tablosu	83
Şekil 3.28. "plan_detail" Tablosu	85
Şekil 4.1. . Algoritmaların Ortalama Maliyet Karşılaştırması.....	116
Şekil 4.2. Algoritmaların Standart Sapma Karşılaştırması	119
Şekil 4.3. . Algoritmaların Ortalama Hız Karşılaştırması.....	121
Şekil 4.4. Algoritmaların Ortalama Maliyet Karşılaştırması.....	123
Şekil 4.5. . Algoritmaların Standart Sapma Karşılaştırması	125
Şekil 4.6. Algoritmaların Ortalama Hız Karşılaştırması.....	126
Şekil 4.7. Başlangıç Durumu.....	133
Şekil 4.8. Rota Güncellemeden Önceki Durum.....	134
Şekil 4.9. Rota Güncellemeden Önceki Durum.....	135

SİMGELER VE KISALTMALAR

ARP	: Araç Rotalama Problemi
DARP	: Dinamik Araç Rotalama Problemi
KKARP	: Kapasite Kısıtlı Araç Rotalama Problemi
ZPARP	: Zaman Pencere Araç Rotalama Problemi
MKARP	: Mesafe Kısıtlı Araç Rotalama Problemi
DTARP	: Dağıtım Toplamalı Araç Rotalama Problemi
ÖDSTARP	: Önce Dağıtım Sonra Topla Araç Rotalama Problemi
KTDARP	: Eş Zamanlı ve Karışık Topla-Dağıtım Araç Rotalama Problemi
ÇDÖDSTARP	: Çok Depolu Önce Dağıtım Sonra Topla Araç Rotalama Problemi
ÇDARP	: Çok Depolu Araç Rotalama Problemi
GA	: Genetik Algoritma
YOA	: Yabani Ot Algoritması
CR1	: Çaprazlama İşlemindeki Yer Değiştirme Oranı
CR2	: Çaprazlama İşlemindeki Tersleme Oranı
CR3	: Çaprazlama İşlemindeki Ekleme Oranı
BKS	: Best Known Solution
P	: Literatürden Alınan P Grubu Veri Seti (Augerat vd., 1995)
CMT	: Literatürden Alınan CMT Grubu Veri Seti (Christofides vd., 1979)
Unif(0,1)	: 0 ile 1 Arasında Rastgele Üretilen Sayı
S_{min}	: Minimum Tohum Sayısı
S_{max}	: Maksimum Tohum Sayısı
n	: Varyans Azaltma Bileşeni
$\sigma_{initial}$: Standart Sapma Başlangıç Değeri
σ_{final}	: Standart Sapma Son Değeri

PSO : Parçacık Sürü Optimizasyonu



1. GİRİŞ

Taşıma maliyetleri lojistikte önemli bir yer tutmaktadır. Bu nedenle, araştırmaların çoğunda taşıma maliyetlerinin en aza indirilmesine çalışılmaktadır. Bu amaçla sıklıkla incelenen problemlerden birisi araç rotalama problemi (Ergülen ve Güngör, 2006).

Araç rotalama probleminde (ARP) amaç, minimum taşıma maliyeti ve maksimum müşteri servis düzeyini sağlamak için aracın ya da araçların izleyeceği rotaların belirlenmesidir. En uygun rotaların belirlenmesiyle toplam uzaklıklar ve toplam araç sayıları minimize edilmiş olur (Pichpibul ve Kawtummachai, 2012).

ARP optimizasyon problemleri arasında üzerinde yoğun çalışılan problemlerden birisidir. Klasik ARP’de, problemin parametreleri önceden bilinir ve planlama boyunca değişmez. Bu haliyle bile ARP, NP-Zor sınıfta yer alan başka bir ifadeyle çözümü zor problem olarak bilinir. Klasik problemde önceden biliniyor olarak kabul edilen bazı bilgiler gerçek hayatta dinamik bir şekilde değişmektedir. Bu tip problemler literatürde dinamik araç rotalama problemi (DARP) olarak adlandırılır. DARP’de klasik yaklaşımdan farklı olarak problemde yer alan verilerin başlangıçta tamamen ya da kısmen bilinmediği varsayılır ve planlama ufku boyunca oluşabilecek değişimler gözlenebilir. Bu durum, DARP’yi ARP’ye göre çözümü daha zor bir probleme dönüştürür (Demirtaş ve Özdemir, 2017).

Dinamik araç rotalama problemi süreç ilerlerken bazı müşterilerin taleplerini iptal etmesi, yeni müşteri taleplerinin oluşması, bazı talep miktarlarının değişmesi, trafik yoğunluğu gibi durumlarda ortaya çıkmaktadır. DARP müşterilerine satış, bakım, onarım, kurulum, kurye gibi hizmetler sağlayan işletmelerde karşı karşıya kalınan önemli problemlerden biridir (Potvin vd., 2004).

Literatürde araç rotalama problemiyle ilgili çalışmalar genellikle statik problemler üzerine yoğunlaşmıştır. Statik araç rotalama problemleri NP-Zor sınıfındadır. Bu problemin dinamik halinin statik halini de kapsadığı ve daha zor

olduğu kanıtlanmıştır. Dinamik araç rotalama problemleri araştırmacılar için yeni ve ilgi çekicidir (Cormen vd., 2009).

1.1.Araç Rotalama Problemi

1.1.1.Araç Rotalama Problemi Tanımı

Araç rotalama problemi, depo veya depolardan müşterilere ürün dağıtımını ve toplanması olarak tanımlanabilir. Araç rotalama problemi, ürünleri depolara dağıtan bir üretici firma, siparişlerin alındığı depo veya depolar ve talepleri olan müşteri bileşenlerinden oluşmaktadır. Lojistik sisteminde bu aşamalara bakıldığında ürün dağıtılmasının ve toplanmasının maliyetli olduğu sonucuna varılmaktadır (Keskintürk vd., 2015).

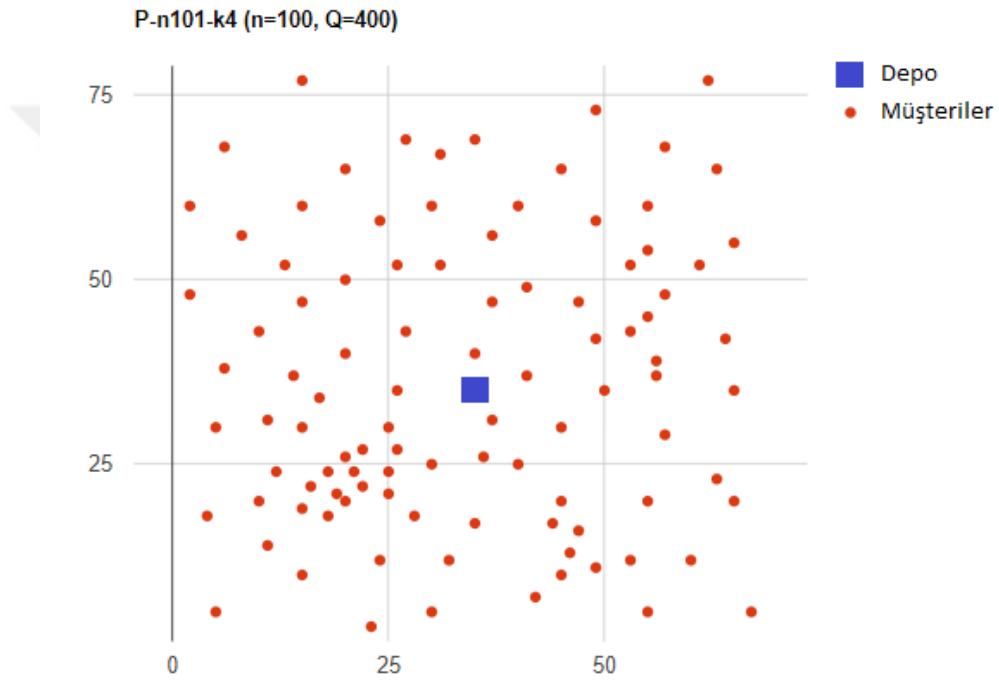
Dağıtım aşaması, üreticiden ürünlerin teslim alınması ve ürünlerin müşterilere teslim edilmesi aşamalarından oluşmaktadır. Ürünlerin dağıtım aşamasında araçlar depodan başlar ve dağıtımını tamamladıktan sonra depoya geri döner. Bundan dolayı depo, araçların rotasına başladığı ve rotası tamamlandıktan sonra geri döndüğü yerdir (Keskintürk vd., 2015).

Literatürde araç rotalama problemi ile ilgili pek çok tanım yapılmıştır. Bu tanımlar incelendiğinde ortak olan nokta; araç filosu ile merkezi depodan müşteri taleplerini, çeşitli kısıtlar altında minimum maliyet ile karşılamaktır. Araç rotalamanın temel amacı; en kısa yol, en kısa süre ve en az maliyet olabilir.

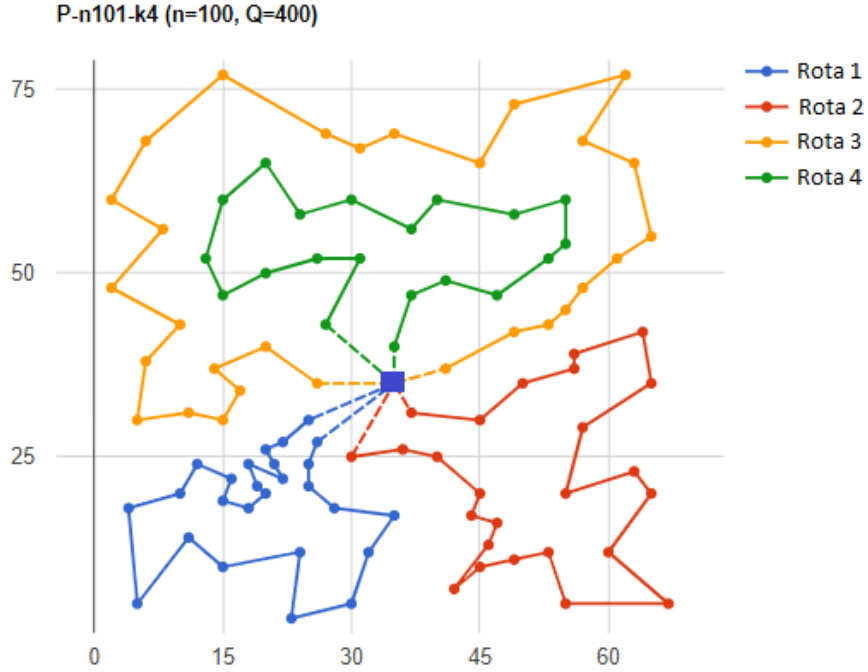
Klasik araç rotalama probleminde homojen araçlar ile merkezi bir depodan müşteri talepleri karşılanmaya çalışılır. Araçlar depodan başlar ve depoya döner. Dolayısıyla rotaların başlangıç ve bitiş noktaları depodur. Araç rotalama probleminin statik versiyonu için müşteri talepleri önceden bilinir ve süreç tamamlanıncaya kadar değişmez. Her müşteriye sadece bir kez uğranmalıdır. Bir rota üzerindeki müşteri talepleri araç kapasitesini geçemez. Bu kısıtlar dışında probleme müşteri önceliği, süre kısıtı gibi farklı kısıtlar da eklenebilir. Bu şartlar altında minimum

maliyetli rotaların oluşturulması amaçlanmaktadır. Araç rotalama probleminde temel amaç toplam uzaklığı küçülterek dağıtım süresini ve maliyetini düşürmektir.

Şekil 1.1’de örnek bir araç rotalama problemi ve Şekil 1.2’de çözümü görülmektedir. Belirli kısıtlar altında Şekil 1.1’deki ARP çözüldüğünde Şekil 1.2’de görülen rotalar oluşmuştur.



Şekil 1.1. Örnek Bir ARP



Şekil 1.2. Örnek Bir ARP Çözümü

1.1.2. Araç Rotalama Probleminin Temel Bileşenleri

Araç rotalama probleminin temel bileşenleri; depo, müşteriler (talep noktaları), araç filosu, talep, rota ve maliyettir. Bir ARP'de yer alan temel bileşenler aşağıda açıklanmıştır.

Depo: Araçların başlangıç ve bitiş noktasıdır. Müşterilere dağıtımın yapıldığı merkezdir. Gerçek hayatta, depo, dağıtım merkezi veya bayi olabilir. Depo sayısına göre araç rotalama problemi tek depolu ve çok depolu olarak adlandırılabilir.

Müşteriler (Talep Noktaları): Müşterilerin talepleri ve lokasyonları vardır. Araç rotalama probleminde bütün müşterilerin talepleri karşılanmalıdır. ARP'de her müşterinin birbiri ile ve depo ile arasındaki uzaklıklar bilinmektedir.

Araç Filosu: Müşteri taleplerini karşılamak için kullanılan araçların tamamı araç filosunu oluşturur. Araç rotalama probleminde araç filosu, aynı kapasiteye sahip (homojen) veya farklı kapasitelere sahip (heterojen) araçlardan oluşabilir.

Talep: Müşterilere sunulması gereken hizmet veya malzemedir. Araç rotalama probleminde talep yapısı; statik veya dinamik olabilir. Talep yapısı statik ise talepler planlama aşamasında bilinir ve süreç boyunca değişmez. Talep yapısı dinamik ise talepler süreç boyunca değişebilir.

Rota: Müşteri taleplerinin karşılanması için bir aracın izleyeceği yolu ifade eder. Rotalar depodan başlar ve depoda biter.

Maliyet: ARP'de maliyet rotaların toplam uzunluğunu ifade etmektedir. Araç rotalama probleminin amacı maliyeti minimize etmektir.

1.1.3. Araç Rotalama Probleminin Kullanım Alanları

Araç rotalama problemi, rotalama kararlarıyla ilgili olarak birçok farklı alanda kullanılabilir. Bu alanlar, dağıtım problemlerini kapsayabileceği gibi toplama veya toplama ve dağıtımın birlikte yapıldığı problem türlerini de kapsayabilir. Genel olarak araç rotalama probleminin kullanım alanları şöyle özetlenebilir: (Keskintürk vd., 2015)

- Tek ya da çok depolu ürün dağıtımı,
- Topla-dağıt problemleri,
- Servis güzergahlarının belirlenmesi,
- Atık toplaması,
- Gemi, uçak vb. araçların rotalama problemleri,
- Kargo ve kurye faaliyetleri,
- Firma içi malzeme akış problemleri.

Süreç ilerlerken bazı müşterilerin taleplerini iptal etmesi, yeni müşteri taleplerinin oluşması, bazı talep miktarlarının değişmesi, trafik yoğunluğu gibi durumlarda dinamik araç rotalama problemi ortaya çıkmaktadır. Dinamik rotalama problemi özelinde konuya bakıldığında ise gerçek hayattaki uygulamalarda daha sık karşılaşılabilecek uygulamalardan oluştuğu görülmektedir. Dinamik rotalama probleminin özellikle şu alanlarda kullanılabileceği görülmektedir (Psaraftis, 1995):

- Petrol ürünlerinin dağıtımı,
- Kurye servisleri,
- Çok modlu taşımacılık,
- Tarifersiz taşımacılıklar,
- Bütünleşik topla-dağıt problemleri,
- Konteyner terminallerinin yönetimi.

1.1.4. Araç Rotalama Problemi Türleri

Araç rotalama probleminde toplam maliyetin en küçüklemesi, toplam rota sayısının en küçüklemesi gibi farklı amaçlar olabilir. Ayrıca kapasite kısıtı, mesafe kısıtı, zaman penceresi gibi kısıtlar altında da incelenebilir. Araç rotalama probleminin çok farklı alanlarda, farklı amaçlarla ve çeşitli kısıtlarla ele alınması bu problemin pek çok çeşidinin oluşmasına olanak sağlamıştır.

1.1.4.1. Kapasite Kısıtlı ARP (KKARP)

Sadece kapasite kısıtının olduğu araç rotalama problemine kapasite kısıtlı araç rotalama problemi denilir. Kapasite; adet, hacim ya da ağırlık şeklinde olabilir. Dağıtma ya da toplama işleminde aracın belirlenen kapasitesi aşılmadan minimum maliyetli rotalama yapılır.

Kapasite kısıtlı araç rotalama problemi (KKARP), tek ya da çok depolu araç rotalama problemlerinde araç yüklerinin sınırlı bir kapasiteye sahip olduğu durumları ifade eder. KKARP probleminin çözümünde araç kullanımının azaltılması için kapasitelere göre sabit araç kullanım maliyetleri gibi farklı durumlar göz önüne alınabilir (Erol, 2006).

1.1.4.2.Zaman Pencereci ARP (ZPARP)

Zaman pencereci araç rotalama problemi, kapasite kısıtlı araç rotalama probleminin genişletilmiş halidir. Bu problemde her müşteriye uygun olan zaman aralığında hizmet verilmelidir. Bu zaman aralığına, zaman penceresi denilmektedir. Her müşteri için belirlenmiş olan en erken ve en geç teslimat başlangıç zamanları vardır. Bu durum problem için zaman kısıtının sınırlarını oluşturur. Araç müşteriye en erken teslimat zamanından önce gelmiş ise beklemek durumda kalır. Ayrıca uygun çözüm olabilmesi için araç müşteriye en geç teslimat zamanından önce varmalıdır. Ayrıca araç hizmet süresince hizmet verilen konumda bulunmalıdır. Endüstriyel atık toplama, posta dağıtımı, kargo taşımacılığı ZPARP sınıfına giren gerçek hayat problemlerinden bazılarıdır.

1.1.4.3. Dağıtım Toplamalı ARP (DTARP)

Dağıtım ve toplamalı araç rotalama problemi (DTARP), ARP'nin bir uzantısıdır. Lojistikte müşterilere ürün dağıtılmasının yanı sıra bazı müşterilerden ürün toplamaları da gerekmektedir. Bu problemde araçlar sadece dağıtım yapmamakta ayrıca toplama işlemi de yapmaktadır.

Önce dağıt sonra topla araç rotalama problemi (ÖDSTARP), dağıtım ve toplamalı ARP'nin bir çeşididir. Önce dağıt sonra topla araç rotalama probleminde, müşterilerin mal alan ve mal gönderen şeklinde ayrıldığı varsayılmaktadır. Ayrıca araçların tüm yüklerini dağıttıktan sonra toplamaya başlayabileceği varsayılmaktadır. Bu varsayımlar uygulamada kolaylık sağlamaktadır.

Eş zamanlı ve karışık topla-dağıt araç rotalama problemi (KTDARP), önce dağıt sonra topla ARP'nin, dağıtım ve toplama işleminin aynı anda gerçekleştiği özel bir halidir. Eş zamanlı ve karışık topla-dağıt ARP'de müşteriler eş zamanlı olarak mal alıp gönderebilirler. Bir müşteriye ürün dağıtımını yapılırken müşteriden ürün teslim alınmasına da izin verilir.

Çok depolu önce dağıt sonra topla araç rotalama problemi (ÇDÖDSTARP), önce dağıt sonra topla ARP'nin gelişmiş halidir ve tek depo yerine birden fazla depo bulunur. Her araç harekete başladığı depoya geri dönmelidir.

1.1.4.4. Mesafe Kısıtlı ARP (MKARP)

Mesafe kısıtlı araç rotalama probleminde (MKARP) aracın bir rota üzerinde gidebileceği mesafe sınırı vardır. Farklı tipteki araçlara göre gidilebilecek mesafe kısıtları değişebilir. Bazı çalışmalarda mesafe yerine yolculuk süresi de kısıt olarak eklenebilir ve böylelikle aracın yükleme boşaltma süreleri de göz önüne alınabilir (Erol, 2006).

Mesafe kısıtlı araç rotalama problemi, gerçek hayatta sürücü kısıtından, araç kısıtından veya ürünün cinsinden dolayı oluşabilir. Araç sürücüsünün sürekli olarak belirli sürelerden fazla araç kullanmaması ya da uzun süre taşınması nedeniyle ürünün bozulması söz konusuysa mesafe kısıtı eklenmelidir (Dursun, 2009).

1.1.4.5. Çok Depolu ARP (ÇDARP)

Birden çok depoya sahip işletmelerde, her depo kendi bölgesindeki müşterilere dağıtım yapabilir. Bu durumda işletme, tek depolu birden çok araç rotalama problemi ile karşı karşıyadır. Her müşteriye kendi bölgesindeki depo tarafından hizmet verilir. Bu problemde her araç rotası aynı depodan başlar ve aynı depoda sonlanır. Bazı durumlarda ise bir araç dağıtım yaptıktan sonra başka

depoda rotasını sonlandırabilir ve tekrar yükleme yaparak diğer rotasına başlayabilir. Bu durumda depolar birbirinden izole olarak düşünülemez. Bu dağıtım problemine çok depolu araç rotalama problemi (ÇDARP) denir. ÇDARP’de amaç kat edilen mesafeyi ve zamanı minimize etmektir. Bunun yanında araç sayısını minimize etmek gibi amaçlar da olabilir. Farklı amaç fonksiyonları olsada temel amaç verimliliği arttırmaktır.

1.1.5.Araç Rotalama Probleminin Matematiksel Modeli

Kapasite kısıtlı araç rotalama probleminin matematiksel modeli aşağıdaki gibidir (Özkök ve Kurul,2014):

Notasyon

N_c	Dağıtım yapılan noktalar kümesi
N	Tüm noktalar kümesi (0 u N_c)
K	Araçlar kümesi
i, j, p	Nokta indisleri
k	Araç indisi
c_{ij}	i-j arası oluşacak taşıma maliyeti
q_k	k. araç kapasitesi
d_i	i. müşterinin talep ettiği ürün miktarı
x_{ijk}	$\begin{cases} 1, \text{Eğer } k. \text{ araç } i - j \text{ arası seyahat ederse} \\ 0, \text{aksi halde} \end{cases}$ (karar değişkeni)
y_i	Alt tur oluşmasını engellemek için kullanılan değişken (karar değişkeni)

Amaç fonksiyonu

$$\min z = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^K c_{ij} * x_{ijk} \quad (1)$$

Kısıtlar

$$\sum_{k=1}^K \sum_{j=1}^{Nc} x_{0jk} = K \quad (2)$$

$$\sum_{i=0}^N x_{ijk} - \sum_{p=0}^N x_{jpk} = 0 \quad \forall k \in K, \quad (3)$$

$$\sum_{k=1}^K \sum_{j=0}^N x_{ijk} = 1 \quad \forall i \in Nc \quad (4)$$

$$\sum_{j=1}^{Nc} x_{0jk} = 1 \quad \forall k \in K \quad (5)$$

$$\sum_{j=1}^{Nc} x_{j0k} = 1 \quad \forall k \in K \quad (6)$$

$$\sum_{j=0}^N \sum_{i=j}^N d_i * x_{ijk} \leq q_k \quad \forall k \in K \quad (7)$$

$$y_i \geq y_j + 1 - N(1 - \sum_{k=1}^K x_{ijk}) \quad i \neq j, \quad (8)$$

$$y_j \geq 0 \quad \forall j \in Nc \quad (9)$$

(1) numaralı denklem amaç fonksiyonunu ifade etmektedir, problemin amacı minimum toplam mesafeyi kat edecek rotaların belirlenmesidir. Kullanılacak araç sayısı (2) nolu kısıtta hesaplanmaktadır. (3) nolu kısıt rotaların oluşmasını

sağlayan bir düğüme giren ve çıkan araçların eşit olmasını sağlayan kısıttır. (4) numaralı kısıt bir noktadan yalnızca tek bir noktaya gidilmesini zorunlu kılar. (5) ve (6) numaralı kısıtlar ise turun depoda başlama ve bitmesini sağlayan kısıtlardır. (7) numaralı kısıt araç kapasitesinin aşılmamasını sağlayan kısıttır. (8) ve (9) nolu kısıtlar ise alt tur oluşmasını engelleyen kısıtlardır.

1.1.6. Statik ve Dinamik Araç Rotalama Problemi

Araç rotalama problemi statik ve dinamik olarak ikiye ayrılabilir. Statik ARP ve dinamik ARP aşağıda sunulmuştur.

1.1.6.1. Statik Araç Rotalama Problemi

Statik araç rotalama probleminde, planlama aşamasında tüm bilgiler bilinir ve süreç boyunca değişmez. ARP'nin bu versiyonu deterministik ARP olarak da tanımlanabilir. Literatürdeki çalışmalar daha çok Statik ARP üzerine yapılmıştır. Problemin bu türü klasik ARP olarak da bilinmektedir.

1.1.6.2. Dinamik Araç Rotalama Problemi

Dinamik araç rotalama probleminde, probleme ait tüm bilgiler planlama aşamasında bilinmeyebilir veya süreç içerisinde değişiklikler (yeni taleplerin gelişi, talep iptalleri ve her iki durumun birlikte oluşması) oluşabilir.

Dinamik ARP literatüre 1988 yılında yapılan çalışma ile tanıtılmıştır. Çalışmada statik ve dinamik araç rotalama problemleri arasındaki farklar ortaya konulmuştur. Statik yaklaşımların dinamik ortama adaptasyonu tartışılmış ve açıklanmıştır. Ayrıca bir kargo şirketinin araçlarının dinamik olarak rotalama problemini çözen bir algoritma önerilmiştir (Psaraftis, 1988).

ARP NP-Zor sınıfındadır. Bu probleme, kesin yöntemler ile makul sürelerde iyi sonuçların üretilmesi pek olanaklı görünmemektedir. Bundan dolayı araç rotalama probleminde sezgisel ve metasezgisel yöntemler kullanılmaya

başlanmıştır. Dinamik araç rotalama probleminin klasik versiyondan daha zor olduğu ortaya konmuştur. Bundan dolayı benzer şekilde DARP için de kesin yöntemler etkisiz kalmaktadır ve sezgisel yöntemler önerilmektedir (Demirtaş ve Özdemir, 2017).

1.2. Problemin Tanımı

Bu çalışmada incelenen araç rotalama problemi,

- Çok araçlı,
- Heterojen
- Kapasite kısıtlı araç rotalama problemidir. Problem, statik ve dinamik talepli olarak ayrı ayrı ele alınmıştır.

Dinamik talepli problem, lojistik firmalarının sıklıkla karşılaştığı problemlerden birisidir. Örneğin Adana bölgesinden yarı dolu olarak İstanbul'a doğru yola çıkan tırın, sipariş oluşması halinde Ankara bölgesinde güzergah değişikliği yaparak İstanbul'a gidişi söz konusu olabilir. Bu tür uygulamaların en sık görüldüğü örnek 3. parti lojistik hizmeti veren firmaların işlemleridir.

Problemin amacı minimum sayıda araç ve minimum uzaklıkla dağıtımın yapılmasıdır. Problemden karşılaşılan kısıtlamalar ise, araç kapasitesinin aşılmaması ve tur kısıtlamalarıdır. Araç kapasitesinin aşılmaması kısıtı, bir araca yapılacak yüklemelerin o aracın kapasitesini aşmaması gerekliliğini ifade etmektedir. Rotalama kısıtları aracın uğrayacağı güzergahın belirlenmesini sağlayan kısıtlamalardır.

1.3. Problemin Çözüm Yöntemleri

Araç rotalama problemi NP-Zor sınıfındadır ve kesin çözümünü matematiksel olarak belirlemek karmaşıktır, zordur. Zorluk derecesinin yüksek

olması ve gerçek hayatta sürekli karşılaşılan bir problem olmasından dolayı kabul edilebilir zaman içinde yüksek kalitede çözümler için sezgisel ve metasezgisel yöntemler tercih edilmektedir. Son zamanlarda yapılan çalışmalarda çoğunlukla KKARP'nin çözümü için sezgisel ve metasezgisel yöntemlerin tercih edildiği görülmektedir (Şahin ve Eroğlu, 2014).

Bu modelin çözümü için bölüm 1.3.1'de anlatıldığı üzere birçok farklı yöntem bulunmaktadır. Bu çalışmada problemin çözümü için metasezgisel ağırlıklı yaklaşımlar tercih edilmiştir.

ARP'nin tasarruf algoritması ile çözümünde tüm müşteriler için tasarruf değerleri hesaplanır. Algoritma adımları işletilir. Sonuçta oluşan rotalar problemin çözümü olarak kabul edilir.

ARP'nin genetik algoritma ile çözümünde müşterilerin numaraları gen olarak ifade edilmiştir. Kromozom, rotalardan oluşan müşteri listesi şeklinde ifade edilmiştir. Kromozom ARP'nin olası çözümlerinden biridir. Bir kromozom üzerinde rotaları barındırır ve rotalar müşteri numaralarından oluşan listedir. Listenin indeksi müşterinin sırasını belirler. Bu kodlama yöntemi belirlendikten sonra algoritma adımları işletilir. Problemin amacı rotaların toplam mesafesini minimize etmek olduğu için algoritmanın durma kriteri sağlandığında hayatta kalan bireylerden uygunluk değeri en düşük olan çözüm en iyi çözüm olarak değerlendirilir. Bu çözüm optimal veya optimale yakın çözümdür.

ARP'nin yabancı ot algoritması ile çözümünde yabancı ot, rotalardan oluşan müşteri listesi şeklinde ifade edilmiştir. Yabancı ot ARP'nin olası çözümlerinden biridir. Bir yabancı ot üzerinde rotaları barındırır ve rotalar müşteri numaralarından oluşan listedir. Listenin indeksi müşterinin sırasını belirler. Bu kodlama yöntemi belirlendikten sonra algoritma adımları işletilir. Problemin amacı rotaların toplam mesafesini minimize etmek olduğu için algoritmanın durma kriteri sağlandığında hayatta kalan yabancı otlardan uygunluk değeri en düşük olan çözüm en iyi çözüm olarak değerlendirilir. Bu çözüm optimal veya optimale yakın çözümdür.

Metasezgisel algoritmalar genellikle rastgele başlangıç çözümleri ile başlar ve aşamalı olarak bu çözümler iyileştirilmeye çalışılır. Durma kriteri sağlandığında optimale yakın çözüm üretilmiş olur. Bu açıdan bakıldığında algoritmanın başlangıç çözümleri iyileştikçe algoritmanın çözüm kalitesi de iyileşecektir. Buradan hareketle bu çalışmada ele alınan tasarruf algoritması, genetik algoritma ve yabani ot algoritması birbirini besleyecek şekilde çeşitli denemeler yapılmıştır. Bir algoritmanın ürettiği sonuç diğer algoritmanın başlangıç popülasyonuna eklenmiştir. Örneğin tasarruf algoritmasının ürettiği çözüm; genetik algortmada ve yabani ot algoritmasında başlangıç popülasyonuna eklenerek çalıştırılmıştır.

Hibrit algoritmalar olarak; tasarruf-genetik algoritması, tasarruf-yabani ot algoritması, tasarruf-genetik-yabani ot algoritması geliştirilmiştir. Algoritmalar, isimlendirmede verilen sıralar ile çalıştırılmıştır.

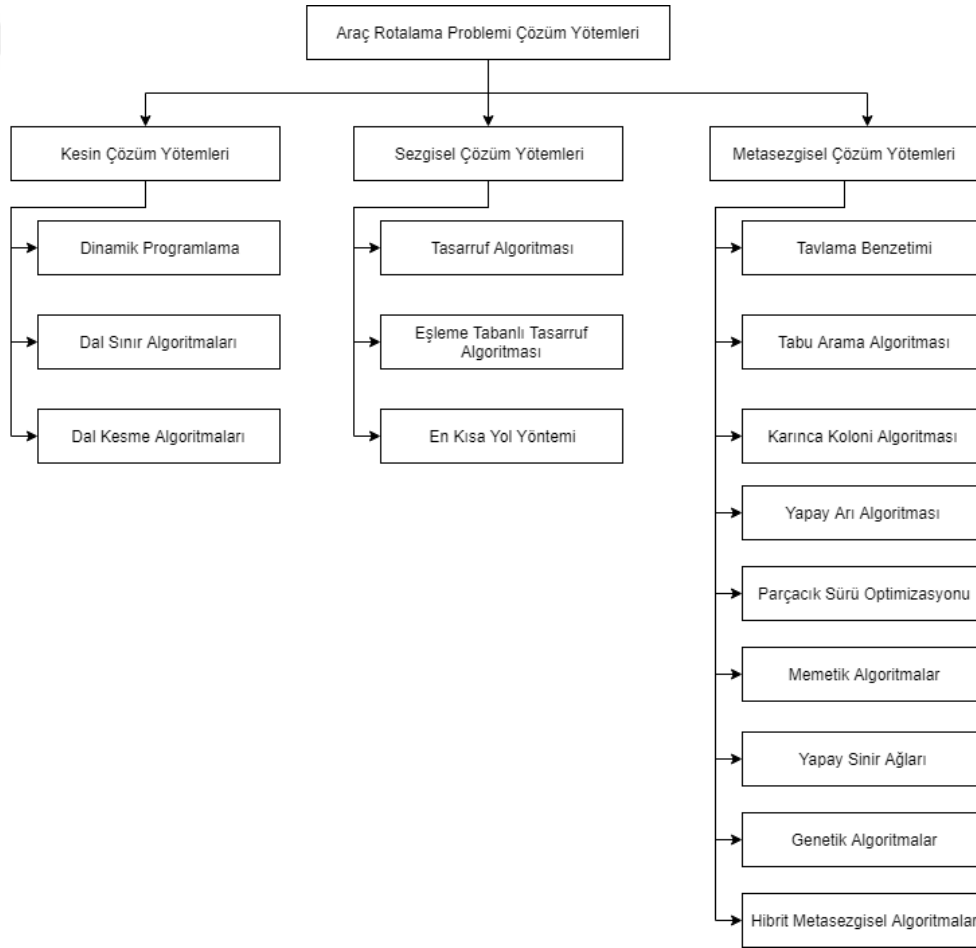
1.3.1.Araç Rotalama Probleminin Çözüm Yöntemleri

Araç rotalama problemlerinin çözümü için birçok yöntem önerilmiştir. Önerilen bu yöntemleri; kesin yöntemler, sezgisel yöntemler ve metasezgisel yöntemler olmak üzere 3 gruba ayırmak mümkündür. Kesin çözüm yöntemleri için en iyi çözüm garanti edilir, diğer yöntemlerle optimal veya optimale yakın çözümler daha kısa sürelerde bulunabilir (Keskintürk vd., 2015).

Kesin çözüm yöntemlerinde probleme ait tüm kısıtlar göz önüne alınarak en uygun çözüm yöntemi bulunur. Ancak araç rotalama probleminde kısıt sayısı ve müşteri sayısı arttıkça çözüme ulaşmak zorlaşır ve çözüm süresini uzatır. Çünkü araç rotalama problemleri NP-Zor sınıfında yer alan bir problemdir. Kesin çözüm yöntemleri küçük çaplı problemler için uygun olacaktır. Gerçek hayat problemleri çok fazla kısıt içerir ve büyük ölçeklidir. Bundan dolayı ARP’de kesin yöntemler yerine sezgisel ve metasezgisel yöntemler kullanılmaya başlanmıştır. Bu yöntemler çok daha kısa sürede optimal veya optimale yakın çözümler üretebilmektedir. Bu

yöntemlerdeki temel amaç problemin çözüm süresini kısaltarak kabul edilebilir kalitede çözümler üretmektir.

Literatürdeki çalışmalar incelendiğinde araç rotalama problemi için birçok algoritmaya rastlanmıştır. Şekil 1.3'te araç rotalama probleminde sıklıkla kullanılan çözüm yöntemleri ve bu çözüm yöntemlerine ait algoritmaların şekilsel gösterimi verilmiştir. Problem NP-Zor sınıfında olduğu için sezgisel ve metasezgisel yaklaşımlar daha çok tercih edilmektedir.



Şekil 1.3. Araç Rotalama Problemi Çözüm Yöntemleri

1.3.1.1.Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri, problem için en iyi çözümü bulmayı garantiler fakat problemin çözüm süresi problemin boyutuna bağlı olarak üstel şekilde artar. ARP’de genelde kesin çözüm yöntemleri ile küçük ve orta boyutlu problemler çözülebilmektedir. Ancak problem boyutu büyüdükçe problemin çözüm süresi üstel olarak artar. Bütün araç rotalama problemlerini başarıyla çözen kesin çözüm yöntemi yoktur (Ropke, 2005).

Araç rotalama probleminin çözümü için kullanılan kesin yöntemlerde; dinamik programlama, dal sınır algoritmaları, dal-kesme algoritmaları yer almaktadır.

1.3.1.2.Sezgisel Yöntemler

ARP’de alternatif rota sayı müşteri sayısının artışıyla artmaktadır. Bu durum problemin hesaplanmasını zorlaştırmaktadır. Problem boyutu büyüdükçe kesin çözüm yöntemleri etkisiz kalmakta ve bu problemde sezgisel, metasezgisel algoritmaları ön plana çıkarmaktadır (Eryavuz ve Gencer, 2001).

ARP’nin çözümünde kısa sürede iyi sonuçlara ulaşmak için genellikle sezgisel yöntemler kullanılmaktadır. Bu yöntemler en iyi çözümü garanti etmezler ancak kısa sürede optimal veya optimale yakın çözümü bulabilirler. Sezgisel yöntemler belirlenen amaca ulaşmak için tanımlanan kriterler veya bilgisayar metotlarıdır (Erol, 2006).

Araç rotalama probleminde tüm kombinasyonlar denenebilirse en iyi çözüme ulaşmak mümkündür. ARP’de en iyi çözüm, üretilecek olası rota kombinasyonlarından biridir. Ancak büyük boyutlu problemlerde tüm kombinasyonların denenmesi ve en iyi çözümün bulunması çözüm süresinin çok uzun olmasına neden olacaktır. Bundan dolayı daha kısa sürelerde optimal veya optimale yakın kabul edilebilir sonuçlara ulaşmak için sezgisel yöntemler geliştirilmiştir (Keskintürk vd., 2015).

Probleme önerilen ilk sezgisel yaklaşım olan tasarruf algoritması ise, Clark ve Wright tarafından sunulmuştur (Clark ve Wright, 1964).

Araç rotalama probleminin çözümü için kullanılan sezgisel yöntemlerde; tasarruf algoritması, eşleme tabanlı tasarruf algoritması, en kısa yol yöntemi gibi algoritmalar yer alır.

1.3.1.3. Metasezgisel Yöntemler

Metasezgisel yöntemler, kesin çözüm yöntemleri ile çözülmesi uzun süren problemlerin, daha kısa sürelerde makul kalitede çözümü için önerilen yaklaşımlardır. Metasezgisel yöntemler, doğadan esinlenerek tasarlanan algoritmalarlardır. Büyük boyutlu gerçek hayat problemlerin çözümünde pratik yol olarak kabul edilebilir. Kolay anlaşılır ve farklı problemlere uygulanabilir olmasından dolayı günümüzde yoğun olarak kullanılmaktadır. Metasezgisel yöntemlerin amacı, çözüm uzayını hızlı ve etkili şekilde araştırmak ve optimale yakın çözümler üretmektir (Şahin ve Eroğlu, 2014).

Metasezgisel yöntemler, sezgisel algoritmaların zekice birleştirilmesi ile oluşturulmuş iteratif problem çözme metotlarıdır. Çözüm uzayında etkili bir şekilde arama yapar. Çoğu metasezgisel yöntem, çözüm uzayında stokastik fakat bilinçli şekilde arama gerçekleştirir. Bu yaklaşımlarda her iterasyonda çözüm adaylarından yola çıkarak yeni çözümler üretilir (Erol, 2006).

Metasezgisel algoritmalarda amaç, arama uzayının en umut verici noktalarında arama yaparak optimum çözüme yakın sonuçlar elde etmektir. Bu yöntemlerle elde edilen sonuçlar sezgisel yöntemlerle elde edilen sonuçlardan daha iyidir ancak genellikle hesaplama işlemi daha uzun sürer. Metasezgisel yöntemler, sezgisel yöntemlerin doğadan esinlenerek tasarlanmış hali olarak düşünebilir (Laporte vd., 2000).

Klasik sezgisel yöntemler, çözüm uzayında belirlenen komşuluk yapısı ile daha iyi bir çözüm bulunmadığı durumlarda sonlanmakta ve lokal minimum

noktalarında takılmaktadır. Metasezgisel yöntemler ise lokal minimumdan kurtulmak amacıyla daha kötü çözüm adaylarının da kabul edildiği global optimizasyon yöntemleridir. Metasezgisel algoritmaların dezavantajı belirli bir durma kriterlerinin bulunmayışıdır. Yani algoritmanın ne zaman duracağını belirlemek zordur (Breedam, 2001).

Araç rotalama probleminin çözümü için kullanılan metasezgisel yöntemlerde; tavlama benzetimi, tabu arama algoritması, karınca kolonisi algoritması, yapay arı algoritması, parçacık sürü optimizasyonu, yapay sinir ağları, genetik algoritmalar ve hibrit metasezgisel algoritmalar yer almaktadır.

1.4. Çalışmanın Amacı

Araç rotalama problemleri, lojistik kararların verilmesinde en sık kullanılan yaklaşımların başında gelmektedir. Önceki çalışmalarda araç rotalama problemlerinin kapasite kısıtları, zaman pencereleri, çok periyotlu, tur kısıtlı, çok araçlı gibi birçok farklı durum için uygulaması yer almaktadır. Pratikte lojistik kararlarının verilmesinde ise daha dinamik kararlar söz konusudur. Bu kararlar, turun oluşmasından sonra zaman içerisinde oluşan taleplerin rota kararlarının değiştirilmesine neden olması olarak düşünülebilir. Lojistik firmaları uygulamalarında sıklıkla statik versiyonu yerine dinamik araç rotalama problemi ile uğraşmaktadır. Önceki çalışmalar incelendiğinde ise dinamik araç rotalama probleminin statik versiyonuna göre daha az irdelendiği görülmektedir. Bu durum, yapılan çalışmanın dinamik araç rotalama literatürüne katkı sağlayacağını ve bu çalışmanın önemini ortaya koymaktadır.

Bu çalışmanın temel amacı, kapasite kısıtlı araç rotalama probleminin statik ve zaman içerisinde oluşan durumları da (yeni talep gelişi, siparişin iptali ve her iki durumun birlikte oluşması gibi durumları) ele alarak rotalama yapılması ve rotalama kararlarının güncellenmesidir.

Yabani ot algoritmasının genetik algoritmaya rakip olabileceği düşünüldüğünden; yabani ot algoritmasını ARP'ye uyarlamak ve genetik algoritma ile kıyaslamasını yapmak bu çalışmanın amaçları arasındadır.

Çalışma amaçları;

1. Dinamik koşullar altında tek depolu, çok araçlı, dinamik talepli, kapasite kısıtlı araç rotalama problemini ele almak.
2. Kapasite kısıtlı statik probleme yeni çözüm yöntemi önermek
3. Bu çözüm yöntemlerini dinamik koşullara adapte etmek
4. Ortaya konulan uygulama için kullanıcı arayüzü tasarlamak

1.5. Çalışmanın Kapsamı

İncelenen problem aşağıdaki karakteristiklere sahip olarak tanımlanmıştır:

- Tek depolu,
- Çok araçlı,
- Statik ya da dinamik talepli,
- Kapasite kısıtlı araç rotalama problemidir.

Problemin karakteristiği göz önüne alındığında bazı kısıtlamalar ile karşılaşılmıştır. Statik ARP için kullanılan varsayımlar aşağıda verilmiştir:

- Filo heterojendir.
- Uzaklıklar bilinmektedir ve gidiş dönüş uzaklıkları simetriktir.
- Talepler tek depodan karşılanmaktadır.
- Dağıtım için yeterli sayıda araç vardır.
- Talepler önceden bilinmektedir ve süreç boyunca değişmemektedir.
- Her bir rota depodan başlar ve depoda biter.

- Her müşteriye yalnızca tek bir araç uğrar.
- Rotada üzerinde müşterilerin talepleri toplamı araç kapasitesini geçemez.

Dinamik ARP için kullanılan varsayımlar aşağıda verilmiştir:

- Filo heterojendir.
- Uzaklıklar bilinmektedir.
- Talepler tek depodan karşılanmaktadır.
- Dağıtım için yeterli sayıda araç vardır.
- Her bir rota depodan başlar ve depoda biter.
- Her müşteriye yalnızca tek bir araç uğrar.
- Rotada üzerinde müşterilerin talepleri toplamı araç kapasitesini geçemez.
- Dinamik problemde zaman içerisinde taleplerde değişim (yeni taleplerin gelişi, talep iptalleri ve her iki durumun birlikte oluşması) oluşabilir.
- Dinamik problemde olası rota değişiklikleri oluşabilir.
- Araçların rotalara göre dağıtıma çıktığı varsayılmıştır.
- Yeni talep geldiğinde veya talep iptali olduğunda; araçların bulunduğu konum rota üzerindeki müşteri konumu veya depo olabileceği varsayılmıştır. (Bu varsayım dinamik problem için üretilen çözümün kalitesini, kolay şekilde anlamak için yapılan bir varsayımdır.)
- Dağıtıma çıkan araçların; rotasında bulunan müşteri talepleri kadar yükü bulunduğu varsayılmıştır.

1.6. Çalışmanın Özgün Katkısı

Çalışmanın temel katkısı üç başlık altında değerlendirilebilir:

- Bu çalışmada, kapasite kısıtlı ve tek depolu araç rotalama problemi ele alınmıştır. Bu problemin çözümü için tasarruf algoritması, genetik algoritma, yabancı ot algoritması ve bu yaklaşımların hibrit versiyonları önerilmiştir. Elde edilen sonuçlar önceki çalışmalarda yer alan test problemleriyle sınanmış ve yapılan karşılaştırmalara göre hibrit yaklaşımla daha hızlı ve tutarlı bir çözüm yaklaşımı önerilmiştir.
- Dinamik araç rotalama problemi ele alınmış ve tek depolu, çok araçlı, kapasite kısıtlı araç rotalama probleminin dinamik versiyonu incelenmiştir. Buna göre talep koşullarındaki değişiklik ele alınmış ve yeni talep oluşması, mevcut talebin iptal edilmesi ve her iki durumun aynı anda olması durumlarına göre oluşacak bir çözüm yaklaşımı önerilmiştir. Klasik problem için elde edilen en iyi hibrit çözüm yaklaşımları dinamik probleme uyarlanmış ve dinamik araç rotalama probleminin çözümü sağlanmıştır.
- Dinamik problemin gerçek hayat uygulamalarında kullanımının sağlanması önemlidir. Bu nedenle bir kullanıcı arayüzü geliştirilmiş ve bu arayüzle birlikte geliştirilen yaklaşımların yaygınlaştırılması sağlanmıştır. Ayrıca gerçek hayat problemleri için noktalar arası uzaklık öklid kuralına göre değil, gerçek araç yolu uzunluğu temel alınarak problem çözümü sağlanmıştır.

1.7. Çalışmanın Adımları ve Organizasyonu

1.7.1. Çalışmanın Adımları

Problemin Belirlenmesi: İncelenecek problemin araştırılması ve belirlenmesi sürecidir. Bu aşamada problem ile ilgili bilgiler toplanır.

Problemin Tanımı: Bu çalışma kapsamında ele alınacak problemin tanımının yapılması, varsayımlarının ortaya konması aşamasıdır.

Önceki Çalışmalar: Araç rotalama problemi ile ilgili literatür taramasının yapıldığı aşamadır.

Geliştirme: Probleme ait algoritmaların analiz edilmesi, çalışma kapsamına uygun olarak tasarımının gerçekleştirilmesi, kodlama ve test süreçlerini içeren adımdır.

Problemin Çözümü: Belirlenen veri setleri ve ortaya konulan yazılımdaki algoritmalarla problemlerin çözülmesi adımdır.

Sonuçlar: Problemin çözümünden elde edilen sonuçların değerlendirildiği aşamadır. Ayrıca bu aşamada gelecek çalışmalar için önerilere de yer verilir.

1.7.2. Çalışmanın Organizasyonu

Çalışmanın birinci bölümünde araç rotalama probleminin tanımı yapılmıştır, problemin varsayımlarına yer verilmiş, çözüm yöntemlerine değinilmiştir. Ayrıca çalışma hakkında bilgilendirilmeler yapılmıştır. Çalışmanın ikinci bölümünde araç rotalama ve dinamik araç rotalama problemi ile ilgili literatürde yer alan çalışmalara yer verilmiş ve bu çalışmaların değerlendirilmesi yapılmıştır. Üçüncü bölümde, algoritmalar hakkında bilgiler verilmiş, algoritmanın adımları ortaya konmuş ve probleme uyarlanması anlatılmıştır. Ayrıca bu bölümde kullanılan veriler ve uygulamanın ekran görüntüleri yer almıştır. Dördüncü bölümde, kullanılan problemlerin çözüm sonuçları tablolar halinde verilmiştir ve elde edilen çözüm sonuçları ile ilgili analizler yapılmıştır. Son bölümünde problemlerin çözümünden elde edilen sonuçlar değerlendirilmiştir ve gelecekte yapılacak çalışmalara önerilerde bulunulmuştur.

2. ÖNCEKİ ÇALIŞMALAR

2.1. Statik Araç Rotalama Problemi

Araç rotalama problemi literatüre 1959 yılında Dantzig ve Ramser tarafından yapılan çalışma ile girmiştir. Bu çalışmalarında benzin istasyonlarına benzin dağıtımını için araç rotalama problemini ele almış ve bu problemin çözümü için ilk kez doğrusal matematiksel model oluşturarak literatüre bir yaklaşım kazandırmışlardır (Dantzig ve Ramser, 1959).

Clarke ve Wright 1964 yılında yaptıkları çalışma ile araç rotalama problemine ilk defa sezgisel bir algoritma önermişlerdir. Çalışmalarında algoritmanın adımını ve işleyişini açıklamışlardır. Algoritmalarını tasarruf algoritması olarak adlandırmışlardır. Tasarruf algoritması, Clarke ve Wright algoritması olarak da bilinmektedir.

Eryavuz ve Gencer 2001 yılında yaptıkları çalışmada; personel servis araçlarının toplam güzergah mesafesini minimize etmeye çalışmışlardır. Problemin çözümü için tasarruf algoritmasını kullanmışlardır ve elde edilen sonuç ile mevcut durumu karşılaştırmışlardır. Sonuç olarak mevcut duruma göre iyileşme kaydetmişlerdir.

Pisinger ve Ropke 2007 yılında yaptıkları çalışmada araç rotalama probleminin beş farklı çeşidi (zaman kısıtlı araç rotalama problemi, kapasiteli araç rotalama problemi, çok depolu araç rotalama problemi, yer-bağımlı araç rotalama problemi ve açık araç rotalama problemi) için birleştirilmiş sezgisel algoritma önermişlerdir.

Tarantilis ve arkadaşları tarafından 2005 yılında yapılan çalışmada; araç rotalama probleminin standart ve çok geniş çalışılmış versiyonu olan kapasiteli araç rotalama problemi için metasezgisel çözüm metodolojilerini incelenmiştir. Her bir metasezgiselin hesaplama performansı gösterilmiştir.

Ergülen ve Güngör tarafından 2006 yılında yapılan çalışmada; bir merkezden çok sayıda müşteriye dağıtım işleminin yapıldığı bir fabrika üzerinde çalışılmıştır. Müşteri talep miktarlarının değişken olduğu durumlardaki ARP için kullanılabilir bir karışık tamsayılı doğrusal programlama modeli önerilmiştir. Önerilen yaklaşımı gerçek hayat problemine uygulamışlardır. Çalışmaları sonucunda; bulanık mantık ile oluşturulan modelin, bulanık mantık içermeyen modelden daha iyi sonuçlar ürettiğini göstermişlerdir. Fabrikanın toplam taşıma maliyetlerini %15 oranında düşürmüşlerdir.

Düzakın ve Demircioğlu 2009 yılında yaptıkları çalışmada araç rotalama problemi için kesin ve sezgisel yöntemleri açıklamışlardır.

Yücenur ve Demirel tarafından 2011 yılında yapılan çalışmada; çok depolu araç rotalama probleminin çözümü için karınca kolonisi optimizasyonu ve genetik algorithmadan oluşan melez bir yaklaşım önerilmiştir. Önerilen melez yaklaşımı literatürden alınan problemler ile test etmişlerdir. Ortaya konulan çözüm sonuçları diğer algoritmaların çözümleriyle karşılaştırılmıştır. Önerilen yöntemin kapasitenin düşük olduğu orta boyutlu problemlerde iyi çözümler ürettiği ortaya konmuştur.

Atmaca tarafından 2012 yılında yapılan çalışmada; eş zamanlı dağıtım toplamalı araç rotalama problemi ele alınmıştır. Bir kargo şirketi için bu problem GAMS programı kullanılarak çözülmüştür. Şirketin önceki durumları ile ortaya konan sonuçlar karşılaştırılmıştır. Önerilen durumda; toplam rotalarda azalma, araç kapasite kullanımında artış, araç ve rota sayılarında azalma sağlamışlardır. Sonuç olarak mevcut sistemde önemli ölçüde iyileştirme gerçekleştirilmiştir.

Kosif ve Ekmekçi 2012 yılında yaptıkları çalışmada; bir lojistik firmasının araç rotalama problemini ele almış ve tasarruf algoritması ile problemi çözerek firmanın lojistik maliyetlerini düşürmüşlerdir.

Bozyer ve arkadaşları 2014 yılında yaptıkları çalışmada; kapasite kısıtlı araç rotalama problemlerinin (KKARP) çözümü için önce grupta sonra rotala prensibine dayanan sezgisel bir yöntem önermişlerdir. KKARP'lerinin gezgin

satıcı problemine dönüştürülerek çözülebileceğini söylemişlerdir. Önerilen yöntemi literatürde yer alan test problemleri ile test etmiş ve bazı problemler için en iyi sonuçlara ulaşmışlardır.

Kiremitçi ve arkadaşları tarafından 2014 yılında yapılan çalışmada; çok araçlı, dağıtım toplamalı ve zaman pencereli araç rotalama problemi ele alınmıştır. Bu problem genetik algoritma ile çözülmüştür. Genetik algoritmaya ait değişkenler farklı bir yapıda gerçek değerli kodlanarak küçük boyutlu kromozom ve daha az değişkenle çözüm üretilmeye çalışılmıştır. Literatürdeki bazı problemler üzerinde çalıştırılmış ve çözümler mevcut algoritmaların çözümleriyle karşılaştırılmıştır.

Şahin ve Eroğlu 2014 yılında, metasezgisel yöntemler ve bunların kapasite kısıtlı araç rotalama problemine uygulanışı hakkında literatür araştırması yapmışlardır.

Yazgan ve arkadaşları 2014 yılında yaptıkları çalışmada; talep ve kapasite kısıtlı araç rotalama problemini ele alınmış ve çözümü için tasarruf algoritmasını esas alan yeni bir hibrit yaklaşım önermişlerdir. Önerilen algoritmayı farklı problem setleri ile denemiş ve önerilen hibrit yöntemin daha iyi sonuçlar verdiğini göstermişlerdir.

Wedyan ve Narayanan 2014 yılında yaptıkları çalışmada kapasite kısıtlı araç rotalama probleminin çözümü için su akışından esinlenen akıllı su damlası metasezgisel algoritmasını önermişlerdir. Literatürden aldıkları bazı test problemleri üzerinde akıllı su damlası algoritmasını uygulamışlardır. Aynı problemleri tasarruf algoritması ile çözüp karşılaştırmalı sonuçlar sunmuşlardır.

Atmaca ve arkadaşları tarafından 2015 yılında yapılan çalışmada ürün dağıtımını yapan bir firmanın zaman pencereli araç rotalama problemi ele alınmış, karışık tamsayı programla ile çözülmüştür. Mevcut duruma göre %67,90 oranında iyileştirme ortaya koyulmuştur.

Keskintürk ve arkadaşları tarafından 2015 yılında yapılan çalışmada; ARP türleri sınıflandırılmıştır. Sınıflandırma yapılan araç rotalama problemlerinin

çeşitleri ve çözüm yöntemleri ele alınmıştır. ARP için çözüm yöntemleri; kesin yöntemler, sezgisel yöntemler ve metasezgisel yöntemler olmak üzere üç grupta ele alınmıştır. Çalışmada, kapasite kısıtlı araç rotalama probleminin iki sezgiselle çözülmesi sağlanmış ve analizleri yapılmıştır.

Karagül ve arkadaşları 2016 yılında yaptıkları çalışmada; başlangıç rotalarının kurulması için Newton'un çekim yasası tabanlı bir algoritma önermişlerdir. Önerilen algoritmayı, genetik algoritmanın başlangıç popülasyonunu oluşturmak ve genetik algoritmanın ürettiği sonuçları iyileştirmek için kullanmıştır. Önerdikleri yaklaşımı Augerat vd. (1995) den aldıkları A, B ve P grubu 74 adet kapasiteli araç rotalama test problemleri üzerinde denemişlerdir. Önerdikleri yaklaşım için bu grupların ortalama saplamalarını sırasıyla %37,95, %32.10 ve %31.45 olarak bulmuşlardır. Önerdikleri yaklaşımı genetik algoritmanın başlangıç popülasyonu için kullandıklarında, ortalama sapmaları sırasıyla % 7.15, %4.33 ve %6.33 olarak bulmuşlardır. Algoritmalarının başka algoritmalar için başlangıç çözümü olarak kullanılabileceğini ifade etmişlerdir.

Ulutaş ve arkadaşları 2017 yılında yaptıkları çalışmada; bir ekmek fırınının araç rotalama problemini ele almış, problemin çözümü için basit ve pratik olmasından dolayı tasarruf algoritması kullanmışlardır. Sonuç olarak fırının dağıtım maliyetini düşürmüşlerdir.

Ahmed ve Sun 2018 yılında yaptıkları çalışmada kapasite kısıtlı araç rotalama probleminin çözümü için iki katlı yerel arama tabanlı parçacık sürü optimizasyonunu (Bilayer Local Search-based Particle Swarm Optimization) önermişlerdir. Önerilen yaklaşımı literatürden alınan problem üzerinde uygulamışlardır. Problemlerin çoğu için bilinen en iyi çözümlere çok kısa hesaplama süreleri ile ulaştıklarını göstermişlerdir. Ayrıca sonuçlara göre, önerilen algoritma ile elde edilen performansın diğer bazı parçacık sürü optimizasyon tabanlı yaklaşımlardan daha iyi performans gösterdiğini kaydetmişlerdir.

Cuevas ve arkadaşları 2018 yılında yaptıkları çalışmada nakliye şirketleri için kapasite kısıtlı araç rotalama problemini incelemişlerdir. Uyarlanmış kapasite kısıtlı araç rotalama test problemleri ve nakliye şirketinden elde edilen gerçek verilerle ile modelin doğrulamasını yapmışlardır. Kurdukları model ile dağıtım maliyetlerinin azaldığını söylemişlerdir.

Zhang ve arkadaşları 2018 yılında yaptıkları çalışmada, kapasite kısıtlı yeşil araç rotalama problemini ele almışlardır. Bu problemde, ürünleri dağıtmak için alternatif yakıtla çalışan araçlar kullanılmıştır. Alternatif yakıtla çalışan araçların düşük yakıt deposu kapasitesine sahip olduğu varsayılmıştır. Bu nedenle dağıtım işlemleri sırasında, araçların yakıt ikmali için alternatif yakıt istasyonlarını ziyaret etme zorunluluğu vardır. Bu sorunu çözmek için iki fazlı sezgisel algoritma ve karınca kolonisi algoritmasına dayalı metasezgisel algoritma iki çözüm yöntemi olarak önerilmiştir. Önerilen algoritmaların performansını değerlendirmek için sayısal deneyler rastgele oluşturulmuş problemlerle gerçekleştirilmiştir. Sayısal deneylerin sonucunda ortalama olarak, karınca kolonisi algoritmasının performansının iki fazlı sezgiselin performansından %38.27 daha iyi olduğu ancak hesaplama süresinin uzun olduğu gösterilmiştir.

Arnold ve arkadaşları 2019 yılında yaptıkları çalışmada; kapasite kısıtlı araç rotalama probleminin çok büyük ölçekli örnekleri için makul bir hesaplama süresi içinde, iyi çözümler üreten yerel bir arama sezgiseli yönteminin nasıl tasarlandığını açıklamışlardır. Budama ve sıralı arama (pruning and sequential search) ile zaman karmaşıklığını ve depolanan bilgi miktarını kısıtlayarak alan karmaşıklığını azaltmanın farklı yollarını araştırmışlardır. Sonuçta ortaya çıkan algoritmanın, 10.000 ve daha fazla müşteri örneği için önceden tanıtılmış sezgisellerden daha iyi performans gösterdiğini ortaya koymuşlardır.

Akakaki ve Usberti 2019 yılında yaptığı çalışmada; Golden ve Wong (1981) tarafından önerilen kapasite kısıtlı ark rotalama problemini (capacitated arc routing problem) ele almışlardır. Kapasite kısıtlı homojen araç

filosu ile depoda başlayan ve depoda biten rotaları optimize etmeyi amaçlamışlardır. Bu problem için verimlilik kuralı kavramını ortaya koyan yeni bir yol tarama (path-scanning) sezgiseli önerilmiştir. Mevcut araç konumu, kat edilen mesafe ve hizmet verilen talep miktarı göz önüne alındığında, verimlilik kuralı bir sonraki hizmet için en uygun kenarları seçtiğini açıklamışlardır. Bir dizi karşılaştırmalı örnek üzerinde yapılan deneyler sonucunda, önerilen sezgiselin literatürdeki tüm yol tarama sezgisellerini geride bıraktığını göstermişlerdir.

Benrahou ve Tairi 2019 yılında yaptıkları çalışmada; atık yağ toplaması için kapasite kısıtlı araç rotalama problemini ele almışlardır. Problemin çözümü için en yakın ekleme algoritması(Nearest Insertion Algorithm) olarak adlandırılan sezgisel yöntemi adapte etmişlerdir. 26 düğümlü modeli çözmek için MATLAB kullanmışlardır. Sonuçta uzaklığı %29.2 oranında azaltmışlardır.

Normasari ve arkadaşları 2019 yılında yaptıkları çalışmada kapasite kısıtlı yeşil araç rotalama problemini ele almışlardır. Problemin çözümü için tavlama benzetimi sezgiselini önermişlerdir ve sayısal deneyler yapmışlardır. Deney sonuçlarına göre tavlama benzetimi algoritması küçük ölçekli ve büyük ölçekli problemlerde kabul edilebilir sürelerde iyi çözümler üretmiştir.

Mulloorakam ve Nidhiry 2019 yılında yaptıkları çalışmada birleşik amaçlı kapasite kısıtlı araç rotalama problemini incelemişlerdir. Probleme araç sayısı ve uzaklık olarak iki amaca odaklanmışlardır. Problemin çözümü için genetik algoritma tabanlı yaklaşım tanıtmışlardır. Önerilen algoritmanın son derece rekabetçi ve verimli olduğunu belirtmişlerdir.

Toffolo ve arkadaşları 2019 yılında yaptıkları çalışmada, kapasite kısıtlı araç rotalama problemini ele almışlardır. Hızlı bir çözüm için çeşitli hızlandırma teknikleri önerilmektedir. Bunlar komşu azaltmaları, dinamik hareket filtreleri, bellek yapıları ve birleştirme teknikleridir. Son olarak, algoritma ilerledikçe arama uzayını yeniden şekillendirmek için bir tünel stratejisi tasarlanmıştır. Önerilen

yaklaşımı literatürden alınan bazı problemlerle test etmiştir. Test edilen problemlerde en iyi çözümler bulunmuştur.

Wang ve Lu 2019 yılında yaptıkları çalışmada kapasite kısıtlı yeşil araç rotalama problemini rekabete dayalı memetik algoritma ile çözmüşlerdir. Önerilen algoritmanın bu problemi çözmede ayrık ateşböceği algoritması ve karınca kolonisi optimizasyonundan daha etkili ve verimli olduğunu göstermişlerdir.

2.2. Dinamik Araç Rotalama Problemi

Dinamik ARP literatüre 1988 yılında yapılan çalışma ile tanıtılmıştır. Çalışmada statik ve dinamik araç rotalama problemleri arasındaki farklar ortaya konulmuştur. Statik yaklaşımların dinamik ortama adaptasyonu tartışılmış ve açıklanmıştır. Ayrıca bir kargo şirketinin araçlarının dinamik olarak rotalama problemini çözen bir algoritma önerilmiştir (Psaraftis, 1988).

Psaraftis 1995 yılında DARP ile ilgili geniş bir derleme çalışması yaparak bu alandaki literatüre katkı sağlamıştır. Ayrıca küresel konumlandırma sistemleri, akıllı araç-karayolu sistemleri, coğrafi bilgi sistemleri, elektronik veri değişimi gibi alanlara değinmiş, sonuçlarını ve avantajlarını incelemiştir (Psaraftis, 1995).

ARP için literatürde yer alan test problemleri Kilby v.d. (1998) tarafından geliştirilerek DARP için test problemleri üretilmiştir. Dinamik hale getirilen test problemlerine, çalışma günü uzunluğu, taleplerin ortaya çıkış zamanları, her bir talebin servis süresi gibi zamana bağlı parametreler eklenmiştir (Demirtaş ve Özdemir, 2017).

Hu ve arkadaşları 2002 yılında yaptıkları çalışmada; stokastik araç rota problemlerini (SVRP), dinamik araç rota problemleri için gerçek zamanlı bilgileri içerecek şekilde genişletmişlerdir. Bu problemi dal sınır tekniği ile çözmüşlerdir.

Erera ve Daganzo 2003 yılında yaptıkları çalışmada tek depolu, homojen araç filolu, stokastik kapasite kısıtlı araç rotalama problemini incelemiştir. Müşteri boyutları ve yük boyutları operasyondan önce dağıtım aşamasında

bilinmektedir. Bu çalışmada eşik küresel paylaşım (threshold global sharing) yöntemiyle maliyetler düşürülmeye çalışılmıştır.

Bayzan 2009 yılında yaptığı çalışmada; rota üzerindeki trafiğin yoğun olması, yolun kapalı olması gibi durumları ele almıştır. Belirli noktalara yerleştirilmiş cihazlardan gelen bilgileri değerlendirerek hesaplama yapan ve kullanıcılara alternatif rota öneren akıllı bir sistem tasarlamıştır.

Demirtaş ve Özdemir 2017 yılında yaptıkları çalışmada; DARP incelenmiş ve parçacık sürü optimizasyonu (PSO) yöntemi ile bir çözüm önerilmiştir. Literatürde çalışılan test problemleri PSO ile çözülmüş ve sonuçlar literatürde yer alan diğer yöntemler ile karşılaştırılmıştır. Çalışmanın son bölümünde ise bir gerçek hayat problemi ele alınmıştır. Ele alınan probleminin çözümü için önerilen PSO algoritması uygulanarak elde edilen sonuçlar hali hazırda kullanılan rotalar ile karşılaştırılmaktadır. Bu çalışma; DARP ile ilgili karşılaştırılabilir sonuçlar sunmaktadır.

Ünlü ve arkadaşları 2017 yılında yaptıkları çalışmada zaman pencereli çok araçlı dinamik rotalama problemini ele almışlardır. Önerilen yöntemle gerçek hayat kısıtları çok fazla basitleştirilmeden problem alanında bir modelleme yapılmış, yeni gelişen gerçek zamanlı müşteri taleplerini destekleyecek dinamik bir rotalama sağlandığını söylemişlerdir.

2.3. Yabani Ot Algoritması İle İlgili Çalışmalar

Mehrabian ve Lucas 2006 yılında yaptığı çalışmada; yabani otların güçlü ve istilacı büyümesinden ilham alan, sağlamlığını ve adaptasyonunu taklit eden yabani ot algoritmasını önermiştir. Bu algoritmayı; genetik algoritmalar, memetik algoritmalar, parçacık sürü optimizasyonu ve kurbağa sıçrama algoritması gibi evrimsel algoritmalarla karşılaştırmıştır. Yabani ot algoritmasını bir mühendislik probleminin çözümü için kullanmıştır. Deneysel sonuçlarda yabani ot

algoritmasının performansının tüm test fonksiyonları için kabul edilebilir olduğunu göstermiştir.

Pahlavani ve arkadaşları 2012 yılında yaptıkları çalışmada; kişiselleştirilmiş kentsel çok ölçütlü yol optimizasyon problemini (personalized urban multi-criteria path optimization problem) ele almışlardır. Bu problem çok ölçütlü en kısa yol probleminin bir çeşididir. Problemin çözümü için değiştirilmiş yabani ot algoritması uygulanmış ve genetik algoritma ile karşılaştırılmıştır. Sonuç olarak genetik algoritma ile karşılaştırıldığında daha iyi sonuçların elde edildiği kaydedilmiştir.

Mohammadi ve arkadaşları 2013 yılında yaptıkları çalışmada yer seçimi-rotalama probleminin bir türü olan stokastik yeşil ana dağıtım merkezi yer seçimi-rotalama problemini ele almışlardır. Problemin çözümü için çok amaçlı karışık tamsayı doğrusal programlama formülasyonları önermişlerdir. Ayrıca hibrit çok amaçlı metasezgisel olan çok amaçlı yabani ot optimizasyonunu uygulamışlardır. Önerilen algoritmanın NSGA-II (nondominated sorting genetic algorithm II), PAES (pareto archived evolution strategy) ve SPEA (strength pareto evolutionary algorithm) gibi literatürdeki temel çok amaçlı algoritmaları geride bıraktığını göstermişlerdir.

Sur ve Shukla 2013 yılında yaptıkları çalışmada grafik tabanlı kombinatoriyal yol ağı yönetim problemini (Graph Based Combinatorial Road Network Management Problem) ele almışlardır. Bu probleminin çözümü için ayrık yabani ot algoritmasını kullanmışlardır. Bu çalışmada ilk defa yabani ot algoritmasının ayrık versiyonu tanıtılmıştır. Ayrık yabani ot algoritmasını, karınca koloni algoritması ve akıllı su damlası algoritması ile karşılaştırmışlardır ve sonuçları sunmuşlardır. Sonuçlara göre ayrık yabani ot algoritmasının diğer iki algoritmadan daha iyi yakınsama oranına sahip olduğunu kaydetmişlerdir.

Zhao ve arkadaşları 2016 yılında yaptıkları çalışmada kapasite kısıtlı araç rotalama problemini ele almışlardır. Problemin çözümü için ayrık hibrit yabani ot

algoritmasını kullanmışlardır. Algoritmanın sonuçlarını iyileştirmek için adaptif mutasyon ve adaptif çaprazlama, iki aşamalı hibrit komşu arama algoritmasını eklemişlerdir. Algoritmanın başarımını değerlendirmek amacıyla Augerat vd. (1995) den aldıkları A grubu, B grubu ve P grubu veri setlerinden bazı problemleri seçmişlerdir. Sonuç olarak kapasite kısıtı göz önüne alınmadığında bilinen en iyi problemde daha iyi sonuçlar elde edilmiştir.

Nath ve arkadaşları 2017 yılında yapılan çalışmada; çok geniş ölçekli tüm devre (VLSI) rotalamaya yabancı ot algoritmasını önermişlerdir. VLSI rotalamada, karınca koloni optimizasyonu, parçacık sürü optimizasyonu, ateşböceği algoritması gibi metasezgisel algoritmalar daha önceki çalışmalarda uygulanmıştır. Yapılan deneylerle, yabancı ot algoritmasının, parçacık sürü optimizasyonu ve ateşböceği algoritması gibi literatürde kullanılan metasezgisellerle karşılaştırıldığında bu problem için daha etkili bir algoritma olduğu gösterilmiştir.

Jahangir ve arkadaşları 2018 yılında yaptıkları çalışmada envanter rotalama problemini ele almışlardır. Büyük ölçekli problemlerin çözümü için ayrıık yabancı ot algoritması ve genetik algoritma olmak üzere iki metasezgisel önermişlerdir. Elde edilen çözüm sonuçlarını, algoritmaları karşılaştırmak için kullanmışlardır. Bu sonuçlara göre; ayrıık yabancı ot algoritmasının, genetik algoritmadan daha iyi yakınsamaya sahip olduğunu göstermişlerdir. Küçük problemlerde çalışma süresi olarak ayrıık yabancı ot algoritmasının daha iyi, büyük problemlerde genetik algoritmanın daha iyi olduğunu aktarmışlardır.

2.4. Önceki Çalışmaların Değerlendirilmesi

Literatürde araç rotalama problemi, çeşitli kısıtlar içeren problemlerin modellenmesi, optimum çözüm için farklı algoritmaların kullanılması ve gerçek hayattaki sorunlar için uygulamalar yapılması şeklindedir (Ergülen ve Güngör, 2006).

ARP'nin çözümü için kullanılan yöntemleri kesin yöntemler, sezgisel yöntemler ve metasezgisel yöntemler olarak ayırmak mümkündür. Kesin yöntemlerde; dinamik programlama, dal sınır algoritmaları, dal-kesme algoritmaları yer almaktadır. Sezgisel yöntemlerde; tasarruf algoritması, eşleme tabanlı tasarruf algoritması, en kısa yol yöntemi gibi algoritmalar yer alır. Metasezgisel yöntemlerde; tavlama benzetimi, tabu arama algoritması, karınca kolonisi algoritması, yapay arı algoritması, parçacık sürü optimizasyonu, yapay sinir ağları, genetik algoritmalar ve hibrit metasezgisel algoritmalar yer almaktadır.

Önceki çalışmalar incelendiğinde dinamik araç rotalama probleminin statik versiyonuna göre daha az irdelendiği görülmektedir. Dinamik araç rotalama probleminin yapısının daha karmaşık olmasından dolayı karmaşık modellemeye ihtiyaç duymaktadır. Bundan dolayı daha az ele alınmıştır. Bununla birlikte, gerçek hayat problemlerine yakın olması problemi ilgi çekici hale getirmektedir.



3. MATERYAL VE METOT

Bu bölümde çalışmada kullanılan materyal ve metotlar hakkında bilgiler verilmiştir.

3.1. Materyal

3.1.1. Problemden Kullanılan Veriler

Literatürde yaygın olarak kullanılan P grubu 24 adet problem seçilmiştir. P grubu KKARP test problemleri Augerat vd. (1995)'ten alınmıştır. Ayrıca literatürde CMT adıyla anılan 5 adet problem seçilmiştir. CMT problemleri Christofides vd. (1979)'ten alınmıştır. Tercih edilen veri setleri, kapasite kısıtlı araç rotalama problemleridir. Her bir problem tek depo, birden fazla araca sahiptir ve araçlar homojendir. Tasarruf algoritması, genetik algoritma, yabani ot algoritması ve hibrit algoritmalar bu veri setleri ile çalıştırılmıştır.

Seçilen veri setlerinde farklı boyutlarda problemler olması algoritmaların problemler üzerindeki etkilerini daha iyi analiz etmeyi sağlamıştır. Veri setlerinde; depo ve düğümler, bunlara ait koordinatlar, kapasite kısıtı ve araç kısıtı bulunmaktadır. Bu veri setlerinin optimum çözüm değerleri de bulunmaktadır.

Çizelge 3.1'de <http://neo.lcc.uma.es/vrp/known-best-results/> adresinden alınan Augerat tarafından (1995) yapılan çalışmaya ait P grubu veri seti verilmiştir. Çizelge 3.2'de <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/> adresinden alınan Christofides v.d. tarafından (1979) tarafından yapılan çalışmaya ait CMT grubu veri seti verilmiştir.

Çizelge 3.1. 24 Adet P Grubu KKARP Problem Bilgileri (Augerat, 1995)

Veri Seti	Lokasyon Sayısı	Araç Sayısı	Kapasite	BKS
P-n16-k8	15	8	35	450
P-n19-k2	18	2	160	212
P-n20-k2	19	2	160	216
P-n21-k2	20	2	160	211
P-n22-k2	21	2	160	216
P-n22-k8	21	8	3000	590
P-n23-k8	22	8	40	529
P-n40-k5	39	5	140	458
P-n45-k5	44	5	150	510
P-n50-k7	49	7	150	554
P-n50-k8	49	8	120	629
P-n50-k10	49	10	100	696
P-n51-k10	50	10	80	741
P-n55-k7	54	7	170	568
P-n55-k8	54	8	160	588
P-n55-k10	54	10	115	694
P-n55-k15	54	15	70	945
P-n60-k10	59	10	120	744
P-n60-k15	59	15	80	968
P-n65-k10	64	10	130	792
P-n70-k10	69	10	135	827
P-n76-k4	75	4	350	593
P-n76-k5	75	5	280	627
P-n101-k4	100	4	400	681

Çizelge 3.2. 5 adet CMT Grubu KKARP Problem Bilgileri (Christofides vd., 1979)

Veri Seti	Lokasyon Sayısı	Araç Sayısı	Kapasite	BKS
CMT1	50	5	160	524.61
CMT2	75	10	140	835.26
CMT3	100	8	200	826.14
CMT4	150	12	200	1028.42
CMT5	199	17	200	1291.29

Dinamik araç rotalama problemini ele almak için P grubu veri seti içinden P-n50-k7 isimli veri seti seçilmiştir. Bu problem, tek depolu, 7 araçlı, 49 müşterili kapasite kısıtlı araç rotalama problemidir. Bu veri setinde belirli varsayımlar yapılmış, senaryolar kurgulanmış ve dinamiklik eklenmiştir.

3.2. Metot

3.2.1. Tasarruf Algoritması

Clarke ve Wright tarafından 1964 yılında önerilen Tasarruf Algoritması sezgisel bir algoritmadır. (Clarke ve Wright, 1964). Clarke ve Wright algoritması olarak da bilinir. Tasarruf algoritması, optimum veya optimuma yakın sonucu üretir.

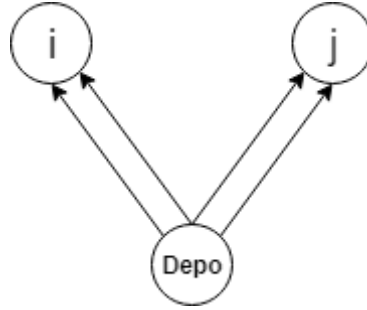
Tasarruf algoritmasının sıralı ve paralel olmak üzere iki versiyonu bulunmaktadır (Yazgan vd., 2014). Sıralı versiyonunda aynı anda tek rota oluşturulur. Paralel versiyonda aynı anda birden fazla rota oluşturulabilir. Paralel versiyonunun aynı anda birden fazla rotayı ele almasından dolayı daha iyi sonuçlar üretilir. Bu çalışmada paralel tasarruf algoritması ele alınacaktır.

3.2.1.1. Algortimaya Genel Bakış

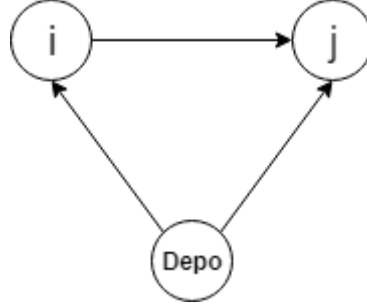
Tasarruf algoritmasında bir adet depo ve birden fazla dağıtım noktası vardır. En büyük tasarruf değerinden başlayarak rotalar belirlenir ve birleştirilir (Ulutaş vd., 2017).

Tasarruf algoritmasında her bir talep noktasına ayrı ayrı hizmet verilerek oluşacak maliyetler hesaplanır. Sonra bütün müşteriler çift oluşturacak şekilde birleştirilir. Bu birleştirme sonucunda oluşacak tasarruf miktarları hesaplanır. Oluşan en büyük tasarruf miktarının, rotaları minimize etmeye en büyük katkıyı sağlayacağı düşünülür. Bu amaçla tasarruf miktarı büyükten küçüğe doğru sıralanır ve müşteri çiftleri bu sıraya göre birleştirilir.

Lysgaard (1997) e göre temel tasarruf kavramı; iki rotanın tek rotada birleştirilerek elde edilen maliyet tasarruflarını ifade eder. Şekil 3.1’de talep noktalarına ayrı ayrı hizmet verilmesi durumu ve Şekil 3.2’de de müşterilerin çift olarak birleştirilmesi durumu gösterilmiştir. Şekil 3.1’in maliyetinin, Şekil 3.2’nin maliyetinden farkı tasarruf değerini oluşturur.



Şekil 3.1. Planlamadan Önce



Şekil 3.2. Planlamadan Sonra

Başlangıçta Şekil 3.1’de i ve j müşterileri ayrı rotalarda ziyaret edilmektedir. Alternatif olarak iki müşterinin de aynı rota üzerinden ziyaret edilmesi Şekil 3.2’de gösterilmektedir. Ulaşım maliyetleri verildiğinde, Şekil 3.1 rotasının yerine Şekil 3.2 rotası kullanıldığında oluşan tasarruf hesaplanabilir.

c_{ij} i ve j arasındaki ulaşım maliyeti, D_a Şekil 3.1’deki toplam ulaşım maliyeti ve depo 0 ile gösterilmek üzere:

$$D_a = c_{0i} + c_{i0} + c_{0j} + c_{j0}$$

Aynı şekilde, D_b Şekil 3.2’deki toplam ulaşım maliyeti olmak üzere:

$$D_b = c_{0i} + c_{ij} + c_{j0}$$

İki rotanın birleştirilmesinden elde edilen tasarruf değeri S_{ij} olmak üzere:

$$S_{ij} = D_a - D_b = c_{i0} + c_{0j} - c_{ij}$$

Büyük tasarruf değerleri çekici ve önemlidir çünkü rota maliyetlerini düşürmede daha etkili olabileceği düşünülür. Tasarruf algoritması, bu tasarruf değerlerine göre rota oluşturma mantığına dayanmaktadır.

3.2.1.2. Algoritma Adımları

Tasarruf algoritmasının paralel versiyonun adımları aşağıda verilmiştir;

1. Tüm (i, j) çiftleri için tasarruf listesini hesapla. ($S_{ij} = c_{i0} + c_{0j} - c_{ij}$)
2. Tasarruf listesini (S_{ij}) büyükten küçüğe sırala.
3. Eğer iki öge bir turda değilse yeni tur yarat

Ögelerden ikisi farklı turdalarsa pas geç

Ögelerden biri turda diğeri turda değilse

Turda olan öge ortadaysa pas geç

Değilse; turda olan öge başta ise başa, sonda ise sona; turda olmayan ögeyi ekle

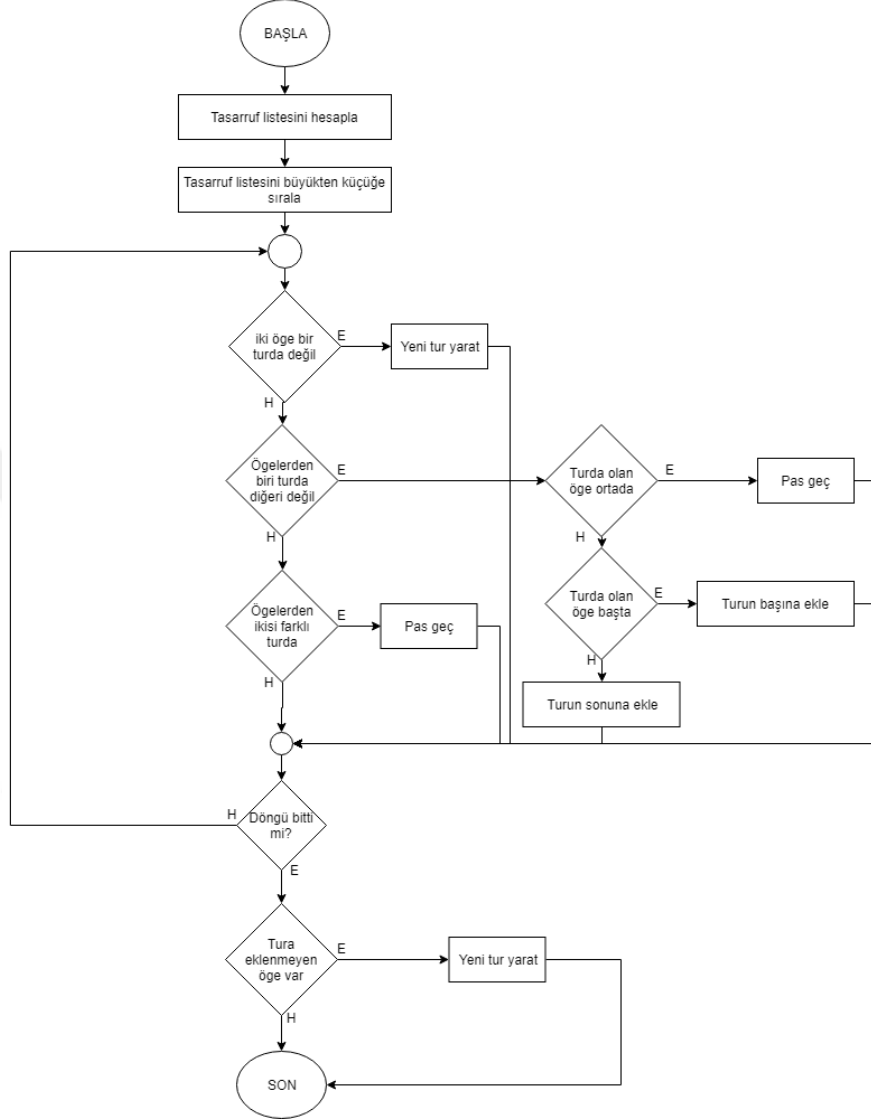
Kazanım sağlayan tüm ögeler tamamlandığında dur.

İşlem bittiğinde tura eklenmeyen öge kalırsa yeni tur yarat.

Ek kurallar

Hiç bir öge ilk iki öge arasına giremez.

Temel adımları verilen tasarruf algoritmasının akış diyagramı Şekil 3.3'te verilmiştir. Şekil 3.3'te görüldüğü üzere probleme ait bilgiler ile tasarruf değerleri hesaplanır ve tasarruf değerleri büyükten küçüğe doğru sıralanır. Tasarruf algoritmasının kuralları işletilir. Çıktı olarak problem çözülmüş ve rotalar oluşturulmuş olur.



Şekil 3.3. Tasarruf Algoritmasının Akış Diyagramı

Rotalar sıralı şekilde veya aynı anda (paralel) oluşturulabilir. Araç kapasitesi dolana kadar talep noktaları rotaya eklenirse, rotalar sıralı şekilde oluşturulmuş olur. Her bir araç için aynı anda rotalar oluşturulursa, rotalar paralel olarak oluşmuş olur.

Tasarruf algoritmasında paralel yöntem, sıralı yönteme göre daha iyi sonuç verir. Bunun sebebi sıralı yöntemde sadece bir rota ele alınırken, paralel yöntemde S_{ij} sıralamasına bakılarak müşteri çiftinin birden fazla rota ile uyumluluğu kontrol edilir. Bu sayede daha fazla alternatif gözden geçirilmiş ve daha iyi sonuçlar ortaya çıkmış olur (Erol, 2006).

3.2.2. Genetik Algoritma

Genetik algoritmalar, Darwin'in evrim teorisinden esinlenerek ortaya çıkmıştır. Bu algoritma, en iyinin korunumu ve doğal seçimle en iyilerin hayatta kaldığı evrimin bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir (Nabiyev, 2003).

Genetik algoritma canlıların evrimine dayanmaktadır. Arama ve optimizasyon algoritmasıdır. GA çözüm uzayındaki çözüm adaylarını kromozom adı verilen yapı ile kodlar. Çözüm adaylarına çaprazlama ve mutasyon gibi genetik operatörleri uygulayarak yeni bir popülasyon oluşturur. Birkaç kuşak sonra, popülasyon daha iyi uygunluk değerlerine sahip üyeleri içerir (Jang, 1997). Genetik algoritma doğadaki evrimi temel alan bir arama yöntemidir (Lawrence, 1990).

Genetik algoritma birçok uygulama alanı bulmuştur. Bunun en önemli nedeni zor problemlere çözüm üretmesi, güçlü ve geniş bir uygulama alanına sahip olmasıdır (Cheng vd., 1999).

3.2.2.1. Algoritmaya Genel Bakış

Genetik algoritmalar probleme tek bir çözüm aramak yerine çözüm kümesi üzerinde çalışır. Genetik algoritmalarda olası çözümler üretilir ve bu olası çözümler kademeli olarak iyileştirilmeye çalışılır. Genetik algoritma deterministik değildir, stokastik yöntemlerdir. Bu algoritma çözüm uzayının farklı noktasında aynı anda arama yapmaya yani paralel çalışmaya uygundur (Dikmen vd., 2014).

Genetik algoritmalar ile ilgili bazı önemli kavramlar; gen, kromozom (birey), popülasyon, seçme yöntemi, çaprazlama, mutasyon, elitizm ve uygunluk değeridir.

Gen: Genetik biliminde bilgi taşıyan DNA bloklardır. Benzer şekilde genetik algoritmalarda olası çözümün temelini oluşturan bir veya birkaç bitlik bilgi taşıyan bloklardır (Dikmen vd., 2014).

Kromozom (Birey): Genetik biliminde topluluğu oluşturan her varlığa birey denir. Bireyler genlerinde bulunan özellikleri taşırlar. Genetik algortmada ise birey genlerden oluşur ve her birey problem için bir çözüm adayıdır. Ayrıca bireyler kromozom olarak da adlandırılmaktadır (Dikmen vd., 2014).

Popülasyon: Genetik algortmada bireylerin bulunduğu topluluğa popülasyon denilmektedir. Bireyler problemin olası çözümünü ifade ettiği için popülasyon olası çözümlerden oluşur. Genetik algortma, popülasyonun kademeli olarak iyileştirilmesi temeline dayanır.

Jenerasyon: Yeni popülasyonu ifade eder.

Çaprazlama: Genetik biliminde bireylerin bir araya gelerek yeni yavru oluşturmalarına çaprazlama denir. Bireyler üreme sırasında bazı genlerini yavru bireye aktarır. Genetik algoritmalarda da benzer şekilde olası çözümlerin farklı bölümlerinin birleştirilmesine çaprazlama denir. Buradaki asıl amaç daha iyi çözümlere ulaşmaktır (Dikmen, 2014). Çaprazlama yöntemi bir bireyin belirli bir bölümü ile diğer bireyin belirli bölümünün birleştirilmesi ile yavru bireyin oluşturulmasıdır.

Seçme Yöntemi: Popülasyondan belirli yöntemle birey seçme işlemidir. Rastgele, turnuva, rulet çarkı gibi seçme yöntemleri vardır (Kahraman ve Özdağlar, 2004). En yaygın olarak kullanılan seçme yöntemi rulet çarkıdır. Rulet çarkı uygun değerlerine göre belirlenir. Turnuva seçiminde topluluktan rastgele bir birey seçilir ve eş olarak seçilecek birey kıyaslama yapılarak bulunur. Rastgele seçim yönteminde çaprazlanacak bireyler rastgele olarak seçilir (Artaç, 2003).

Mutasyon: Mutasyon ile kromozomda rastgele değişiklikler yapılır. Böylelikle çeşitlik sağlanmış olur. Ayrıca çaprazlama ile değerli genetik özellikler kaybolabilir mutasyon bu kaybedilen özelliklerin geri kazanılmasını sağlayabilir. Bu şekilde global optimuma ulaşılması olasılığını artırır. Mutasyon işlemi kromozomun rastgele seçilen genlerinin değiştirilmesi ile yapılır (Artaç, 2003).

Seçkinlik (Elitizm): Popülasyondaki en iyi bireyin çaprazlama ve mutasyon sonucunda kaybolma olasılığı vardır. Bunun önüne geçmek için topluluktaki en iyi birey sonraki jenerasyona aktarılır. Böylece en iyi bireyin korunması ve diğer kuşağa aktarılması sağlanır. Bir sonraki kuşaktaki popülasyonun bir öncekinden kötü olma olasılığı da ortadan kalkmış olur (Kahraman ve Özdağlar, 2004).

Uygunluk Değeri: Uygunluk değeri bireyin çözüm için uygunluğunun ölçütüdür. Her problem için bir uygunluk fonksiyonu mevcuttur. Bu fonksiyon ile popülasyondaki bireylerin uygunluk değerleri bulunur. Uygunluk değeri her jenerasyonda daha iyi çözümler üretilmesi için yol gösterir. Olası çözümler kümesinde yani popülasyonda bireyin uygunluk değeri yüksekse yaşama ve diğer jenerasyona aktarılma olasılığı artar (Yeniay, 2001). Uygunluk fonksiyonu probleme özgüdür. Uygunluk değeri iyi olan çözümler bir sonraki nesle aktarılmak için seçilmektedirler.

3.2.2.2. Algoritma Parametreleri

Genetik algoritmada parametre değerleri problem türüne göre değişiklik gösterebilir. Parametre değerleri algoritmanın çalışma süresini ve başarımını etkilemektedir. Bundan dolayı uygun parametre değerlerinin seçilmesi gerekmektedir. Parametre değerlerini belirlemenin kesin yöntemleri bulunmamaktadır. Problem türüne göre farklılık gösterebildiğinden dolayı bu değerlerin test edilerek bulunması gerekebilir. Jenerasyon sayısı, birey sayısı, birey

seçim yöntemi, çaprazlama olasılığı, mutasyon olasılığı algoritma tasarlayıcısı tarafından belirlenir.

Jenerasyon Sayısı: Algoritmanın iterasyon sayısıdır. Belirlenen jenerasyon sayısı kadar algoritmanın adımları tekrarlanır. Genetik algoritmanın durma kriteri sayılabilir. Bu parametrenin düşük olması optimum çözüme ulaşmadan algoritmanın sonlanmasına, yüksek olması ise çözüm süresinin uzamasına neden olur.

Birey Sayısı: Popülasyondaki maksimum birey sayısını belirtir, popülasyon büyüklüğünü tanımlar. Birey sayısının fazla olması programın çalışma süresini ve yükünü artırır. Popülasyondaki birey sayısının gerektiğinden az sayıda belirlenmesi ise, çözüm uzayının kısıtlı bölümünün taranmasına ve çözüm kalitesinin düşük olmasına neden olabilmektedir.

Birey Seçim Yöntemi: Çaprazlama işlemine girecek bireylerin hangi yöntem ile seçileceğini belirler. Genetik algoritmada rulet tekeri, rastgele seçim ve turnuva yöntemi çaprazlama işleminde yaygın olarak kullanılan seçim yöntemlerindedir.

Çaprazlama Olasılığı: Yeni popülasyon oluşturulurken bireylerin çaprazlama işleminde girip girmeyeceğini belirleyen parametredir.

Mutasyon Olasılığı: Popülasyondaki bireylerin mutasyona uğrayıp uğramayacağını belirleyen olasılıktır.

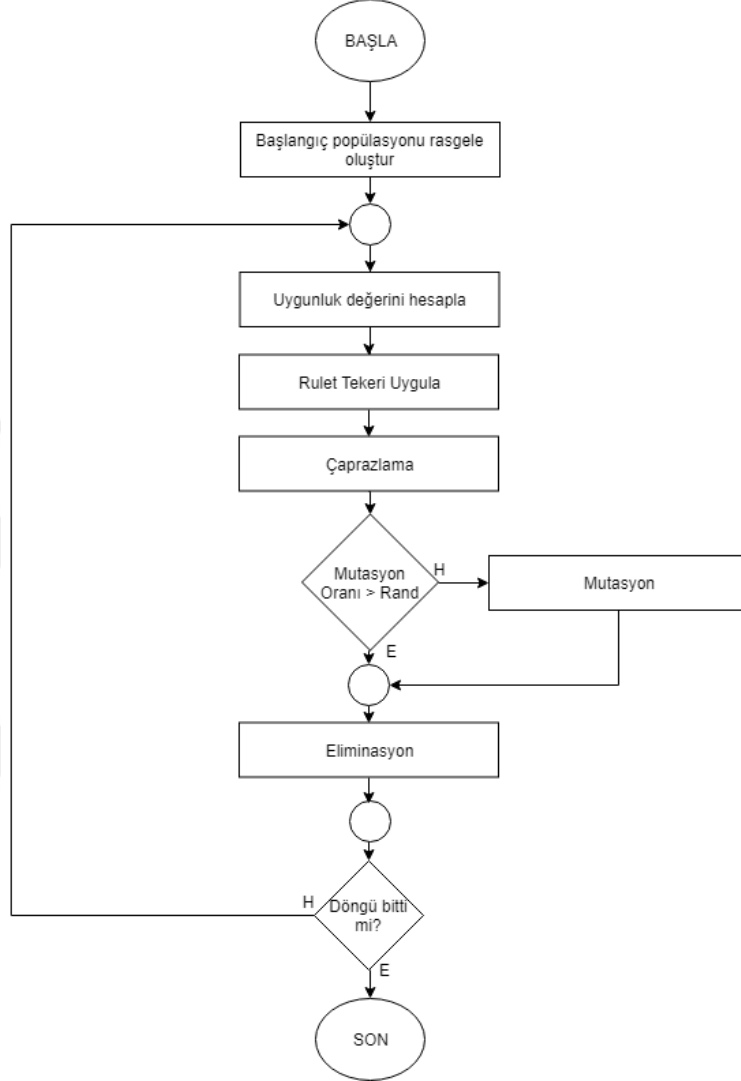
3.2.2.3. Algoritma Adımları

Farklı varyantlara sahip olsa da genetik algoritmanın genel adımları aşağıda sunulmuştur.

1. Başlangıç popülasyonunu oluştur.
2. Uygunluk değerlerini hesapla.
3. Rulet tekeri uygula

4. Çaprazlama yap.
5. Mutasyon oranı $>$ Unif(0,1) ise mutasyon uygula
6. Eliminasyon yap. (Popülasyonu uygunluk değerine göre sırala ve maksimum popülasyon sayısı kadar bireyi sakla, geri kalan bireyleri yok et)
7. Durma kriteri sağlanana kadar 2. adımdan itibaren adımları tekrarla, durma kriteri sağlandığında dur.

Genetik algoritmayı probleme uygulamak için öncelikle genin taşıyacağı bilgi ve kromozom kodlama yöntemi belirlenir. Bu aşamadan sonra problemin uygunluk fonksiyonu belirlenir. Genetik algoritma rastgele üretilen ve popülasyon olarak adlandırılan olası çözümler ile işleme başlar. Daha sonra popülasyondaki bireylere çaprazlama uygulayarak daha iyi çözümler üretmek amaçlanır. Belirlenen mutasyon oranında bireylere mutasyon uygulanarak çeşitlilik sağlanır. Popülasyon büyüklüğünü kontrol altında tutmak için popülasyon büyüklüğü belirlenir. Her jenerasyonda popülasyon artacağı için bazı bireylerin yok edilmesi gerekir. Bireylerin yok edilmesi aşaması eliminasyondur. Bu aşamada iyi bireylerin korunumu ve diğer jenerasyonlara aktarımı önemlidir ve bu elitizm kavramını anlatır. Eliminasyon aşamasında uygunluğu yüksek bireyler korunur, düşük bireyler yok edilir. Belirlenen jenerasyon sayısınca bireylere çaprazlama işlemi, belirlenen oranda mutasyon işlemi ve eliminasyon işlemi uygulanır. Bu şekilde başlangıçtaki popülasyon kademeli olarak iyileştirilir. Algoritma sonucunda popülasyonda hayatta kalan en iyi birey problemin çözümüdür. Şekil 3.4'te genetik algoritmanın akış diyagramı verilmiştir.



Şekil 3.4. Genetik Algoritmanın Akış Diyagramı

3.2.2.4. Algoritmanın Probleme Uyarlanması

Genetik algoritma KKARP'ine uyarlanırken permütasyon kodlama kullanılmıştır. Permütasyon kodlamada, ilgili değerler için oluşturulan belirli bir sıralamada, değerlerin konumlarını ifade eden sayılar dizi şeklinde kromozomu

oluştururlar. Permütasyon kodlama, genelde iş çizelgeleme, GSP gibi sıralamanın önemli olduğu problemlerde kullanılır.

KKARP'nin genetik algoritma ile çözümünde müşterilerin numaraları gen olarak ifade edilmiştir. Kromozom rotalardan oluşan müşteri listesi şeklinde ifade edilmiştir. Kromozom KKARP'nin olası çözümlerinden biridir. Bir kromozom üzerinde rotaları barındırır ve rotalar müşteri numaralarından oluşan listedir. Listenin indeksi müşterinin sırasını belirler. Açıklamalarda birey ve kromozom eş anlamlı olarak kullanılmıştır. Şekil 3.5'te 10 müşterili problem için örnek gen ve kromozom gösterimi sunulmuştur. Şekil 3.5'te her birim geni, genlerin tamamı da kromozomu ifade etmektedir. Ayrıca bu gösterim olası çözümlerden biridir.

3	2	4	7	9	6	8	1	10	5
---	---	---	---	---	---	---	---	----	---

Şekil 3.5. Gen ve Kromozom Gösterimi

Müşteriler rastgele sıralanarak bir birey elde edilmiş olur. Belirlenen birey sayısı parametresi kadar bu işlem yapılır. Bu işlem sonucunda oluşan bireyler popülasyona atılır. Böylece başlangıç popülasyonu oluşturulmuş olur. Şekil 3.6'da 10 müşterili problem için genetik algoritmanın; gen, kromozom, popülasyon ve uygunluk değeri gibi bileşenleri gösterilmiştir.

											Uygunluk Değeri
Gen	1	4	6	2	3	8	9	5	10	7	125
Kromozom (Birey)	1	10	8	2	5	4	9	3	6	7	132
	6	1	9	7	3	2	10	4	8	5	160
	1	9	7	10	8	6	4	3	2	5	198
Popülasyon											

Şekil 3.6. Genetik Algoritma Bileşenleri (Gen, Kromozom, Popülasyon ve Uygunluk Değeri)

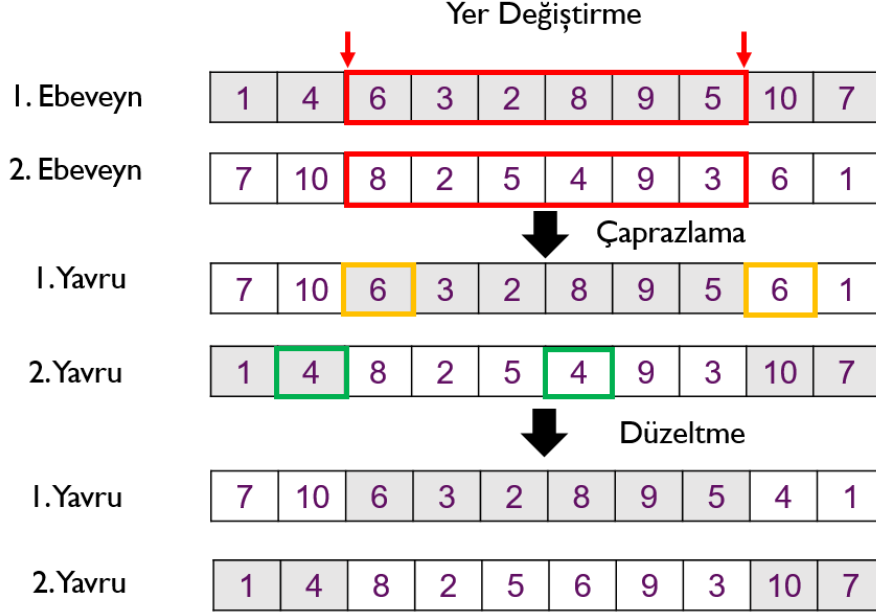
Popülasyondaki kromozomların uygunluk değeri rotaların toplam maliyeti olarak belirlenmiştir. Kromozomlar rotalardan oluşan müşteri listesidir ve kapasite kısıtına uygun rotalara bölünmesi için bir ayırma işlemi gereklidir. Uygunluk değerini hesaplamak için kromozom üzerindeki genler yani müşteriler sırasıyla alınır. Müşteri talebi araç kapasitesini aşmıyorsa rotaya eklenir. Aracın yükü güncellenir. Aracın kapasitesi doluncaya kadar kromozomdaki genler yani müşterilerle rota oluşturmaya devam edilir. Talebin miktarı araç kapasitesini aşarsa yeni rota oluşturulur. Kromozomdaki tüm genler tamamlanıncaya kadar bu işlem tekrar eder. Kromozomdaki tüm genler rotalara atandıktan sonra rota ayırma işlemi gerçekleştirilmiş olur. Rotaların başlangıcına ve sonuna depolar eklenir ve bu rotaların maliyeti hesaplanır. Bu işlem sonucunda popülasyondaki kromozomların (bireylerin) uygunluk değerleri hesaplanmış olur. Rotaların toplam mesafesi uygunluk değeri olarak belirlenmiştir. Bir kromozom için yapılan bu işlem tüm popülasyona uygulanır.

Çaprazlama işleminde; popülasyonun bireyleri sırayla alınır ve bu bireye rastgele eş seçilir. Bu eş seçim yöntemine rastgele seçim denir. Rasgele seçim popülasyon içerisinde, hiçbir kısıt ve ya kural olmadan, kromozom seçimi ile

gerçekleştirilen yöntemdir. Bu yöntemde, kromozomları seçmek için belirli aralıklarda verilen uniform bir sayı üreticisi kullanılmaktadır. Bu aşamada sıra ile alınan bir kromozom ve rastgele seçilen diğer kromozom çaprazlama işlemi için hazırdır.

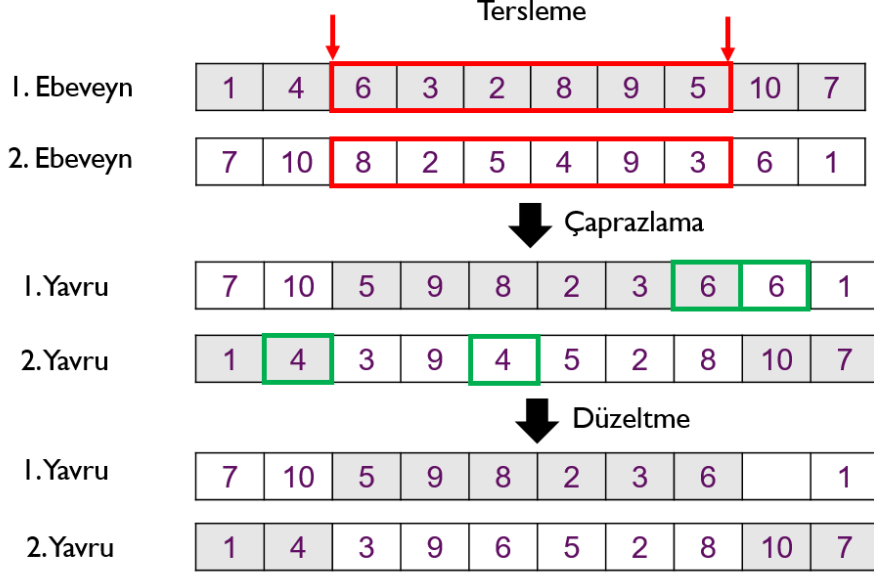
Bu çalışmada yer değiştirme, ters çevirme ve ekleme olmak üzere 3 farklı çaprazlama yöntemi kullanılmıştır. Çaprazlama sonucunda iki yavru oluşacaktır. Çaprazlama sonrasında yavru kromozomlarda tekrar eden genler bulunabilir. Tekrar eden genler aynı müşteriye birden fazla kez uğramak anlamında geldiği için çözüm olarak kabul edilemez ve düzeltilmesi gerekir. Düzeltme işleminde kromozom üzerinde tekrar eden genler belirlenir, tekrar eden genlerden biri bırakılır diğerlerinin yerine kromozomda kullanılmamış olan genlerden rastgele seçim yapılarak düzeltme işlemi uygulanır. Düzeltme sonrasında yavru bireyler tamamıyla oluşturulmuş olur. Yer değiştirme, tersleme ve ekleme yöntemi ile çaprazlamanın detayları aşağıda anlatılmıştır.

Yer değiştirme yöntemi ile çaprazlamada, rastgele iki nokta seçilir. Bu iki noktanın aynı olmaması sağlanır. Birinci ebeveynin iki nokta arasında kalan genleri alınır. Seçilen alt dizi ikinci ebeveynin aynı noktalarına denk gelen bölümüne sıra bozulmadan yerleştirilir ve birinci yavru oluşturulmuş olur. Sonra ikinci ebeveynin iki nokta arasında kalan genleri alınır. Seçilen alt dizi birinci ebeveynin aynı noktalarına denk gelen bölümüne sıra bozulmadan yerleştirilir ve ikinci yavru oluşturulmuş olur. Bu çaprazlama sonrasında yavru kromozomlarda tekrar eden genler bulunabilir. Bundan dolayı düzeltme işlemi uygulanır ve yavru kromozomlar popülasyona eklenir. Şekil 3.7’de yer değiştirme ile çaprazlama gösterilmiştir.



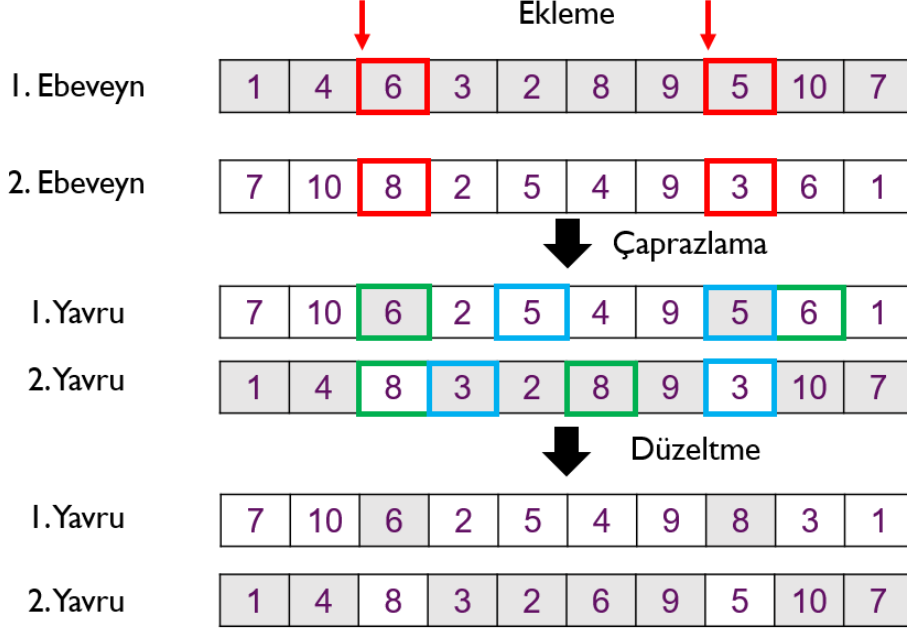
Şekil 3.7. Yer Değişirme ile Çaprazlama

Tersleme yöntemi ile çaprazlamada, rasgele iki nokta seçilir. Bu iki noktanın aynı olmaması sağlanır. Birinci ebeveynin iki nokta arasında kalan genleri alınır. Seçilen alt dizi ikinci ebeveynin aynı noktalarına denk gelen bölümüne ters çevrilerek yerleştirilir ve birinci yavru oluşturulmuş olur. Sonra ikinci ebeveynin iki nokta arasında kalan genleri alınır. Seçilen alt dizi birinci ebeveynin aynı noktalarına denk gelen bölümüne ters çevrilerek yerleştirilir ve ikinci yavru oluşturulmuş olur. Bu çaprazlama sonrasında yavru kromozomlarda tekrar eden genler bulunabilir. Bundan dolayı düzeltme işlemi uygulanır ve yavru kromozomlar popülasyona eklenir. Şekil 3.8’de tersleme ile çaprazlama gösterilmiştir.



Şekil 3.8. Tersleme ile Çaprazlama

Ekleme yöntemi ile çaprazlamada, rasgele iki nokta seçilir. Bu iki noktanın aynı olmaması sağlanır. Birinci ebeveyninden seçilen iki noktaya denk gelen genler alınır ve ikinci ebeveynin aynı noktalarına denk gelen bölümüne yerleştirilir. Böylelikle birinci yavru oluşturulmuş olur. Sonra ikinci ebeveyninden seçilen iki noktaya denk gelen genler alınır, birinci ebeveynin aynı noktalarına denk gelen bölümüne yerleştirilir ve ikinci yavru oluşturulmuş olur. Bu çaprazlama sonrasında yavru kromozomlarda tekrar eden genler bulunabilir. Bundan dolayı düzeltme işlemi uygulanır ve yavru kromozomlar popülasyona eklenir. Şekil 3.9’da tersleme ile çaprazlama gösterilmiştir.

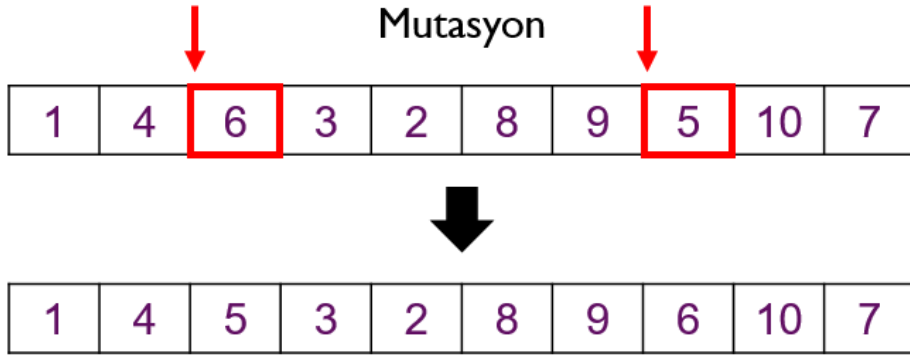


Şekil 3.9. Ekleme ile Çaprazlama

Çaprazlama yöntemlerinin her birine aralık olarak değer atanmıştır ve bu aralığın toplamı 1'dir. Çaprazlama yönteminin seçimi için 0-1 aralığında rastgele sayı üretilir. Üretilen sayının denk geldiği aralıktaki çaprazlama yöntemi seçilir. Örneğin yer değiştirme ile çaprazlamanın aralığı 0-0.6, tersleme ile çaprazlamanın aralığı 0.6-0.8, ekleme ile çaprazlama aralığı 0.8-1 olsun. Bu durumda üretilen sayı 0-0.6 aralığında bulunursa yer değiştirme, 0.6-0.8 aralığında bulunursa tersleme ve 0.8-1 aralığında bulunursa ekleme çaprazlama yöntemi seçilecektir.

Rastgele bir sayı üretilir ve bu sayıya karşılık gelen çaprazlama yöntemi tercih edilir. Yukarıda detayları verilen yer değiştirme, tersleme ve ekleme ile çaprazlamadan biri gerçekleştirilir. Çaprazlama sonrası oluşan iki yavru popülasyona dahil edilir. Bu işlem tüm popülasyon bireyelerine uygulanıncaya kadar tekrar eder.

Mutasyon aşamasında rastgele sayı üretilir. Bu sayı belirlenmiş olan mutasyon oranından küçükse mutasyon gerçekleştirilir, değilse mutasyon gerçekleştirilmez. Mutasyon işlemi gerçekleştirilirken rastgele kromozom seçilir ve bu kromozomdan rastgele 2 gen seçilir. Bu genlerin aynı olmaması sağlanır. Seçim işleminden sonra genler yer değiştirilir ve mutasyon işlemi tamamlanmış olur. Şekil 3.10'da mutasyon işlemi gösterilmiştir.



Şekil 3.10. Mutasyon İşlemi

Eliminasyon aşamasında popülasyon, uygunluk değerine göre sıralanır. Maksimum popülasyon sayısını aşan birey sayısı kadar; düşük uygunluk değerine sahip bireyler yok edilir.

Uygunluk değeri hesaplama, rulet tekeri uygulaması ile çaprazlama yöntemi seçimi ve çaprazlama, mutasyon, eliminasyon işlemleri; durma kriteri sağlanıncaya kadar tekrar uygulanır. Problemin amacı rotaların toplam mesafesini minimize etmek olduğu için algoritmanın durma kriteri sağlandığında hayatta kalan bireylerden uygunluk değeri en düşük olan çözüm en iyi çözüm olarak değerlendirilir. Bu çözüm optimal veya optimale yakın çözümdür.

3.2.3. Yabani Ot Algoritması

Yabani ot algoritması Mehrabian ve Lucas tarafından 2006 yılında yapılan çalışma ile ortaya konulmuş metasezgisel bir algoritmadır. Bu algoritma, doğadaki yabani otların zor şartlara rağmen ayakta kalıp soyunu güçlendirerek devam etmesinden esinlenerek geliştirilmiştir. Yabani ot ekolojisinde iyilerin daima ayakta kaldığı mekanizma vardır. Yabani otların bu özelliği yabani ot algoritmasının temelini oluşturur (Urul ve Cengiz, 2014).

YOA, yabani otların sağlamlık, adaptasyon ve rastgelelik gibi özelliklerini simüle eden basit ama etkili bir optimizasyon algoritmasıdır. Bu algoritma, bir kolonideki yabani ot karakteristiklerini (tohum üretme, büyüme ve özel rekabet gibi) uygulayarak etkili bir şekilde global optimuma yaklaşır (Mehrabian ve Lucas, 2006).

Yabani ot algoritması; doğadaki istilacı yabani otların zor şartlara rağmen ayakta kalmasından ve türünü devam ettirmesinden esinlenilerek geliştirilmiştir. Algoritma temel olarak, iyi yabani otların daha fazla tohum üretirken kötü yabani otların daha az tohum (yavru) üretmesine dayanır. Yani iyi yabani otların aşamalı bir şekilde daha iyi olmasını ve koloniye hakim olmasını sağlayan mekanizma vardır. Bu açıdan bakıldığında genetik algoritmayla benzerlik göstermektedir. Yabani ot algoritmasının; iyi çözümlere daha fazla yavru üretme imkanı tanınması, iyi çözümlerin hızlı bir şekilde daha iyi çözümlere ulaşılacağını düşündürmüştür. Yabani ot algoritmasında genetik algoritmadan daha az aşama olması da algoritmanın daha kolay uygulanabileceğini ve daha hızlı çalışabileceğini düşündürmüştür.

3.2.3.1. Algoritmaya Genel Bakış

Yabani ot algoritması ile ilgili bazı önemli kavramlar; tohum, yabani ot, koloni, jenerasyon ve uygunluk değeridir.

Tohum: Yabani ot algoritmasında yabani ottan üretilen yavruya tohum denir. Tohum da yabani ot gibi problemin olası çözümünden biridir. Çoğalma aşamasında oluşacak yavruları belirtmek için kullanılan terimdir.

Yabani Ot (Birey): Her bir yabani ot problem için bir çözüm adayıdır.

Koloni: Yabani otlardan meydana gelen topluluğu tanımlamak için kullanılmıştır.

Jenerasyon: Yeni koloniyi ifade eder.

Uygunluk Değeri: Uygunluk değeri yabani otun çözüm için uygunluğunun ölçütüdür. Her problem için bir uygunluk fonksiyonu mevcuttur. Bu fonksiyon ile kolonideki yabani otların uygunluk değerleri bulunur. Uygunluk değeri her jenerasyonda daha iyi çözümler üretilmesi için yol gösterir. Olası çözümler kümesinde yani kolonide yabani otun uygunluk değeri yüksekse daha fazla tohum üretme şansına sahiptir. Bu şekilde yaşama ve diğer jenerasyona aktarılma olasılığı artar.

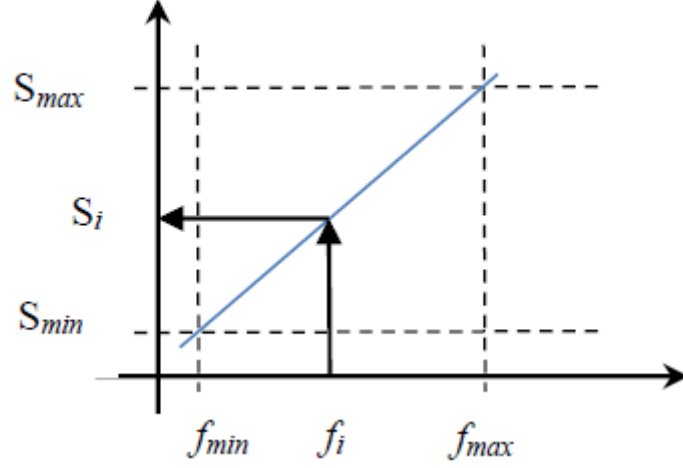
Yabani ot algoritmasının, başlatma, çoğalma süreci, rastgele dağıtım ve rekabetçi eleme olmak üzere 4 temel adımı bulunur. Her bir adım aşağıda açıklanmıştır.

3.2.3.1.(1).Başlatma (Initialization)

Belirli sayıda yabani ot çözüm uzayının her tarafına rastgele dağıtılarak başlangıç kolonisi oluşturulur.

3.2.3.1.(2).Çoğalma Süreci (Reproduction)

Bu aşamada kolonideki yabani otlardan kaç tane tohum oluşturulacağı belirlenir. Daha iyi olan yabani ot, daha fazla tohum üretirken, daha kötü olan yabani ot, daha az tohum üretir. Şekil 3.11’de uygunluk değeri ve tohum sayısı arasındaki ilişkinin grafiği verilmiştir.



Şekil 3.11. Uygunluk Değeri ve Tohum Sayısı Arasındaki İlişki

Yeniden çoğalma formülü Denklem 1’de verilmiştir:

$$S_i = \left[\frac{f_i - f_{min}}{f_{max} - f_{min}} (S_{max} - S_{min}) \right] + S_{min} \quad (3.1)$$

Burada $\lfloor \rfloor$ aşağı yuvarlamayı ifade etmektedir. f_i , mevcut yabancı otun uygunluk değeridir. f_{max} ve f_{min} , koloninin maksimum ve minimum uygunluk (fitness) değerlerini temsil etmektedir. S_{max} ve S_{min} sırasıyla bir yabancı otun üretebileceği maksimum ve minimum tohum sayısını ifade eder. S_i , yabancı otun üreteceği tohum sayısını ifade etmektedir.

3.2.3.1.(3).Rastgele Dağıtım (Spatial Dispersal)

Bu aşamada, belirlenen sayıdaki tohumlar rastgele olarak problem uzayına dağıtılır ve ortalamaya göre sifıra eşit ve değişken varyantına dayalı olarak, üretilen bu tohumlar ana yabancı ot yakınında bulunurlar. Burada, fonksiyonun standart sapması (σ) önceden belirlenmiş bir başlangıç değerinden ($\sigma_{initial}$) başlayıp, son bir

değere (σ_{final}) kadar iterasyon boyunca azaltılacaktır. σ_{iter} hesaplama yöntemi Denklem 2’de verilmiştir (Koç vd., 2018).

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{(iter_{max})^n} (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \quad (3.2)$$

Burada, $iter_{max}$ maksimum jenerasyon (iterasyon) sayısıdır, σ_{iter} belirli andaki iterasyona ait standart sapma ve n varyans azaltma bileşenidir.

3.2.3.1.(4).Rekabetçi Eleme (Competitive Exclusion)

Hayatta kalmak için bitkiler arasında bir rekabet vardır. Kolonideki yabancı otlar çok hızlı çoğalır ve tüm yabancı otlar koloni olarak kabul edilir. Fakat kolonideki toplam yabancı ot sayısı koloninin maksimum yabancı ot değerini aşmamalıdır. Bu nedenle, daha fazla uygunluk gösteren yabancı otlar koloni içine dâhil edilirken, daha az uygunluk gösteren yabancı otlar koloniden çıkarılır (Koç vd., 2018). Bu şekilde zayıfın elenip kuvvetlilerin ayakta kaldığı mekanizma elde edilir.

Bu işlemler belirlenen iterasyon boyunca yani durma kriteri sağlanıncaya kadar devam eder. Bunun sonucunda, hayatta kalan en iyi yabancı ot, problemin aranan en iyi çözümü olarak değerlendirilir (Urul ve Cengiz, 2014).

3.2.3.2.Algoritma Parametreleri

Yabancı ot algoritmasında parametre değerleri problem türüne göre değişiklik gösterebilir. Parametre değerleri algoritmanın çalışma süresini ve başarımını etkilemektedir. Bundan dolayı uygun parametre değerlerinin seçilmesi gerekmektedir. Parametre değerlerini belirlemenin kesin yöntemleri bulunmamaktadır. Problem türüne göre farklılık gösterebildiğinden dolayı bu değerlerin test edilerek bulunması gerekebilir. Jenerasyon sayısı, başlangıç yabancı

ot sayısı, maksimum yabancı ot sayısı, minimum tohum sayısı, maksimum tohum sayısı, varyans azaltma bileşeni, standart sapma başlangıç değeri, standart sapma son değeri algoritma tasarlayıcısı tarafından belirlenir.

Jenerasyon Sayısı: Algoritmanın iterasyon sayısıdır. Belirlenen jenerasyon sayısı kadar algoritmanın adımları tekrarlanır. Algoritmanın durma kriteri sayılabilir. Bu parametrenin düşük olması optimum çözüme ulaşmadan algoritmanın sonlanmasına, yüksek olması ise çözüm süresinin uzamasına neden olur.

Başlangıç Yabancı Ot Sayısı: Algoritmanın başlangıç aşamasında üretilen yabancı ot sayısını belirtir.

Maksimum Yabancı Ot Sayısı: Kolonideki maksimum yabancı ot sayısını belirtir, koloni büyüklüğünü tanımlar. Yabancı ot sayısının fazla olması programın çalışma süresini ve yükünü artırır. Kolonideki yabancı ot sayısının gerektiğinden az sayıda belirlenmesi ise, çözüm uzayının kısıtlı bölümünün taranmasına ve çözüm kalitesinin düşük olmasına neden olabilmektedir.

Minimum Tohum Sayısı (S_{min}): Kolonideki yabancı otların üretebileceği minimum tohum sayısını belirtir. Koloninin en kötü yabancı otu, minimum tohum sayısı kadar tohum üretebilir.

Maksimum Tohum Sayısı (S_{max}): Kolonideki yabancı otların üretebileceği maksimum tohum sayısını belirtir. Koloninin en iyi yabancı otu, maksimum tohum sayısı kadar tohum üretebilir.

Varyans Azaltma Bileşeni (n): Rastgele dağıtım aşamasında kullanılan, varyansın iterasyon boyunca azalmasını sağlayan parametredir.

Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): Rastgele dağıtım aşamasında kullanılan standart sapmanın başlangıç değeridir.

Standart Sapma Son Değeri (σ_{final}): Rastgele dağıtım aşamasında kullanılan standart sapmanın son değeridir.

3.2.3.3. Algoritma Adımları

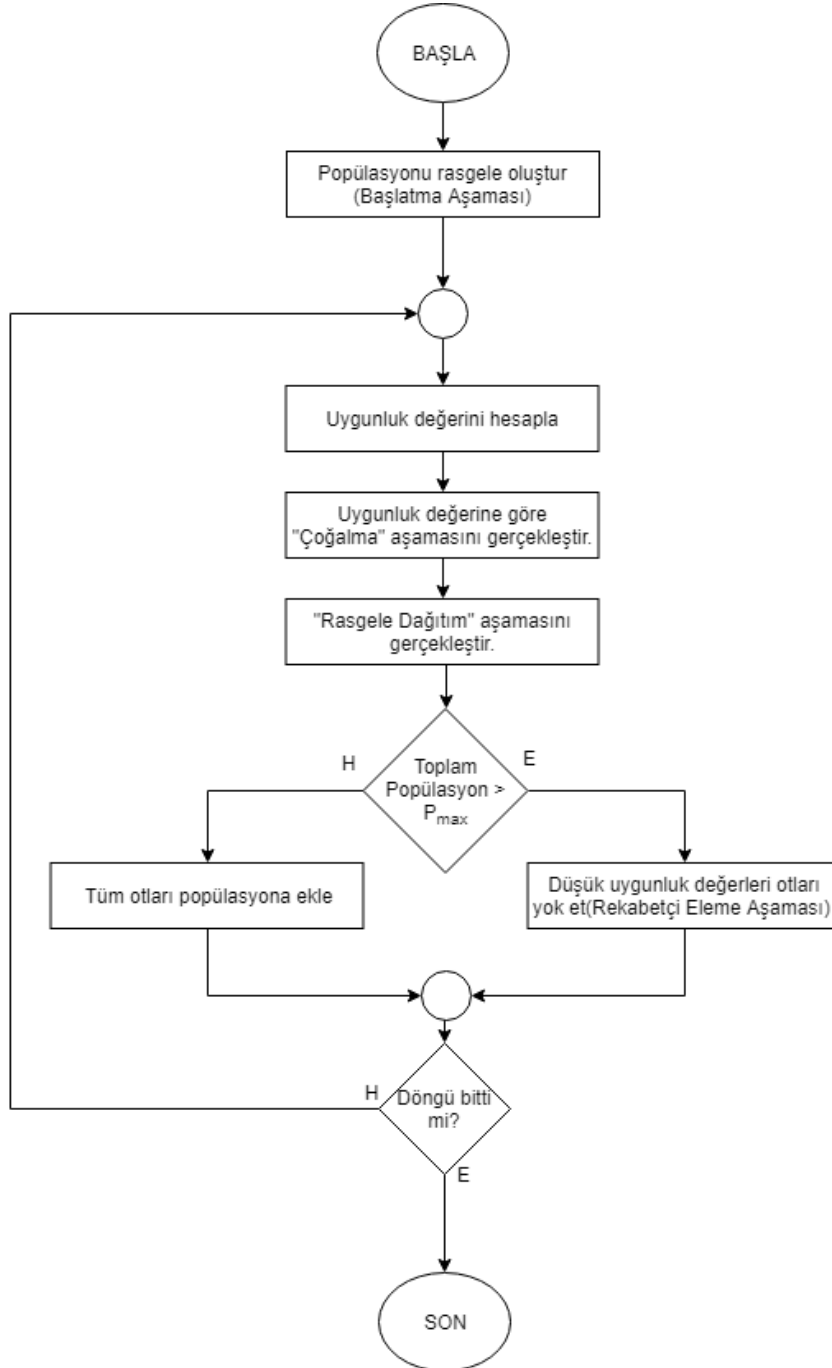
Yabani ot algoritmasının adımları aşağıda verilmiştir;

1. Rastgele başlangıç popülasyonunu oluştur.
2. Uygunluk değerlerini hesapla.
3. Popülasyondaki bireylerin uygunluğuna göre üretilecek yavru sayılarını belirleyerek çoğalma aşamasını gerçekleştir (Popülasyonun bireyelerinden belirlenen sayılarda yavru üret).
4. Rastgele dağıtım aşamasını gerçekleştir.
5. Popülasyon büyüklüğü kadar en iyi bireyleri koru diğerlerini yok et.
6. Durma kriteri sağlanana kadar 2. adımdan itibaren adımları tekrarla, durma kriteri sağlandığında dur.

Yabani ot algoritmasını probleme uygulamak için öncelikle yabani otun taşıyacağı bilgi yani kodlama belirlenir. Bu aşamadan sonra problemin uygunluk fonksiyonu ortaya konur. Yabani ot algoritması rasgele üretilen ve koloni olarak adlandırılan olası çözümler ile işleme başlar. Daha sonra otların üretebileceği tohum sayıları belirlenerek çoğalma aşaması gerçekleştirilir. Bu aşamada otlar uygunluk değeriyle doğru orantılı olacak sayıda tohum üretir. Rasgele dağıtım aşamasında ata yabani ottan uygunluk değerine göre belirlenecek sayıda tohumlar üretilir. Bu şekilde çözüm uzayındaki çözümlerin yakınındaki çözümler aranmış olur. Koloni büyüklüğünü kontrol altına almak için maksimum yabani ot sayısı belirlenir. Her jenerasyonda yabani ot sayısı artacağı için düşük uygunluk gösteren bazı yabani otların yok edilmesi gerekir. Yabani otların yok edilmesi, rekabetçi eleme aşamasında gerçekleştirilir. Bu aşamada iyi otların korunması ve diğer jenerasyona aktarılması önemlidir. Belirlenen jenerasyon sayısınca çoğalma aşaması, rasgele dağıtım ve rekabetçi eleme gerçekleştirilir. Bu şekilde

başlangıçtaki koloni kademeli olarak iyileştirilir. Algoritma sonucunda kolonide hayatta kalan en iyi yabancı ot problemin çözüdür. Şekil 3.12’de algoritmanın akış diyagramı verilmiştir.





Şekil 3.12. Yabani Ot Algoritmasının Akış Diyagramı

3.2.3.4. Algoritmanın Probleme Uyarlanması

Yabani ot algoritması KKARP'ine uyarlanırken genetik algoritmada kullanılan permütasyon kodlamaya benzer kodlama kullanılmıştır.

KKARP'nin yabani ot algoritması ile çözümünde yabani ot, rotalardan oluşan müşteri listesi şeklinde ifade edilmiştir. Yabani ot KKARP'nin olası çözümlerinden biridir. Bir yabani ot üzerinde rotaları barındırır ve rotalar müşteri numaralarından oluşan listedir. Listenin indeksi müşterinin sırasını belirler. Şekil 3.13'te 10 müşterili problem için örnek yabani ot gösterimi sunulmuştur. Şekil 3.13'te her birim müşterinin numarasını, birimlerin tamamı da yabani otu ifade etmektedir. Ayrıca bu gösterim olası çözümlerden biridir.

3	2	4	7	9	6	8	1	10	5
---	---	---	---	---	---	---	---	----	---

Şekil 3.13. Yabani Ot Gösterimi

Müşteriler rastgele sıralanarak bir yabani ot elde edilmiş olur. Başlangıç yabani ot sayısı kadar bu işlem yapılır. Bu işlem sonucunda oluşan yabani otlar koloniye eklenir. Böylece başlangıç kolonisi oluşturulmuş olur. Şekil 3.14'te 10 müşterili problem için yabani ot algoritmasının; yabani ot, koloni ve uygunluk değeri gibi bileşenleri gösterilmiştir.

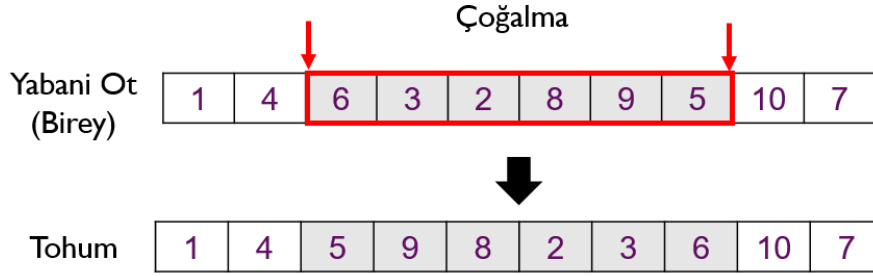
Müşteri Numarası	1	4	6	2	3	8	9	5	10	7	Uygunluk Değeri	125
Yabani Ot (Birey)	1	10	8	2	5	4	9	3	6	7		132
	6	1	9	7	3	2	10	4	8	5		160
Koloni	1	9	7	10	8	6	4	3	2	5		198

Şekil 3.14. Yabani Ot Algoritması Bileşenleri (Yabani Ot, Koloni, Uygunluk Değeri)

Kolonideki yabancı otların uygunluk değeri rotaların toplam maliyeti olarak belirlenmiştir. Yabancı otlar rotalardan oluşan müşteri listesidir ve kapasite kısıtına uygun rotalara bölünmesi için bir ayırma işlemi gereklidir. Uygunluk değerini hesaplamak için yabancı ot üzerindeki müşteriler sırasıyla alınır. Müşteri talebi araç kapasitesini aşmıyorsa rotaya eklenir. Aracın yükü güncellenir. Aracın kapasitesi doluncaya kadar yabancı ottaki müşterilerle rota oluşturmaya devam edilir. Talebin miktarı araç kapasitesini aşarsa yeni rota oluşturulur. Yabancı ottaki tüm müşteriler rotalara atandıktan sonra rotalar oluşturulmuş olur. Rotaların başlangıcına ve sonuna depolar eklenir ve bu rotaların maliyeti hesaplanır. Bu işlem sonucunda kolonideki yabancı otların (olası çözümlerin) uygunluk değerleri hesaplanmış olur. Rotaların toplam mesafesi uygunluk değeri olarak belirlenmiştir. Bir yabancı ot için yapılan bu işlem tüm koloniye uygulanır.

Çoğalma aşamasında kolonideki her yabancı otun uygunluk derecesine göre üretebileceği tohum sayısı belirlenir. Yani iyi otlardan çok tohum oluşturulurken kötü otlardan az tohum oluşturulur. Koloninin en iyi yabancı otu maksimum tohum sayısı parametresi kadar, en kötü yabancı otu minimum tohum sayısı parametresi kadar tohum üretir. Yabancı otlar sırayla alınarak üreteceği tohumların

oluşturulması sağlanır. Tohumlar oluşturulurken rastgele dağıtım aşamasına uygun olarak oluşması istenir. Yani yabancı otun çevresindeki çözümlere ulaşabilmek için rastgele dağıtım kullanılır. Bu işlem ilgili yabancı otun iki noktası seçilerek ve terslenerek sağlanır. Seçilen iki nokta rasgele dağıtım formülü yardımı ile bulunur. Böylelikle ata yabancı otun çevresindeki çözümlere ulaşılmış olur. Şekil 3.15’de çoğalma aşamasında bir yabancı ottan bir tohum üretilmesi gösterilmiştir.



Şekil 3.15. Çoğalma Aşaması

Rekabetçi eleme aşamasında koloni, uygunluk değerine göre sıralanır. Maksimum yabancı ot sayısını aşan yabancı ot sayısı kadar; düşük uygunluk değerine sahip yabancı otlar yok edilir.

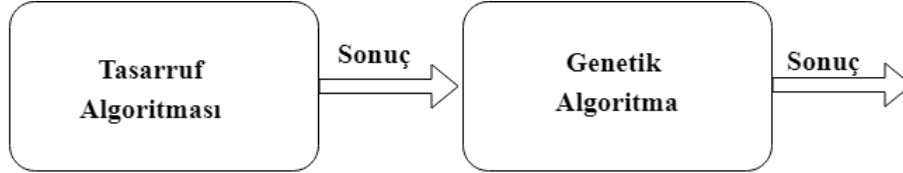
Uygunluk değeri hesaplama, çoğalma rastgele dağıtım ve rekabetçi eleme aşamaları durma kriteri sağlanıncaya kadar tekrar uygulanır. Problemin amacı rotaların toplam mesafesini minimize etmek olduğu için algoritmanın durma kriteri sağlandığında hayatta kalan yabancı otlardan uygunluk değeri en düşük olan çözüm en iyi çözüm olarak değerlendirilir. Bu çözüm optimal veya optimale yakın çözümdür.

3.2.4.Hibrit Yöntemler

Bu çalışma kapsamında ele alınan metasezgisel algoritmalar rastgele başlangıç çözümleri ile başlar ve aşamalı olarak bu çözümler iyileştirilmeye çalışılır. Durma kriteri sağlandığında optimale yakın çözüm üretilmiş olur. Bu açıdan bakıldığında algoritmanın başlangıç çözümleri iyileştikçe algoritmanın çözüm kalitesi de iyileşecektir. Buradan hareketle bu çalışmada ele alınan tasarruf algoritması, genetik algoritma ve yabani ot algoritması birbirini besleyecek şekilde çeşitli denemeler yapılmıştır. Bir algoritmanın ürettiği sonuç diğer algoritmanın başlangıç popülasyonuna eklenmiştir. Örneğin tasarruf algoritmasının bulduğu çözüm; genetik algortmada ve yabani ot algoritmasında başlangıç popülasyonuna eklenerek çalıştırılmıştır.

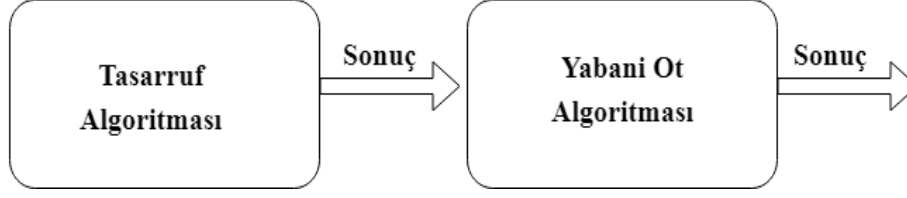
Hibrit algoritmalar olarak; tasarruf-genetik algoritması, tasarruf-yabani ot algoritması, tasarruf-genetik-yabani ot algoritması geliştirilmiştir. Algoritmalar isimlendirmede verilen sıralar ile çalıştırılmıştır.

Şekil 3.16'da tasarruf ve genetik algortmadan oluşan hibrit yaklaşımın şekilsel gösterimi verilmiştir.



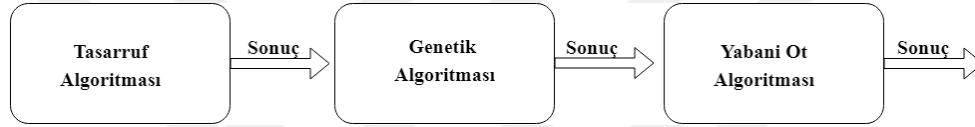
Şekil 3.16. Tasarruf ve Genetik Algoritma Hibrit Yaklaşımı

Şekil 3.17'de tasarruf ve yabani ot algoritmasından oluşan hibrit yaklaşımın şekilsel gösterimi verilmiştir.



Şekil 3.17. Tasarruf ve Yabani Ot Algoritması Hibrit Yaklaşımı

Şekil 3.18’de tasarruf, genetik ve yabani ot algoritmasından oluşan hibrit yaklaşımın şekilsel gösterimi verilmiştir. Bu hibrit yaklaşımda genetik algoritmanın ve yabani ot algoritmasının jenerasyon sayısı yarıya indirilmiştir.



Şekil 3.18. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Yaklaşımı

3.2.5. Dinamik Araç Rotalama

Dinamik araç rotalama problemi için literatürde yaygın olarak kullanılan P grubu veri setinden P-n50-k7 isimli problem kullanılmıştır. Bu problem tek depolu, 49 müşterili 7 araçlı kapasite kısıtlı araç rotalama problemidir. Problemi dinamikleştirmek için dağıtım süreci ilerlerken 3 farklı senaryo oluşturulmuştur. İlk olarak bazı müşterilerin taleplerini iptal etmesi durumu, ikinci olarak yeni müşteri taleplerinin gelmesi durumu, son olarak da aynı anda bazı müşterilerin taleplerini iptal etmesi ve sisteme yeni müşteri talepleri gelmesi durumları ele alınmıştır. Bu senaryolar gerçek hayat problemlerine yakın senaryolardır.

Problem dinamikleştirilirken bazı varsayımlarda bulunulmuştur. Bu problemde araçların çözüm olarak bulunan rotalara göre dağıtıma çıktığı varsayılmıştır. Yeni talep geldiğinde veya talep iptali olduğunda; araçların bulunduğu konum rota üzerindeki müşteri konumu veya depo olabileceği varsayılmıştır. Bu varsayım dinamik problem için üretilen çözümün kalitesini

kolay şekilde anlamak için yapılan bir varsayımdır. Dağıtıma çıkan araçların; rotasında bulunan müşteri talepleri kadar yükü bulunduğu varsayılmıştır.

Araç rotalama probleminin statik hali için geliştirilen hibrit yaklaşımlar genişletilerek dinamik probleme uygun hale getirilmiştir. Bu değişikliklerde algoritmanın çalışma adımları değiştirilmemiş olup sadece dinamik olan girdileri kabul edecek şekilde ayarlamalar yapılmıştır. Kurgulanan senaryolar; P-n50-k7 isimli probleme uygulanarak algoritmaya girdi olarak verilmiş ve problem çözülmüştür.

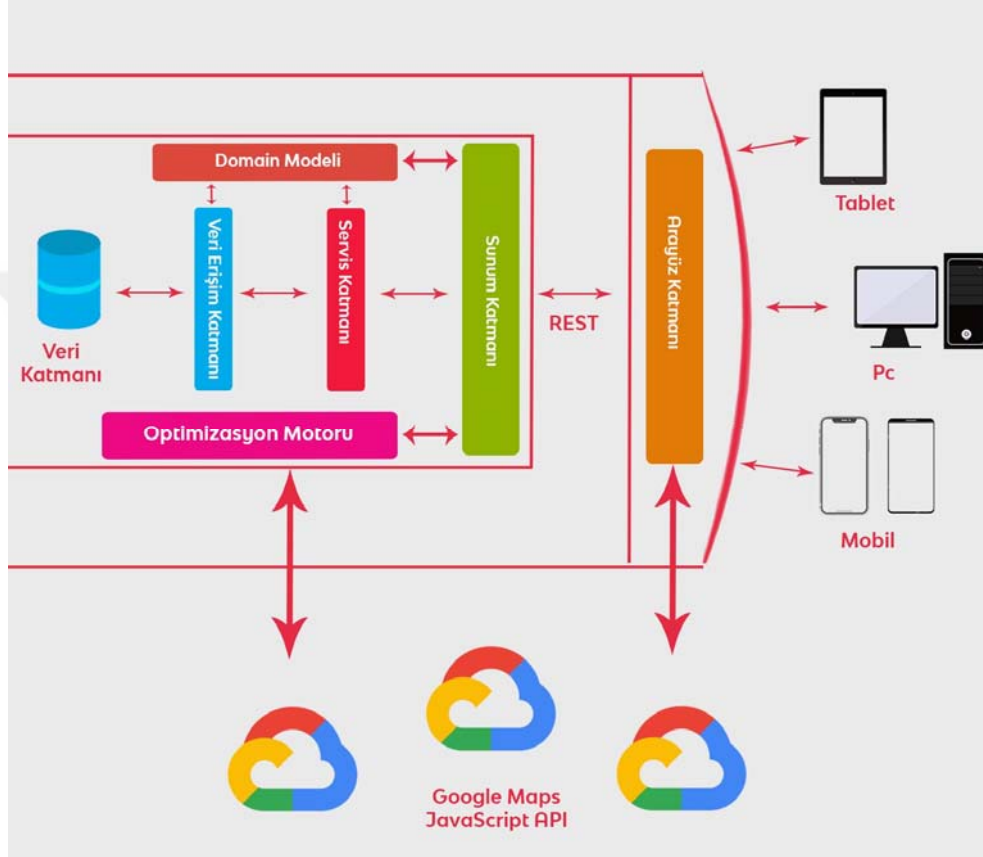
Sisteme yeni talepler geldiğinde veya var olan talepler iptal edildiğinde planlanmış olan rotalara devam edilmesi ile rota güncellemesi yapılması arasında kıyaslama yapılmıştır. Bu sonuçlara göre algoritmanın faydalı olup olmadığı tartışılmıştır.

3.2.6. Sistem Tasarımı

Bu bölümde kapasite kısıtlı araç rotalama probleminin çözümü için kullanılan algoritmaların kullanımını kolaylaştırmak amacıyla tasarlanan web tabanlı uygulamanın sistem mimarisi, kullanıcı arayüzleri ve veri tabanı hakkında bilgiler sunulmuştur.

Web tabanlı uygulamanın geliştirilmesinde çok katmanlı mimari kullanılmıştır. Uygulamanın dış sistemlerle ve kullanıcılarla etkileşimi arayüz katmanı tarafından ele alınarak, sunum katmanına iletilir. Sunum katmanı yapılan işlemi değerlendirerek servis katmanı veya optimizasyon motoruna yönlendirme yapar. Sunum katmanı tarafından optimizasyon motoruna yönlendirme yapılması durumunda, girdiler algoritmalara verilir. Algoritma sonucunda üretilen çıktılar sunum katmanına iletilir. Sunum katmanı tarafından servis katmanına yönlendirme yapılması durumunda, servis katmanı sınıfları aracılığıyla veri erişim katmanında gerekli işlemleri gerçekleştirir. İşlem sonucu veya oluşturulan bilgi, tersine sıralı

akışla arayüz katmanı üzerinden işlemi isteyen dış sistem veya kullanıcıya iletilir. Şekil 3.19’da sistem mimarisi verilmiştir.



Şekil 3.19. Sistem Mimarisi

Arayüz Katmanı: Web arayüzlerinin oluşturulması için JavaScript, HTML, CSS teknolojilerinden faydalanılmıştır. Ayrıca Google Maps Javascript API ile entegrasyon, arayüz katmanında yapılmıştır. Arayüz katmanında sorgulama, tanımlama, silme ve düzenleme işlevleri bulunan standart ekran modelleri ve optimizasyon için kullanılan ekran bulunmaktadır.

Sunum Katmanı: Uygulamanın arayüzlerle iletişim kurduğu katmandır. Bu katmanda REST servisler bulunmaktadır. Arayüz katmanından gelen istekleri karşılar. Servis katmanı veya optimizasyon motoruna yönlendirme yapar. Ayrıca gelen ve giden bilgileri taşımak için domain modelini kullanır.

Servis Katmanı: Uygulamanın iş mantığı, veri tabanına ait operasyonel işlemler bu katmanda yapılır. Süreçler içinde tanımlanan işlemlerin veri erişim katmanı ile iletişimini kullanarak veri tabanı üzerinde değişiklik yapılmasını sağlar.

Veri Erişim Katmanı: Bu katmanda sadece veri tabanı işlemleri yapılmaktadır. Bu katmanın görevi veriyi ekleme, silme, güncelleme ve veri tabanından çekme işlemidir. Bu katmanda bu işlemlerden başka herhangi bir işlem yapılmamaktadır. Veri erişim katmanında Hibernate kullanılarak HQL ile sorgular oluşturulmuştur. Bu sorgular veri tabanına Hibernate aracılığıyla yansıtılmış, dönen sonuçların bu katmanı çağıran katmana iletilerek veri saklama katmanının soyutlanması sağlanmıştır.

Veri Saklama Katmanı: Uygulamada yönetilen verilerin tutulduğu katmandır. Veri tabanı olarak MySQL veri tabanı yönetim sistemi kullanılmıştır. Veri Erişim katmanı ile etkileşim kurarak gelen sorguları çalıştırılması ve gereken verilerin ya da işlem sonucunun ilgili bağlantı üzerinden dönülmesini sağlamaktadır.

Domain Modeli: Veri tabanında tutulan tablo ve kolonların uygulama için yönetilebilir ve okunabilir halde olmasını sağlar. Domain modeli için Hibernate/JPA kullanılarak annotation tabanlı veri-domain dönüşümleri sağlanmıştır. Servis katmanı, sunum katmanı ve veri erişim katmanı bu ortak dili kullanarak iletişim kurabilmektedir. Domain sınıfları, veri erişim katmanının veri tabanı ile olan iletişimi sonucunda elde edilen verileri uygulamanın tamamında yönetilebilir şekilde taşımaktadır.

Optimizasyon Motoru: Algoritmaların bulunduğu katmandır. Optimizasyon motorunun görevi girdi olarak verilen rotalama problemini optimize

edip sonuçları kendisini çağıran katmana dönmektir. Optimizasyon motoru katmanına; tasarruf algoritması, genetik algoritma, yabani ot algoritması ve bu algoritmalarından oluşan hibrit yaklaşımlar eklenmiştir. Ayrıca optimizasyon motoru genişlemeye açıktır. Yani uygulamanın yapısı değiştirilmeden bu katmana uygun girdi çıktı yapısıyla başka bir algoritma kolayca entegre edilebilir.

Google Maps Javascript API: Geliştiricilere Google Haritalar'ı sayfalarına entegre etmek, Google Haritalar'dan veri almak gibi olanaklar sağlayan bir Google ürünüdür. Bu çalışma kapsamında; harita altyapısı olarak kullanılmıştır. Ayrıca adresler arasındaki araç yolu uzaklık bilgisi, süre bilgisi gibi bilgilerin elde edilmesi, adres-koordinat dönüşümleri, adres işaretleme, rota gösterme gibi işlemlerde de kullanılmıştır.

Literatürdeki çalışmalarda koordinatlar ile noktalar arası mesafelerin hesaplanması öklid kuralına göre yapılmaktadır. Bu yöntem ile tasarlanan uygulamalar gerçek hayat probleminde kullanıma uygun değildir. Gerçek hayat problemlerinde noktalar arası uzaklıkta, araç yollarının mesafesi temel alınmalıdır. Bu çalışma kapsamında bu sorunu aşmak için Google Maps API'leri kullanılmıştır. Her yeni adres kayıt işleminde yeni kaydedilen adres ile veri tabanındaki tüm adresler arasındaki uzaklık bilgisi, süre bilgisi gibi bilgiler Google Maps API'leri ile bulunur ve veri tabanına kaydedilir. Böylelikle gerçek verilerle problem çözümü sağlanabilir.

Bu çalışmada ele alınan algoritmalar Java programlama dili ile kodlanmıştır. Java programlama dili kullanıldığı için platform bağımsızdır, Linux veya Windows işletim sistemlerinde çalışabilir. Kullanıcının girdilerini tutmak amacıyla veri tabanı olarak MySQL seçilmiştir. Veri tabanı işlemleri için Hibernate kullanılmıştır. Hibernate teknolojisi ile farklı veri tabanları arasında geçiş kolaylıkla sağlanabilir. Uygulamaya Spring Boot çatısı eklenerek konfigürasyonun kolaylaştırılması ve bazı konularda kolaylık sağlanmıştır. Proje kütüphanelerini yönetmek için Maven kullanılmıştır. Uygulama REST mimarisi üzerine

kurulmuştur. Kullanıcı arayüzü JavaScript, HTML, CSS ve Google Maps JavaScript API kullanılarak web tabanlı olarak gerçekleştirilmiştir. Bu çalışma sonucunda ortaya çıkan uygulama web tabanlı, kullanıcı arayüzüne sahip gerçek hayattaki kullanıma uygundur.

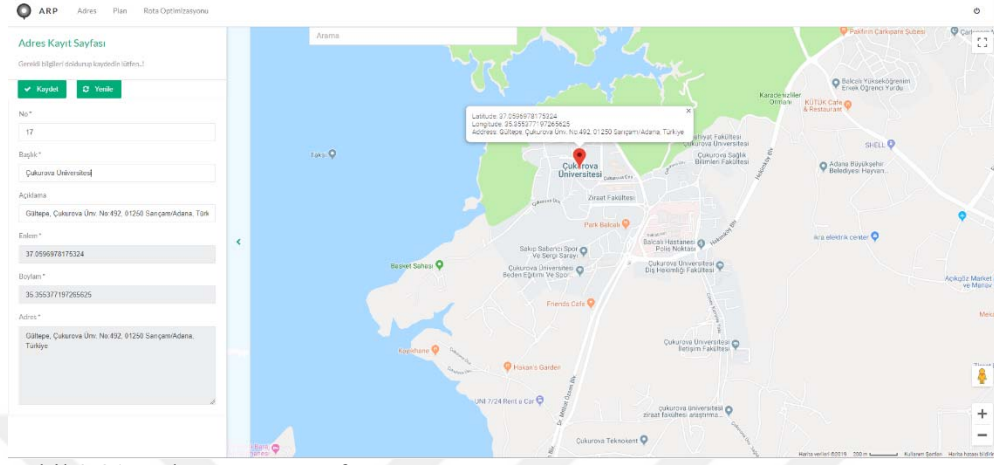
3.2.6.1.Kullanıcı Arayüzü Tasarımı

Bu çalışma sonucunda ortaya koyulan uygulamanın kullanımı kolaylaştırmak için web tabanlı kullanıcı arayüzü tasarlanmıştır. Bu arayüz ile kullanıcı harita üzerinden koordinat seçerek veya adres bilgisi girerek adres tanımlaması yapabilir. Tanımlaması yapılan adresler seçilerek uygulamaya girildi olarak verilir. Uygulamanın oluşturmuş olduğu optimum rota harita üzerinde gösterilerek kullanıcı için anlaşılır sonuçlar üretir.

3.2.6.1.(1).Adres Kayıt Sayfası

Şekil 3.20’de gösterilen sayfanın üst kısmında bulunan “Adres” linkine tıklanır. “Adres Kayıt Sayfası” açılır.

Şekil 3.20’de gösterilen sayfada adres, harita alanının sol üst köşesinde bulunan arama alanından aranılarak veya konum olarak biliniyorsa haritadan arama yapmadan direk işaretlenir. İşaretleme yapıldığında veya arama yapıldıktan sonra adres seçimi gerçekleştirildiğinde sol panelde bulunan adres bilgileri (açıklama, enlem, boylam, adres, no) otomatik dolar. Başlık bilgisi girilerek ve isteğe bağlı olarak açıklama alanı değiştirilerek kaydet butonuna basılır. Başarılı şekilde yeni adres kaydı tanımlanmış olur.



Şekil 3.20. Adres Kayıt Sayfası

3.2.6.1.(2).Planlama Sayfası

Menüden “Plan” linkine tıklanır. Daha önceden kayıt edilmiş adresler listelenir.

Şekil 3.21’de listelenen adresler tıklanarak seçilir. Her adres seçiminde o adresle ilgili talepler girilir ve depo seçimi yapılır. Seçim işlemi tamamlandıktan sonra plana isim verilerek kaydedilir. Böylece plan listesi hazır hale gelir.

#No	Başlık	Açıklama	Enlem	Boylam	Adres
1	Çukurova Üniversitesi	Gültepe Mahallesi, Çukurova Ün. No:492, 01250 Sarıçam/Adana, Türkiye	37.0596978175324	35.355377197265625	Gültepe Mahallesi, Çukurova Ün. No:492, 01250 Sarıçam/Adana, Türkiye
2	Havaalanı	Adana Şakirpaşa Havalimanı, Turhan Cemal Beriker Blv., 01000 Seyhan/Adana, Türkiye	36.98582581985	35.2880859375	Adana Şakirpaşa Havalimanı, Turhan Cemal Beriker Blv., 01000 Seyhan/Adana, Türkiye
3	Mersin	Mithatpaşa Mahallesi, 5506. Sk. No:20, 33040 Akdeniz/Mersin, Türkiye	36.81323387340213	34.63553109723648	Mithatpaşa Mahallesi, 5506. Sk. No:20, 33040 Akdeniz/Mersin, Türkiye
4	Silifke	Burunucu Mahallesi, Unnamed Road, 33900 Silifke/Mersin, Türkiye	36.34845881374465	33.92518804922463	Burunucu Mahallesi, Unnamed Road, 33900 Silifke/Mersin, Türkiye
5	Pınar	Pınar Mahallesi, 74211. Sk. 3-2, 01160 Seyhan/Adana, Turkey	37.0140568389993	35.290289867431625	Pınar Mahallesi, 74211. Sk. 3-2, 01160 Seyhan/Adana, Turkey
6	test	Sarıhuğlar Mahallesi, Algıda 1 N, 01355 Seyhan/Adana, Türkiye	37.01364562932162	35.22746180346678	Sarıhuğlar Mahallesi, Algıda 1 N, 01355 Seyhan/Adana, Türkiye
7	Reğatbey	Reğatbey Mahallesi, Ordu Cd. 12 C, 01120 Seyhan/Adana, Türkiye	36.9914194	35.3308285000005	Reğatbey Mahallesi, Ordu Cd. 12 C, 01120 Seyhan/Adana, Türkiye
8	Hacettepe	Hacettepe, A.Adnan Saygun Cd, Altındağ/Ankara, Türkiye	39.9333635	32.8597419000002	Hacettepe, A.Adnan Saygun Cd, Altındağ/Ankara, Türkiye
9	Yaprak	Yaprak Mahallesi, İstasyon Cd. No:32, 27080 Şehitkamil/Gaziantep, Türkiye	37.065953	37.378109999999999	Yaprak Mahallesi, İstasyon Cd. No:32, 27080 Şehitkamil/Gaziantep, Türkiye

Şekil 3.21.Planlama Sayfası

3.2.6.1.(3).Rotalama Sayfası

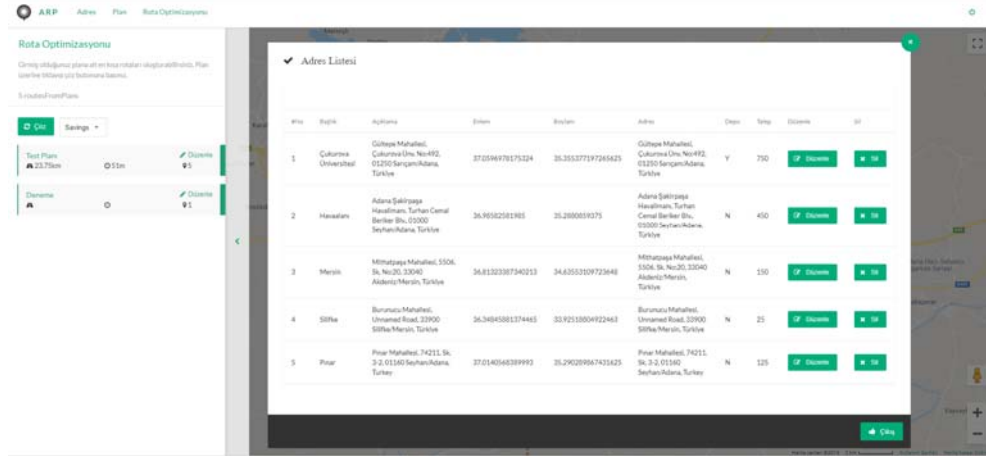
Şekil 3.22’de sayfanın yukarısında bulunan “Rota Optimizasyonu” linkine tıklanır. Daha önceden kaydedilen planlar listelenir.

Şekil 3.22’de listelenen planlar üzerine tıklanarak seçilir. Seçilen plana ait lokasyonlar harita üzerinde işaretlenir. Planlar üzerinde herhangi bir değişiklik yapılmayacaksa çözüme butonuna basılarak optimum rota oluşturulur. Oluşan rota harita üzerinde yol çizimi yapılarak gösterilir. Rota ile ilgili detaylar sayfanın altında gösterilir.



Şekil 3.22. Rotalama Sayfası

Eğer planlar üzerinde değişiklik yapılacaksa; Şekil 3.23'te görülen sayfanın sol panelinde listelenen planların sağ köşesinde bulunan düzenle butonuna basılır. Bu işlemten sonra Şekil 3.23'te görülen plana ait lokasyonların listesinin bulunduğu ekran açılır.



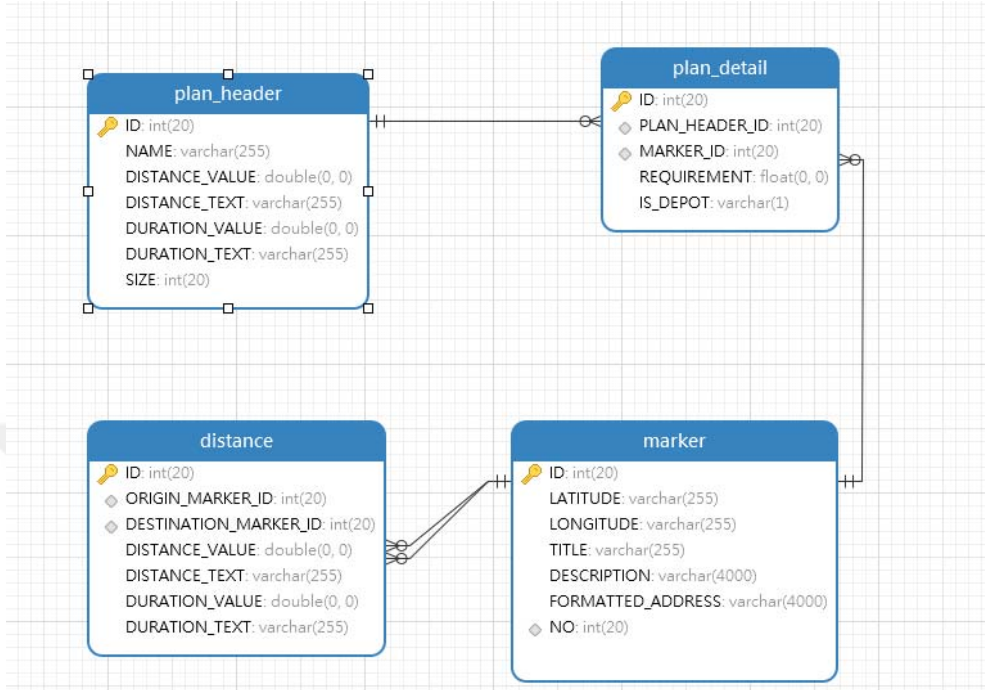
Şekil 3.23. Rota Güncelleme

Şekil 3.23'te gösterilen ekranda sil butonu ile adres silinip plandan çıkartılabilir. Yani müşteri çıkarma işlemi yapılabilir. Düzenle butonu ile ilgili plana ait depo bilgisi veya müşteri talep miktarı güncellenebilir. Güncellemeler yapıldıktan sonra kaydet butonuna basılarak plan güncellenir. Plan bu haliyle çözülmeye hazırdır. İlgili plana tıklanarak plana ait müşteriler harita üzerinde işaretlenir. Çöz butonuna basılarak optimum rota bulunur ve oluşan rota yol çizimi ile harita üzerinde görüntülenir.

Şekil 3.23'te gösterilen ekranda plana ait müşterilerin dağılımını ve lokasyonlarını görmek için planın üzerine tıklanabilir ve ilgili planın müşteri dağılımı harita üzerinde görüntülenebilir.

3.2.6.2. Veri Tabanı Tasarımı

Uygulamada kullanıcı verilerini tutmak için veri tabanı olarak MySQL kullanılmıştır. Veri tabanındaki tablolar ve tablolar arasındaki ilişkiler Şekil 3.24'te verilmiştir.



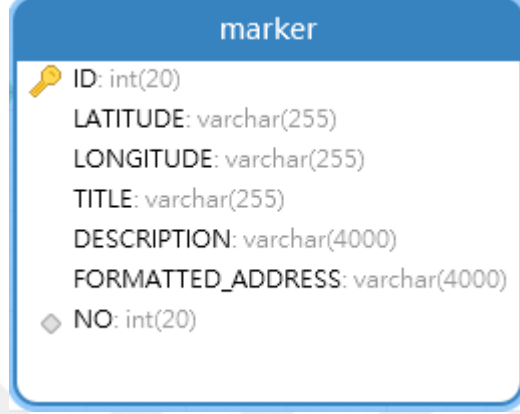
Şekil 3.24. Veri tabanı Diyagramı

“marker” isimli tablo lokasyon bilgisini tutmak üzere oluşturulmuştur. Tabloya ait alanlar Çizelge 3.3’te verilmiştir.

Çizelge 3.3. "marker" Tablosu Alanları

Alan	Tip	Açıklama
ID	int	Birincil anahtardır. Otomatik olarak üretilir.
LATITUDE	varchar	Lokasyonların enlem bilgisi bu alanda tutulur.
LONGITUDE	varchar	Lokasyonların boylam bilgisi bu alanda tutulur.
TITLE	varchar	Müşteri veya depoyu yani lokasyonu tanımlayacak bilgi bu alanda tutulur.
DESCRIPTION	varchar	Lokasyon ile ilgili açıklamanın tutulacağı alandır.
FORMATTED_ADDRESS	varchar	Lokasyonun adres bilgisinin kaydedileceği alandır. Bu alan haritaya tıkladığında, koordinatlar kullanılarak Google Map API yardımıyla bulunur. Eğer kullanıcı harita yerine adres arama alanını kullanırsa, bu alanda girdiği kelimeler Google Maps API ile sorgulanarak öneriler getirilir ve kullanıcının listelenen adreslerden seçim yapmasıyla bu alana kaydedilecek bilgi sağlanmış olur.
NO	int	Her müşteriye otomatik olarak bir no atanır. Bu no haritada müşteri gösterilirken kullanılır.

Şekil 3.25’de “marker” tablosu verilmiştir.



marker	
ID	int(20)
LATITUDE	varchar(255)
LONGITUDE	varchar(255)
TITLE	varchar(255)
DESCRIPTION	varchar(4000)
FORMATTED_ADDRESS	varchar(4000)
NO	int(20)

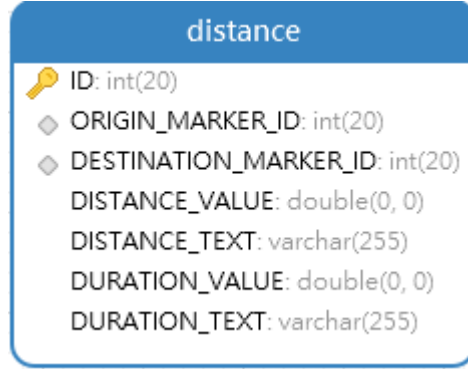
Şekil 3.25. "marker" Tablosu

“distance” isimli tablo müşterilerin birbirleri ve depo ile arasındaki uzaklık bilgisini tutmak üzere oluşturulmuştur. Her yeni adres kayıt işleminde yeni kaydedilen adres ile veri tabanındaki tüm adresler arasındaki uzaklık bilgisi, süre bilgisi gibi bilgiler Google Maps API’leri ile bulunur ve “distance” tablosuna kaydedilir. “distance” tablosu “marker” tablosu ile ilişkilidir. Tabloya ait alanlar Çizelge 3.4’te verilmiştir.

Çizelge 3.4."distance" Tablosu Alanları

Alan	Tip	Açıklama
ID	İnt	Birincil anahtardır. Otomatik olarak üretilir.
ORIGIN_MARKER_ID	int	Kaynak lokasyonun ID'sini tutar. Yabancı anahtardır. "marker" tablosunun ID'si ile ilişkilidir.
DESTINATION_MARKER_ID	int	Hedef lokasyonun ID'sini tutar. Yabancı anahtardır. "marker" tablosunun ID'si ile ilişkilidir.
DISTANCE_VALUE	double	Kaynak ve hedef lokasyon arasındaki uzaklığı metre olarak tutar.
DISTANCE_TEXT	varchar	Gerçek hayattaki kullanımda uzaklıkları metre şeklinde göstermek anlaşılır olmayabilir. Bundan dolayı bu alan oluşturulmuştur. Uzaklıklar kullanıcı için anlamlı olacak formatta bu alanda tutulur. Örneğin DISTANCE_VALUE'de 21715 olan değer bu alanda "21 km 715 m" şeklinde tutularak kullanıcıya anlamlı bilgi sunulur.
DURATION_VALUE	double	Kaynak ve hedef lokasyon arasındaki araç ile tahmini varış süresidir. Süre bilgisi dakika birimiyle tutulur.
DURATION_TEXT	varchar	Süre bilgisinin yazı formatında tutulmuş halidir. Örneğin DURATION VALUE' de 127 olan değer bu alanda "2 s 7 dk" olarak tutulur.

Şekil 3.26'da “distance” tablosu verilmiştir.



The image shows a screenshot of a database table definition for a table named "distance". The table has the following columns and data types:

Column Name	Data Type
ID	int(20)
ORIGIN_MARKER_ID	int(20)
DESTINATION_MARKER_ID	int(20)
DISTANCE_VALUE	double(0, 0)
DISTANCE_TEXT	varchar(255)
DURATION_VALUE	double(0, 0)
DURATION_TEXT	varchar(255)

Şekil 3.26. "distance " Tablosu

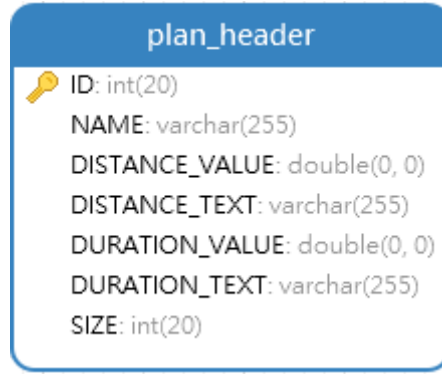
“plan_header” isimli tablo dağıtım planının genel bilgilerini tutmak üzere oluşturulmuştur. Bir dağıtım planına ait bir satır veri tutulur: Tabloya ait alanlar Çizelge 3.5’de verilmiştir.

Çizelge 3.5. "plan_header" Tablosu Alanları

Alan	Tip	Açıklama
ID	int	Birincil anahtardır. Otomatik olarak üretilir.
NAME	varchar	Kullanıcın plana ait isimlendirmesi bu alanda tutulur.
DISTANCE_VALUE	double	Planın toplam uzaklığını metre olarak tutar.
DISTANCE_TEXT	varchar	DISTANCE_VALUE'nin yazı formatındaki halidir. Uzaklıklar kullanıcı için anlamlı olacak formatta bu alanda tutulur. Örneğin DISTANCE_VALUE'de 892536 olan değer bu alanda "892 km 536 m" şeklinde tutularak kullanıcıya anlamlı bilgi sunulur.
DURATION_VALUE	double	İlgili plan ile oluşan rotanın tahmini tamamlanma süresidir. Süre bilgisi dakika birimiyle tutulur.
DURATION_TEXT	varchar	Süre bilgisinin yazı formatında tutulmuş halidir. Örneğin DURATION VALUE' de 436 olan değer bu alanda "7 s 16 dk" olarak tutulur.
SIZE	İnt	Rota sayısını tutmak için kullanılır.

"plan_header" tablosu ile ilgili DISTANCE_VALUE, DISTANCE_TEXT, DURATION_VALUE, DURATION_TEXT alanları plan oluşturulurken "null" (boş değer)dir. Dağıtım planı oluşturulduktan sonra uygulama ile çözüldüğünde bu alanlar otomatik olarak dolacaktır.

Şekil 3.27'de "plan_header" tablosu verilmiştir.



plan_header	
ID:	int(20)
NAME:	varchar(255)
DISTANCE_VALUE:	double(0, 0)
DISTANCE_TEXT:	varchar(255)
DURATION_VALUE:	double(0, 0)
DURATION_TEXT:	varchar(255)
SIZE:	int(20)

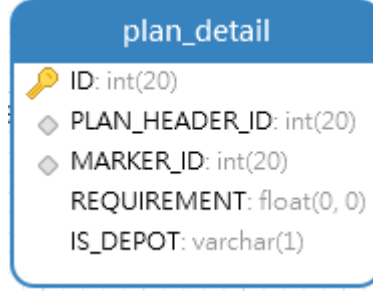
Şekil 3.27. "plan_header" Tablosu

“plan_detail” isimli tablo dağıtım bilgilerinin detaylarını kaydetmek amacıyla oluşturulmuştur. Bir dağıtım planına ait müşterilerin tamamı ve depo bu tabloda bulunur. “plan_detail” tablosu, “plan_header” ve “marker” tablolarıyla ilişkilidir. Tabloya ait alanlar Çizelge 3.6’da verilmiştir.

Çizelge 3.6. "plan_detail" Tablosu Alanları

Alan	Tip	Açıklama
ID	int	Birincil anahtardır. Otomatik olarak üretilir.
PLAN_HEADER_ID	int	Dağıtım planının ID'sini tutar. Yabancı anahtardır. "plan_header" tablosunun ID'si ile ilişkilidir.
MARKER_ID	int	Müşteri veya deponun lokasyonun ID'sini tutar. Yabancı anahtardır. "marker" tablosunun ID'si ile ilişkilidir.
REQUIREMENT	double	İlgili kayıt müşteri ise talep miktarını, depo ise ilgili depoya ait bir aracın kapasitesini tutar.
IS_DEPOT	varchar	Dağıtım planında seçilenin lokasyonun müşteri veya depo bilgisi bu alana kaydedilir. Bu alanda tutulan değer "Y" ise lokasyon depo, "N" ise müşteridir. Dağıtım planı yapılırken kullanıcıya seçtiği lokasyonun depo olup olmadığı sorulur. Evet cevabını verilirse diğer lokasyon seçimlerinde bu soru sorulmaz ve bu şekilde bir dağıtım planına ait tek depo girişi garanti edilir.

Şekil 3.28’de “plan_detail” tablosu verilmiştir.



plan_detail	
🔑	ID: int(20)
◊	PLAN_HEADER_ID: int(20)
◊	MARKER_ID: int(20)
	REQUIREMENT: float(0, 0)
	IS_DEPOT: varchar(1)

Şekil 3.28. "plan_detail" Tablosu



4. BULGULAR VE TARTIŞMA

Bu bölümde bulgular ve tartışmalara yer verilmiştir. Kullanıcı tanımlı parametrelerin belirlenmesi için testler yapılmıştır. Belirlenen parametreler ile algoritmalar literatürden alınan test problemleri ile test edilmiştir.

Algoritmalar; veri setindeki her bir veri için 100 defa çalıştırılmıştır. Bu işlem sonucunda; minimum maliyet, ortalama maliyet, maksimum maliyet, maliyetin standart sapması, minimum hız, ortalama hız, maksimum hız gibi değerler bulunmuştur. Algoritmaların performansı ve kalitesi değerlendirilirken; maliyet, standart sapma ve hız önemli olduğu için bu değerler her algoritma için hesaplanmıştır. Bu değerler algoritmanın performansını karşılaştırmak için gerekli metrikler olarak kullanıp algoritmaların performansları karşılaştırılmıştır. Ayrıca seçilen farklı ölçeklerde veriler ile testler gerçekleştirildiğinden dolayı algoritmanın farklı koşullarda davranışı incelenebilmiştir.

Algoritmaların performansı ve başarımını ölçmek amacıyla kullanılacak metrikler aşağıda açıklanmıştır:

Maliyet; algoritmanın bulmuş olduğu rotaların toplam mesafesini ifade etmektedir. Algoritmaların minimum, ortalama ve maksimum maliyetleri bulunmuştur.

Standart sapma; algoritmaların maliyetinin standart sapması hesaplanmıştır. Standart sapma algoritmanın kararlılığı hakkında bilgi vericidir. Küçük standart sapmaya sahip algoritmaların daha kararlı sonuçlar ürettiği düşünülür.

Hız; algoritmanın çalışma süresini ifade etmektedir. Algoritmaların minimum, ortalama ve maksimum hızları bulunmuştur. Algoritmaların hız ifadesi için milisaniye tercih edilmiştir. Çalışma süresi ve hız aynı anlamda kullanılmıştır.

4.1. Kullanıcı Tanımlı Parametrelerin Belirlenmesi

Metasezgisel algoritmanın iyi sonuç üretebilmesi için parametrelerin doğru değerlerde belirlenmesi oldukça önemlidir. Uygulanan probleme göre parametre değerleri değişiklik gösterebilir. Parametre değer belirlenmesi için kesin bir yöntem yoktur. Bundan dolayı parametrelerin farklı değerleri test edilerek en uygun değer belirlenir.

Bu çalışmada parametre değerlerinin belirlenmesi için; değeri belirlenecek parametre dışında kalanlar sabit tutulurken, değeri belirlenecek parametreye çeşitli değerler verilmiştir. Parametre değerlerinin belirlenmesi için literatürde yaygın olarak kullanılan P grubu veri setinden P-n101-k4 isimli tek depolu, 100 müşterili ve 4 araçlı kapasite kısıtlı araç rotalama problemi kullanılmıştır. Orta ölçekteki bu problem ile test edilmesi planlanan her bir değer, algoritmada 100 defa çalıştırılmıştır. Bunun sonucunda; minimum maliyet, ortalama maliyet, maksimum maliyet, maliyetin standart sapması, minimum hız, ortalama hız, maksimum hız değerleri bulunmuştur. Sonuçlara göre hız, maliyet ve kararlılık açısından (standart sapma değerinin düşük olması algoritmanın daha kararlı olduğu anlamına gelmektedir) en uygun değer seçilmiştir.

4.1.1.Genetik Algoritma Parametrelerinin Belirlenmesi

Genetik algoritmada belirlenmesi gereken parametreler; jenerasyon sayısı, birey sayısı, mutasyon oranı, çaprazlama yöntemi oranlarıdır.

4.1.1.1.Birey Sayısının Belirlenmesi

Birey sayısına 25-250 aralığında 25'er artacak şekilde değerler verilmiştir. Birey sayısının belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1500

- Mutasyon Oranı: 0.3
- Çaprazlama İşlemindeki Yer Değiştirme Oranı (CR1) : 0.33
- Çaprazlama İşlemindeki Tersleme Oranı (CR2) : 0.66
- Çaprazlama İşlemindeki Ekleme Oranı (CR3) : 1

Çizelge 4.1'deki sonuçlar içerisinde minimum maliyeti olan, ortalama maliyeti, ortalama hızı ve standart sapması kabul edilebilir düzeyde olan seçilmiştir. Birey sayısı 125 olarak belirlenmiştir.

Çizelge 4.1. Genetik Algoritma Birey Sayısı Test Sonuçları

Birey Sayısı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
25	813	912.6	1054	43.41	322	350	945
50	764	866.45	1004	47.11	650	670	789
75	768	855.84	961	39.89	984	1003	1090
100	779	856.54	953	40.11	1319	1356	1675
125	755	854.14	961	42.60	1657	1695	1892
150	761	854.32	1022	45.54	1991	2045	2685
175	758	847.45	951	43.79	2311	2358	2505
200	741	847.17	999	44.96	2651	2689	2876
225	759	842.88	954	39.37	2983	3048	3737
250	755	853.32	984	47.33	3337	3401	3740

4.1.1.2.Mutasyon Oranının Belirlenmesi

Mutasyon oranına 0-1 aralığında 0.2'şer artacak şekilde değerler verilmiştir. Mutasyon oranının belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1500

- Birey Sayısı: 125
- Çaprazlama İşlemindeki Yer Değiştirme Oranı (CR1) : 0.33
- Çaprazlama İşlemindeki Tersleme Oranı (CR2) : 0.66
- Çaprazlama İşlemindeki Ekleme Oranı (CR3) : 1

Çizelge 4.2'deki sonuçlar içerisinde ortalama maliyeti minimum olan, ortalama hızı ve standart sapması kabul edilebilir düzeyde olan seçilmiştir. Mutasyon oranı 0.2 olarak belirlenmiştir.

Çizelge 4.2. Genetik Algoritma Mutasyon Oranı Test Sonuçları

Mutasyon Oranı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
0.0	782	897.4	1054	54.66	1568	1612	2560
0.2	779	841.1	978	42.38	1578	1621	2093
0.4	770	855.15	975	45.08	1590	1615	1726
0.6	777	859.85	1036	52.21	1593	1629	1889
0.8	761	862.95	955	38.81	1607	1643	1962
1.0	852	948.27	1049	42.46	1693	1738	2620

4.1.1.3. Çaprazlama Yöntemi Oranlarının Belirlenmesi

Çaprazlama işlemindeki yer değiştirme oranı (CR1), tersleme oranı (CR2), ekleme oranı (CR3) olasılık toplamı 1 olacak şekilde farklı kombinasyonlarda değerler atanmıştır. Atanan değerler Çizelge 4.4'de gösterilmiştir. Bu parametrelerin değerlerinin belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1500
- Birey Sayısı: 125

- Mutasyon Oranı: 0.2

Çizelge 4.3'deki sonuçlar içerisinde ortalama maliyeti minimum olan, ortalama hızı ve standart sapması kabul edilebilir düzeyde olan seçilmiştir. Çaprazlama işlemindeki tersleme oranı (CR2) 0.6, yer değiştirme oranı (CR1) 0.8, ekleme oranı (CR3) 1 olarak belirlenmiştir.

Çizelge 4.3. Genetik Algoritma Çaprazlama Oranı Test Sonuçları

Oranlar	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
CR1:0.2 CR2:0.4 CR3:1	771	850.3	945	39.88	1585	1645	3116
CR1:0.4 CR2:0.6 CR3:1	754	858.41	1003	45.46	1651	1687	2174
CR1:0.6 CR2:0.8 CR3:1	778	860.19	1004	44.47	1669	1714	2120
CR2:0.2 CR1:0.4 CR3:1	768	853.06	959	41.56	1538	1578	2320
CR2:0.4 CR1:0.6 CR3:1	769	857.05	972	40.13	1597	1632	1998
CR2:0.6 CR1:0.8 CR3:1	770	841.25	948	38.01	1649	1692	1920
CR3:0.2 CR2:0.4 CR1:1	760	854.06	994	41.30	1634	1680	2640
CR3:0.4 CR2:0.6 CR1:1	784	854.38	979	37.40	1595	1653	2164
CR3:0.6 CR2:0.8 CR1:1	779	851.62	1014	42.24	1583	1616	1824

4.1.1.4.Jenerasyon Sayısının Belirlenmesi

Jenerasyon sayısına 500-5000 aralığında 500'er artacak şekilde değerler verilmiştir. Jenerasyon sayısının belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Birey Sayısı: 125
- Mutasyon Oranı: 0.2
- Çaprazlama İşlemindeki Tersleme Oranı (CR2) : 0.6
- Çaprazlama İşlemindeki Yer Değiştirme Oranı (CR1) : 0.8
- Çaprazlama İşlemindeki Ekleme Oranı (CR3) : 1

Çizelge 4.4'deki sonuçlar incelendiğinde ortalama maliyetler arasındaki farklar düşük olduğu için standart sapması en düşük ve ortalama hızı kabul edilebilir olan kayıt seçilmiştir. Jenerasyon sayısı 2500 olarak belirlenmiştir.

Çizelge 4.4. Genetik Algoritma Jenerasyon Sayısı Test Sonuçları

Jenerasyon Sayısı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
500	756	878.25	1040	45.07	610	651	1633
1000	776	846.99	949	39.24	1211	1236	1384
1500	781	843.48	973	39.80	1796	1840	2382
2000	760	844.02	968	39.76	2384	2429	2552
2500	768	835.08	924	34.98	2957	3005	3385
3000	770	840.66	1031	47.82	3546	3593	3776
3500	753	838.42	946	37.91	4099	4160	4363
4000	760	839.41	971	37.05	4684	4745	4871
4500	757	838.5	926	36.82	5278	5346	5713
5000	742	827.79	917	36.89	5862	6055	7444

4.1.1.5.Parametre Test Sonuçları

Testler sonucunda genetik algoritmanın parametre değerleri aşağıdaki gibi bulunmuştur:

- Birey Sayısı: 125
- Mutasyon Oranı: 0.2
- Çaprazlama İşlemindeki Tersleme Oranı (CR2) : 0.6
- Çaprazlama İşlemindeki Yer Değiştirme Oranı (CR1) : 0.8
- Çaprazlama İşlemindeki Ekleme Oranı (CR3) : 1
- Jenerasyon Sayısı: 2500

4.1.2.Yabani Ot Algoritması

Yabani ot algoritmasında belirlenmesi gereken parametreler; jenerasyon sayısı, başlangıç yabani ot sayısı, maksimum yabani ot sayısı, minimum tohum sayısı (S_{min}), maksimum tohum sayısı (S_{max}), varyans azaltma bileşeni (n), standart sapma başlangıç değeri ($\sigma_{initial}$), standart sapma son değeridir (σ_{final}).

4.1.2.1.Başlangıç Yabani Ot Sayısının Belirlenmesi

Başlangıç yabani ot sayısına 500-5000 aralığında 500'er artacak şekilde değerler verilmiştir. Başlangıç yabani ot sayısının belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1000
- Maksimum Yabani Ot Sayısı: 30
- Minimum Tohum Sayısı (S_{min}): 1
- Maksimum Tohum Sayısı (S_{max}): 15
- Varyans Azaltma Bileşeni (n): 2

- Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): 1
- Standart Sapma Son Değeri (σ_{final}): 1000

Çizelge 4.5’da sonuçlar içerisinde minimum maliyeti, ortalama maliyeti ve standart sapması minimum olan, ortalama hızı uygun olan durum seçilmiştir. Başlangıç yabancı ot sayısı 1500 olarak belirlenmiştir.

Çizelge 4.5. YOA Başlangıç Yabancı Ot Sayısı Testi

Başlangıç Yabancı Ot Sayısı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
500	802	891.33	1037	54.86	78	110	311
1000	775	887.61	1068	54.42	82	105	137
1500	756	869.53	1019	48.17	94	117	161
2000	782	887.64	1080	61.35	101	123	175
2500	766	879.68	1006	54.79	106	130	168
3000	790	887.92	1049	54.89	106	135	185
3500	770	883.72	1020	50.50	108	145	189
4000	789	876.78	1042	53.99	131	152	185
4500	771	880.95	1043	54.80	137	163	235
5000	762	878.58	1025	61.04	135	168	209

4.1.2.2. Maksimum Yabancı Ot Sayısının Belirlenmesi

Maksimum yabancı ot sayısına 25-250 aralığında 25’er artacak şekilde değerler verilmiştir. Maksimum yabancı ot sayısının belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1000
- Başlangıç Yabani Ot Sayısı: 1500
- Minimum Tohum Sayısı (S_{min}): 1
- Maksimum Tohum Sayısı (S_{max}): 15
- Varyans Azaltma Bileşeni (n): 2
- Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): 1
- Standart Sapma Son Değeri (σ_{final}): 1000

Çizelge 4.6'daki sonuçlar içerisinde minimum maliyete yakın, minimum standart sapmalı, ortalama hızı uygun olan durum seçilmiştir. Maksimum yabancı ot sayısı 125 olarak belirlenmiştir.

Çizelge 4.6. YOA Maksimum Yabani Ot Sayısı Testi

Maksimum Yabani Ot Sayısı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
25	779	894.48	1061	57.16	77	115	360
50	758	864.46	980	50.09	130	157	203
75	738	852.32	1040	53.58	155	206	342
100	748	850.32	1034	59.25	197	253	326
125	751	839.14	961	44.60	252	295	359
150	751	843.39	1025	53.75	279	349	616
175	750	845.71	963	51.57	331	387	628
200	728	844.61	1036	49.39	359	423	515
225	751	838.66	996	45.14	405	478	709
250	746	839.66	997	50.12	447	543	722

4.1.2.3.Maksimum Tohum Sayısının Belirlenmesi

Maksimum tohum sayısına (S_{max}) 10-100 aralığında 10'ar artacak şekilde değerler verilmiştir. Maksimum tohum sayısının (S_{max}) belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1000
- Başlangıç Yabani Ot Sayısı: 1500
- Maksimum Yabani Ot Sayısı: 125
- Minimum Tohum Sayısı (S_{min}): 1
- Varyans Azaltma Bileşeni (n): 2
- Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): 1
- Standart Sapma Son Değeri (σ_{final}): 1000

Çizelge 4.7'de sonuçlar içerisinde minimum maliyeti, ortalama maliyeti ve standart sapması minimum olan, ortalama hızı uygun olan durum seçilmiştir. Maksimum tohum sayısı (S_{max}) 90 olarak belirlenmiştir.

Çizelge 4.7. YOA Maksimum Tohum Sayısı (S_{max}) Değer Testi

S_{max}	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
10	736	836.81	958	48.04	218	274	697
20	743	839.77	1051	50.76	262	333	423
30	754	826.15	932	36.89	342	413	529
40	741	818.44	916	40.15	421	494	663
50	747	814.81	964	39.89	454	566	680
60	745	822.04	911	39.11	490	632	866
70	723	811.99	907	38.17	596	704	850
80	732	808.88	905	37.73	614	775	940
90	729	799.46	918	33.59	704	860	1133
100	746	806.72	928	38.23	750	916	1095

4.1.2.4.Varyans Azaltma Bileşeninin Belirlenmesi

Varyans azaltma bileşenine (n) 1-5 aralığında 1'er artacak şekilde değerler verilmiştir. Varyans azaltma bileşeninin (n) belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1000
- Başlangıç Yabani Ot Sayısı: 1500
- Maksimum Yabani Ot Sayısı: 125
- Minimum Tohum Sayısı (S_{min}): 1
- Maksimum Tohum Sayısı (S_{max}): 90
- Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): 1
- Standart Sapma Son Değeri (σ_{final}): 1000

Çizelge 4.8'de sonuçlar içerisinde ortalama maliyeti ve standart sapması minimum olan, ortalama hızı uygun olan durum seçilmiştir. Varyans azaltma bileşeninin (n) değeri 3 olarak belirlenmiştir.

Çizelge 4.8. YOA Varyans Azaltma Bileşeni (n) Değer Testi

n	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
1	756	856.44	1009	53.57	261	321	675
2	742	842.56	1087	56.55	275	345	503
3	763	841.54	985	42.34	264	321	475
4	769	847.75	980	42.61	273	331	805
5	743	851.51	1020	50.58	260	333	472

4.1.2.5. Standart Sapma Son Değerinin Belirlenmesi

Standart sapma son değerine (σ_{final}); 1, 10, 100, 1000, 10000 verilmiştir. Standart sapma son değerinin (σ_{final}) belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Jenerasyon Sayısı: 1000
- Başlangıç Yabani Ot Sayısı: 1500
- Maksimum Yabani Ot Sayısı: 125
- Minimum Tohum Sayısı (S_{min}): 1
- Maksimum Tohum Sayısı (S_{max}): 90
- Varyans Azaltma Bileşeni (n): 3
- Standart Sapma Başlangıç Değeri (σ_{initial}): 1

Çizelge 4.9 sonuçları içerisinde ortalama maliyeti minimum olan, standart sapma ve ortalama hızı uygun olan durum seçilmiştir. Standart sapma son değeri (σ_{final}) 100 olarak belirlenmiştir.

Çizelge 4.9. YOA Standart Sapma Son Değer (σ_{final}) Testi

σ_{final}	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
1	2766	2921.04	3034	52.78	146	160	433
10	909	1316.93	1841	204.29	197	240	381
100	765	841.86	985	46.33	256	315	450
1000	760	851.32	972	41.26	270	314	395
10000	751	844.04	1003	48.65	249	316	398

4.1.2.6.Jenerasyon Sayısının Belirlenmesi

Jenerasyon sayısına 500-5000 aralığında 500'er artacak şekilde değerler verilmiştir. Jenerasyon sayısının belirlenmesi için sabit tutulacak parametreler ve değerleri aşağıdaki gibi belirlenmiştir:

- Başlangıç Yabani Ot Sayısı: 1500
- Maksimum Yabani Ot Sayısı: 125
- Minimum Tohum Sayısı (S_{min}): 1
- Maksimum Tohum Sayısı (S_{max}): 90
- Varyans Azaltma Bileşeni (n): 3
- Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): 1
- Standart Sapma Son Değeri (σ_{final}): 100

Çizelge 4.10'de sonuçlar içerisinde ortalama maliyeti, standart sapması ve ortalama hızı minimum olan durum seçilmiştir. Jenerasyon sayısı 500 olarak belirlenmiştir.

Çizelge 4.10. YOA Maksimum Jenerasyon Sayısı

Jenerasyon Sayısı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Standart Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
500	733	800.77	879	31.53	618	817	1642
1000	739	805.02	975	41.38	690	856	1348
1500	736	804.94	936	37.48	769	914	1158
2000	742	807.39	908	35.88	830	964	1165
2500	731	804.27	921	38.06	870	1011	1223
3000	744	808.06	946	39.21	887	1073	1246
3500	735	803.14	924	39.31	974	1133	1395
4000	732	804.01	941	40.49	1067	1199	1413
4500	709	809.08	999	48.84	1094	1246	1437
5000	733	803.17	982	37.14	1112	1311	1504

4.1.2.7.Parametre Test Sonuçları

Minimum tohum sayısı koloninin en kötü bireyine tohum üretme şansı vermek için 1 olarak kabul edilmiştir. Standart sapma başlangıç değeri, standart sapma son değerine göre anlamlı hale gelen parametre olduğu için 1 olarak kabul edilmiştir. Testler sonucunda yabancı ot algoritmasının parametre değerleri aşağıdaki gibi bulunmuştur:

- Jenerasyon Sayısı: 500
- Başlangıç Yabancı Ot Sayısı: 1500
- Maksimum Yabancı Ot Sayısı: 125
- Minimum Tohum Sayısı (S_{min}): 1
- Maksimum Tohum Sayısı (S_{max}): 90
- Varyans Azaltma Bileşeni (n): 3
- Standart Sapma Başlangıç Değeri ($\sigma_{initial}$): 1
- Standart Sapma Son Değeri (σ_{final}): 100

4.2.Senaryo Analizi

Bu bölümde deney tasarımına ve deney sonuçlarına yer verilmiştir.

Algoritmalar; Intel Core i7-7600U CPU 2.80GHz işlemcili, 16GB RAM'e sahip, 64 bit Windows 10 Enterprise işletim sistemli bilgisayar üzerinde çalıştırılmıştır. Algoritmaların test edilmesi için P grubu 24 adet ve CMT grubu 5 adet problem örneği seçilmiştir.

Deneysel çalışmalara göre elde edilen sonuçlar P grubu ve CMT grubu veri setleri için Çizelge 4.11 – 4.23'de verilmiştir. Bu çizelgelerde sütunlar sırasıyla; problem adını, en küçük maliyeti, ortalama maliyeti, maksimum maliyeti, algoritmanın standart sapmasını, algoritmanın minimum çözüm süresini, ortalama çözüm süresini ve maksimum çözüm süresini göstermektedir.

4.2.1.P Grubu Veri Seti Sonuçları

P grubu veri seti seçilmiştir ve algoritmalar bu problemler üzerinde çalıştırılmıştır. P grubu veri setine ait problemlerin farklı boyutlarda olması, algoritmaların etkinliğini ve başarımını anlamak için faydalı olacaktır.

4.2.1.1.Tasarruf Algoritması Çözüm Sonuçları

Çizelge 4.11’de tasarruf algoritmasının sonuçları verilmiştir. Tasarruf algoritması, tasarruf değerlerini hesaplayıp işlem yapar ve tasarruf değerleri aynı problem için değişmediğinden dolayı çözüm olarak üretilen sonuçlar aynı değerdedir. Bir problem için minimum, ortalama, maksimum maliyetler aynıdır ve standart sapma değeri 0’dır. Bundan dolayı tablo sadeleştirilmiştir, ortalama maliyet ve ortalama çalışma süresi verilmiştir. Ayrıca elde edilen sonuçlara göre problem boyutunun büyümesi algoritmanın çalışma süresine fazla etki etmemiştir.

Çizelge 4.11. Tasarruf Algoritması Çözüm Sonuçları

Adı	Ort. Maliyet	Ort. Hız(ms)
P-n16-k8	482	≈0
P-n19-k2	226	≈0
P-n20-k2	222	≈0
P-n21-k2	224	≈0
P-n22-k2	232	≈0
P-n22-k8	590	≈0
P-n23-k8	537	≈0
P-n40-k5	561	≈0
P-n45-k5	676	≈0
P-n50-k7	693	≈0
P-n50-k8	716	≈0
P-n50-k10	775	≈0
P-n51-k10	825	≈0
P-n55-k7	702	≈0
P-n55-k8	722	≈0
P-n55-k10	792	≈0
P-n55-k15	1000	≈0
P-n60-k10	859	≈0
P-n60-k15	1020	≈0
P-n65-k10	982	≈0
P-n70-k10	1047	≈0
P-n76-k4	930	≈0
P-n76-k5	989	≈0
P-n101-k4	1266	1

4.2.1.2.Genetik Algoritma Çözüm Sonuçları

Çizelge 4.12'deki genetik algoritmanın sonuçları incelendiğinde problem boyutu büyüdükçe algoritmanın standart sapması genel olarak artmaktadır. Bu durum algoritmanın ürettiği sonucun kararsızlaştığı anlamına gelmektedir. Ayrıca problem boyutu büyüdükçe algoritmanın çalışma süresi de artmıştır. Bu sonuçlara göre genetik algortmada problem boyutu büyüdükçe standart sapma ve çalışma süresi artmaktadır yorumu yapılabilir.

Çizelge 4.12. Genetik Algoritma Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
P-n16-k8	450	450.22	461	1.21	629	704	1868
P-n19-k2	212	224.56	263	9.59	689	710	811
P-n20-k2	216	226.12	250	8.61	722	736	870
P-n21-k2	211	221.32	250	9.41	728	752	1149
P-n22-k2	216	226.56	252	9.26	770	784	982
P-n22-k8	590	595.71	624	8.94	775	799	878
P-n23-k8	529	536.74	570	7.09	809	835	943
P-n40-k5	469	521.24	573	22.75	1241	1264	1377
P-n45-k5	535	583.07	671	28.73	1356	1377	1462
P-n50-k7	574	626.53	732	25.92	1588	1616	1772
P-n50-k8	658	703.17	767	24.72	1559	1588	1692
P-n50-k10	720	774.72	883	27.08	1551	1584	1670
P-n51-k10	770	833.27	887	28.02	1571	1598	1699
P-n55-k7	594	647.10	777	29.94	1688	1713	1815
P-n55-k8	602	651.41	758	27.78	1669	1701	1789
P-n55-k10	723	763.49	817	20.51	1675	1704	1805
P-n55-k15	976	1020.65	1107	24.79	1709	1747	1861
P-n60-k10	780	833.91	908	27.37	1801	1835	1981
P-n60-k15	1021	1060.69	1132	24.69	1787	1824	1900
P-n65-k10	828	884.73	975	27.20	1907	1971	2920
P-n70-k10	883	959.49	1049	36.48	2040	2075	2349
P-n76-k4	629	706.76	783	32.45	2141	2198	2424
P-n76-k5	669	739.00	828	37.94	2203	2235	2373
P-n101-k4	743	846.84	980	45.37	2731	2904	5001
Standart Sapmaların Ortalaması				22.74			

4.2.1.3.Yabani Ot Algoritması Çözüm Sonuçları

Çizelge 4.13'deki yabancı ot algoritmasının sonuçları incelendiğinde problem boyutu büyüdükçe algoritmanın çalışma süresi uzamaktadır ve genel olarak standart sapma artmaktadır. Bu sonuçlara göre yabancı ot algoritmasında problem boyutu büyüdükçe standart sapma ve çalışma süresi artmaktadır yorumu yapılabilir.

Çizelge 4.13. Yabancı Ot Algoritması Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
P-n16-k8	450	450.42	463	1.63	50	62	233
P-n19-k2	212	223.13	263	9.05	61	73	123
P-n20-k2	216	224.70	250	8.41	67	80	152
P-n21-k2	211	218.51	243	8.19	70	83	168
P-n22-k2	216	224.07	256	8.92	75	87	156
P-n22-k8	590	598.83	645	11.17	70	83	139
P-n23-k8	529	538.64	581	8.90	72	87	105
P-n40-k5	472	503.19	586	19.58	138	174	269
P-n45-k5	519	551.67	615	20.90	159	206	325
P-n50-k7	571	610.10	715	21.39	169	241	331
P-n50-k8	651	683.58	738	16.39	173	246	395
P-n50-k10	707	758.27	850	27.26	190	240	378
P-n51-k10	771	824.22	955	33.35	190	245	365
P-n55-k7	591	629.66	692	21.37	230	283	456
P-n55-k8	592	630.02	719	22.05	230	284	414
P-n55-k10	713	747.44	811	19.16	219	287	503
P-n55-k15	969	1018.94	1092	24.57	191	259	437
P-n60-k10	768	811.29	915	23.09	255	332	434
P-n60-k15	1011	1058.29	1162	25.32	232	310	442
P-n65-k10	811	865.20	914	18.77	303	385	776
P-n70-k10	859	939.20	1077	38.02	319	437	633
P-n76-k4	628	673.43	738	26.56	395	480	674
P-n76-k5	659	709.47	810	30.36	391	503	643
P-n101-k4	740	811.81	921	37.48	613	828	1186
Standart Sapmaların Ortalaması				20.08			

4.2.1.4.Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları

Çizelge 4.14'deki tasarruf ve genetik algoritma hibrit yaklaşım sonuçları incelendiğinde büyük ölçekli problemlerde standart sapmanın genel olarak daha büyük olduğu gözlenmektedir. Bu durum büyük ölçekli problemlerde algoritmanın ürettiği sonucun kararsızlaştığı anlamına gelmektedir. Ayrıca problem boyutu büyüdükçe algoritmanın çalışma süresi de artmıştır. Tasarruf ve genetik algoritma hibrit yaklaşımında genel olarak genetik algoritma karakteristiğini göstermeye devam etmiş ancak genetik algoritmanın ürettiği sonuçlar iyileşmiş, standart sapma oranları düşmüştür.

Çizelge 4.14. Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
P-n16-k8	450	450.55	461	2.40	642	775	1432
P-n19-k2	212	218.75	219	1.13	700	712	799
P-n20-k2	216	217.98	218	0.20	727	741	989
P-n21-k2	211	213.75	222	4.76	737	755	896
P-n22-k2	216	217.93	227	4.14	772	787	849
P-n22-k8	590	590.00	590	0.00	797	816	906
P-n23-k8	529	530.95	537	2.87	825	849	917
P-n40-k5	464	499.60	522	10.48	1255	1274	1416
P-n45-k5	511	533.46	584	13.03	1373	1395	1491
P-n50-k7	569	590.43	624	6.42	1607	1631	1744
P-n50-k8	662	670.57	675	3.20	1579	1597	1698
P-n50-k10	711	726.44	762	8.36	1572	1595	1707
P-n51-k10	753	771.52	798	9.76	1588	1613	1714
P-n55-k7	585	596.20	609	5.16	1694	1722	1844
P-n55-k8	599	617.70	656	9.04	1689	1719	1842
P-n55-k10	712	725.48	739	3.89	1697	1721	1838
P-n55-k15	966	983.98	992	4.11	1734	1784	1980
P-n60-k10	774	786.05	820	7.81	1814	1839	1948
P-n60-k15	988	1000.29	1001	2.16	1812	1840	1947
P-n65-k10	828	846.47	903	10.51	1935	2062	3445
P-n70-k10	871	905.23	938	13.71	2067	2248	4071
P-n76-k4	627	662.95	724	16.76	2184	2265	3757
P-n76-k5	675	699.72	743	10.35	2244	2421	3030
P-n101-k4	741	794.06	849	25.63	2744	2989	3016
Standart Sapmaların Ortalaması				7.33			

4.2.1.5.Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Çizelge 4.15’deki tasarruf ve yabani ot algoritma hibrit yaklaşım sonuçları incelendiğinde büyük ölçekli problemlerde standart sapmanın genel olarak daha büyük olduğu gözlenmektedir. Ayrıca problem boyutu büyüdükçe algoritmanın çalışma süresi de artmıştır. Tasarruf algoritmasının ürettiği sonucun yabani ot algoritmasının başlangıç çözümüne eklenmesi yabani ot algoritmasının standart sapmasını oldukça düşürmüştür. Hibrit yaklaşımın yabani ot algoritmasının kararlılığını yükselttiği ve sonuçlarını iyileştirdiği söylenebilir.



Çizelge 4.15. Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
P-n16-k8	450	450.85	462	2.60	49	65	566
P-n19-k2	212	212.03	215	0.30	57	74	162
P-n20-k2	216	219.62	222	2.05	61	79	112
P-n21-k2	211	214.42	224	5.63	66	83	145
P-n22-k2	216	219.12	232	5.46	75	87	145
P-n22-k8	590	590.00	590	0.00	71	81	139
P-n23-k8	529	535.16	537	3.18	75	86	153
P-n40-k5	470	497.02	521	11.18	138	177	298
P-n45-k5	526	532.73	548	4.47	162	217	433
P-n50-k7	573	590.61	604	5.07	193	246	383
P-n50-k8	662	670.47	674	2.08	206	252	399
P-n50-k10	711	715.94	732	7.21	191	243	366
P-n51-k10	754	767.70	783	7.60	186	271	1038
P-n55-k7	583	591.27	619	4.31	226	295	440
P-n55-k8	588	612.11	632	7.54	235	306	628
P-n55-k10	709	722.77	728	4.36	229	295	519
P-n55-k15	958	972.07	990	10.00	197	265	391
P-n60-k10	782	790.12	797	2.59	234	341	480
P-n60-k15	993	1008.13	1020	6.99	240	313	505
P-n65-k10	822	861.20	873	8.01	328	391	602
P-n70-k10	880	906.92	936	10.86	328	446	569
P-n76-k4	611	650.35	680	13.47	409	481	631
P-n76-k5	664	693.00	718	13.13	383	509	1272
P-n101-k4	719	762.99	839	22.99	683	835	1114
Standart Sapmaların Ortalaması				6.71			

4.2.1.6. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Çizelge 4.16'deki tasarruf, genetik ve yabani ot algoritma hibrit yaklaşım sonuçları incelendiğinde problem boyutu büyüdükçe algoritmanın çalışma süresi de artmıştır. Ayrıca büyük ölçekli problemlerde küçük ölçekli problemlere göre standart sapmanın genel olarak daha büyük olduğu gözlenmektedir.

Çizelge 4.16. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
P-n16-k8	450	450.05	455	0.50	371	406	1240
P-n19-k2	212	215.14	219	3.15	396	407	431
P-n20-k2	216	217.98	218	0.20	408	420	497
P-n21-k2	211	213.42	222	4.56	410	420	530
P-n22-k2	216	219.15	230	5.05	432	440	461
P-n22-k8	590	590.00	590	0.00	445	461	559
P-n23-k8	529	531.19	537	3.06	455	474	567
P-n40-k5	468	496.15	513	9.48	693	714	797
P-n45-k5	511	530.17	571	12.90	753	771	869
P-n50-k7	567	588.73	605	6.59	887	940	1276
P-n50-k8	658	668.61	673	3.74	864	893	1110
P-n50-k10	708	718.14	744	8.45	864	885	995
P-n51-k10	751	769.32	795	9.44	871	902	1012
P-n55-k7	585	594.17	609	4.93	931	954	1036
P-n55-k8	593	615.24	667	9.68	922	976	1391
P-n55-k10	712	724.30	737	4.14	932	950	1062
P-n55-k15	958	979.13	990	8.09	937	968	1060
P-n60-k10	769	785.74	811	7.58	978	1004	1117
P-n60-k15	995	1000.75	1001	1.09	976	1002	1144
P-n65-k10	823	844.77	861	9.41	1038	1063	1163
P-n70-k10	873	903.69	946	13.63	1106	1134	1340
P-n76-k4	614	655.26	697	15.61	1161	1210	1317
P-n76-k5	670	696.79	723	10.44	1181	1225	1316
P-n101-k4	730	787.54	861	26.82	1525	1590	1852
Standart Sapmaların Ortalaması				7.44			

4.2.2.CMT Grubu Veri Seti Sonuçları

CMT grubu veri setinden farklı boyutlarda problemler seçilmiştir ve algoritmalar bu problemler üzerinde çalıştırılmıştır. Problemlerin farklı boyutlarda olması, algoritmaların etkinliğini ve başarımını anlamak için faydalı olacaktır.

4.2.2.1.Tasarruf Algoritması Çözüm Sonuçları

Tasarruf algoritmasının bir problem için minimum, ortalama, maksimum maliyetler aynıdır ve standart sapma değeri 0 olmuştur. P grubu verileri için yapılan yorum burada da yapılabilir; algoritma aynı problem için her çalıştırılmasında aynı sonucu üretir denilebilir. Çizelge 4.17’de tasarruf algoritmasının sonuçları verilmiştir. Problem boyutu büyüdükçe algoritmanın çalışma sürelerinde az da olsa yükselme gözlenmektedir.

Çizelge 4.17. Tasarruf Algoritması Çözüm Sonuçları

Adı	Ort. Maliyet	Ort. Hız(ms)
CMT1(50)	731	0
CMT2(75)	1053	1
CMT3(100)	1321	1
CMT4(150)	1382	3
CMT5(199)	1722	7

4.2.2.2.Genetik Algoritma Çözüm Sonuçları

Çizelge 4.18’deki genetik algoritmanın sonuçları incelendiğinde P grubu için yapılan yorumların CMT grubu içinde uygun olduğu görülmektedir. Problem boyutu büyüdükçe algoritmanın standart sapması artmaktadır. Bu durum algoritmanın ürettiği sonucun kararsızlaştığı anlamına gelmektedir. Ayrıca problem boyutu büyüdükçe algoritmanın çalışma süresi de artmıştır. Bu sonuçlara göre

genetik algoritmada problem boyutu büyüdükçe standart sapma ve çalışma süresi artmaktadır yorumu yapılabilir.

Çizelge 4.18. Genetik Algoritma Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
CMT1(50)	535	612.84	740	35.84	1531	1576	2437
CMT2(75)	903	971.31	1097	35.26	2045	2092	2377
CMT3(100)	902	1012.02	1142	56.95	2733	2783	2957
CMT4(150)	1186	1334.25	1489	62.32	3886	3945	4203
CMT5(199)	1599	1745.73	1897	70.77	5327	5403	5671

4.2.2.3.Yabani Ot Algoritması Çözüm Sonuçları

Çizelge 4.19'daki yabani ot algoritmanın sonuçları incelendiğinde büyük ölçekli problemlerde algoritmanın çalışma süresinde artış olmaktadır. Standart sapma oranları büyük ölçekli problemlerde daha büyük olmaktadır.

Çizelge 4.19. Yabani Ot Algoritması Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
CMT1(50)	526	582.72	656	27.35	208	252	385
CMT2(75)	869	962.99	1096	40.44	375	505	835
CMT3(100)	865	980.74	1195	57.70	628	857	1135
CMT4(150)	1146	1301.4	1653	86.75	1622	2220	2965
CMT5(199)	1560	1709.48	1954	94.21	3125	3945	4600

4.2.2.4. Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları

Çizelge 4.20'deki tasarruf ve genetik algoritma hibrit yaklaşım sonuçları incelendiğinde problem boyutu büyüdükçe algoritmanın standart sapmasının ve çalışma süresinin arttığı gözlenmektedir. P grubu için ifade edildiği gibi CMT grubu için de; genel olarak genetik algoritma karakteristiğini göstermeye devam etmiş ancak genetik algoritmanın ürettiği sonuçlar hibrit yaklaşımla iyileşmiş, standart sapma oranları düşmüştür denilebilir.

Çizelge 4.20. Tasarruf ve Genetik Algoritma Hibrit Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
CMT1(50)	534	577.64	613	12.20	1569	1622	2437
CMT2(75)	876	906.84	948	14.52	2123	2173	2420
CMT3(100)	891	935.34	982	16.59	2820	2883	3235
CMT4(150)	1109	1181.2	1261	27.28	4087	4231	4509
CMT5(199)	1432	1498.17	1558	30.63	5644	5757	6407

4.2.2.5. Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Çizelge 4.21'deki tasarruf ve yabani ot algoritma hibrit yaklaşım sonuçları incelendiğinde çalışma süresi ve standart sapma oranları problemin ölçeği büyüdükçe artmaktadır. Tasarruf algoritmasının ürettiği sonucun yabani ot algoritmasının başlangıç çözümüne eklenmesi yabani ot algoritmasının standart sapmasını oldukça düşürmüştür. Hibrit yaklaşımın YOA'sının kararlılığını yükselttiği söylenebilir.

Çizelge 4.21. Tasarruf ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
CMT1(50)	546	567.56	582	5.50	209	292	788
CMT2(75)	877	894.69	931	10.15	418	525	696
CMT3(100)	879	925.96	954	15.84	698	965	1506
CMT4(150)	1105	1149.19	1215	20.21	1485	2439	4090
CMT5(199)	1414	1480.5	1557	29.48	3197	3988	5139

4.2.2.6. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Çizelge 4.22'deki tasarruf, genetik ve yabani ot algoritma hibrit yaklaşım sonuçları incelendiğinde problem boyutu büyüdükçe algoritmanın çalışma süresi ve standart sapması artmıştır.

Çizelge 4.22. Tasarruf, Genetik ve Yabani Ot Algoritması Hibrit Çözüm Sonuçları

Adı	Min. Maliyet	Ort. Maliyet	Maks. Maliyet	Std. Sapma	Min. Hız(ms)	Ort. Hız(ms)	Maks. Hız(ms)
CMT1(50)	533	575.33	613	12.82	808	845	1646
CMT2(75)	881	904.83	939	13.96	1077	1109	1217
CMT3(100)	862	928.23	970	20.30	1429	1494	1732
CMT4(150)	1126	1191.38	1280	27.46	2111	2197	2531
CMT5(199)	1439	1502.67	1581	26.53	2855	3083	3940

4.3. Tartışma

4.3.1. P Grubu Veri Seti Karşılaştırması

Bu bölümde P grubu veri seti üzerinde algoritmaların çalıştırılmasından elde edilen; ortalama maliyet, standart sapma ve ortalama çalışma süreleri

karşılaştırılmıştır. Algoritmaların performansı ve başarımı üzerine tartışmalar yürütülmüştür.

4.3.1.1.Algoritmaların Ortalama Maliyet Karşılaştırması

Şekil 4.1 ve Çizelge 4.23’de algoritmaların her bir problem için ortalama maliyetleri değerlendirilmiştir.

Tekil olarak çalıştırılan tasarruf algoritması, genetik algoritma ve yabani ot algoritması arasından, yabani ot algoritmasının çoğunlukla en iyi sonuçları ürettiği söylenebilir. Kapasite kısıtlı araç rotalama problemine uyarlanan yabani ot metasezgisel algoritmasının son derece rekabetçi ve verimli sonuçlar ürettiği söylenebilir. Sezgisel algoritma olan tasarruf algoritmasının çoğunlukla en kötü başarıma sahip olduğu söylenebilir. Metasezgisel algoritma olan genetik algoritma ve yabani ot algoritmasının başarımı sezgisel olan tasarruf algoritmasından daha iyidir. Genelde metasezgisellerin çözüm sonuçlarının sezgisellerden daha iyi olduğu bilinmektedir. Sonuçlar bu durumu göstermektedir.

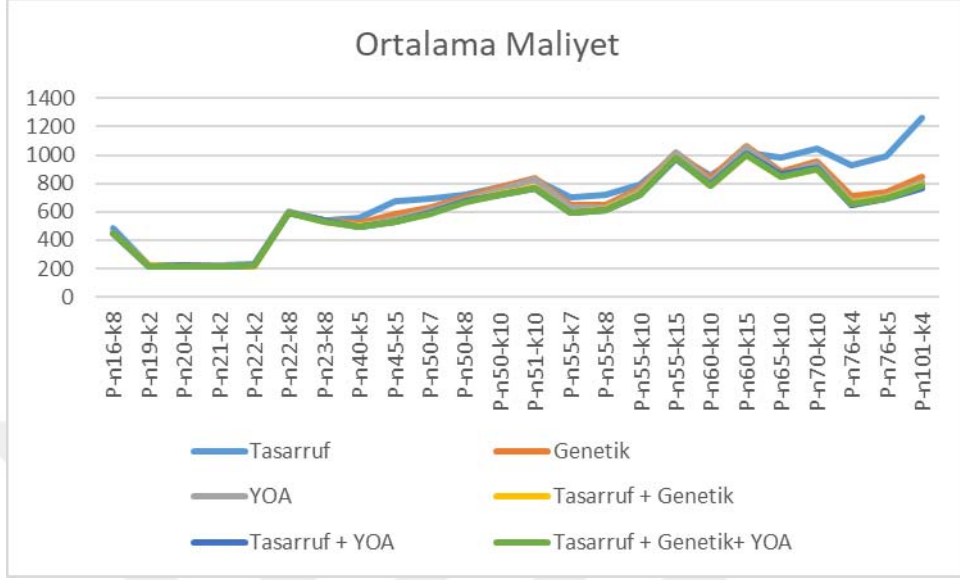
Tüm sonuçlar genel olarak incelendiğinde, hibrit yaklaşımların diğer algoritmalarından daha iyi sonuçlar verdiği görülmektedir. Hibrit yaklaşımlar bilinen iyi çözümlere yakın sonuçlar üretebilmiştir. Tasarruf ve genetik hibrit yaklaşımı diğer hibritlerden genel olarak daha kötü sonuçlar üretmiştir. Hibrit algoritmalar arasından tasarruf, yabani ot hibrit yaklaşımı; tasarruf, genetik ve yabani ot hibrit yaklaşımı sonuçları birbirlerine genel olarak yakındır. 24 problemin 13’ünde tasarruf, genetik ve yabani ot hibriti; tasarruf, yabani ot hibritinden daha iyi sonuç vermiştir.

Genetik algoritma ve yabani ot algoritmasının başlangıç popülasyonlarına tasarruf algoritmasının ürettiği çözümün eklenmesi algoritmaların çözümlerini iyileştirmiş, ürettiği sonuçların daha iyi olmasını sağlamıştır. Buradan hareketle metasezgisel algoritmaların başlangıç popülasyonlarına başka bir algoritmanın

çözümünün eklenmesinin metasezgisel algoritmalarının başarımını arttırdığı söylenebilir.

Çizelge 4.23. Algoritmaların Ortalama Maliyet Karşılaştırması

Adı	BKS	Tasarruf	Genetik	YOA	Tasarruf + Genetik	Tasarruf + YOA	Tasarruf + Genetik + YOA
P-n16-k8	450	482	450.22	450.42	450.55	450.85	450.05
P-n19-k2	212	226	224.56	223.13	218.75	212.03	215.14
P-n20-k2	216	222	226.12	224.70	217.98	219.62	217.98
P-n21-k2	211	224	221.32	218.51	213.75	214.42	213.42
P-n22-k2	216	232	226.56	224.07	217.93	219.12	219.15
P-n22-k8	590	590	595.71	598.83	590.00	590.00	590.00
P-n23-k8	529	537	536.74	538.64	530.95	535.16	531.19
P-n40-k5	458	561	521.24	503.19	499.60	497.02	496.15
P-n45-k5	510	676	583.07	551.67	533.46	532.73	530.17
P-n50-k7	554	693	626.53	610.10	590.43	590.61	588.73
P-n50-k8	629	716	703.17	683.58	670.57	670.47	668.61
P-n50-k10	696	775	774.72	758.27	726.44	715.94	718.14
P-n51-k10	741	825	833.27	824.22	771.52	767.70	769.32
P-n55-k7	568	702	647.10	629.66	596.20	591.27	594.17
P-n55-k8	588	722	651.41	630.02	617.70	612.11	615.24
P-n55-k10	694	792	763.49	747.44	725.48	722.77	724.30
P-n55-k15	945	1000	1020.65	1018.94	983.98	972.07	979.13
P-n60-k10	744	859	833.91	811.29	786.05	790.12	785.74
P-n60-k15	968	1020	1060.69	1058.29	1000.29	1008.13	1000.75
P-n65-k10	792	982	884.73	865.20	846.47	861.20	844.77
P-n70-k10	827	1047	959.49	939.20	905.23	906.92	903.69
P-n76-k4	593	930	706.76	673.43	662.95	650.35	655.26
P-n76-k5	627	989	739.00	709.47	699.72	693.00	696.79
P-n101-k4	681	1266	846.84	811.81	794.06	762.99	787.54



Şekil 4.1. .Algoritmaların Ortalama Maliyet Karşılaştırması

4.3.1.2. Algoritmaların Standart Sapma Karşılaştırması

Şekil 4.2 ve Çizelge 4.24'de algoritmaların her bir problem için 100 defa çalıştırılması sonucunda hesaplanan standart sapma değerleri değerlendirilmiştir. Tasarruf algoritmasının hesaplaması tasarruf değerlerine dayandığından ve bir problem için bu değerler farklı koşullarda bile değişmemesinden dolayı standart sapma değerleri her problem için 0 olarak bulunmuştur. Bundan dolayı standart sapma değerleri kıyaslaması diğer algoritmalar arasında yapılacaktır.

Genetik ve yabani ot algoritma karşılaştırmasında yabani ot algoritmasının daha düşük standart sapmalara sahip olduğu görülmektedir. Kapasite kısıtlı araç rotalama problemine uyarlanan yabani ot algoritmasının genetik algoritmadan daha kararlı sonuçlar ürettiği söylenebilir.

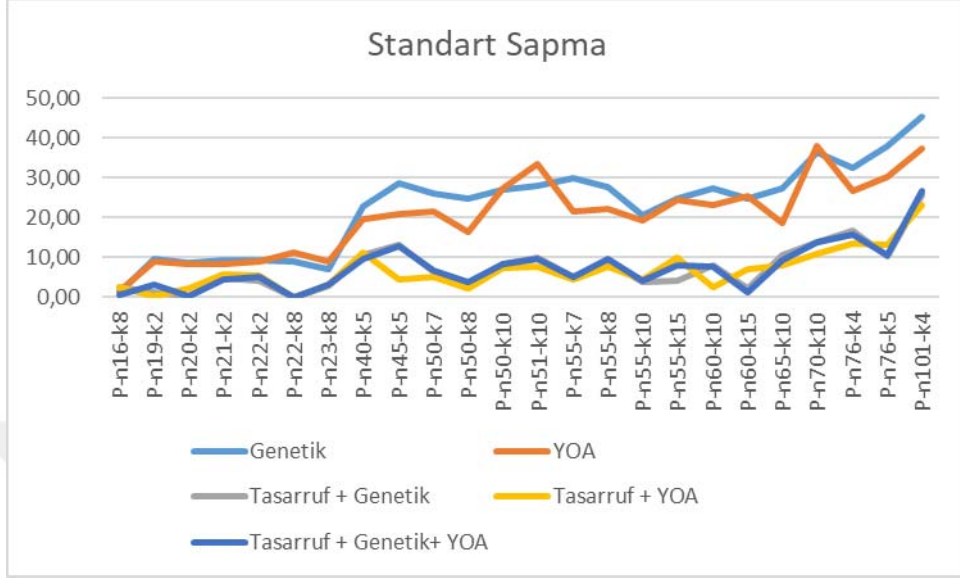
Sonuçlar incelendiğinde hibrit yaklaşımların standart sapmaları daha düşüktür. Hibrit yaklaşımların diğer algoritmalarından daha kararlı olduğu söylenebilir. Hibrit yaklaşımların standart sapmalarının ortalamaları birbirlerine yakındır. Hibrit yaklaşımlar arasında en düşük standart sapma sonucunu tasarruf ve

yabani ot algoritmasından oluşan yaklaşım vermiştir. Standart sapmaların ortalamasına bakılacak olursa en düşük değer tasarruf ve yabani ot algoritmasından oluşan hibrit yaklaşıma aittir. Sonuç olarak tasarruf ve yabani ot algoritmasının ele alınan diğer algoritmalarından daha kararlı olduğu söylenebilir.

Genetik algoritma ve yabani ot algoritmasının başlangıç popülasyonlarına tasarruf algoritmasının ürettiği çözümün eklenmesi algoritmaların standart sapmalarını düşürmüştür. Buradan hareketle metasezgisel algoritmaların başlangıç popülasyonlarına başka bir algoritmanın çözümünün eklenmesinin metasezgisel algoritmalarının kararlılığını da arttırdığı söylenebilir. Yani standart sapması yüksek metasezgisel algoritmaların başlangıç çözümleri başka bir algoritma ile beslendiğinde standart sapmasında düşme olduğu ve algoritmanın kararlılığını arttırdığı yorumu yapılabilir.

Çizelge 4.24. Algoritmaların Standart Sapma Karşılaştırması

Adı	Genetik	YOA	Tasarruf + Genetik	Tasarruf + YOA	Tasarruf + Genetik + YOA
P-n16-k8	1.21	1.63	2.40	2.60	0.50
P-n19-k2	9.59	9.05	1.13	0.30	3.15
P-n20-k2	8.61	8.41	0.20	2.05	0.20
P-n21-k2	9.41	8.19	4.76	5.63	4.56
P-n22-k2	9.26	8.92	4.14	5.46	5.05
P-n22-k8	8.94	11.17	0.00	0.00	0.00
P-n23-k8	7.09	8.90	2.87	3.18	3.06
P-n40-k5	22.75	19.58	10.48	11.18	9.48
P-n45-k5	28.73	20.90	13.03	4.47	12.90
P-n50-k7	25.92	21.39	6.42	5.07	6.59
P-n50-k8	24.72	16.39	3.20	2.08	3.74
P-n50-k10	27.08	27.26	8.36	7.21	8.45
P-n51-k10	28.02	33.35	9.76	7.60	9.44
P-n55-k7	29.94	21.37	5.16	4.31	4.93
P-n55-k8	27.78	22.05	9.04	7.54	9.68
P-n55-k10	20.51	19.16	3.89	4.36	4.14
P-n55-k15	24.79	24.57	4.11	10.00	8.09
P-n60-k10	27.37	23.09	7.81	2.59	7.58
P-n60-k15	24.69	25.32	2.16	6.99	1.09
P-n65-k10	27.20	18.77	10.51	8.01	9.41
P-n70-k10	36.48	38.02	13.71	10.86	13.63
P-n76-k4	32.45	26.56	16.76	13.47	15.61
P-n76-k5	37.94	30.36	10.35	13.13	10.44
P-n101-k4	45.37	37.48	25.63	22.99	26.82
Standart Sapmaların Ortalaması	22.74	20.08	7.33	6.71	7.44



Şekil 4.2. Algoritmaların Standart Sapma Karşılaştırması

4.3.1.3. Algoritmaların Ortalama Hız Karşılaştırması

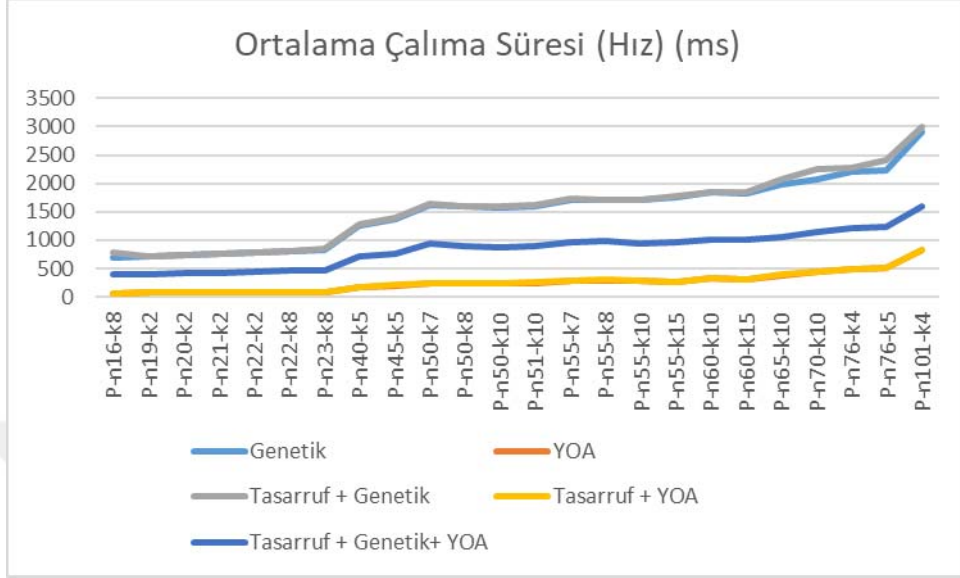
Tasarruf algoritmasının ortalama çalışma süresi P grubu problemleri için yaklaşık 0'dır. Bundan dolayı ortalama çalışma süresi karşılaştırması diğer algoritmalar arasında yapılacaktır. Tasarruf algoritması sezgisel; genetik algoritma ve yabani ot algoritması metasezgisel algoritmalarıdır. Metasezgisellerin çalışma süreleri genellikle sezgisellerden daha yüksek olmaktadır. Sonuçlar da bu duruma kanıt oluşturmaktadır.

Şekil 4.3 ve Çizelge 4.25'de algoritmaların her bir problem için 100 defa çalıştırılması sonucunda hesaplanan ortalama çalışma süreleri (hız değerleri) değerlendirilmiştir. Algoritmalar arasında çalışma süresi en düşük olan yani en hızlısı yabani ot algoritmasıdır. Tasarruf ve yabani ot algoritmasından oluşan hibrit yaklaşımın sonuçları da tekil çalıştırılan yabani ot algoritmasına yakındır. Kapasite kısıtlı araç rotalama problemine uyarlanan metasezgisel bir algoritma olan yabani ot algoritmasının son derece hızlı olduğu söylenebilir.

Sonuçlara göre algoritmaların çalışma süresinin problem boyutuna bağlı olduğu açıkça görülmektedir. Problem boyutu büyüdükçe çalışma süreleri doğru orantılı olarak artmıştır.

Çizelge 4.25. Algoritmaların Ortalama Hız Karşılaştırması

Adı	Genetik	YOA	Tasarruf + Genetik	Tasarruf + YOA	Tasarruf + Genetik + YOA
P-n16-k8	704	62	775	65	406
P-n19-k2	710	73	712	74	407
P-n20-k2	736	80	741	79	420
P-n21-k2	752	83	755	83	420
P-n22-k2	784	87	787	87	440
P-n22-k8	799	83	816	81	461
P-n23-k8	835	87	849	86	474
P-n40-k5	1264	174	1274	177	714
P-n45-k5	1377	206	1395	217	771
P-n50-k7	1616	241	1631	246	940
P-n50-k8	1588	246	1597	252	893
P-n50-k10	1584	240	1595	243	885
P-n51-k10	1598	245	1613	271	902
P-n55-k7	1713	283	1722	295	954
P-n55-k8	1701	284	1719	306	976
P-n55-k10	1704	287	1721	295	950
P-n55-k15	1747	259	1784	265	968
P-n60-k10	1835	332	1839	341	1004
P-n60-k15	1824	310	1840	313	1002
P-n65-k10	1971	385	2062	391	1063
P-n70-k10	2075	437	2248	446	1134
P-n76-k4	2198	480	2265	481	1210
P-n76-k5	2235	503	2421	509	1225
P-n101-k4	2904	828	2989	835	1590



Şekil 4.3. .Algoritmaların Ortalama Hız Karşılaştırması

4.3.2.CMT Grubu Veri Seti Karşılaştırması

Bu bölümde CMT grubu veri seti üzerinde algoritmaların çalıştırılmasından elde edilen; ortalama maliyet, standart sapma ve ortalama çalışma süreleri karşılaştırılmıştır. Algoritmaların performansı ve başarımı üzerine tartışmalar yürütülmüştür.

4.3.2.1.Algoritmaların Ortalama Maliyet Karşılaştırması

Şekil 4.4 ve Çizelge 4.27’de algoritmaların CMT grubu her bir problem için 100 defa çalıştırılması sonucunda hesaplanan ortalama maliyetleri değerlendirilmiştir. Sonuçlar incelendiğinde tasarruf algoritması çoğu problemde en kötü başarıma sahiptir.

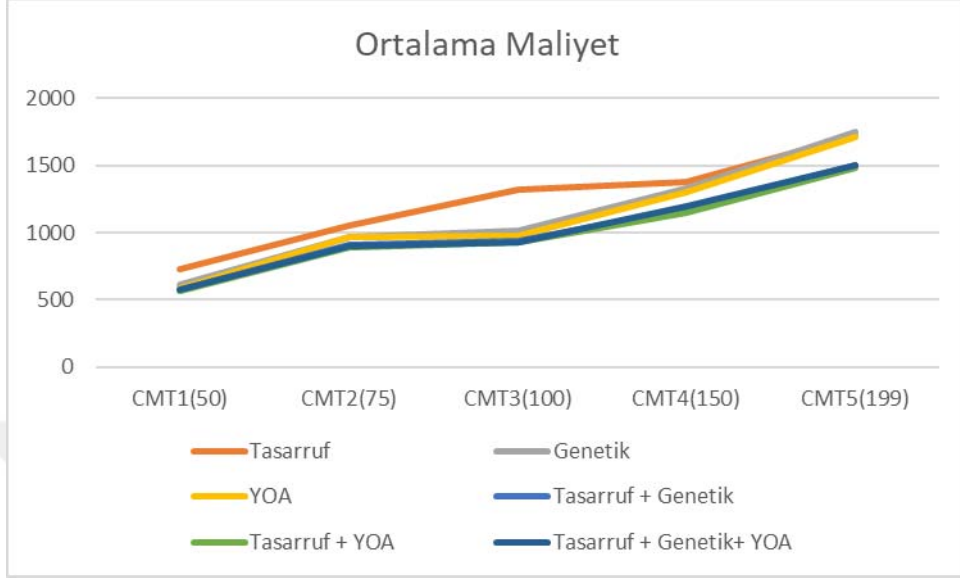
Bütün problemler için, hibrit yaklaşımlar daha iyi sonuçlar üretmiştir. Hibrit yaklaşımların başarımı birbirine yakın olmasına rağmen tasarruf ve yabancı ot algoritmasından oluşan hibrit yaklaşımın en iyi sonuçları ürettiği söylenebilir. Tekil

çalıştırılan genetik algoritma ve yabani ot algoritmasının tasarruf algoritması ile hibritlenmesi algoritmaların başarımını yükseltmiştir.

Genetik algoritma ve yabani ot algoritması karşılaştırmasında yabani ot algoritması daha iyi sonuçlar vermiştir. Kapasite kısıtlı araç rotalama problemine uyarlanan metaevrimsel bir algoritma olan yabani ot algoritmasının CMT grubu verileri için de son derece rekabetçi ve verimli sonuçlar ürettiği söylenebilir.

Çizelge 4.26. Algoritmaların Ortalama Maliyet Karşılaştırması

Adı	BKS	Tasarruf	Genetik	YOA	Tasarruf + Genetik	Tasarruf + YOA	Tasarruf + Genetik + YOA
CMT1(50)	524.61	731	612.84	582.72	577.64	567.56	575.33
CMT2(75)	835.26	1053	971.31	962.99	906.84	894.69	904.83
CMT3(100)	826.14	1321	1012.02	980.74	935.34	925.96	928.23
CMT4(150)	1028.42	1382	1334.25	1301.4	1181.2	1149.19	1191.38
CMT5(199)	1291.29	1722	1745.73	1709.48	1498.17	1480.5	1502.67



Şekil 4.4. Algoritmaların Ortalama Maliyet Karşılaştırması

4.3.2.2. Algoritmaların Standart Sapma Karşılaştırması

Şekil 4.5 ve Çizelge 4.28’de algoritmaların CMT grubu her bir problem için 100 defa çalıştırılması sonucunda hesaplanan standart sapma değerleri değerlendirilmiştir. Tasarruf algoritmasının standart sapma değerleri her problem için 0 olmuştur. Standart sapma değerleri kıyaslaması diğer algoritmalar arasında yapılacaktır.

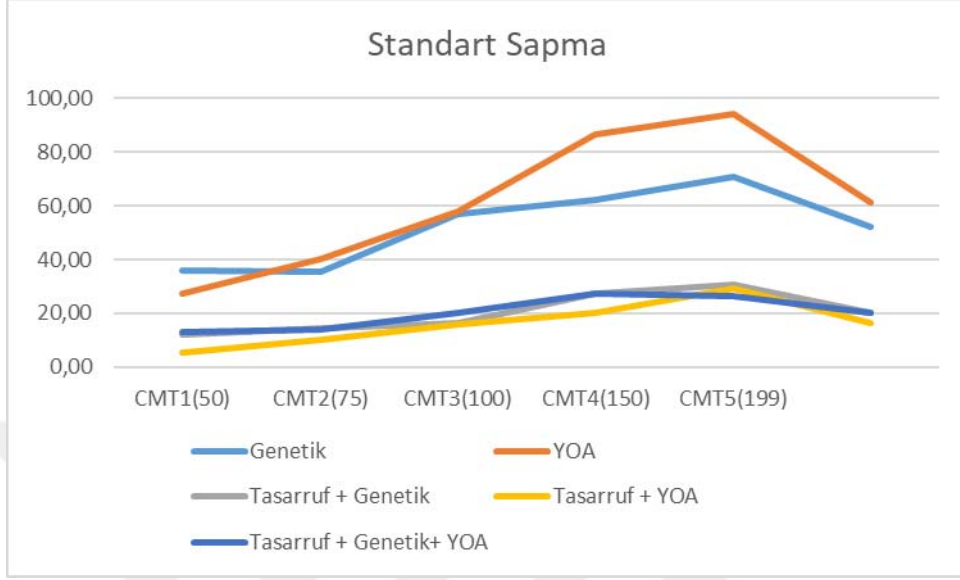
Tekil çalıştırılan genetik algoritma ve yabani ot algoritmasının standart sapması hibrit yaklaşımlara göre yüksektir. Sonuçlar incelendiğinde hibrit yaklaşımların standart sapmalarının daha iyi olduğu söylenebilir. Yani standart sapması yüksek metasezgisel algoritmaların başlangıç çözümleri başka bir algoritma ile beslendiğinde standart sapmasında düşme olduğu ve algoritmanın kararlılığını arttığı yorumu yapılabilir.

Tasarruf ve yabani ot algoritmasından oluşan hibrit yaklaşımı 5 adet CMT probleminin 4 tanesinde en düşük standart sapma oluşmuştur. Standart sapmaların ortalamasına bakılacak olursa en düşük değer tasarruf ve yabani ot algoritmasından

oluşan hibrit yaklaşıma aittir. Sonuç olarak tasarruf ve yabancı ot algoritmasının ele alınan diğer algoritmalarından daha kararlı olduğu söylenebilir.

Çizelge 4.27. Algoritmaların Standart Sapma Karşılaştırması

Adı	Genetik	YOA	Tasarruf + Genetik	Tasarruf + YOA	Tasarruf + Genetik + YOA
CMT1(50)	35.84	27.35	12.20	5.50	12.82
CMT2(75)	35.26	40.44	14.52	10.15	13.96
CMT3(100)	56.95	57.70	16.59	15.84	20.30
CMT4(150)	62.32	86.75	27.28	20.21	27.46
CMT5(199)	70.77	94.21	30.63	29.48	26.53
Standart Sapmaların Ortalaması	52.23	61.29	20.24	16.24	20.21



Şekil 4.5. .Algoritmaların Standart Sapma Karşılaştırması

4.3.2.3. Algoritmaların Ortalama Hız Karşılaştırması

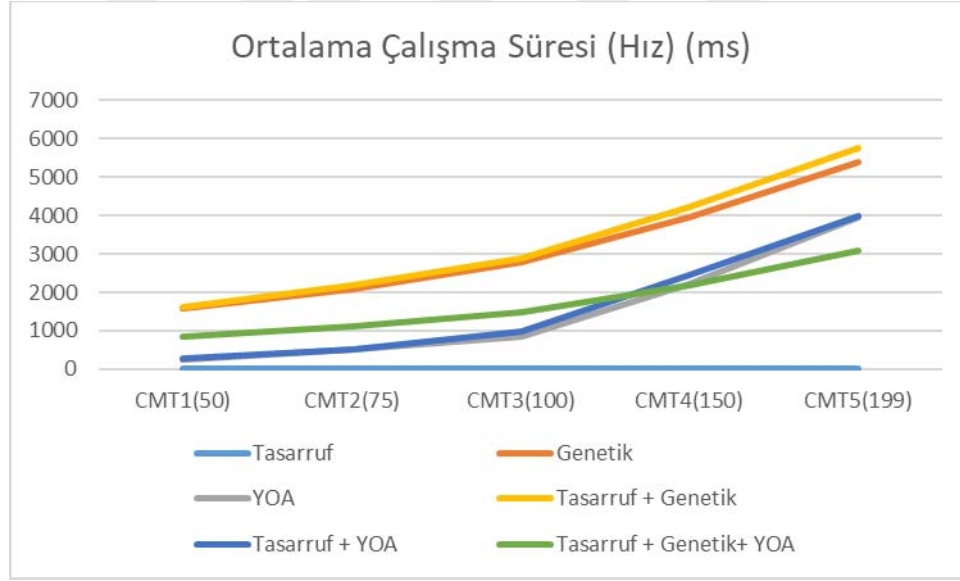
Şekil 4.6 ve Çizelge 4.29’da algoritmaların CMT grubu her bir problem için 100 defa çalıştırılması sonucunda hesaplanan çalışma süreleri (hız değerleri) değerlendirilmiştir. Algoritmalar arasında çalışma süresi en düşük olan yani en hızlısı tasarruf algoritmasıdır.

Tasarruf algoritması dışında kalan algoritmalar arasında en hızlısının seçilmesi pek mümkün görünmemektedir. Orta ölçekli problemler için yabani ot algoritmasının çalışma süresi düşüktür ancak büyük boyutlu problemlerde aynı performansı gösterememiştir. CMT grubu içindeki 199 müşterili problemde en iyi çalışma süresi tasarruf, genetik ve yabani ot algoritmasından oluşan hibrit yaklaşıma aittir.

Genetik algoritma ve yabani ot algoritmasının ayrı ayrı tasarruf algoritması ile hibritlenmesi çalışma sürelerini fazla etkilemiştir. Tasarruf algoritmasının çalışma süresi düşük olduğu için hibritleme işleminde kullanılması çalışma süresini uzatmamaktadır yorumu yapılabilir.

Çizelge 4.28. Algoritmaların Ortalama Hız Karşılaştırması

Adı	Tasarruf	Genetik	YOA	Tasarruf + Genetik	Tasarruf + YOA	Tasarruf + Genetik+ YOA
CMT1(50)	0	1576	252	1622	292	845
CMT2(75)	1	2092	505	2173	525	1109
CMT3(100)	1	2783	857	2883	965	1494
CMT4(150)	3	3945	2220	4231	2439	2197
CMT5(199)	0	5403	3945	5757	3988	3083



Şekil 4.6. Algoritmaların Ortalama Hız Karşılaştırması

4.3.3. Farklı Çalışmalar ile Kıyaslamalar

Karagül ve arkadaşları (2016) yaptıkları çalışmada; kapasite kısıtlı araç rotalama probleminin çözümü için Newton'un çekim yasasını temel alan bir algoritma önermiş ve bu algoritmayı genetik algoritmanın başlangıç

popülasyonunu oluşturmak için kullanmıştır. Bu yöntem genetik algoritmanın başarımını iyileştirmiştir. Önerdikleri yaklaşımı P grubu veri seti problemlerine uygulamış ve çözümleri değerlendirmişlerdir.

Ahmed ve Sun (2018) yaptıkları çalışmada kapasite kısıtlı araç rotalama probleminin çözümü için iki katlı yerel arama tabanlı parçacık sürü optimizasyonunu (Bilayer Local Search-based Particle Swarm Optimization) önermişlerdir. Önerilen yaklaşımı literatürden alınan P grubu veri seti problemleri üzerinde uygulamışlardır.

Kapasite kısıtlı araç rotalama problemi üzerinde yapılan çalışmalar, bu çalışmada ele alınan algoritmalarından elde edilen çözüm sonuçlarının durumunu değerlendirmek için bilgi vericidir.

Ekler bölümünde sunulan Çizelge 4.29'da Karagül, arkadaşları (2016) ve Ahmed, Sun (2018) tarafından yapılan çalışmaların sonuçları ile birlikte çalışma kapsamında ele alınan tasarruf, yabani ot hibriti ve tasarruf, genetik, yabani ot hibritinin sonuçları verilmiştir. Algoritmaların probleme ait en iyi sonucu bulması durumu çözüm değerinin koyulaştırılması ile gösterilmiştir. Değişim olarak ifade edilen kavramın formülasyonu (4.1) nolu denklemde sunulmuştur.

$$\text{Değişim} = \frac{(\text{Algoritma Maliyeti} - \text{BKS})}{\text{BKS}} * 100 \quad (4.1)$$

Değişim olarak ifade edilen kavram problemin bilinen en iyi çözüme (BKS, Best Known Solution) göre sapmaların ortalamasını ifade etmektedir. Değişim değeri küçüldükçe bilinen en iyi çözüme yaklaşılr. Bundan dolayı değişim kavramı, algoritmanın çözümünün kalitesini göstermektedir.

Çizelge 4.29'da önerilen en iyi hibritlerin önceki çalışmada yer alan algoritma sonuçlarıyla kıyaslanması sunulmuştur. Sonuçlara göre bu çalışma kapsamında önerilen tasarruf, yabani ot hibrit yaklaşımı ve tasarruf, genetik, yabani

ot hibrit yaklaşımı Karagül ve arkadaşlarının önerdiği yaklaşımdan daha iyi sonuç vermiştir. Çözüm kalitesi olarak yapılan karşılaştırmada en iyi sonucu Ahmet ve Sun'ın önerdiği yaklaşım sağlamıştır. Ancak çalışma tekrarı bilinmediğinden algoritmaların oluşturduğu maliyetlerin standart sapmaları kıyaslanamamıştır.



Çizelge 4.29. P Grubu Veri Setlerinin Çözüm Sonuçlarının Farklı Çalışmalar ile Karşılaştırılması (Karagül ve arkadaşları, 2016; Ahmed ve Sun,

Adı	BKS	Karagül Maliyet	Karagül Değişim (%)	Ahmed ve Sun Maliyet	Ahmed ve Sun Değişim (%)	Ahmed ve Sun Hız (sn)	Tasarruf + YOA Maliyet	Tasarruf + YOA Değişim (%)	Tasarruf + YOA Hız (ms)	Tasarruf + Genetik + YOA Maliyet	Tasarruf + Genetik + YOA Değişim (%)	Tasarruf + Genetik + YOA Hız (ms)
P-n16-k8	450	451.3	0.30	450	0.00	0.11	450	0.00	65	450	0.00	406
P-n19-k2	212	212.7	0.31	212	0.00	0.1	212	0.00	74	212	0.00	407
P-n20-k2	216	217.4	0.66	216	0.00	0.35	216	0.00	79	216	0.00	420
P-n21-k2	211	219.1	3.83	211	0.00	0.32	211	0.00	83	211	0.00	420
P-n22-k2	216	217.9	0.87	216	0.00	0.71	216	0.00	87	216	0.00	440
P-n22-k8	590	595.8	0.98	603	2.20	0.83	590	0.00	81	590	0.00	461
P-n23-k8	529	531.2	0.41	529	0.00	1.02	529	0.00	86	529	0.00	474
P-n40-k5	458	475.3	3.78	458	0.00	1.33	470	2.62	177	468	2.18	714
P-n45-k5	510	542.1	6.30	510	0.00	1.45	526	3.14	217	511	0.20	771
P-n50-k7	554	593.2	7.07	554	0.00	1.48	573	3.43	246	567	2.35	940
P-n50-k8	629	659.5	4.85	631	0.32	1.05	662	5.25	252	658	4.61	893
P-n50-k10	696	708.9	1.85	696	0.00	2.23	711	2.16	243	708	1.72	885

P-n51-k10	741	767.4	3.56	741	0.00	3.38	754	1.75	271	751	1.35	902
P-n55-k7	568	629.5	10.82	568	0.00	4.32	583	2.64	295	585	2.99	954
P-n55-k8	588	619.6	5.38				588	0.00	306	593	0.85	976
P-n55-k10	694	729.7	5.14	694	0.00	4.94	709	2.16	295	712	2.59	950
P-n55-k15	945	985.6	4.30	989	4.66	4.29	958	1.38	265	958	1.38	968
P-n60-k10	744	820.5	10.28	744	0.00	5.83	782	5.11	341	769	3.36	1004
P-n60-k15	968	1033	6.71	968	0.00	5.37	993	2.58	313	995	2.79	1002
P-n65-k10	792	864.3	9.12	792	0.00	6.44	822	3.79	391	823	3.91	1063
P-n70-k10	827	905.4	9.48	833	0.73	9.24	880	6.41	446	873	5.56	1134
P-n76-k4	593	711.6	20.00	598	0.84	16.11	611	3.04	481	614	3.54	1210
P-n76-k5	627	752	19.94	636	1.44	15.85	664	5.90	509	670	6.86	1225
P-n101-k4	681	838	23.05	692	1.62	20.17	719	5.58	835	730	7.20	1590
Ortalama Değişim			6.63		0.51			2.37			2.23	

Karagül ve arkadaşları (2016) küçük problemler için bilinen en iyi çözümlere yaklaşmışlardır ancak hiçbir çözüm bilinen en iyi çözüme ulaşmamıştır. Ahmet ve Sun (2018) önerdikleri yaklaşım ile 23 adet problemin 16 tanesinde bilinen en iyi sonuçlara ulaşmışlardır. Tasarruf ve yabani ot algoritmasından oluşan hibrit yaklaşım 24 problemin 8 tanesinde; tasarruf genetik ve yabani ot algoritmasından oluşan hibrit yaklaşım 24 problemin 7 tanesinde en iyi sonuçlara ulaşmıştır.

Ahmet ve Sun (2018) önerdikleri yaklaşım ile bu çalışma kapsamında ele alınan hibrit yaklaşımların çalışma süreleri karşılaştırıldığında bu çalışma kapsamında önerilen hibrit yaklaşımların çok daha hızlı olduğu görülmektedir. 100 müşterili bir problem için Ahmet ve Sun (2018) önerdikleri yaklaşımın çalışma süresi 20.17 saniye iken; tasarruf, yabani ot algoritması hibritinin 835 milisaniye, tasarruf, genetik ve yabani ot algoritması hibritinin 1590 milisaniye olduğu görülmektedir. Ahmet ve Sun (2018) önerdikleri yaklaşımın çalışma süreleri incelendiğinde büyük ölçekli problemler için ciddi oranda artış gösterdiği fark edilmektedir. Karagül ve arkadaşları (2016) algoritmanın çalışma süreleri bulunmadığı için kıyaslama yapılamamıştır.

Sonuç olarak bu çalışma kapsamında önerilen hibrit yaklaşımların çözüm kalitelerini koruyorken çok hızlı şekilde bilinen en iyi çözümlere yaklaştığı söylenebilir.

4.3.4.Genel Değerlendirme

P grubu ve CMT grubu için genel bir değerlendirme yapmak gerekirse; tasarruf algoritmasının standart sapması ve çalışma süresi düşüktür ancak diğer algoritmalara kıyasla kötü sonuçlar üretmektedir.

Yabani ot algoritması; tasarruf algoritmasından ve genetik algoritmadan daha iyi çözümler üretmiştir. Ayrıca yabani ot algoritması genetik algoritmadan daha hızlı çalışmıştır. Kapasite kısıtlı araç rotalama problemine uyarlanan

metasezgisel bir algoritma olan yabani ot algoritmasının son derece rekabetçi ve verimli sonuçlar ürettiği söylenebilir. Yabani ot algoritmasının çalışma süresi genel olarak bu çalışmada ele alınan diğer algoritmalara kıyasla düşüktür. Ancak büyük boyutlu problemlerde bu avantajını kaybedebilir. Bu durumun netleştirilmesi için daha fazla büyük ölçekli problem ile teste ihtiyaç vardır.

Genetik algoritma ve yabani ot algoritmasının başlangıç popülasyonlarına tasarruf algoritmasının ürettiği çözümün eklenmesi algoritmaların çözümlerini iyileştirmiş, ürettiği sonuçların daha iyi olmasını sağlamıştır. Ayrıca standart sapmayı düşürmüş, algoritmanın daha kararlı olmasını sağlamıştır. Tasarruf algoritmasının çalışma süresi oldukça düşük olduğu için hibritleme işleminde algoritmaların çalışma sürelerini etkilememiştir. Sonuç olarak hibritleme işlemi algoritmaların başarımını yükseltmiştir. Buradan hareketle metasezgisel algoritmaların başlangıç popülasyonlarına başka bir algoritmanın çözümünün eklenmesinin metasezgisel algoritmalarının başarımını arttırdığı söylenebilir.

Denenen hibrit yaklaşımlar arasından; tasarruf, yabani ot algoritması hibriti ve tasarruf, genetik, yabani ot algoritması hibriti iyi sonuçlar üretmiştir. Bu yaklaşımların çalışma süreleri de diğer algoritmalara kıyasla daha kısadır. Bu iki hibrit yaklaşım arasında tasarruf, yabani ot algoritması hibritinin genel olarak çalışma süresinin düşük olmasından kaynaklı daha tercih edilebilir bir yaklaşım olduğu düşünülmektedir.

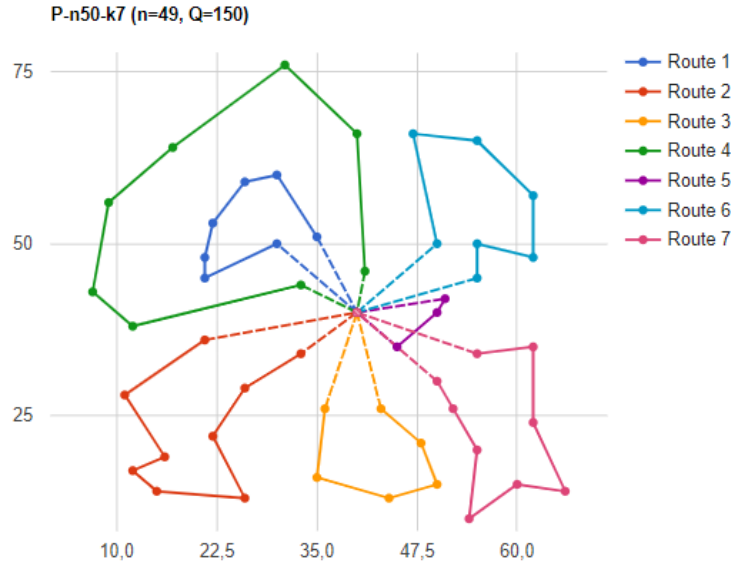
4.4.Dinamik ARP Çözümü

Bu bölümde dinamik araç rotalama probleminin çözümüne yer verilmiştir. Algoritmaların dinamik araç rotalama problemine uygulanması için P grubu veri setinden P-n50-k7 isimli veri seti seçilmiştir. Bu problem, tek depolu, 7 araçlı, 49 müşteri kapasite kısıtlı araç rotalama problemidir. Talep iptali, yeni talep gelmesi ve her iki durumun aynı anda oluşması gibi dinamik durumlarda problemin yeniden çözülmesi aşamasında araçlar sahip olduğu yük miktarı kadar dağıtım

yapabilirler. Bundan dolayı dinamik problemde araç kapasiteleri, aracın sahip olduğu yük miktarı olarak kabul edilir. Araçların anlık olarak sahip olacakları yük miktarları farklı olabileceği için, problem heterojen araç filosuna sahip bir probleme dönüşür.

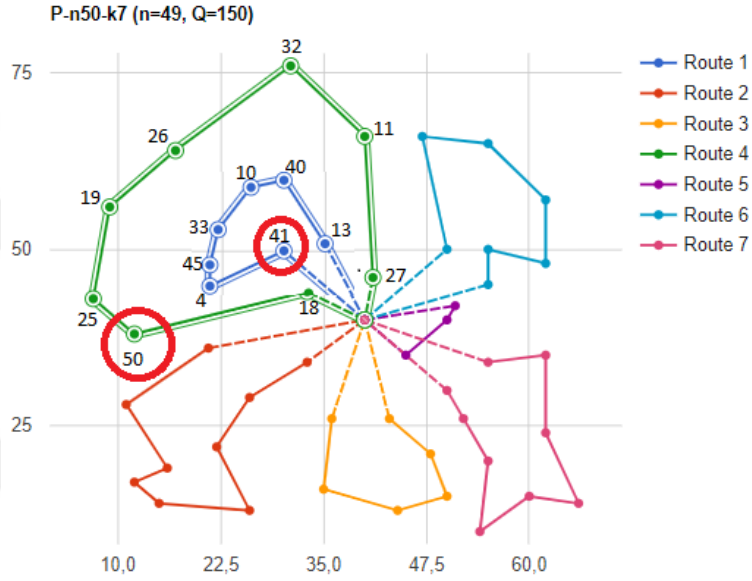
Dinamik araç rotalama problemi çözümünde belirli varsayımlarla dinamiklik ele alınmıştır. Bu varsayımlar “1.5. Çalışmanın Kapsamı” başlığı altında dinamik ARP için kısıtlamalar açıklamasıyla verilmiştir. Bu bölümde belirlenen varsayımlar ve kısıtlar altında problem çözülmüştür ve tartışması yürütülmüştür. Dinamik problem çözümünde hibrit yaklaşımlar kullanılmıştır ve bulunan en iyi çözüm çıktı olarak verilmiştir.

P-n50-k7 isimli klasik problemin çözüm sonucunda oluşan rotalar Şekil 4.7’de görülmektedir.



Şekil 4.7. Başlangıç Durumu

Şekil 4.8’de verilen şekilde 1. rota ve 4. rota dışındaki rotalar tamamlanmıştır. 1. rotada araç 41 nolu müşteride, 4. rotada araç 50 nolu müşteride bulunmaktadır. Şekil 4.8’de rotalar, müşteriler ve araç gösterilmiştir. Tamamlanmayan rotalar daha koyu şekilde, diğer rotalar normal olarak gösterilmiştir. Araçların bulunduğu noktalar kırmızı daire içine alınmıştır.



Şekil 4.8. Rota Güncellemeden Önceki Durum

4.4.1.Müşteri Talep İptalleri

Başlangıç durumu Şekil 4.7’de ve anlık durumu Şekil 4.8’de verilen problem için 4. rotadan; 25, 19, 26, 32 ve 11 nolu müşterilerin talebi iptal ettiğini varsayalım. 4. rota üzerinde bulunan aracın rotasına devam etmesi ve rotaların güncellenerek yeniden rotalama yapılması durumlarında oluşacak rotalar ve maliyetleri ortaya konmuştur. Bu duruma ait sonuçlar aşağıdadır. Hibrit yaklaşımların ürettiği çözüm ve rotaya devam edilmesi durumunda oluşacak çözümün rotaları ve rotanın maliyetleri verilmiştir.

Başlangıç durumu Şekil 4.7’de ve anlık durumu Şekil 4.8’de verilen problem yeni müşteri talebinde de kullanılacaktır. 1. rotada araç 41 nolu müşteride, 4. rotada araç 50 nolu müşteride bulunmaktadır. Bu anda 50 nolu düğümün bulunduğu noktada 3 farklı müşteri (1000, 1001, 1002) talebi oluştuğunu varsayalım. Yeni eklenen 3 müşterinin talepleri toplamının; 4. rotada bulunan 27 nolu müşteri talebinden az olacak şekilde belirlenmiştir. Yapılan hesaplamalarda yeni gelen taleplere depodan araç atamak yerine 4. rotada 50 nolu müşteride bulunan aracın, yeni gelen müşteri taleplerini karşılaması ve talebi karşılanmayan 27 nolu müşteriye depodan araç atanması beklenmektedir.

Yeni gelen siparişlerin depodan kalkacak araçla karşılanması durumunda; “41-4-45-33-10-40-13-1”, “50-25-19-26-32-11-27-1”, “1-100-1001-1002-1” şeklinde 3 adet rota oluşacaktır. Bu rotaların maliyeti 197.78’dir.

Bu problemin hibrit yaklaşımlarla çözülmesi sonucunda; “41-4-45-33-10-40-13-1”, “50-1001-1001-1000-25-19-26-32-11-1”, “1-27-1” şeklinde 3 adet rota bulunmuştur. Bu rotaların maliyeti 153.70’dür.

Sonuçların maliyetine bakıldığında; yeni gelen siparişlerin depodan kalkacak araçla karşılanması durumunda oluşan maliyet 197.78, hibrit yaklaşımlarla çözümün maliyeti 153.70’dür. Hibrit yaklaşımlar; yeni gelen siparişlerin depodan kalkacak araçla karşılanması durumundan daha iyi sonuç üretebilmiştir. Hibrit yaklaşımların yeni talep gelmesi durumuna hızlı ve daha iyi sonuç ürettiği söylenebilir.

4.4.3.Yeni Müşteri Talepleri ve Talep İptalleri

4.4.1 bölümünde verilen müşteri talep iptalleri ve 4.4.2 de verilen yeni müşteri talepleri senaryoları birleştirilerek çözülmüştür. Bu duruma ait sonuçlar verilmiştir.

Müşteri iptallerinin olduğu rotadaki aracın rotasına devam etmesi ve yeni gelen müşterilere depodan araç atanması durumunda oluşacak sonuç; “41-4-45-33-

10-40-13-1”, “ 50-27-1”, “1-100-1001-1002-1” şeklinde 3 adet rotadır. Bu rotaların maliyeti 144.41’dir.

Bu problemin hibrit yaklaşımlarla çözülmesi sonucunda;“50-1001-1000-1001-4-45-33-10-40-1”, “41-13-27-1” şeklinde 2 adet rota bulunmuştur. Bu rotaların maliyeti 72.18’dir.

Sonuçların maliyetine bakıldığında; hibrit yaklaşımlar sonucunun rota sayısı ve maliyeti daha düşüktür. Bu bölümde gerçek hayatta yaşanabilecek senaryolar oluşturulmuş ve hibrit yaklaşımla çözülmüştür. Bu sonuçlardan görüleceği gibi klasik araç rotalama problemi için önerilen hibrit yaklaşım, dinamik durumlar için de başarılı sonuçlar üretmiştir.



5. SONUÇLAR VE ÖNERİLER

5.1. Çalışmanın Özeti

Bu çalışmada, kapasite kısıtlı araç rotalama problemi ele alınmış ve farklı metasezgisel yaklaşımlarla çözülmüştür. Problemin çözümü için tasarruf algoritması, genetik algoritma, yabancı ot algoritması ve bu algoritmalarından oluşturulan hibrit algoritmalar oluşturulmuştur. Bu çalışmanın en önemli katkılarından birisi dinamik araç rotalama problemlerinde kullanılacak metasezgisel yaklaşımların ele alınması ve en uygun yaklaşımın ortaya konulmasıdır. Bu amaçla, önerilen metasezgisellerin kullanıldığı dinamik araç rotalama problemi incelenmiştir. Buna ek olarak geliştirilen yaklaşımların kullanılabilirliğini sağlamak için optimum ya da optimuma yakın rotayı bulan ve gerçek hayat uygulamalarında kullanılacak bir kullanıcı arayüzü geliştirilmiştir.

5.2. Sonuçlar

Bu çalışmadan elde edilen sonuçlar aşağıda özetlenmiştir:

- Kapasite kısıtlı araç rotalama problemleri çözümü zor problem sınıfı olan NP-Hard sınıf problemlerdendir. Bu problemlerin kesin olarak çözülmesi genellikle çok zordur. Bu nedenle önceki çalışmalarda sıklıkla sezgisel yaklaşımlar önerilmiştir. Bu yaklaşımların bazıları probleme özgü olarak önerilen yaklaşımlardır. Bu çalışmada, bu kapsamda tasarruf algoritması tabanlı bir yaklaşım ile kapasite kısıtlı araç rotalama probleminin çözümü sağlanmıştır. Bu yaklaşımla maliyet olarak genellikle diğer yaklaşımlara göre daha yüksek maliyetli rotalar oluşturulmaktadır. Buna ek olarak çözüm süresi incelendiğinde, tek bir iterasyonda çözüm bulunduğundan çözüm süreleri açısından oldukça hızlı bir yöntem olduğu görülmüştür.

- Genetik algoritma araç rotalama problemlerinde sıklıkla kullanılan yaklaşımlardan birisidir. Bu amaçla genetik algoritma yaklaşımıyla da problemin modellenmesi sağlanmıştır. Buna göre elde edilen sonuçların görece olarak tasarruf algoritmasından iyi olduğu görülmektedir. Buna karşın hız incelendiğinde genetik algoritmanın diğer yaklaşımlara göre daha uzun sürede çözüm bulabildiği görülmektedir.
- Araç rotalama problemlerine yabani ot algoritması uyarlanmıştır. Yabani ot algoritmasının sürekli çözüm uzayındaki problemlerde iyi sonuçlar verdiği ve buna ek olarak gezgin satıcı probleminde de uyarlandığı önceki çalışmalardan görülmüştür. Bu çalışmada incelenen kapasite kısıtlı araç rotalama problemine yabani ot algoritmasının eklenmesiyle birlikte çözüm açısından tasarruf algoritmasından ve genetik algoritmadan iyi maliyetler elde edilmiştir. Ayrıca diğer algoritmalarından çözüm süresinin oldukça hızlı olması yabani ot algoritmasının etkin bir yaklaşım olabileceğini göstermiştir. Kapasite kısıtlı araç rotalama problemine uyarlanan metsezgisel bir algoritma olan yabani ot algoritmasının çok hızlı şekilde, son derece rekabetçi ve verimli sonuçlar ürettiği söylenebilir.
- Tasarruf ve genetik algoritmaları hibrit hale geldiğinde çözümlerdeki maliyetlerin iyileştiği görülmüştür, ancak çözüm süreleri genetik algoritmadan dolayı halen oldukça uzun olmuştur.
- Tasarruf ve yabani ot algoritmasının hibritlenmesi ise maliyet ve çözüm süresi açısından oldukça iyi sonuçlar vermiştir. Bu yaklaşım ile iyi çözüm maliyetlerine oldukça düşük standart sapmalarla ve kabul edilebilir çözüm süreleriyle ulaşılmıştır.
- Önerilen yaklaşımların sonuncusu ise tasarruf, genetik ve yabani ot algoritmalarının hibritlendiği yaklaşımdır. Bu yaklaşıma göre, iyi çözüm maliyetlerine oldukça düşük standart sapmalarla ve kabul edilebilir çözüm süreleriyle ulaşılmıştır.

- Dinamik problemin çözümlenmesi için örnek bir problem ele alınmış ve talebin azaldığı, arttığı ve her iki durumun birlikte olduğu durumda problem yeniden çözümlenmiştir. Buna göre, önerilen yaklaşımla dinamik ele alınması durumunda dinamik olmaması durumuna göre daha düşük maliyetler oluşmuştur.
- Bu sonuçlara ek olarak elde edilen sonuçlar göstermiştir ki metasezgiselerde başlangıç popülasyonun iyi çözümler olması, sonuç çözümün verimliliğini ve başarımını artırmaktadır. Metasezgisel bir algoritmanın çözümünün başka bir metasezgiselde kullanılması çözüme olumlu katkılar yapmıştır. Hibrit algoritmalar, diğer algoritmalarından daha iyi sonuçlar üretmiştir.
- Ortaya konan uygulamanın kullanımı için web tabanlı kullanıcı arayüzü tasarlanmıştır. Kullanıcı arayüzü uygulamaya kolaylığı sağlamıştır. Ayrıca web tabanlı olmasından dolayı kurulum gerektirmeden farklı cihazlardan ve farklı ortamlardan web üzerinden erişim sağlanmıştır. Kullanıcı için anlaşılır sonuçlar üretilmiş ve bu sonuçlar harita üzerinde gösterilmiştir. Ayrıca gerçek hayat problemleri için noktalar arası uzaklık öklid kuralına göre değil, gerçek araç yolu uzunluğu temel alınarak problem çözümü sağlanmıştır.

5.3. Öneriler

Araç rotalama problemine uyarlanan yabancı ot algoritmasının çözüm kalitesi için parametre seçimi önemlidir. Çoğalma aşamasında tohumlar üretilirken farklı yaklaşımların kullanılması da çözüm kalitesini etkileyebilecek diğer bir faktördür. Gelecek çalışmalarda uygun parametre seçimi ve çoğalma aşamasında farklı yaklaşımların kullanılması için detaylı çalışmalar yapılırsa algoritmanın çözüm kalitesi yükseltilebilir. Yabancı ot algoritmasının çok büyük ölçekli problemlerdeki başarımı araştırılabilir.

İncelenen araç rotalama probleminin karakteristikleri zaman pencereli, belirsiz talep koşullarına sahip, aynı anda topla-dağıt durumu oluşması durumlarına uyarlanabilir.

Algoritmaların dinamik araç rotalama problemindeki başarımını ortaya koymak için daha fazla problem ile test edilmesi gerekir. Ayrıca bir işletmenin dinamik araç rotalama problemi için algoritmalar uyarlanabilir.

Dinamik probleme yolun durumu, trafik durumu dahil edilebilir ve bu durum için bir çözüm aranabilir. Makine öğrenmesi ile gelecek taleplerin nerelerden geleceğini tahmin eden ve öğrenen bir algoritma geliştirilebilir.

KAYNAKLAR

- Ahmed, A. K. M. F., Sun, J., 2018. Bilayer Local Search Enhanced Particle Swarm Optimization for the Capacitated Vehicle Routing Problem. *Algorithms*, 11(3):31.
- Arakaki, R. K., Usberti, F. L., 2019. An Efficiency-Based Path-Scanning Heuristic for the Capacitated arc Routing Problem. *Computers & Operations Research*, 103:288-295.
- Arnold, F., Gendreau, M., Sörensen, K., 2019. Efficiently Solving Very Large-Scale Routing Problems. *Computers & Operations Research*, 107:32-42.
- Artaç, T., 2003. Genetik Algoritma İle Dağıtım Şebekelerinin Optimum Tasarımı. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul (Yayınlanmamış).
- Atmaca, E., 2012. Bir Kargo Şirketinde Araç Rotalama Problemi ve Uygulaması. *Türk Bilim Araştırma Vakfı Dergisi*, 5(2):12-27.
- Atmaca, H. E., Vardar, S., Akbabaöz, S., Vural, A., Gözde, U, 2015. Ankara İlinde Ürün Dağıtımını Yapan Bir Beyaz Eşya Yetkili Servisinin Araç Rotalama Problemine Çözüm Yaklaşımı. *Politeknik Dergisi*, 18(2):99-105.
- Augerat, P., Belenguer, J., Benavent, E., Corberan, A., Naddef, D., Rinaldi, G., 1995. Computational Results with A Branch And Cut Code for the Capacitated Vehicle Routing Problem. Tech. Rep. 949-M, Universit'e Joseph Fourier, Grenoble, France.
- Bayzan, Ş., 2009. GPRS Verileri Yardımıyla Araç Rotalarının Belirlenmesi Problemine Farklı Bir Yaklaşım. *Akademik Bilişim*, 9(1):243-249.
- Benrahou, F., Tairi, A., 2019. Capacitated Vehicle Routing Problem For Collection Waste Lube Oil In Algiers. *Feb-Fresenius Environmental Bulletin*, 4500.

- Bozyer, Z., Alkan, A., Fıglalı, A., 2014. Kapasite Kısıtlı Araç Rotalama Probleminin Çözümü İçin Önce Grupla Sonra Rotala Merkezli Sezgisel Algoritma Önerisi. *International Journal of Informatics Technologies*, 7(2).
- Breedam, A.V., 2001. Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem. *Computers & Operations Research*, 28:289–315.
- Cheng, R., Gen, M., Tsujimiray, Y., 1999. A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms, Part II: Hybrid Genetic Search Strategies. *Computers and Industrial Engineering*, 36:343-364.
- Christofides, N., Mingozzi, A., Toth, P., Sandi, C., 1979. *Combinatorial Optimization*. Wiley, Chichester, 315-338.
- Clarke, G., Wright, J. W., 1964. Scheduling Of Vehicles From A Central Depot To A Number Of Delivery Points. *Operations Research*, 12(4):568-581.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. Ve Stein, C., 2009. *Introduction to Algorithms*. Cambridge: MIT, 3:1049-1050.
- Dantzig, G. B., & Ramser, J. H., 1959. The Truck Dispatching Problem. *Management Science*, 6(1):80-91.
- Demirtaş, Y.E., Özdemir, E., 2017. Dinamik Araç Rotalama Problemleri İçin Yeni Bir Çözüm Önerisi. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 22(3):807-823.
- Dikmen, H., Dikmen, H., Elbir, A., Ekşi, Z., Çelik, F., 2014. Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Yüksek Lisans*, 9s.
- Dursun, P., 2009. Zaman Pencereli Araç Rotalama Problemi'nin Genetik Algoritma İle Modellenmesi. *Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü (Yayınlanmamış)*.

- Düzakın, E., Demircioğlu, M., 2009. Araç Rotalama Problemleri ve Çözüm Yöntemleri. Çukurova Üniversitesi İİBF Dergisi, 13(1):68-87.
- Erera, A. L., Daganzo, C. F., 2003. A Dynamic Scheme for Stochastic Vehicle Routing. Report, Georgia Institute of Technology.
- Ergülen, Güngör, İ., , A., 2006. Bulanık Araç Rotalama Problemlerine Bir Model Önerisi ve Bir Uygulama. Yönetim ve Ekonomi: Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 13(1):53-60.
- Erol, V., 2006. Araç Rotalama Problemleri İçin Popülasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması. Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul (Yayınlanmamış).
- Eryavuz, M., Gencer, C., 2001. Araç Rotalama Problemine Ait Bir Uygulama. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 6(1).
- Golden, B.L., Wong, R. T., 1981. Capacitated arc Routing Problems. Networks 11:305–315.
- Hu, T. Y., Liao, T. Y., Lu, Y. C., 2003. Study of Solution Approach for Dynamic Vehicle Routing Problems with Real-Time Information. Transportation Research Record, 1857(1):102-108.
- Jahangir, H., Mohammadi, M., Pasandideh, S. H. R., Nobari, N. Z., 2018. Comparing Performance of Genetic and Discrete Invasive Weed Optimization Algorithms for Solving The Inventory Routing Problem with An Incremental Delivery. Journal of Intelligent Manufacturing, 1-27.
- Jang, J.S.R., 1997. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Chapter 7: Derivative-Free Optimization, Prentice-Hall, USA, s. 173-196.

- Karagül, K., Tokat, S., Aydemir, E., 2016. Kapasite Kısıtlı Araç Rotalama Problemlerinde Başlangıç Rotalarının Kurulması İçin Yeni Bir Algoritma. Mühendislik Bilimleri ve Tasarım Dergisi, 4(3):215-226.
- Keskintürk, T., Topuk, N., Özyeşil, O., 2015. Araç Rotalama Problemleri ile Çözüm Yöntemlerinin Sınıflandırılması ve Bir Uygulama. İşletme Bilimi Dergisi, 3(2):77-107.
- Kiremitçi, B., Kiremitçi, S., Keskintürk T., 2014. Zaman Pencereci Çok Araçlı Dağıtım Toplamalı Rotalama Problemi İçin Gerçek Değerli Genetik Algoritma Yaklaşımı. İstanbul Üniversitesi İşletme Fakültesi Dergisi, 43(2):391-403.
- Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F., 2000. Classical And Modern Heuristics For The Vehicle Routing Problem. International Transactions in Operation Research, 7:285-300.
- Lawrence, D., 1990. Handbook of Genetic Algorithms. Addison Wesley.
- Li, J., Pardalos, P. M., Sun, H., Pei, J., Zhang, Y., 2015. Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. Expert Systems with Applications, 42:3551-3561.
- Lysgaard, J., 1997. Clarke & Wright's savings algorithm. Department of Management Science and Logistics The Aarhus School of Business, 2-3.
- Kahraman, A.M., Özdağlar, D., 2004. Su Dağıtım Sistemlerinin Genetik Algoritma İle Optimizasyonu. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 6(3):1-18.
- Kilby, P., Prosser, P., Shaw, P., 1998. Dynamic VRPs: A study of scenarios. University of Strathclyde Technical Report, 1-11.
- Kosif, B., Ekmekçi, İ., 2012. Araç Rotalama Sistemleri ve Tasarruf Algoritması Uygulaması. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 11(21):41-51.

- Koç, İ., Nureddin, R., Kahramanlı, H., 2018. Türkiye'de Enerji Talebini Tahmin Etmek İçin Doğrusal Form Kullanarak GSA (Yerçekimi Arama Algoritması) Ve IWO (Yabani Ot Optimizasyon Algoritması) Tekniklerinin Uygulanması. Selçuk Üniversitesi Mühendislik, Bilim Ve Teknoloji Dergisi, 6(4):529-543.
- Mehrabian, A. R., Lucas, C., 2006. A Novel Numerical Optimization Algorithm Inspired From Weed Colonization. Ecological informatics, 1(4):355-366.
- Mohammadi, M., Razmi, J., Tavakkoli-Moghaddam, R., 2013. Multi-Objective Invasive Weed Optimization for Stochastic Green Hub Location Routing Problem with Simultaneous Pick-Ups and Deliveries. Economic Computation & Economic Cybernetics Studies & Research, 47(3).
- Mulloorakam, A. T., Nidhiry, N. M., 2019. Combined Objective Optimization for Vehicle Routing Using Genetic Algorithm. Materials Today: Proceedings, 11:891-902.
- Nabiyev, V. V., 2003. Yapay Zeka – Problemler Yöntemler Algoritmalar. Seçkin Yayınevi, Ankara.
- Nath, S., Chakravarty, A. K., Ghosh, S., Sarkar, S. K., 2017. Invasive Weed Optimization Approach To VLSI Routing. Devices for Integrated Circuit, 615-619.
- Normasari, N. M. E., Yu, V. F., Bachtayar, C., 2019. A Simulated Annealing Heuristic for the Capacitated Green Vehicle Routing Problem. Mathematical Problems in Engineering.
- Özkök, B. A., Kurul, F. C., 2014. Araç rotalama problemine tam sayılı lineer programlama modeli ve gıda sektöründe bir uygulama. Istanbul University Journal of the School of Business Administration, 43(2):251-260.
- Pahlavani, P., Delavar, M. R., Frank, A. U. 2012. Using A Modified Invasive Weed Optimization Algorithm for a Personalized Urban Multi-Criteria

- Path Optimization Problem. *International Journal of Applied Earth Observation and Geoinformation*, 18:313-328.
- Pichpibul, T., Kawtummachai, R., 2012. An Improved Clarke and Wright Savings Algorithm for the Capacitated Vehicle Routing Problem. *ScienceAsia*, 38(3):307-318.
- Pisinger, D., Ropke, S., 2007. A General Heuristic for Vehicle Routing Problems. *Computers & Operations Research*, 34(8):2403-2435.
- Potvin, J. Y., Xu, Y., Benyahia, I., 2004. Vehicle Routing and Scheduling with Dynamic Travel Times. *Computers & Operations Research*, 0305-0548.
- Psaraftis, H., 1988. Dynamic vehicle routing problems. *Vehicle routing: Methods and studies*, 16:223-248.
- Psaraftis, H. N., 1995. Dynamic vehicle routing: Status and prospects. *Annals of operations research*, 61(1):143-164.
- Rojas-Cuevas, I. D., Caballero-Morales, S. O., Martinez-Flores, J. L., Mendoza-Vazquez, J. R., 2018. Capacitated Vehicle Routing Problem Model for Carriers. *Journal of Transport and Supply Chain Management*, 12(1):1-9.
- Ropke, S., 2005. Heuristic and Exact Algorithms for Vehicle Routing Problems.
- Sur, C., Shukla, A., 2013. Discrete Invasive Weed Optimization Algorithm for Graph Based Combinatorial Road Network Management Problem. *International Symposium on Computational and Business Intelligence*, 254-257.
- Şahin, Y., Eroğlu, A., 2014. Kapasite Kısıtlı Araç Rotalama Problemi İçin Metasezgisel Yöntemler: Bilimsel Yazın Taraması. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 19(4):337-355.
- Tarantilis, C. D., Ioannou, G. ve Prostacos, G., 2005. Advanced Vehicle Routing Algorithms for Complex Operations Management Problems. *Journal of Food Engineering*, 70:455-471.

- Toffolo, T. A., Vidal, T., Wauters, T., 2019. Heuristics for Vehicle Routing Problems: Sequence Or Set Optimization?. *Computers & Operations Research*, 105:118-131.
- Ulutaş, A., Bayrakçıl, A. O., Kutlu, B., 2017. Araç Rotalama Probleminin Tasarruf Algoritması İle Çözümü: Sivas'ta Bir Ekmek Fırını İçin Uygulama. *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 18(1):185-197.
- Urul, B., Cengiz, Y., 2014. H Şekilli Mikroşerit Antenin Yabani Ot Algoritması İle Rezonans Frekansının Belirlenmesi. *Deü Mühendislik Fakültesi Mühendislik Bilimleri Dergisi*, 16(48):109-117.
- Ünlü, N., Uçar, E., Akkuş, G. B., Şen, B., 2017. Kargo İşlemede Zaman Pencereci Çok Araçlı Dinamik Rotalama. *Yönetim Bilişim Sistemleri Dergisi*, 3(2):105-113.
- Wang, L., Lu, J., 2019. A Memetic Algorithm With Competition for the Capacitated Green Vehicle Routing Problem. *IEEE/CAA Journal of Automatica Sinica*, 6(2):516-526.
- Wedyan, A. F., Narayanan, A., 2014. Solving Capacitated Vehicle Routing Problem Using Intelligent Water Drops Algorithm. *10th International Conference on Natural Computation*, 469-474.
- Yazgan, H. R., Ercan, S., Arslan, C., 2014. Talep Ve Kapasite Kısıtlı Optimizasyon Problemi İçin Yeni Bir Melez Algoritma. *Endüstri Mühendisliği Dergisi*, 25(1/2):16-28.
- Yeniay, Ö., 2001. An Overview of Genetic Algorithms. *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, 2(1):37-49.
- Yücenur, G. N., Demirel, N. Ç., 2011. A Hybrid Algoritm With Genetic Algorithm And Ant Colony Optimization For Solving Multi-Depot Vehicle Routing Problems. *Journal of Engineering and Natural Sciences Mühendislik ve Fen Bilimleri Dergisi*, 29:340-350.

Zhang, S., Gajpal, Y., Appadoo, S. S., 2018. A Meta-Heuristic for Capacitated Green Vehicle Routing Problem. *Annals of Operations Research*, 269(1-2):753-771.

Zhao, Y., Leng, L., Qian, Z., Wang, W., 2016. A Discrete Hybrid Invasive Weed Optimization Algorithm for the Capacitated Vehicle Routing Problem. *Procedia Computer Science*, 91:978-987.

<http://neo.lcc.uma.es/vrp/known-best-results/>, Son Erişim Tarihi: 01.07.2019.

<http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>, Son Erişim Tarihi: 01.07.2019.



ÖZGEÇMİŞ

07.07.1991 yılında Elazığ'da doğdu. İlk, orta ve lise öğrenimini Adana'da tamamladı. 2009 yılında Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü'nde üniversite eğitimine başladı. 2011 yılında Çukurova Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü'ne geçiş yaptı ve 2014 yılında başarıyla mezun oldu. Aynı yıl çift anadal programı kapsamında Endüstri Mühendisliği Bölümü'nden de mezun oldu.



