



**HETEROJEN FİLOLU HAREKETLİ MÜŞTERİLİ ARAÇ ROTALAMA
PROBLEMİ İÇİN ÇÖZÜM YAKLAŞIMLARI**

Ukbe Üsame UÇAR

**DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

EKİM 2019

Ukbe Üsame UÇAR tarafından hazırlanan “HETEROJEN FİLOLU HAREKETLİ MÜŞTERİLİ ARAÇ ROTALAMA PROBLEMİ İÇİN ÇÖZÜM YAKLAŞIMLARI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Endüstri Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Selçuk Kürşat İŞLEYEN

Endüstri Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Başkan: Prof. Dr. Hadi GÖKÇEN

Endüstri Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Prof. Dr. Cevriye TEMEL GENCER

Endüstri Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. M. Alp ERTEM

Endüstri Mühendisliği Ana Bilim Dalı, Çankaya Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Ümit Sami SAKALLI

Yöneylem Araştırması Ana Bilim Dalı, Kırıkkale Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 16/10/2019

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ukbe Üsame UÇAR

16/10/2019

HETEROJEN FİLOLU HAREKETLİ MÜŞTERİLİ ARAÇ ROTALAMA PROBLEMİ İÇİN ÇÖZÜM YAKLAŞIMLARI

(Doktora Tezi)

Ukbe Üsame UÇAR

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Ekim 2019

ÖZET

Bu tez çalışmasında, SİHA ve İHA'lar tarafından hareketli nesnelerin takibi, gözetlenmesi ve etkisiz hale getirilmesi için uygun görev rotalarının belirlenmesi gereksiniminden ortaya çıkan Çok Amaçlı-Zaman Pencere-Kapasite Kısıtlı-Heterojen Filolu-Hareketli Müşterili Araç Rotalama Problemi ele alınmıştır. Tez çalışmasının ilk aşamasında operasyon alanında bir tane vurucu ve “n” tane hareketli hedefin bulunduğu problem türü dikkate alınmış ve problemin çözümü için Tavlama Benzetimine dayalı bir çözüm yaklaşımı geliştirilmiştir. Önerilen Tavlama Benzetimi Algoritması içerisinde çözüm kalitesini artırmak ve çözüm hızını iyileştirmek için yeni çözüm stratejileri geliştirilmiş ve farklı başlangıç çözümleri kullanılmıştır. Tezin ikinci aşamada, bir tane vurucu ve “n” tane hareketli hedeften ya da düğümden oluşan problem, zaman penceresi kısıtlaması altında incelenmiş ve her bir hedefin ilgili zaman aralığı içerisinde imha edildiği minimum zamanlı tur rotası Tavlama Benzetimi Algoritması ile belirlenmeye çalışılmıştır. Önerilen algoritmanın etkinliği, Ege Denizindeki gemilerin gözetlendiği bir uygulama çalışması üzerinde test edilmiştir. Belirtilen iki aşamada da zaman biriminin sürekli olduğu varsayılmıştır. Tezin üçüncü aşamasında, “m” sayıda vurucu ve “n” sayıda hedeften oluşan Hareketli Müşterili Araç Rotalama Problemi, minimum görev zamanı ve görev maliyetleri amaçları doğrultusunda çözülmeye çalışılmıştır. Problemden, vurucuların farklı hızlara, havada kalma kapasitelerine ve görev maliyetlerine sahip oldukları varsayılmış ve ilgili zaman penceresi içerisinde tüm hedeflerin etkisiz hale getirileceği ideal tur rotasının belirlenmesi için sezgisel algoritmalar geliştirilmiş (ÇARA, RASA) ve metasezgisel algoritmalar (Tavlama Benzetimi, Genetik Algoritma ve NSGA-II Algoritması) yararlanılmıştır. Önerilen algoritmalar, farklı problem setleri üzerinde test edilmiş ve ilgili problemlerin hepsinde zaman biriminin sürekli olduğu varsayılmıştır.

Bilim Kodu : 90610

Anahtar Kelimeler : Hareketli hedefli gezgin satıcı problemi, hareketli müşterili araç rotalama problemi, metasezgisel algoritmalar

Sayfa Adedi : 99

Danışman : Doç. Dr. Selçuk Kürşat İŞLEYEN

SOLUTION APPROACHES FOR HETEROGENEOUS FLEET MOVING CUSTOMER VEHICLE ROUTING PROBLEM

(Ph. D. Thesis)

Ukbe Üsame UÇAR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

October 2019

ABSTRACT

In this thesis, Multi Objective-Time Window-Capacity Restricted-Heterogeneous Fleet-Moving Customer Vehicle Routing Problem, which arises from the need to determine the appropriate task routes for tracking, monitoring and neutralizing moving objects by UCAVs and UAVs, is discussed. In the first stage of the thesis, the problem type which has one pursuer and "n" moving target in the field of operation has been taken into consideration and a solution approach based on Simulated Annealing has been developed for the solution of the problem. In the proposed Simulated Annealing Algorithm, new solution strategies have been developed and different initial solutions have been used in order to improve the solution quality and improve the solution speed. In the second stage of the thesis, the problem consisting of one pursuer and "n" moving target or node is examined under the time window constraint and the minimum time tour route in which each target is destroyed within the relevant time interval is tried to be determined by the Simulated Annealing Algorithm. The efficiency of the proposed algorithm was tested on an application study in which the vessels in the Aegean Sea were monitored. In both of these stages, the time unit was continuously considered. In the third stage of the thesis, the Moving Customer Vehicle Routing Problem consisting of "m" number of pursuers and "n" number of targets is tried to be solved for the purposes of minimum operation time and mission costs. In the problem, the pursuers are assumed to have different speeds, endurance and mission costs. In the problem, Heuristic (ÇARA, RASA) and Metaheuristic (Tavlama Benzetimi, Genetik Algoritma ve NSGA-II Algoritması) Algorithms have been developed to determine the ideal tour route in which all targets will be destroyed within the relevant time window. The proposed algorithms are tested on different problem sets and it is assumed that the time unit is continuous in all test problems.

Science Code : 90610

Key Words : Moving target traveling salesman problem, moving customer vehicle routing problem, metaheuristic algorithms.

Page Number : 99

Supervisor : Assoc. Prof. Dr. Selçuk Kürşat İŞLEYEN

TEŞEKKÜR

Doktora tezi bilimsel anlamda uzun, zor ve yorucu bir süreç olmakta, bu süreçte akademik ve sosyal anlamda birçok kişinin desteğine ihtiyaç duyulmaktadır. Öncelikle lisans mezunu olduğum, Atatürk Üniversitesi Endüstri Mühendisliği Bölümü'nün değerli hocalarına çok teşekkür ederim. ÖYP Asistanı olarak görevlendirildiğim, yüksek lisans ve doktora eğitimimi aldığım Gazi Üniversitesi Endüstri Mühendisliği Bölümündeki Değerli Hocalarıma, Asistan Arkadaşlarıma ve İdari Personel Çalışanlarına çok teşekkürler ederim. Değerli Hocalarımla ve Asistan Arkadaşlarımla birlikte çalışmış olmamın benim açımdan büyük bir mutluluk, gurur ve şeref olduğunu belirtmek isterim. Doktora tez çalışmalarında, değerli bilgi birikimleri ile tezimin bilimsel anlamda daha anlamlı ve değerli olmasını sağlayan, tez süreci boyunca desteklerini ve anlayışlarını esirgemeyen Değerli Hocalarım Prof. Dr. Hadi GÖKÇEN Hocama ve Doç. Dr. M. Alp ERTEM Hocama çok teşekkürler ederim. Lisans eğitimim zamanında öğrencisi olduğum, Yüksek Lisans ve Doktora Eğitimi gibi uzun ve yorucu bir süreç boyunca sürekli yanımda olan, maddi ve manevi yardımlarını esirgemeyen, bana akademik ve kreatif düşünme becerisini kazandıran, asistanı olmaktan büyük onur, gurur ve mutluluk duyduğum Değerli Danışmanım Doç. Dr. Selçuk Kürşat İŞLEYEN Hocama çok teşekkürler ederim. Fırat Üniversitesi Endüstri Mühendisliği Bölümü'nde göreve başladığım 2013 yılından itibaren her zaman yanımda olan, bilimsel ve manevi desteği ile değerli anlayışını hiçbir zaman esirgemeyen Değerli Bölüm Başkanımız Prof. Dr. Figen BALO' ya çok teşekkür ederim. Değerli Dostlarım Murat ŞAHİN, İsmet SÖYLEMEZ, Oğuz ALTUNYUVA, Refik GÖKTUĞ, Remzi ÇALIŞIR, Sinan TURGUT ve Tolga ISPARTALIOĞLU'na desteklerinden dolayı çok teşekkürler ederim. Tezimin yazımı ve tez ile ilgili süreç prosedürü hakkında bilgi ve desteğini esirgemeyen Arş. Gör. Esra ÖZKAN AKSU ve Arş. Gör. Özlem ALPAY' a çok teşekkür ederim. Doktora tez çalışmalarına sağladığı maddi destekten dolayı TÜBİTAK-BİDEB 2211 Doktora Burs Programına teşekkürü bir borç bilirim. Son olarak, üzerimde sonsuz emekleri olan, benim için hiçbir zaman hiçbir fedakârlıktan kaçınmayan, bana sabır ile katlanan ve hayatımın her anında destekleri ile yanımda olan Annem Mine UÇAR, Babam Osman UÇAR, Kardeşlerim Mücahid UÇAR ve Deryanur UÇAR ile Teyzem Fatıma YURT ve Dayım Ayhan ASLAN başta olmak üzere anneanneme, dayılarıma ve teyzelerime çok teşekkürler ederim. Saygılarımla...

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ.....	xii
RESİMLERİN LİSTESİ.....	xiv
HARİTALARIN LİSTESİ.....	xv
SİMGELER VE KISALTMALAR.....	xvi
1. GİRİŞ.....	1
2. LİTERATÜR ARAŞTIRMASI	3
2.1. Hareketli Hedefli Gezgin Satıcı Problemi	3
2.2. Dinamik Araç Rotalama Problemi	10
3. TEK VURUCULU-HAREKETLİ HEDEFLİ GEZGİN SATICI PROBLEMİ İÇİN TAVLAMA BENZETİMİNE DAYALI BİR ÇÖZÜM YAKLAŞIMI	13
3.1. Çözüm Metodolojisi	16
3.1.1. Minimum yol sezgiseli	17
3.1.2. Bağlı zaman sezgiseli	17
3.1.3. Bölge imha sezgiseli	18
3.1.4. Güncelleme tabanlı bölge imha sezgiseli	18
3.1.5. Rassal arama.....	19
3.1.6. Minimum zaman sezgiseli.....	19
3.1.7. Strateji tabanlı tavlama benzetimi algoritması	20

	Sayfa
3.2. Uygulama Çalışması	23
3.2.1. Problem tasarımı.....	23
3.2.2. Başlangıç çözümünün belirlenmesi.....	25
3.2.3. Parametre konfigürasyonu	28
3.2.4. Deneysel sonuçlar	32
3.3. Sonuç.....	33
4. ZAMAN PENCERELİ HHGSP’NİN TAVLAMA BENZETİMİ ALGORİTMASI İLE ÇÖZÜMÜ: DENİZ GÖZETLEME OPERASYONU UYGULAMASI	35
4.1. Problemin Tanımı	36
4.2. Metodoloji.....	40
4.3. Uygulama Çalışması	41
4.3.1. Problem tasarımı	41
4.3.2. Parametre konfigürasyonu.....	44
4.3.3. Deneysel sonuçlar.....	45
4.4. Sonuç.....	48
5. UÇUŞ KAPASİTESİ VE ZAMAN PENCERESİ KISITLARI ALTINDA ÇOK AMAÇLI-HETEROJEN FİLOLU-HAREKETLİ MÜŞTERİLİ ARAÇ ROTALAMA PROBLEMİNİN METASEZGİSEL ALGORİTMALARLA ÇÖZÜMÜ	51
5.1. HM-ARP İçin Önerilen Çözüm Metodolojileri.....	57
5.1.1. Çok hedefli çok vuruculu rotalama algoritması	58
5.1.2. Rassal hedef imha algoritması	59
5.1.3. Genetik algoritma.....	59
5.1.4. Pareto Optimalite kavramı ve problem için geliştirilen NSGA-II algoritmasına dayalı bir çözüm yaklaşımı.....	67
5.1.5. Tavlama benzetimi.....	73

	Sayfa
5.2. Uygulama Çalışması	74
5.2.1. Problem tasarımı	74
5.2.2. Parametre konfigürasyonu.....	74
5.2.3. Uygulama çalışması.....	77
6. SONUÇ VE ÖNERİLER.....	83
KAYNAKLAR	87
ÖZGEÇMİŞ	95

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Minimum yol sezgiseli	17
Çizelge 3.2. Bağlı zaman sezgiseli	17
Çizelge 3.3. Bölge imha sezgiseli.....	18
Çizelge 3.4. Güncelleme tabanlı bölge imha sezgiseli.....	19
Çizelge 3.5. Minimum zaman sezgiseli	20
Çizelge 3.6. Strateji tabanlı tavlama benzetimi algoritması	24
Çizelge 3.7. Tek vuruculu HHGSP için senaryo bilgileri.....	24
Çizelge 3.8. Çözüm kurucu sezgisellerin karşılaştırılmasına ilişkin deney sonuçları	25
Çizelge 3.9. Çözüm kurucu sezgisel algoritmalar için wilcoxon signed ranked test analiz sonuçları	28
Çizelge 3.10. Tek vuruculu HHGSP'nin deneysel analizi için konfigürasyonda kullanılan parametreler ve seviyeleri	29
Çizelge 3.11. L9 ortogonal dizilimi	29
Çizelge 3.12. Parametre konfigürasyonunda kullanılan senaryolar.....	30
Çizelge 3.13. SA-MTH algoritması için parametre analizi	31
Çizelge 3.14. SA-MDH algoritması için parametre analizi sonuçları	31
Çizelge 3.15. SA-MDH(A1) ve SA-MTH(A2) metotlarının karşılaştırılması	32
Çizelge 4.1. Zaman pencereli-MTH başlangıç çözümlü-tavlama benzetimi algoritması.....	40
Çizelge 4.2. Harita 4.7'de belirtilen probleme ilişkin bilgiler	42
Çizelge 4.3. Konfigürasyonda kullanılan parametreler ve onların seviyeleri.....	44
Çizelge 4.4. $\lambda=5$ için analiz sonuçları	45
Çizelge 4.5. $\lambda=10$ için analiz sonuçları	47
Çizelge 5.1. 3 vuruculu HM-ARP için hedef sayısı arttıkça çözüm sayısında meydana gelen artış	53

Çizelge	Sayfa
Çizelge 5.2. Bu tez çalışması ve Groba ve ark., çalışmaları arasındaki fark	55
Çizelge 5.3. HM-ARP için uygunluk fonksiyonuna ilişkin bilgiler	63
Çizelge 5.4. Amaçların analizinde kullanılan senaryolar.....	63
Çizelge 5.5. Amaç 1 için algoritma sonuçları	63
Çizelge 5.6. Amaç 2 için algoritma sonuçları	64
Çizelge 5.7. HM-ARP için amaçların karşılaştırılmasında dikkate alınan algoritma parametreleri	64
Çizelge 5.8. Amaçların karşılaştırılmasına ilişkin deneysel analiz sonuçları	65
Çizelge 5.9. HM-ARP için önerilen genetik algoritma.....	67
Çizelge 5.10. Tavlama benzetimi, genetik algoritma ve NSGA-II algoritmaları için dikkate alınan parametreler	75
Çizelge 5.11. HM-ARP için konfigürasyonda kullanılan parametreler ve seviyeleri.....	76
Çizelge 5.12. Tavlama benzetimi algoritması için ideal parametre değerleri.....	77
Çizelge 5.13. NSGA-II ve genetik algoritma için ideal parametre değerleri.....	77
Çizelge 5.14. HM-ARP’de amaç-1 için wilcoxon signed ranked test analiz sonuçları ..	79
Çizelge 5.15. HM-ARP’de amaç-2 için wilcoxon signed ranked test analiz sonuçları ..	80

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Hareketli hedefli gezgin satıcı problemi.....	3
Şekil 2.2. Araç rotalama problemi	10
Şekil 3.1. Tek vuruculu-hareketli hedefli gezgin satıcı problemi.....	14
Şekil 3.2. Bölge imha sezgiseli	18
Şekil 3.3. UBRDH sezgiseli (başlangıç ve ilk küme imha edildikten sonraki durum)...	19
Şekil 3.4. Herhangi iki rassal düğüm arası çözüm bozma mekanizması	23
Şekil 3.5. Herhangi iki rassal düğüm arası çözüm bozma mekanizması	23
Şekil 3.6. SA-MTH algoritması için parametre analizi	30
Şekil 3.7. SA-MDH algoritması için parametre analizi.....	31
Şekil 4.1. $\lambda=5$ için amaç değerleri açısından analiz sonuçları	46
Şekil 4.2. $\lambda=10$ için amaç değerleri açısından analiz sonuçları	46
Şekil 4.3. $\lambda=5$ için her bir senaryo için ortalama çözüm zamanı.....	47
Şekil 4.4. $\lambda=10$ için her bir senaryo için ortalama çözüm zamanı.....	48
Şekil 5.1. Hareketli müşteri araç rotalama problemi	52
Şekil 5.2. HM-ARP için çizelge 5.1'deki denemelere ilişkin birim çözüm zaman grafiği	54
Şekil 5.3. Hareketli Sistemde hedef bulunmadığı durum için vurucuya ait rota planı hareketler	54
Şekil 5.4. Çok hedefli çok vuruculu rotalama algoritması.....	59
Şekil 5.5. 10 hedeften oluşan bir kromozom için çözüm gösterimi hareketler.....	60
Şekil 5.6. PMX operatörüne göre çaprazlamadan önce kromozomların durumu.....	61
Şekil 5.7. PMX operatörüne göre çaprazlamadan sonra kromozomların durumu.....	61
Şekil 5.8. Maksimizasyon ve minimizasyon problemleri için bastırılmış ve bastırılmamış çözümler	68

Şekil	Sayfa
Şekil 5.9. Hızlı baskın olmayan sıralama yaklaşımı	70
Şekil 5.10. Yığılma uzaklığı hesaplaması	71
Şekil 5.11. Yığılma uzaklığı (crowding-distance) hesaplama algoritması	72
Şekil 5.12. NSGA-II Algoritmasına dayalı çözüm yaklaşımı.....	73
Şekil 5.13. Algoritmaların, amaç-1 için en iyi değerden sapma miktarları(%)	77
Şekil 5.14. Algoritmaların, amaç-2 için en iyi değerden sapma miktarları(%)	78
Şekil 5.15. HM-ARP’de algoritmaların çözüm zamanlarına ilişkin bilgiler	81

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 3.1. Çözüm kurucu sezgisel algoritmalar için kolmogorov-smirnov normallik test sonuçları.....	26
Resim 3.2. Çözüm kurucu sezgisel algoritma için friedman Anova test sonuçları	27
Resim 4.1. S-70-B2 Seahawk deniz helikopteri	39
Resim 5.1. Amaç-1 ve amaç-2 için görev zamanları arasındaki anlamlı farklılığın analizi	66
Resim 5.2. Amaç-1 ve amaç-2 için görev maliyetleri arasındaki anlamlı farklılığın analizi	67
Resim 5.3. HM-ARP’de amaç-1 için friedman anova testi	78
Resim 5.4. HM-ARP’de amaç-2 için friedman anova testi	80

HARİTALARIN LİSTESİ

Harita	Sayfa
Harita 4.1. Ege denizindeki karasuları sınırları.....	36
Harita 4.2. Ege denizindeki yasadışı göç güzergahı	36
Harita 4.3. Ege denizi kıyı bazında yasadışı göç güzergâhı.....	37
Harita 4.4. 2013 yılında ilgili bölgelerde yakalanan yasadışı göçmen sayısı	37
Harita 4.5. İlgili bölgedeki kıyılardan gerçekleşebilecek göç senaryoları	38
Harita 4.6. Deniz gözetleme operasyonuna ait örnek bir tur rotası.....	39
Harita 4.7. Uygulama çalışmasına ilişkin senaryo görseli	41
Harita 4.8. Deniz-trafik verilerine göre 2017 yılında Ege denizi için yoğunluk haritası.....	43

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklamalar
A	HHGSP için hedefler arasındaki yolların kümesi
a_i	HHGSP ve HM-ARP için “i” numaralı hedefin açısı
b_i, n_i	HM-ARP için “i.” vurucunun başlangıç maliyeti
c_i, f_i	HM-ARP için “i.” vurucunun uçuş maliyeti
$c_{i,j}$	Vurucunun “i.” hedeften, “j.” hedefe varma süresi
$c_i[c_{i,x}, c_{i,y}]$	HM-ARP de vurucular için başlangıç koordinatları
b_i, n_i	HM-ARP için “i.” vurucunun başlangıç maliyeti
D	Klasik ARP için düğümlerin kümesi
H	HM-ARP için hedeflerin kümesi
h_i	HM-ARP için “i” numaralı hedef
km	kilometre
M	Klasik ARP için düğümler arası mesafeler matrisi
m_i	HM-ARP de “i.” hedef için mühimmat gereksinimi
n	HHGSP için hedef sayısı
p_i	HHGSP’de “i” numaralı hedefin başlangıç pozisyonu
q_i	HM-ARP de “i.” vurucunun hızı
r_i	HM-ARP’de “i.” hedefin hızı
S	HHGSP için hedeflerin kümesi
sa	Saat
s_i	HHGSP için “i” numaralı hedef
$s_{i,j}$	Vurucunun “i.” hedeften ayrılma zamanı
$t_{i,j}$	Vurucunun “j.” hedefe varma zamanı
u_i	HM-ARP’de “i.” vurucunun havada kalma kapasitesi
v	HHGSP’de vurucunun hızı
v_i	HHGSP’de “i.” hedefin hızı
V	HM-ARP’de vurucuların kümesi

Simgeler $w_i[w_{i,x}, w_{i,y}]$ x_i $x_{i,j}$ Y $z_i[z_{i,g}, z_{i,\zeta}]$ **Açıklamalar**

HM-ARP’de “i.” hedefin başlangıç koordinatları

HM-ARP’de “i.” vurucunun toplam uçuş süresi

Vurucunun i düğümünden j düğümüne hareket ederse

Klasik ARP için düğümler arasındaki yolların kümesi

HM-ARP’de “i.” hedef için zaman penceresi

Kısaltmalar**AHP****ARP****ÇARA****D-ARP****GA****GSP****HHGSP****HHGSP-ZP****HM-ARP****İHA****MDH****MTH****NSGA-II****RASA****RDH****RS****RTH****SA-MDH****SA-MTH****SİHA****TB****UBRDH****Açıklamalar**

Analitik Hiyerarşik Proses

Araç Rotalama Problemi

Çok Hedefli-Çok Vuruculu Rotalama Algoritması

Dinamik Araç Rotalama Problemi

Genetik Algoritma

Gezgin Satıcı Problemi

Hareketli Hedefli Gezgin Satıcı Problemi

Zaman Pencereci-HHGSP

Hareketli Müşterili Araç Rotalama Problemi

İnsansız Hava Aracı

Minimum Yol Sezgiseli

Minimum Zaman Sezgiseli

Non-dominated Sorting Genetic Algorithm-II

Rassal Hedef İmha Algoritması

Bölge İmha Sezgiseli

Rassal Arama

Bağıl Zaman Sezgiseli

MDH’ye dayalı TB Algoritması

MTH’ye dayalı TB Algoritması

Silahlı İnsansız Hava Aracı

Tavlama Benzetimi

Güncelleme Tabanlı Bölge İmha Sezgiseli

1. GİRİŞ

Hareketli Müşteri Araç Rotalama Problemi (HM-ARP), filo halinde hareket eden insansız hava araçları ve dronelar tarafından deniz gözetleme operasyonları, askeri operasyonlar ve arama-kurtarma görevleri başta olmak üzere, hareketli nesnelerin takibi ve etkisiz hale getirilmesi için uygun rota planının belirlenmesi gereksiniminden ortaya çıkan, dinamik yapıları bir problem türüdür. Problem geniş bir uygulama alanına sahip olmakta ve uygulama alanları ile bağlantılı olarak içerisinde birçok amaç ve kısıt barındırabilmektedir.

Problemin en önemli uygulama alanlarından birini, Silahlı (SİHA) ve Silahsız İnsansız Hava Araçları (İHA) ile operasyon alanındaki hareketli nesnelerin gözetlenmesi, takibi ve imhası için minimum görev zamanlı ve görev maliyetli tur rotasının belirlenmesi problemi oluşturmaktadır. Ulusal güvenliğin sağlanması ve ilgili tehdit unsurlarının hızlı ve etkili bir şekilde etkisiz hale getirilmesi için filodaki araçlardan uygun olanların seçilmesi ve bu araçların uygun bir şekilde rotalanması gerekmektedir. Bu tez çalışmasında, Heterojen SİHA ve İHA filoları için hedeflere ait zaman penceresi ve göreve çıkacak araçların uçuş kapasiteleri dikkate alınarak, minimum görev süresi ve görev maliyeti amaçları doğrultusunda ideal tur rotasını belirleyecek uygun çözüm stratejilerinin geliştirilmesi amaçlanmaktadır. Askeri operasyonlara ek olarak, denizlerdeki güvenlik botlarının rotalanması, deniz gözetleme operasyonları, uzaydaki uydulara yakıt ikmali yapılması ve güvenlik kameraları için uygun hareket planının belirlenmesi gibi dinamik günlük hayat problemleri de, tez kapsamında ele alınan problemin uygulama alanlarına örnek olarak verilebilmektedir.

HM-ARP'nin temellerini Hareketli Hedefli Gezgin Satıcı Problemi(HHGSP) ve Dinamik Araç Rotalama Problemi(D-ARP) oluşturmaktadır. Çalışmanın ikinci bölümünde HHGSP, ve D-ARP tanımlamaları yapılmakta ve problemler ile ilgili literatürde bulunan çalışmalar hakkında bilgiler verilmektedir. Üçüncü bölümde, HM-ARP'nin temel versiyonlarından, Tek Vuruculu-HHGSP ele alınmakta ve problem için önerilen çözüm stratejileri takdim edilmektedir. Dördüncü bölümde, ilgili problem zaman penceresi kısıtları altında çözülmeye çalışılmakta ve probleme ilişkin bir uygulama çalışması gerçekleştirilmektedir. Beşinci bölümde ise, önceki bölümde ifade edilen problemlerden hareketle geliştirilmiş ve literatüre ilk kez önerilmiş HM-ARP'e ele alınmaktadır. Önceki bölümdeki problemlere ek

olarak, sistemde birbirinden farklı özelliklere sahip birçok vurucu bulunmakta ve bu vuruculara ait bir uçuş kapasitesi tanımlanmaktadır. Ayrıca vurucuların tamamı operasyonlarda kullanılmayıp, minimum görev zamanı ve maliyeti amacı doğrultusunda az sayıda vurucu kullanılarak zaman penceresi kısıtları altında görevler gerçekleştirilmeye çalışılmaktadır. Önceki bölümlerde, problemler için tek bir amaç (minimum görev zamanı) dikkate alınmışken, bu bölümde ifade edilen problemin amaç fonksiyonuna görev zamanı amacına ek olarak görev maliyeti amacı da eklenmiş ve iki amaç eş zamanlı minimize edilmeye çalışılmıştır. Problemin çözümünde çözüm kurucu sezgisel algoritmalar geliştirilmiş ve metasezgisel algoritmalarından (NSGA-II, Genetik Algoritma ve Tavlama Benzetimi) yararlanılmıştır. Bu sayede ilgili problem üzerinde, popülasyon tabanlı çözüm yaklaşımları ile diğer çözüm yaklaşımlarının etkinliğinin karşılaştırılması amaçlanmıştır. Altıncı ve son bölümde ise tez çalışmasına ilişkin genel değerlendirmeler yapılmış ve gelecekteki çalışmalara yönelik önerilerde bulunulmuştur.

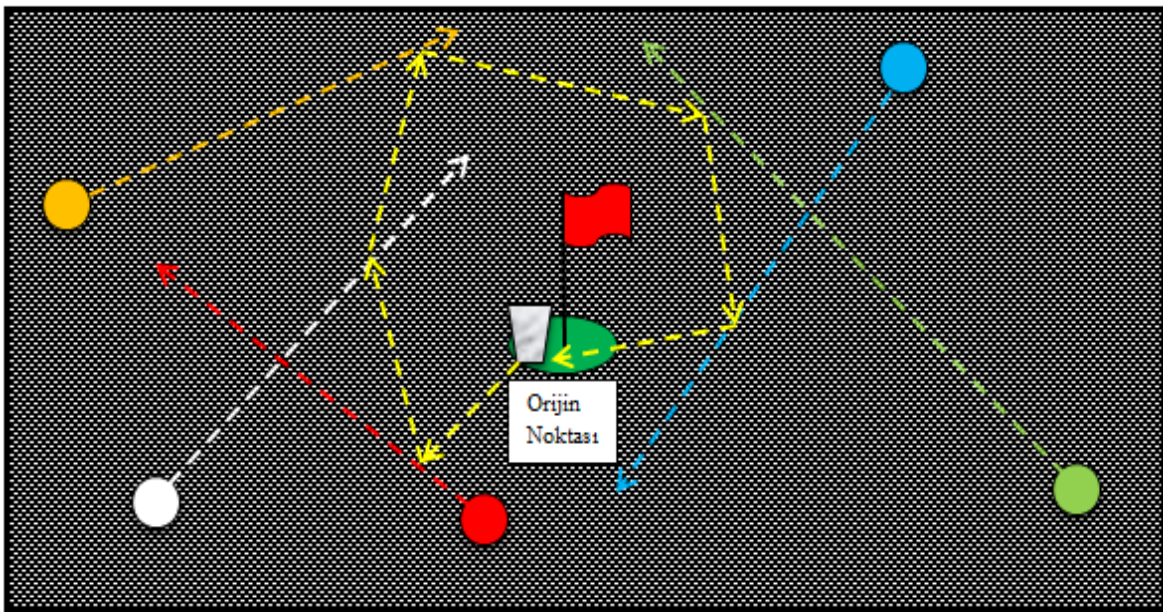
2. LİTERATÜR ARAŞTIRMASI

Bu bölümde D-ARP ve HHGSP'nin tanımlamaları yapılmakta ve ilgili problemlerle ilgili literatürde bulunan çalışmalara ilişkin bilgiler verilmektedir.

2.1. Hareketli Hedefli Gezgin Satıcı Problemi

HHGSP, ilk kez 1988 yılında Helving ve ark., tarafından ortaya atılan dinamik yapıli bir problem olup, problemin özellikleri aşağıda ifade edilmektedir [1].

Problemde, $S = \{s_1, s_2, s_3, \dots, s_n\}$, sistem içerisindeki " s_i " hedeflerinin oluşturduğu kümeyi tanımlamakta, her bir " s_i " hedefi, önceden tanımlanmış " p_i " başlangıç pozisyonundan, sabit bir " a_i " açısı ve " v_i " hızıyla harekete başlamaktadır. Hedefleri yakalayacak olan vurucu ya da gezgin satıcı ise sabit bir v hızına sahip olmakta ve bu hızın, diğer tüm hedef hızlarından daha büyük olduğu ($v > |v_i|$) varsayılmaktadır. Problemde temel amaç, orijinden harekete başlayan vurucu tarafından sistem içerisindeki tüm hedeflerin imha edilip, vurucunun tekrardan orijine döneceği minimum zamanlı, maliyetli ya da mesafeli tur rotasının belirlenmesidir [1]. Problemin temel versiyonu şekil 2.1'de gösterilmektedir.



Şekil 2.1. Hareketli hedefli gezgin satıcı problemi

Şekil 2.1’deki problemde, sistemde beş adet hedef bulunmakta ve bu hedefler belirli bir açı ve hız ile hareket etmektedir. Orijin noktasından harekete başlayan vurucu, sırasıyla kırmızı, beyaz, turuncu, yeşil ve mavi renkli hedefleri imha ederek orijin noktasına dönmekte ve minimum görev zamanı içerisinde ilgili turunu tamamlamaktadır. Problem ile ilgili literatürde birçok çalışma bulunmakta, bu çalışmaların sayısı son yıllarda giderek artmaktadır. HHGSP ile ilgili literatürde bulunan çalışmalar aşağıda ifade edilmektedir.

HHGSP’nin tanımlaması, literatürde ilk defa Helvig ve ark., tarafından yapılmış, problemin varsayımları ve özellikleri bu çalışmada ortaya konulmuştur. Bu çalışmada problem; hedef ve vurucuların tek boyutlu düzlemde(x düzleminde) hareket etmesi, vurucunun yeniden ikmal yapmak için orijine dönmesi varsayımı altında birden fazla vurucunun operasyon alanında bulunması ve vurucuların yeniden yakıt ikmali yapmak için orijine gelmesi gereksinimleri açısından incelenmiş, bu versiyonlar için çeşitli çözüm yaklaşımları geliştirilmiştir [1].

HHGSP için ilk matematiksel model 2014 yılında Fügenschuh ve ark., tarafından geliştirilmiştir. Modelde, hedef ve vurucu sayısının birden fazla olduğu HHGSP dikkate alınmış ve problemde zaman biriminin kesikli olduğunu varsayılmıştır. Matematiksel modelin küçük boyutlu problemlerde başarılı sonuçlar ürettiği tespit edilmiş, problem boyutu büyüdükçe modelin yeterli olmadığı ifade edilmiştir. Belirtilen durumun üstesinden gelmek için minimum operasyon zamanı ve minimum görev mesafesi amaçlarının dikkate alındığı iki farklı sezgisel algoritma geliştirilmiştir. Önerilen matematiksel model ve geliştirilen sezgisel algoritmaların başarısı, lazer silahları kullanılarak hareketli hedeflerin etkisiz hale getirilmeye çalışıldığı, farklı sayı ve özelliklere sahip hedefler ve vuruculardan oluşan 36 örnek seti üzerinde test edilmiştir. Önerilen sezgisel algoritmaların 30 örnek seti üzerinde optimale ulaştığı tespit edilmiştir [2].

Stieber ve ark., çok hedefli çok silahlı atama problemi için zaman biriminin kesikli olarak dikkate alındığı bir matematiksel model önermişleridir. Buna ek olarak, üç farklı sezgisel algoritma geliştirmişler ve algoritmaların başarısını 36 farklı örneklem seti üzerinde önerilen matematiksel modelleme yaklaşımıyla karşılaştırarak test etmişlerdir. Problemde enerji tüketim durumunu amaç fonksiyonuna bir parametre olarak yansıtmışlardır [3].

Stieber ve Fügenschuh, hedeflerin belirli bir zaman aralığı içerisinde bulunduğu Zaman

Pencereli-HHGSP'yi incelemiş ve tüm hedeflerin ilgili zaman aralığı içerisinde yakalandığı, minimum zamanlı seyahat turunu belirlemeye çalışmışlardır. Problemede, zaman biriminin kesikli ve sürekli olarak dikkate alındığı iki farklı durum için çözüm yaklaşımı geliştirmişlerdir. Zaman biriminin kesikli olarak dikkate alındığı durumda, kesikli birim sayısı arttıkça çözüm zamanının önemli ölçüde arttığını tespit etmişlerdir. Zaman biriminin sürekli olduğu modelde ise, zaman birimlerindeki artışın ihmal edildiği durumda, vurucu sayısındaki artışın çözüm süresini artırdığını belirlemişlerdir [4].

Jiang ve ark., klasik HHGSP'yi çözmek için genetik algoritmaya dayalı bir çözüm yaklaşımı geliştirmiş, çaprazlama operatörü, mutasyon operatörü ve bakım onarım mekanizmaları kullanarak çözüm kalitesini iyileştirmeye çalışmışlardır. Önerilen çözüm metodolojisini, Tek vuruculu-10 şehirli, 30 farklı örnek seti üzerinde test etmişlerdir. Analiz sonucunda, 30 problem setinin 18'inde optimal sonuçlara ulaşıldığını, 5 problem setinde CX operatörünün, 7 problem setinde ise OX operatörünün daha iyi sonuçlara ulaştığı belirtmişlerdir [5].

Jindal ve ark., hedeflerin ve vurucunun düz bir çizgide hareket ettiği HHGSP'yi, vurucunun iki hedefi imha ettikten sonra orijine dönmesi gerektiği varsayımı altında sezgisel algoritma ile çözmüş ve amaç fonksiyonunda toplam seyahat zamanını minimize etmeyi amaçlamışlardır. Önerilen çözüm metodolojisinde, hedeflere ait öncelik sıralarını karşılaştırmış ve analiz sonucunda ilgili metodun uygun sonuçlar ürettiğini ifade etmişlerdir [6].

Englot ve ark., HHGSP ile Kinetik GSP'nin aynı problem özelliklerine sahip olduğunu ifade etmiş ve HHGSP'nin çözümü için açgözlü sezgisel algoritma ile Lin-Kernighan Sezgiselini(LKH) sezgisellerini kullanmışlardır. İHA'lar ile hareketli hedeflerin imha edildiği uygulama çalışması sonucunda, hareketin hızlı olduğu sistemlerde açgözlü olmayan sezgisellerin, hareketin yavaş olduğu sistemlerde ise LKH sezgiselinin daha iyi çözümler ürettiğini belirtmişlerdir. Buna ek olarak, hedeflerin anlık olarak takibinde ajanlardan yararlanmışlardır [7].

Khosravi ve ark., hareketli hedeflerin belirsiz bir yörüngede hareket ettiği, operasyon alanında ise engellemeler ve düşman unsurların bulunduğu ödül toplama problemini, sezgisel algoritma ve simülasyon yardımıyla çözmeye çalışmışlardır [8].

Hammar ve Nilsson, Kinetik-GSP'yi, düğümlerin sabit bir doğrultuda ve hızda hareket ettiği varsayımı altında incelemişlerdir [9].

Zhou ve ark., şehir lokasyonlarının Gauss dağılımına göre değiştiği, 100 düğümlü Dinamik GSP'yi, matematiksel modelleme ve DIOEA algoritmasını kullanarak çözmüş ve analiz çalışması sonucunda DIOEA'nın kabul edilebilir zaman içerisinde optimal rotaya ulaştığını ifade etmişlerdir [10].

Choubey, HHGSP'yi, vurucuya ait görev zamanının minimize edilmesi amacı doğrultusunda genetik algoritma ve açgözlü sezgisel algoritma kullanarak çözmüştür. Açgözlü sezgisel algorithmada ziyaret edilecek düğümleri (yol/hız) oranına göre belirlemiş, analiz sonucunda, genetik algoritmanın açgözlü algorithmadan daha iyi sonuçlara ulaştığını tespit etmiştir [11].

Lee ve ark., Bağışıklık Sistemine Dayalı Karınca Kolonisi Optimizasyonu yöntemini kullanarak silah-hedef atama problemini çözmüşler ve amaç fonksiyonunda beklenen tahribatı minimize etmeyi amaçlamışlardır. Analiz çalışması sonucunda, ilgili algoritmanın başarılı sonuçlar ürettiğini belirlemişlerdir [12].

Agharkar ve Bullo, birim çemberlik bir görev alanı içerisinde, poisson dağılımına göre rastgele hareket eden hedeflerin imha edilmesi problemini, maksimum sayıda hedefin yakalanması amacı doğrultusunda, CNN (Capturable Nearest Neighbor), SW(Sector Wise) ve SNB (Stay Near Boundary) algoritmalarını kullanarak çözmüşler ve CNN algoritmasının diğer algoritmalarından daha iyi olduğunu belirtmişlerdir [13].

Bengt, Kinetik TSP'nin bazı versiyonları için karmaşıklık hesaplaması yapmış ve bu versiyonların P ya da NP sınıfına ait olma durumlarını incelemiştir [14].

Knapp ve Rothe, düşman unsurlardan gelen top, roket gibi tehditlerin lazer silahları yardımıyla etkisiz hale getirilmeye çalışıldığı Silah-Hedef Atama Problemini simülasyon yöntemini kullanarak çözmeye çalışmışlardır [15].

Bourjolly ve ark., dünya yörüngesinde sabit bir açı ve hızla hareket eden 20 adet uyduya, gezici bir yakıt ikmali deposu tarafından, minimum görev zamanı ve enerji kullanımı

amacı doğrultusunda yakıt ikmali yapıp, tekrardan gezicinin orijine döneceği tur rotasını, küçük boyutlu problemlerde kapsamlı bir algoritma, büyük boyutlu problemlerde tabu arama algoritması ile belirlemişlerdir. Bu bağlamda, uyduların 2002-2007 yılları arasında, üç boyutlu düzlemde her bir dakika içerisindeki koordinatlarını hesaplamışlar ve analiz sonucunda her iki algoritmanın uygun sonuçlar ürettiğini tespit etmişlerdir [16].

Blough ve ark., Jüpiter gezegeninin L4 ve L5 yörüngesinde hareket eden asteroidlerin izlenmesi ve ziyaret edilmesi görevlerini gerçekleştirecek uzay aracının 2013-2022 yılları arasında rotalanması problemini, minimum yakıt tüketimi amacı doğrultusunda Genetik Algoritma ve Sıralı Buluşma Noktası Metodunu kullanarak çözmüşlerdir. Uygulama çalışması sonucunda, Sıralı Buluşma Noktası Metodunun, Genetik Algoritmadan daha iyi sonuçlara ulaştığını belirtmiştir [17].

Mei ve ark., uzayda hareket eden nesnelerin uydular tarafından izlenmesi problemini, hedeflerin hareketlerindeki belirsizliği azaltmaya amaçlayan bir çizelgele algoritması ile çözmeye çalışmışlardır. Önerilen algoritmanın etkinliğini, Rassal Seçim Algoritması(RA - Random Selection Algorithm), Algılama Olasılık Algoritması(Detection Probability Algorithm - DPA), Bilgi Algoritması(Information Algorithm-IA) ve Kapsam Algoritması (Comprehensive Algorithm) ile karşılaştırarak test etmişlerdir. Önerilen çözüm yaklaşımında, hareketli nesnelere ilişkin bilgiler ve nesnelerin algılanma olasılıklarının birlikte değerlendirilmesi nedeniyle başarılı sonuçlar üretildiğini belirtmişlerdir [18].

Groba ve ark., nehirlerdeki balık toplama cihazlarının izleyeceği rotayı ve ağların toplanacağı zamanın belirlenmesi problemini, tahmin tekniklerine dayalı Genetik Algoritma (GATP) yöntemiyle çözmüşlerdir. Önerilen çözüm yaklaşımının(GATP), GA-GSP ve NN algoritmalarından daha başarılı sonuçlara ulaştığını belirtmişlerdir [19].

Mercer ve arkadaşları, Avusturalya'daki deniz gözetleme problemini, genetik algoritma yöntemini kullanarak, uçakların toplam tur uzunluğunun minimize edilmesi amacı doğrultusunda çözmüştür. Çalışma sonucunda, uçakların minimum tur uzunluğunu elde etmesi için, tarama aralıklarının farklı olması gerektiğini, tarama alanı daraldıkça gözden kaçan gemilerin olabileceğini ve uçakların minimum yarıçap ile dönüş yapabilmesi için toplam tur uzunluğunu %20 oranında artırılması gerektiğini tespit etmişlerdir [20].

Killby ve ark., deniz gözetleme operasyonunda kullanılan bir uçak tarafından denizde harekete eden gemilerin izlenmesi problemini, minimum taranamayan gemi sayısı ve minimum toplam uçuş zamanı amacı doğrultusunda, genetik algoritma kullanarak çözmüşlerdir. Problemde, uçak ve gemilere ait hız, yön ve başlangıç lokasyonunun bilindiğini varsaymışlardır. Buna ek olarak, sistem içerisinde hareketli ve sabit nesnelerin bulunduğunu ifade etmişler ve zaman penceresi ile öncelik kısıtlamalarının dikkate alındığını belirtmişlerdir [21].

Marlov ve ark., sınırlı bir görev alanı içerisinde, sabit hızlı ve tarama aralıklı uçak tarafından, gemi sayısı ve hızının zaman içerisinde değiştiği deniz gözetleme problemini, NN tekniği ve “2-opt”, “2-opt Stationary Ships”, “2-opt Jumping Ships” sezgisellerini kullanarak çözmüş ve uygulama çalışması sonucunda, “2-opt” ve “2-opt Stationary” algoritmalarının daha iyi sonuçlara ulaştığını belirtmişlerdir [22].

Fang ve ark., denizlerdeki hareketli nesnelerin, deniz güvenlik araçları tarafından tehlikelere karşı korunması problemini, tek ve iki boyutlu koordinat ekseninde, güvenlik aracı sayısının tek ve çoklu olması varsayımı altında, CASS algoritmasını kullanarak çözmüşler ve zarar gören hareketli nesne sayısını minimize etmeyi amaçlamışlardır [23].

Cross ve ark., okyanustaki sınırlı bir alan içerisinde rassal açı ve hızlarda hareket eden gemilerin, sabit hızla hareket eden deniz gözetleme uçağı tarafından tamamının tarandığı ve sınıflandırıldığı, minimum uçuş zamanlı tur rotasını belirlemeye çalışmışlardır [24].

Bimbo ve Pernici, zoom yapabilme özelliğine sahip kameranın görüntüleme açısı içerisinde maksimum sayıda hareketli nesnenin gözetlenmesi problemini, minimum gözlem süresi amacı doğrultusunda optimal bir çözüm yaklaşımıyla çözmüşler ve kamera sayısı arttıkça, gözlem alanının arttığını ve maliyetin düştüğünü belirtmişlerdir [25].

Bimbo ve Pernici, PTZ kamerasını kullanarak hareketli hedeflerin ilgili görüş aralığı içerisinde izlenmesi için uygun tur rotasını sezgisel bir algoritma ile belirlemeye çalışmışlar ve kaçan hedef sayısının tespitinde Monte Carlo Simülasyonundan yararlanmışlardır [26].

Ilavarasi ve Joseph, Fayda Tabanlı, Zaman Pencere, Maksimizasyon Tabanlı ve Kinetik Tabanlı GSP problemlerinin, özelliklerini ve formülasyonlarını tanımlamışlardır. Buna ek olarak, Kinetik GSP'yi, vurucu ve hedeflerin tek bir eksen üzerinde hareket ettiği ve sistemde bazı hedeflerin hareketli bazı hedeflerin sabit olduğu durumlar açısından incelemişlerdir [27].

Asahiro ve ark., orijin noktasından düz bir doğrultuda hareket eden k tane robotun, n tane hareketli topu yakalaması problemini öklid düzlemi içerisinde incelemiş ve HHGSP'nin NP-Zor problem grubunda olduğunu belirtmişlerdir. Problemi, (1) k tane robotun, n tane topu toplayabildiği durum, (2) k tane robot tarafından toplanabilecek maksimum top sayısının belirlenmesi ve (3) n tane topu yakalayabilecek minimum robot sayısının tespit edilmesi olmak üzere 3 farklı durum açısından ele almışlardır. Bu durumları, hareket rotasının bilindiği, bilinmediği ve aynı olduğu varsayımı altında değerlendirmişlerdir. Buna ek olarak, robotların, sınırlı uzunlukta bir yol üzerinde hareket ettiği ve bu sınırları geçemeyeceği belirtmişlerdir. Problemin çözümü için sezgisel bir algoritma geliştirmişler ve algoritmanın polinom zamanı içerisinde çözüm üretebildiğini ifade etmişlerdir [28].

Asahiro ve ark., hızı zaman içerisinde değişmeyen ve önceden belirlenmiş bir rota üzerinde hareket eden n adet ürünün, kapasiteli bir robot kolu tarafından yakalanıp depoya iletilmesi için uygun tur rotasının belirlenmesi problemini, maksimum sayıda ürünün yakalanması amacı doğrultusunda 2 faktörlü yaklaşım algoritmasını kullanarak çözmüşler ve ilgili problemin MAXSNP-Hardness sınıfında olduğunu belirtmişlerdir [29].

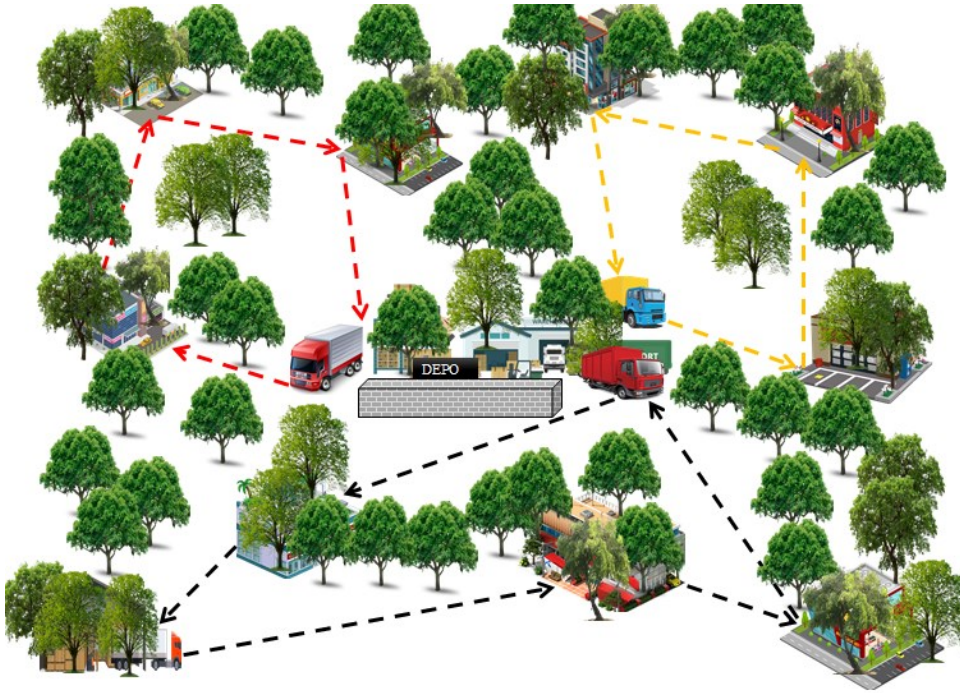
Chalasani ve ark., sınırlı kapasiteli bir robot tarafından konveyör bandı üzerinde gelen ürünlerin kavranması ve ilgili teslimat noktasına yerleştirilmesi problemini, ürünlerin sabit bir hızla veya hareketsiz olduğu varsayımı altında bir yaklaşım algoritması kullanarak çözmüşler ve minimum görev süresi içerisinde maksimum sayıda ürünün yakalanmasını amaçlamışlardır [30].

Uçar ve İşleyen, zaman penceresi ve araç kapasite kısıtları altında, hareketli hedefli, heterojen filolu İHA rotalama problemini, minimum araç sayıcı amacı doğrultusunda AHP yöntemine dayalı sezgisel bir algoritma ile çözmüşlerdir. AHP yöntemini kullanarak, operasyonda görev alacak araçlar ve imha edilecek hedefler için bir öncelik sırası oluşturmuşlardır. Önerilen algoritmayı, farklı sayıda vurucular ve hedeflerden oluşan

problem setleri üzerinde test etmişler, analiz sonucunda algoritmanın kabul edilebilir çözüm süreleri içerisinde uygun çözümler ürettiğini belirtmişlerdir [31].

2.2. Dinamik Araç Rotalama Problemi

Dantzig ve Ramser'in 1959 yılında literatüre önerdiği Araç Rotalama Problemi, merkezi lokasyonda bulunan bir depodan harekete başlayan gezici bir araç ya da araç filosu tarafından sistem içerisindeki müşteri taleplerinin minimum maliyetle karşılanacağı tur rotasının belirlenmesi problemi olarak tanımlanmaktadır [32,33]. Probleme ilişkin örnek bir gösterim Şekil 2.2'de gösterilmektedir.



Şekil 2.2. Araç rotalama problemi

Klasik Araç Rotalama Probleminde, $G(D, Y)$ grafında, $D = \{d_1, d_2, \dots, d_n\}$ düğümler kümesini ve $Y = \{(y_i, y_j) : i \neq j, y_i, y_j \in D\}$ düğümler arasındaki yolların kümesini ifade etmektedir. D kümesi içerisinde " d_1 " elemanı " n " tane aracın bulunduğu merkez depoyu, diğer " d_i " elemanları ise sistemdeki müşterileri temsil etmektedir. Buna ek olarak, Y kümesine göre tanımlanan $M = (m_{i,j})$ matrisi, iki düğüm arasındaki seyahat zamanını, seyahat maliyetini ya da uzaklığını belirtmektedir. Klasik ARP'de müşterilere ya da düğümlere ilişkin talepler önceden bilinmekte ve bu talepler ile sistemdeki müşteri

sayılarında zaman içerisinde herhangi bir değişme olmamaktadır. Merkez depodaki araçlar homojen olmakta, kapasite, yakıt maliyeti vb. parametreler cinsinden araçların arasında herhangi bir farklılık bulunmamaktadır. Problemdeki temel amaç, araç kapasitesinin aşılmadan, merkezi depodan başlanıp, tüm müşterilerin ziyaret edilip tekrar merkezi depoya döneceği minimum maliyetli araç rotasını belirlemektir [33]. Literatürde ARP farklı kriterlere göre çeşitli şekillerde sınıflandırılabilmekte, çevre durumu kriterine göre ARP, Statik ve Dinamik ARP olmak üzere iki gruba ayrılmaktadır. Statik ARP 'de problem içerisindeki parametrelerde zaman içerisinde değişme meydana gelmezken, 1988 yılında Psaraftis'in literatüre önerdiği D-ARP'de problem içerisindeki parametreler zaman içerisinde değişmekte ve problem ARP'nin genel bir versiyonu olarak ifade edilmektedir [34]. Örneğin iki düğüm arasındaki uzaklık ya da seyahat zamanı, trafik yoğunluğuna ya da alternatif yol güzergâhlarına göre değişiklik gösterebilmektedir. Buna ek olarak müşteri taleplerinin zaman içerisinde değişmesi veya herhangi bir anda yeni bir müşteri düğümünün sisteme dahil olması, zaman içerisindeki parametre değişikliklerine örnek olarak verilebilmektedir [35-47]. İHA'ların, filo halindeki operasyonlarda rotalanması problemi de Dinamik ARP kapsamında incelenebilmekte, literatürde belirtilen problemin çözümü için matematiksel modelleme, tamsayılı programlama, sezgisel ve metasezgisel algoritmalar gibi çözüm yaklaşımları kullanılabilmektedir [25-31,48].

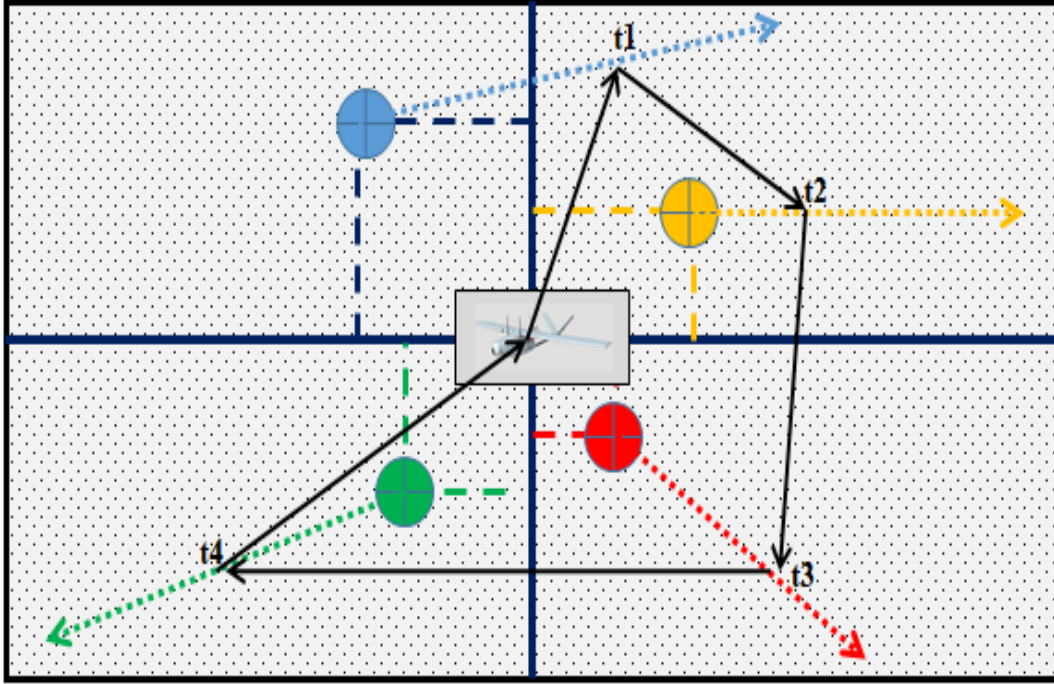
3. TEK VURUCULU-HAREKETLİ HEDEFLİ GEZGİN SATICI PROBLEMİ İÇİN TAVLAMA BENZETİMİNE DAYALI BİR ÇÖZÜM YAKLAŞIMI

Günümüzde birçok ülke, olası terörist sızmalarını, mobil terörist birliklerini, mühimmat depolarını ya da drone ve İHA gibi casus araçlarını, havadan operasyonlarla hızlı ve kayıpsız bir şekilde imha etmeye çalışmakta, bunları yaparken de büyük maliyetlere katlanmaktadır. Operasyonların etkin, verimli ve hızlı bir şekilde gerçekleşmesi, vurucu araçların doğru bir şekilde rotalanması ile mümkün olmaktadır. Tez çalışmasının bu bölümünde hareketli hedeflerin, tek bir vurucu tarafından hava operasyonu ile imha edilmeye çalışıldığı HHGSP ele alınmıştır. Problemin temel varsayımları aşağıda ifade edilmiştir.

- Vurucu, tüm hedeflerde daha hızlı olmalıdır.
- Vurucu, tüm hedefleri imha ettikten sonra başlangıç noktasına geri dönmelidir.
- Hedefler, (x,y) koordinat ekseninde, herhangi bir hız ve açı ile hareket edebilirler.
- Vurucu ve hedeflerin hızları zaman içerisinde değişmemektedir.
- Hedef açısı zaman içerisinde değişmemektedir.
- Vurucu ve hedeflerin yakıtları sınırsızdır.
- Vurucu ve hedefler için görev zamanı ve görev alanı sınırlandırılmamıştır.
- Vurucu ilgili hedefi yakaladığı an, ilgili hedefin etkisiz hale getirildiği ya da gözlemlendiği varsayılmakta ve bu andaki operasyon için görev süresi “0” olarak tanımlanmaktadır.
- Hedef hızlarının [300-400] birim yol/birim zaman, vurucun hızının [401-600] birim yol/birim zaman, hedef ve vurucu açılarının [0-360] derece arasında, hedeflerin başlangıç koordinatlarının ise $[\pm 400, \pm 400]$ birim yolları arasında olduğu varsayılmaktadır.

İlgili problemin görsel gösterimi Şekil 3.1’de gösterilmektedir. Bu şekilde, belirli bir açı ve hızla hareket eden 4 adet hedef ve 1 tane vurucu bulunmaktadır. Temel amaç, vurucunun orijin noktasından başlayıp, tüm hedefleri imha edip, tekrar orijine döneceği minimum görev zamanlı turu bulmaktır. Şekil 3.1’e göre vurucu ilk olarak “t1” zamanında mavi hedefi, “t2” anında turuncu hedefi, “t3” anında kırmızı hedefi ve “t4” anında yeşil hedefi imha edip, orijin noktasına dönmektedir. Şekil 3.1’de belirtilen problem, jet uçağının,

helikopterin ya da mühimmat yüklü bir insanız hava aracının, hareket halindeki hedefleri imha etmesi ile aynıdır. Buna ek olarak, tez çalışmasında hedefler, koordinat düzleminin dört farklı bölgesi içerisinde hareket edebilmektedir.



Şekil 3.1. Tek vuruculu-hareketli hedefli gezgin satıcı problemi

Problemin matematiksel modelinin genel formülasyonu aşağıda ifade edilmektedir (Stieber et al., 2015).

Notasyonlar

- N : hedef sayısı
- I : hedef indeksi
- V : hedef seti $\{1, \dots, n\}$
- A : $A \subseteq V \times V$
- $x_{i,j}$: $\begin{cases} 1, & \text{Eğer vurucu } i \text{ hedefinde } j \text{ hedefine hareket ederse} \\ 0, & \text{diğer durumda} \end{cases}$
- $s_{i,j}$: Vurucunun "i". hedeften ayrılma zamanı
- $t_{i,j}$: Vurucunun "j". hedefe varma zamanı
- $c_{i,j}$: "s" zamanında "i" düğümünden ayrılıp, "t" zamanında "j" düğümüne ulaşması için gereken seyahat zamanı , $[0, T] \times [0, T]$
- T : $R^+ \cup \{\infty\}$

Amaç fonksiyonu ve kısıtlar

$$\text{Min} \sum_{i=0}^n \sum_{j \neq i, j=0}^n c_{i,j}(s_{i,j}, t_{i,j}) * x_{i,j} \quad (3.1)$$

$$\sum_{i:(i,j) \in A} x_{i,j} = 1 \quad \forall j \in V \quad (3.2)$$

$$\sum_{j \in V} x_{i,j} \leq 1 \quad \forall i \in V \quad (3.3)$$

$$\sum_{i \in V} x_{i,j} \geq \sum_{i \in V} x_{j,i} \quad \forall j \in V \quad (3.4)$$

$$s_{i,j} \leq T * x_{i,j}, \quad t_{i,j} \leq T * x_{i,j} \quad \forall (i,j) \in V \quad (3.5)$$

$$s_{i,j} \leq t_{i,j} \quad \forall (i,j) \in V \quad (3.6)$$

$$\sum_{i \in V} t_{i,j} + \sum_{i \in V} c_{j,i}(s_{j,i}, t_{j,i}) * x_{j,i} \leq \sum_{i \in V} t_{j,i} + T * \left(1 - \sum_{i \in V} x_{j,i}\right) \quad \forall j \in V \quad (3.7)$$

Amaç fonksiyonunda (3.1), toplam görev zamanının minimize edilmesi amaçlanmıştır. Denklem (3.2)'ye göre, her hedef bir kez ziyaret edilmelidir. Denklem (3.3)'de, bir hedefin en fazla bir kez ziyaret edilebileceği ifade edilmektedir. Denklem (3.4), vurucunun, düğümler (hedefler) arasındaki geçişini sağlamaktadır. Bu denkleme göre, vurucu bir düğümü ziyaret ettikten sonra, bir sonraki düğüme geçiş yapabilir. Denklem (3.5)' de, eğer vurucu “i” düğümünden “j” düğüme geçiş yaparsa, ilgili düğümler için ayrılış ve varış zamanları tanımlamaktadır. Denklem (3.6)'da, eğer vurucu “i” düğümünden sonra “j” düğüme hareket ederse, “j” düğüme varış zamanı, “i” düğümünden ayrılış zamanından daha büyük olmalıdır. Son olarak denklem (3.7)'de, herhangi bir düğüme varış zamanı, önceki düğüme varış zamanından daha büyük olmalıdır.

Problemde, hedefler hareket etmekte ve çözüm süreci boyunca vurucu ile hedeflere ait koordinat bilgilerinin sürekli olarak hesaplanması gerekmektedir. Bu nedenden dolayı problemi matematiksel modelleme yaklaşımlarıyla çözmek zor olmaktadır. Buna ek olarak, sürekli hesaplama gereksinimi altında, hangi hedefin, ne zaman, hangi koordinatlarda imha edileceğinin belirlenmesi ise büyük bir önem arz etmektedir. Problemin, “n!” çözümü bulunmakta, bu çözümler arasından minimum zamanlı tur rotasını belirlemek oldukça zaman almaktadır. Problemdeki hedef sayısının az olduğu durumlarda enumerasyon yöntemi ile optimal çözümlere ulaşılabilen, çözüm sayısı arttıkça ilgili çözüm

yöntemiyle kabul edilebilir çözüm süresi içerisinde uygun çözümler bulmak oldukça zor olmaktadır. Askeri operasyonlar, problemin önemli uygulama alanlarından birini oluşturmakta, bu operasyonlarda ulusal güvenliğin sağlanması için hızlı ve etkili çözümler üretilmesi gerekmektedir. Bir sonraki bölümde, problemi hızlı ve etkili bir şekilde çözmek için kullanılacak çözüm metodolojilerine ilişkin bilgiler verilmektedir.

3.1. Çözüm Metodolojisi

Literatürde, önceki bölümde ifade edilen problemi çözmek için farklı çözüm yaklaşımları bulunmaktadır. Problemin dinamik bir yapıya sahip olması ve problem içerisindeki parametrelerin sürekli değişme nedeniyle, önerilen çözüm yaklaşımlarının büyük bir çoğunluğunu probleme özgü sezgiseller ya da metasezgisel algoritmalar oluşturmaktadır. Buna ek olarak, problemin güncel olması ve geniş bir uygulama alanına sahip olması nedeniyle jenerik ve etkin çözüm metodolojilerine sıklıkla ihtiyaç duyulmaktadır.

Bu tez çalışmasında, etkin ve jenerik çözümler üretmek için tavlama benzetimine dayalı bir çözüm yaklaşımı geliştirilmiş, algoritmanın daha iyi sonuçları üretmesi için başlangıç çözümünde kullanmak amacıyla altı farklı çözüm kurucu sezgisel algoritma önerilmiştir. Bu sezgisel algoritmalar aşağıda ifade edilmektedir.

- Minimum Yol Sezgiseli (The Minimum Distance Heuristic – MDH)
- Bağıl Zaman Sezgiseli (Relative Time Heuristic - RTH)
- Bölge İmha Sezgiseli (Region Destruction Heuristic – RDH)
- Güncelleme Tabanlı Bölge İmha Sezgiseli (Update Based Region Destruction Heuristic-UBRDH)
- Rassal Arama (Random Search – RS)
- Minimum Zaman Sezgiseli (The Minimum Time Heuristic) - MTH

Geliştirilen çözüm kurucu sezgisellerden RTH, RDH ve UBRDH algoritmalarının HHGSP literatüründe bilindiği kadarıyla ilk kez kullanıldığı tahmin edilmektedir. Algoritmalara ilişkin çözüm adımları aşağıda ifade edilmektedir.

3.1.1. Minimum yol sezgiseli

Numara Minimum Yol Sezgiselinde, vurucu konum olarak kendine en yakın hedefi imha

ederek tura başlamakta ve sonraki hedefleri de yakınlıklarına göre imha edip orijine dönerek turu sonlandırmaktadır. Yöntemin adımları Çizelge 3.1’de gösterilmektedir.

Çizelge 3.1. Minimum yol sezgiseli

Adım 1: İmha edilmiş hedefler ve imha edilmemiş hedefler listesini tanımla.
 Adım 2: Vurucu ve hedefler arasındaki uzaklığı hesapla.
 Adım 3: En yakın uzaklığa sahip hedefi seç ve imha et.
 Adım 4: Seçilen hedefi imha edilmiş hedefler listesine ekle ve imha edilmemişler listesinden çıkar.
 Adım 5: Vurucu ve imha edilmemiş hedeflerin konumları ile görev zamanı bilgisini güncelle.
 Adım 6: İmha edilmemiş hedefler listesi boş küme ise Adım 1’e, değilse Adım 7’ye git.
 Adım 7: Bitir.

3.1.2. Bağlı zaman sezgiseli

Bağlı Zaman Sezgiselinde, hedeflerin koordinatları ve açısı ile vurucunun koordinatları dikkate alınarak her bir hedefi etkisiz hale getirmek için bir “ t_i ” zamanı hesaplanmakta ve bu zamanlar içerisinde minimum “ t_i ” değerine sahip hedef seçilerek imha edilmektedir. Tüm hedefler imha edilinceye kadar bu işlemler tekrarlanmakta ve sistemde hedef kalmayınca tur sonlanmaktadır. İlgili yöntemin adımları Çizelge 3.2’de gösterilmektedir.

Çizelge 3.2. Bağlı zaman sezgiseli

Adım 1: Her bir hedef için vurucuya göre minimum zamanını aşağıdaki kurala göre bul ve adım 2 ye git.
 Kural 1: Eğer vurucunun y eksenindeki konumu, hedefin y eksenindeki konumundan büyük ve hedefin açısı 180° küçük ise:

$$\text{zaman} = \frac{d_i}{V_{\text{pursuer}} + V_{\text{target}}}$$

 Kural 2: Eğer vurucunun y eksenindeki konumu, hedefin y eksenindeki konumundan büyük ve hedefin açısı 180° büyük ise:

$$\text{zaman} = \frac{d_i}{V_{\text{pursuer}} - V_{\text{target}}}$$

 Kural 3: Eğer vurucunun y eksenindeki konumu, hedefin y eksenindeki konumundan küçük ve hedefin açısı 180° büyük ise:

$$\text{zaman} = \frac{d_i}{V_{\text{pursuer}} + V_{\text{target}}}$$

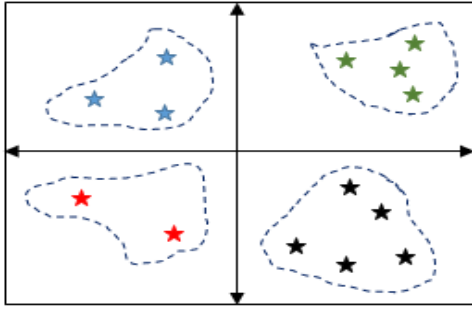
 Kural 4: Eğer vurucunun y eksenindeki konumu, hedefin y eksenindeki konumundan küçük ve hedefin açısı 180° küçük ise:

$$\text{zaman} = \frac{d_i}{V_{\text{pursuer}} - V_{\text{target}}}$$

 Adım 2: Minimum zamana sahip hedefi seç ve imha et.
 Adım 3: Seçilen hedefi imha edilmiş listesine ekle ve imha edilmemişler listesinden çıkar
 Adım 4: Vurucu ve imha edilmemiş hedeflerin konumları ile görev zamanı bilgisini güncelle.
 Adım 5: İmha edilmemiş hedefler listesi boş küme ise Adım 1’ e değilse Adım 6 ya git.
 Adım 6: Bitir

3.1.3. Bölge imha sezgiseli

Bölge İmha Sezgiseli'nde, hedefler, başlangıç koordinatlarına göre kümelenmekte, her bir küme içerisinde hedefler $\min\left(\frac{d_i}{v_i}\right)$ oranına göre sıralanmakta ve en fazla elemana sahip kümeden başlanarak, tüm hedefler imha edilmektedir. Bölge İmha Sezgiseli'nin görsel gösterimi Şekil 3.2'de, sezgiselin adımları ise Çizelge 3.3'de gösterilmektedir.



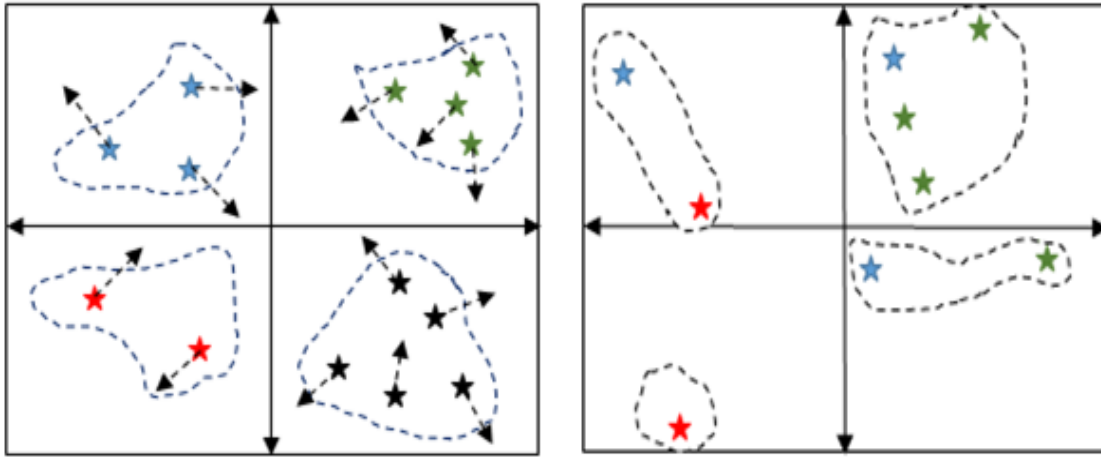
Şekil 3.2. Bölge imha sezgiseli

Çizelge 3.3. Bölge imha sezgiseli

- | |
|--|
| <p>Adım 1: Hedefleri bölgelerine göre kümelenir.</p> <p>Adım 2: Kümeleri eleman sayısına göre küçükten büyüğe doğru sırala ve en fazla elemana sahip olan kümeyi imha etmek için seç.</p> <p>Adım 3: Seçilen kümedeki her bir hedef için, vurucuya göre (yol/hız) oranını hesapla ve en düşük oranlı hedefi vurmak için seç. $\text{Oran} = (\text{hedef} - \text{vurucu yol}) / (\text{hedef hız})$</p> <p>Adım 4: İlgili bölgedeki tüm hedefler imha edildiyse, sıradaki en fazla elemana sahip kümeyi imha için seç ve Adım 3 deki işlemleri uygula, değilse adım 5 e git</p> <p>Adım 5: Tüm kümeler imha edildiyse Adım 6' ya, değilse Adım 3' e git.</p> <p>Adım 6: Bitir.</p> |
|--|

3.1.4. Güncelleme tabanlı bölge imha sezgiseli

UBRDH sezgiselinde hedefler, bölgelere göre kümelere ayrılmış, kümeler eleman sayısına göre büyükten küçüğe sıralanmış ve en fazla elemana sahip küme imha için seçilmiştir. Seçilen kümedeki hedefler $\min(d_i/v_i)$ oranına göre imha edilmiştir. Kümedeki tüm elemanlar imha edildikten sonra, imha edilmeyen hedefler, o anda bulundukları bölgelere göre yeniden kümelendirilmiş ve en fazla elemana sahip küme imha için seçilip, küme içerisindeki elemanlar $\min(d_i/v_i)$ oranına göre etkisiz hale getirilmiştir. Tüm hedefler imha edildikten sonra ise algoritma sonlandırılmıştır. Eğer, eleman sayıları aynı ise önce 4, sonra 3, sonra 2 ve en son 1. küme imha için seçilmiştir. Algoritmaya ilişkin görseller Şekil 3.3'de, algoritmanın adımları ise Çizelge 3.4'de gösterilmiştir.



Şekil 3.3. UBRDH sezgiseli (başlangıç ve ilk küme imha edildikten sonraki durum)

Çizelge 3.4. Güncelleme tabanlı bölge imha sezgiseli

Adım 1: Hedefleri bölgelerine göre kümelendir. Adım 2: Kümeleri, eleman sayısına göre büyükten küçüğe sırala ve en fazla eleman sayısına sahip kümeyi imha için seç. Adım 3: Seçilen kümedeki hedefleri $\min(d_i/v_i)$ oranına göre imha et. Adım 4: Kümedeki tüm elemanlar imha edildikten sonra, imha edilmeyen hedefleri, o anda bulundukları bölgelere göre yeniden kümelendir ve adım 3 deki işlemi uygula. Adım 5: Kümelerdeki tüm hedefler imha edildiyse Adım 6'ya, değilse Adım 4'e git. Adım 6: Bitir.
--

3.1.5. Rassal arama

Rassal Arama Metodunda, algoritma farklı hedef sıralamaları için n kez çalıştırılmakta ve minimum görev süresi ya da görev maliyetine sahip tur rotası algoritma çıktısı olarak sunulmaktadır. Uygulama çalışmasında her bir senaryo için algoritma 300 kez koşturulmakta ve en iyi rota, o problem için uygun tur rotası olarak sunulmaktadır.

3.1.6. Minimum zaman sezgiseli

Minimum zaman sezgiselinde, vurucunun her bir hedefi imha etme süresi hesaplanmakta ve en kısa sürede yok edeceği hedef, imha etmek için seçilmektedir. Tüm hedefler imha edildikten sonra ise algoritma sonlanmaktadır. MTH sezgiseli, HHGSP problem kapsamında çokça başvurulan yöntemlerden biri olmaktadır. Bu sezgisele ilişkin algoritmanın adımları Çizelge 3.5'de gösterilmektedir.

Çizelge 3.5. Minimum zaman sezgiseli

Adım 1: İmha edilmiş hedefler ve imha edilmemiş hedefler listesini tanımla.
Adım 2: Her bir hedef için vurucunun imha edeceği süreyi hesapla.
Adım 3: En kısa sürede imha edeceği hedefi seç ve imha et.
Adım 4: Seçilen hedefi imha edilmiş hedefler listesine ekle ve imha edilmemiş hedefler listesinden çıkar
Adım 5: Vurucu ve imha edilmemiş hedeflerin konumları ile görev zamanı bilgisini güncelle.
Adım 6: İmha edilmemiş hedefler listesi boş küme ise Adım 1' e, değilse Adım 7 ye git.
Adım 7: Bitir

3.1.7. Strateji tabanlı tavlama benzetimi algoritması

Tezin bu aşamasında belirtilen problemin çözümü için Strateji Tabanlı Tavlama Benzetimi Algoritması yöntemi geliştirilmiştir. Algoritma içerisinde çeşitli çözüm stratejileri uygulandığı için yöntem bu isimle adlandırılmıştır. Tavlama benzetimi, katıların fiziksel tavlama sürecinden hareketle, 1983 yılında, Kirckpatrick ve ark., tarafından geliştirilmiş, stokastik bir arama yöntemidir. Algoritma iteratif olarak ilerlemekte ve çözüm kalitesini artırmak için çeşitli hareket mekanizmaları kullanılmaktadır. Çözüm kalitesini artıran hareketler sonraki adımda başlangıç çözümü olarak kabul edilirken, çözüm de iyileştirme yapmayan hareketler yerel optima takılmamak için belirli bir yüzde ile kabul edilmektedir. Kötü çözümün kabul olasılığında ise Boltzmann olasılık faktörü ($P = e^{(-\Delta/T)}$) kullanılmaktadır [49]. Büyük boyutlu problemin çözümünde, bilgiyi kullanması ve aramayı iyi yapması nedeniyle, sıklıkla başvurulmuş bir çözüm yöntemi olmaktadır. Bu çalışmada istatistiksel olarak optimal bulduğunu garanti etmesi, sürekli optimizasyon problemlerini ele alması ve çeşitli maliyet fonksiyonları ile bu maliyetlere ait kısıtlamalar altında kullanılabilir olması nedeniyle belirtilen problemin çözümünde tavlama benzetimi algoritması kullanılmıştır [50-51]. Algoritmanın etkinliğini artırmak için Bölüm 3.6 içerisinde belirtilen çözüm kurucu sezgisellerden faydalanılmıştır. Belirlenen sezgisel algoritma ile tavlama benzetimi algoritması için hızlı ve kalite bir şekilde başlangıç çözümü oluşturulmaya çalışılmıştır.

Tavlama benzetimi algoritması uygulanırken verilmesi gereken bazı kararlar vardır. Bunlar çözümün kalitesine ve çözüm hızına etki eden kararlardır. Bu bağlamda verilmesi gereken kararlardan ilkinin başlangıç sıcaklığı oluşturmaktadır. Başlangıç sıcaklığı, kötü çözümleri kabul edecek ve elde edilen son çözümün, başlangıç çözümünden bağımsız olmasını sağlayacak kadar yüksek olmalıdır. Başlangıç sıcaklığından sonra karar verilmesi gereken bir diğer parametre soğutma planıdır. Soğutma planı, optimizasyon algoritmasının

başarısına etki eden bir diğer önemli karardır. Bu planda en önemli parametrelerden birini soğutma oranı oluşturmaktadır. Bu oran, kabul olasılığına etki etmekte ve her bir iterasyondaki sıcaklık değerini belirlemektedir. Soğutma oranı denemelerde genel olarak 0.5 ile 0.99 arasında olmaktadır [51]. Ayrıca geliştirilen algorithmada sıcaklık değerlerini güncellemek için geometrik soğutma çizelgesi kullanılmaktadır. Durdurma kriteri olarak son sıcaklık değeri dikkate alınmakta ve algoritma her bir senaryo için 30 kez çalıştırılarak ortalama ve en iyi çözüm değerleri belirlenmeye çalışılmaktadır. Komşuluk değişim mekanizması olarak ise swap ve insert yöntemi kullanılmakta, her bir iterasyonda aranacak komşu sayısı, swap ve insert mekanizması ile elde edilecek olası tüm çözümlerin sayısı kadar olmaktadır. Algoritma içerisindeki parametrelerin belirlenmesinde Taguchi yönteminden yararlanılmakta ve parametre seti hakkındaki bilgiler Bölüm 3.2.2 içerisinde verilmektedir. Buna ek olarak, çalışmada çözüm süresini kısaltmak ve yerel çözüme takılmamak için üç farklı strateji kullanılmakta, bu stratejiler aşağıda ifade edilmektedir.

- Tur hesaplamasının eliminasyonu
- Hafızalama
- Çözüm bozma mekanizması.

Tur hesaplamasının eliminasyonu

Bu bölümde belirtilen problemin çözümünde tavlama benzetimi yöntemi kullanılmakta ve bu yöntem tek bir çözüm üzerinden farklı komşu çözümler üreterek ilerleyen iteratif bir yöntem olmaktadır. Burada komşuluk mekanizması olarak swap ve insert yöntemleri kullanılmakta, başlangıç çözümünden hareketle swap mekanizması ile üretilen tüm komşu çözümlerden en iyisi seçilerek insert mekanizmasına aktarılmakta, insert mekanizmasında swaptan gelen çözümün tüm komşu çözümleri üretilmekte ve en iyisi seçilerek çözüm kabul fonksiyonuna aktarılmaktadır.

Önerilen algorithmada tüm çözümler, başlangıç çözümünden hareketle üretilmekte ve bu çözümünden daha iyi bir çözüm var ise ilgili arama mekanizması içerisinde en iyi çözüm bilgisi kendini güncellemektedir. İlgili çözümün tespiti için ise komşu çözüme ait tüm turlar hesaplanmaktadır. Bu çalışmada bir komşuya ait tur hesaplaması yapılırken, en iyi çözüm bilgisi hedef imha mekanizması işlerken güncellenmekte ve bu sayede gereksiz tur hesaplamalarından kaçınılmaktadır.

Örneğin bir senaryoya ait başlangıç çözümü $0 - 4 - 1 - 3 - 2 - 5 - 0$ olsun. Bu çözümünden swap mekanizmasına göre ilk olarak $0 - 1 - 4 - 3 - 2 - 5 - 0$ çözümü elde edilsin ve ilgili komşuluk için en iyi çözüm değeri 15 olsun. Daha sonra swap mekanizmasına göre ikinci çözüm(4 ile 3 yer değiştiği zaman) için $0 - 3 - 1 - 4 - 2 - 5 - 0$ turu üretilirken 0-3-1 değeri hesaplandıktan sonra en iyi çözümünden(15) daha büyük bir tur zamanı elde edilmişse hesaplama sonlanmakta, sonraki hedefler imha edilmemekte ve bir sonraki komşu çözüme(0-2-1-3-4-5-0) geçilmektedir. Eğer küçükse sonraki hedefin imha edilmesi durumu yani 0-3-1-4 turu hesaplanıp yine aynı karşılaştırma işlemi gerçekleştirilmekte ve bu sayede gereksiz tur hesabının elimine edilmesi amaçlanmaktadır. Burada sadece belirli hedefler için değil ilgili çözümdeki her bir hedefin imhasından sonra karşılaştırma işlemi gerçekleştirilmektedir.

Hafızalama

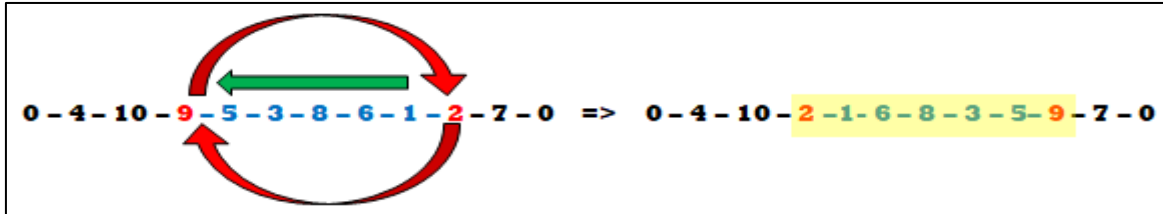
Tavlama benzetiminde çözüm, başlangıç çözümünden hareketle iteratif olarak ilerlemekte ve aynı işlemler farklı veriler üzerinde sürekli olarak tekrarlanmaktadır. Bu çalışmada başlangıç çözümündeki her bir hedefe ait imha süreleri ile hedefin imha edildiği zamanda sistemde bulunan vurucu ve hedeflere ait tüm lokasyon bilgileri hafızaya alınmaktadır. Bu sayede üretilen komşu çözümlerde hafızada tutulmuş olan hedef sıraları ilgili tura aktararak turun gereksiz hesaplamalar yapması engellenmektedir. Bu strateji özellikle hedef sayısının çok olduğu durumlarda büyük bir kolaylık sağlamaktadır.

Örneğin $0-11-20-9-13-17-8-6-15-7-18-3-2-14-19-4-1-12-5-16-10-15-0$ başlangıç çözümünde kırmızı renkli rakamlar yer değiştirildiğinde tüm turun ilgili değişikliğe göre yeniden hesaplanması gerekmektedir. Hafızalama strateji sayesinde $0-11-20-9-13-17-8-6-15-7-18-3-2-14-19-4-1-12-5-16$ olan tur değeri ve 16 numaralı hedefin imha edildiği andaki lokasyon bilgileri hafızada olduğu için bu bilgilerden hareketle sadece $10-15-0$ sırası için tur hesaplama işlemi gerçekleştirilmekte ve 18 tane hedef için tur hesabı yapılmamaktadır.

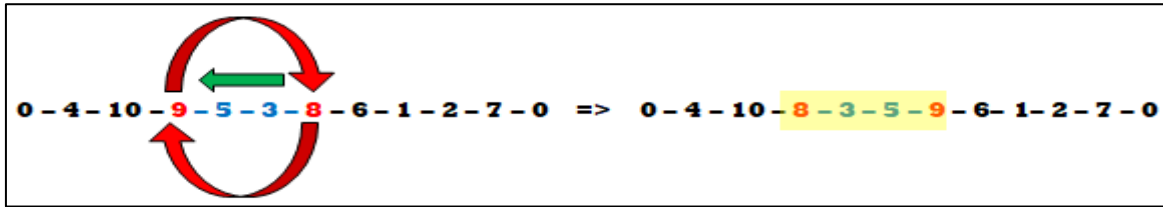
Çözüm bozma mekanizması

Tavlama benzetimi yönteminde çözüm belirli bir iterasyondan sonra yerel optimuma takılabilmektedir. Bu çalışmada belirtilen durumun önüne geçmek için belirli iterasyondan

sonra çözüm iyileşmezse çözüm bozma metodolojisi (perturbation) kullanılmaktadır. Kullanılan mekanizma şekil 3.4’de belirtilmektedir. Bu şekilde ilgili çözümde rassal olarak herhangi iki nokta seçilmekte ve bu iki nokta arasında kalan çözümler tersten sıralanıp yeni bir çözüm oluşturulmaktadır. Bazı durumlarda üretilen rassal sayılar mevcut çözümde çok büyük değişikliğe neden olmakta, bu nedenden dolayı algoritmanın etkinliğini sağlanamamakta ve algoritma rassal aramaya kaymaktadır. Bu çalışmada belirtilen problemlerin önüne geçmek için şekil 3.5’de gösterildiği üzere iki nokta arasındaki fark 3 olarak belirlenmekte ve bu bağlamda iki nokta arasındaki hedefler yer değiştirilerek etkinlik sağlanmaya çalışılmaktadır.



Şekil 3.4. Herhangi iki rassal düğüm arası çözüm bozma mekanizması



Şekil 3.5. Herhangi iki rassal düğüm arası çözüm bozma mekanizması

Belirtilen stratejiler doğrultusunda oluşturulan algoritma Çizelge 3.6’da ifade edilmekte ve bu çözüm yöntemleri kullanılarak Bölüm 3.1.7 içerisindeki algoritmanın etkinliği çeşitli senaryolar üzerinde test edilmektedir.

3.2. Uygulama Çalışması

3.2.1. Problem tasarımı

Önerilen algoritmanın etkinliğini test etmek için, hedef sayısının 5 ile 25 arasında değiştiği 20 farklı senaryo geliştirilmiştir. Senaryolarda, küçük (hedef sayısı 5 ve 10), orta (15 hedef) ve büyük (hedef sayısı 20 ve 25) boyutlu problemler olarak sınıflandırılmıştır.

Çizelge 3.6. Strateji tabanlı tavlama benzetimi algoritması

```

f(Sbest) = double. Maxvalue, Sbest,
while(çözüm zamanı < durdurma kriteri)
    İlgili çözüm kurucu sezgisele göre başlangıç çözümünü(S0) oluştur.
    Başlangıç sıcaklığını(T)100 ve soğutma oranını( $\alpha$ )belirle, çözüm bozma sayısı( $\text{çbs} = 0$ ).
    S  $\leftarrow$  S0, f(S)  $\leftarrow$  f(S0)
    Sgood  $\leftarrow$  S0, f(Sgood)  $\leftarrow$  f(S0)
    while(T  $\geq$  1)
        Başlangıç çözümündeki her bir hedefin vurulduğu anda sistemde bulunan tüm
        hedef ve vurucuların konum ve zaman bilgilerini hafızaya ata.
        Swap mekanizmasına göre çözüm kısaltma ve hafıza özelliğini kullanarak S nin
        tüm komşularını üret, en iyisini S' ye ata, S' yü inserte başlangıç çözümü yap ve S' için
        hafızalama işlemini uygula
        İinsert mekanizmasına göre çözüm kısaltma ve hafıza özelliğini kullanarak S' nün tüm komşu
        çözümlerini üret ve en iyi çözümü S* a ata.
         $\Delta \leftarrow f(S^*) - f(S)$ 
        if  $\Delta \leq 0$  then
            S  $\leftarrow$  S*
            vurucu ve hedeflerin pozisyonlarını ile görev zamanını güncelle
        Else
            u  $\sim$  u(0,1)
            if u <  $e^{(-\Delta/T)}$  then
                s  $\leftarrow$  s*
                vurucu ve hedeflerin pozisyonlarını ve görev zamanını güncelle
            Else
                S  $\leftarrow$  S
            End
        End
        if f(S*) < f(Sgood) then bestsay ++
        if bestsay > çbs then invers mekanizmasını kullanarak S* çözümünü çbs kere
        perturbation yap, en iyi çözümü sonraki iterasyonadaki S başlangıç
        çözümü olarak ata ve çbs = 0 yap.
        if f(S*) < f(Sgood) then
            f(Sgood)  $\leftarrow$  f(S*)
        if f(Sgood) < f(Sbest) then
            f(Sbest)  $\leftarrow$  f(Sgood)
        n  $\leftarrow$  n + 1

    T = T *  $\alpha$ 
End
End

```

Çizelge 3.7. Tek vuruculu HHGSP için senaryo bilgileri

Senaryo	Hedef Sayısı	Araç No	Araç hızı	Araç açısı	Başlangıç Pozisyonu
Senaryo_1	5	Vurucu	538	-	[0, 0]
		Hedef_1	106	64	[-24, 353]
		Hedef_2	106	221	[-60, 306]
		Hedef_3	247	318	[-365, -178]
		Hedef_4	112	177	[-228, -21]
		Hedef_5	91	254	[397, -251]

Bu sayede, hedef sayısı arttıkça algoritmanın çözüm zamanı ve kalitesi üzerindeki etkisi belirlenmeye çalışılmıştır. Geliştirilen senaryolar için örnek bir gösterim Çizelge 3.7’de gösterilmektedir.

3.2.2. Başlangıç çözümünün belirlenmesi

Başlangıç çözümünün belirlenmesi için geliştirilmiş 6 farklı çözüm kurucu sezgisel algoritma, bir önceki bölümde belirtilen senaryolar üzerinde test edilmiş ve bu algoritmalarından en iyisi, Tavlama Benzetimi Algoritması içerisinde başlangıç çözümü olarak kullanılmıştır.

Çizelge 3.8. Çözüm kurucu sezgisellerin karşılaştırılmasına ilişkin deney sonuçları

Senaryo	Hedef Sayısı	MDH	UBRDH	RTH	RDH	RS	MTH
		Zaman	Zaman	Zaman	Zaman	Zaman	Zaman
Senaryo 1	5	7,41	8,76	6,44	6,62	6,13	6,76
Senaryo 2	5	3,84	17,11	4,53	7,75	3,73	4,43
Senaryo 3	5	14,52	21,88	14,52	25,96	8,42	12,59
Senaryo 4	5	12,80	9,86	8,85	12,38	9,86	8,85
Senaryo 5	10	113,37	216,86	165,26	290,20	56,78	165,26
Senaryo 6	10	44,49	72,27	26,08	54,13	56,97	57,64
Senaryo 7	10	58,50	133,57	59,11	82,84	51,48	59,11
Senaryo 8	10	9,90	40,58	22,39	70,30	20,42	22,39
Senaryo 9	15	270,46	1647,60	835,38	11781,98	2100,13	406,15
Senaryo 10	15	156,63	396,55	527,87	518,26	222,95	197,31
Senaryo 11	15	245,93	1440,82	275,74	693,24	685,19	160,74
Senaryo 12	15	1952,46	3113,30	882,99	4325,13	1426,99	1870,04
Senaryo 13	20	295,10	2401,12	492,72	9825,65	9504,54	161,12
Senaryo 14	20	9,43	19,30	17,06	33,42	60,87	7,34
Senaryo 15	20	7,01	27,06	11,62	31,60	26,83	11,56
Senaryo 16	20	308,30	12819,73	399,74	4809,27	11241,48	371,06
Senaryo 17	25	7997,00	24960,24	10825,46	161189,72	77513,61	790,88
Senaryo 18	25	598,56	2580,79	1004,95	12769,00	33890,02	465,11
Senaryo 19	25	501,65	20225,22	5407,66	49071,16	33681,09	1432,93
Senaryo 20	25	5,20	6,13	5,62	9,25	30,89	7,85

Deneylerin tamamı 4 GB RAM ve Intel(R) Core(TM) i5-6200@ 2.3 GHz işlemcili bilgisayarlarda C# programlama dili kullanılarak gerçekleştirilmiştir. Deneysel çalışmalarına ilişkin sonuçlar Çizelge 3.8’de gösterilmektedir.

Başlangıç çözümünde kullanılacak algoritmayı seçmek için Çizelge 3.8’deki sonuçlar istatistiksel olarak analiz edilmiş ve çözüm kurucu sezgisellerin performansını değerlendirmek için Friedman Anova Testi kullanılmıştır. Verilerin normal dağılmaması, grup sayısının üç ya da daha fazla olması ve tüm grup//koşullarda aynı deneklerin kullanılması nedeniyle bu testin kullanılmasına karar verilmiştir. İlgili veri setinin normal dağılıma uygunluğunu test etmek için SPSS programındaki Kolmogorov-Smirnov testinden yararlanılmıştır. İlgili istatistiksel analiz için aşağıdaki hipotez testi kurulmuş, SPSS programında yapılan uygunluk analizi sonuçları Resim 3.1’de ifade edilmiştir.

H0: Veriler normal dağılıma uygunluk göstermektedir.

H1: Veriler normal dağılıma uygunluk göstermemektedir.

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
MDH	,407	20	,000	,379	20	,000
UBRDH	,372	20	,000	,554	20	,000
RTH	,407	20	,000	,452	20	,000
RDH	,400	20	,000	,396	20	,000
RS	,381	20	,000	,520	20	,000
MTH	,289	20	,000	,654	20	,000

a. Lilliefors Significance Correction

Resim 3.1. Çözüm kurucu sezgisel algoritmalar için kolmogorov-smirnov normallik test sonuçları

Resim 3.1’e baktığımızda, 6 çözüm kurucu sezgisel için hem Kolmogorv-Smirnov hem de Shapiro-Wilk testlerinde p değerinin 0.05 den düşük olduğu görülmekte ve H0 hipotezi (verilerin normal dağılıma uygunluk göstermektedir) reddedilmektedir. Yani tüm gruplar için %95 güven düzeyinde veriler normal dağılıma uymamaktadır. Bu nedenden dolayı, grupların ortalamaları arasında anlamlı bir farklılığın olup olmadığının belirlenmesinde Friedman Anova Testine başvurulmaktadır. Friedman Anova Testi için kurulan hipotez aşağıda belirtilmektedir.

H0:Çözüm kurucu sezgisellerin ortalamaları arasında anlamlı bir farklılık yoktur.

H1: Çözüm kurucu sezgisellerin ortalamaları arasında anlamlı bir farklılık vardır.

Descriptive Statistics						Ranks		Test Statistics ^a	
	N	Mean	Std. Deviation	Minimum	Maximum		Mean Rank		
MDH	20	630,6280	1788,55220	3,84	7997,00	MDH	2,18	N	20
UBRDH	20	3507,9375	7172,48467	6,13	24960,24	UBRDH	4,78	Chi-Square	45,029
RTH	20	1049,6995	2592,23014	4,53	10825,46	RTH	2,93	df	5
RDH	20	12780,39	36681,00654	6,62	161189,72	RDH	5,20	Asymp. Sig.	,000
RS	20	8529,9190	19276,11964	3,73	77513,61	RS	3,53	a. Friedman Test	
MTH	20	310,9560	507,15072	4,43	1870,04	MTH	2,40		

Resim 3.2. Çözüm kurucu sezgisel algoritma için friedman Anova test sonuçları

SPSS programında yapılan Friedman Anova Testi sonucunda elde edilen sonuçlar Resim 3.2’de gösterilmiştir. Resim 3.2’deki sıralı ortalamalara baktığımızda, MDH ile MTH sezgisellerinin birbirlerine yakın olduğunu ve en düşük ortalamaya sahip sezgiseller olduğunu görmekteyiz. Anlamlı farklılığın neden kaynaklandığının belirlenmesi ve sezgisellerin ikili karşılaştırılması yapmak için ise post hoc testlerinden Wilcoxon testi kullanılmaktadır.

Wilcoxon Signed Ranked Test, aynı veri kaynağından el edilen iki ölçüm sonuçları arasında anlamlı bir farklılığın olup olmadığının belirlenmesinde kullanılan istatistiksel bir yöntemdir. Eğer veri sayısı az, veri dağılım ölçüleri t-testinin koşullarını sağlamıyor ya da veriler sıralama ölçeğinde ise bu durumlarda ihtiyaç duyulmaktadır. Analiz çalışması için kurulan hipotez aşağıdaki gibidir.

H0:Sezgisel-1 ve Sezgisel-2 ortalamaları arasında anlamlı bir farklılık yoktur.

H1: Sezgisel-1 ve Sezgisel-2 ortalamaları arasında anlamlı bir farklılık vardır.

SPSS programında Wilcoxon yöntemi ile yapılan ikili karşılaştırmalar sonucunda elde edilen sonuçlar Çizelge 3.9’da gösterilmektedir. Tabloda, “Sezgisel 1>Sezgisel2” sütunu yapılan 20 senaryo analizinin kaçında Sezgisel 1’in, Sezgisel 2’den daha yüksek zamanlı çözüm verdiğini, “Sezgisel 1<Sezgisel2” sütunu kaç senaryoda Sezgisel 1 in, Sezgisel 2’den düşük zamanlı çözüm ürettiğini ve “Sezgisel 1=Sezgisel2” sütunu ise problemlerin kaçında eşit çözümlerin üretildiğini göstermektedir. “Asymo. Sig” ise karşılaştırılan sezgisel ortalamalar arasında anlamlı bir farklılık olup olmadığını ifade etmektedir.

Çizelge 3.9. Çözüm kurucu sezgisel algoritmalar için wilcoxon signed ranked test analiz sonuçları

Sezgisel_1	Sezgisel_2	Sezgisel_1>Sezgisel_2	Sezgisel_2>Sezgisel_1	Sezgisel_1=Sezgisel_2	Asymp. Sig.
MDH	UBRDH	1	19	0	0
MDH	RTH	4	15	1	0,016
MDH	RDH	2	18	0	0
MDH	RS	7	13	0	0,015
MDH	MTH	9	11	0	0,881
UBRDH	RTH	19	1	0	0,001
UBRDH	RDH	6	14	0	0,057
UBRDH	RS	12	7	1	0,936
UBRDH	MTH	19	1	0	0
RTH	RDH	1	19	0	0
RTH	RS	7	13	0	0,017
RTH	MTH	12	4	4	0,049
RDH	RS	15	5	0	0,062
RDH	MTH	18	2	0	0
RS	MTH	12	8	0	0,037

Çizelge 3.9’da, MDH-MTH sezgiselleri için anlamlılık değeri 0,05 den büyük çıkmakta ve sezgisellerinin ortalamaları arasında anlamlı bir fark olmadığı anlaşılmaktadır. Bu nedenden dolayı tavlama benzetimi algoritmasında başlangıç çözümünün üretilmesinde iki yönteminde kullanılmasına karar verilmiştir.

3.2.3. Parametre konfigürasyonu

Parametre konfigürasyonu, algoritmanın etkinliği açısından büyük bir öneme sahiptir. Tavlama Benzetimi Algoritmasında da çözümün hızına ve kalitesine etki eden birçok parametre bulunmaktadır. Klasik deneysel tasarım metotları çok karmaşıktır ve uygulamada çeşitli zorluklarla karşılaşılmaktadır. Buna ek olarak, parametre sayısı arttıkça uygun parametre setini elde etmek için çok sayıda deney yapılması gerekmektedir. Bu çalışmada, uygun parametre setini belirlemek için Taguchi metodu kullanılmıştır. Literatürde parametre optimizasyonu için Taguchi metodunun kullanıldığı birçok çalışma bulunmaktadır [52-56]. Bu metotta, küçük sayıda denemelerle parametre uzayının tamamının taranması için özel tasarlanmış ortagonal diziler kullanılmaktadır. Bu diziler sayesinde, daha az denemelerle daha uygun parametre setleri elde edilebilmektedir [57].

Önerilen tavlama benzetimi algoritmasında beş parametre vardır: durdurma kriteri (S.C.), soğutma oranı (α), her bir sıcaklıktaki komşu çözüm sayısı (N), başlangıç sıcaklığı (T_0) ve son sıcaklık (T_f). Bu parametreler arasından N parametresi, deterministik (swap ve insert mekanizması ile elde edilen tüm çözümlerin sayısı), olarak belirlenmektedir. Diğer dört parametre için parametre konfigürasyonu yapılması gerekmektedir. Konfigürasyonda kullanılan parametreler ve bunların seviyeleri Çizelge 3.10’da gösterilmektedir.

Çizelge 3.10. Tek vuruculu HHGSP’nin deneysel analizi için konfigürasyonda kullanılan parametreler ve seviyeleri

Problem Boyutu	Hedef Sayısı	Seviye	A	B	C	D
			T_0	T_f	S.C.	A
Küçük	5 ve 10	1	20	0,005	30 saniye	0,8
		2	60	0,5	90 saniye	0,85
		3	100	1	3 dakika	0,90
Orta	15	1	100	0,005	5 dakika	0,85
		2	150	0,5	10 dakika	0,90
		3	200	1	15 dakika	0,95
Büyük	20 ve 25	1	200	0,005	10 dakika	0,90
		2	300	0,5	20 dakika	0,95
		3	400	1	30 dakika	0,99

Taguchi metodunda denemelerin sayısı, parametre sayısı ve seviyesine göre değişiklik gösterebilmektedir. Bu çalışmada 4 faktörün dikkate alınması ve her bir faktörün 3 seviyeye sahip olması nedeniyle 9 adet deneyin yapılması gerekmektedir. İlgili denemeler için L9 ortogonal dizisi kullanılmakta ve bu diziye ilişkin bilgiler Çizelge 3.11’de gösterilmektedir.

Çizelge 3.11. L9 ortogonal dizilimi

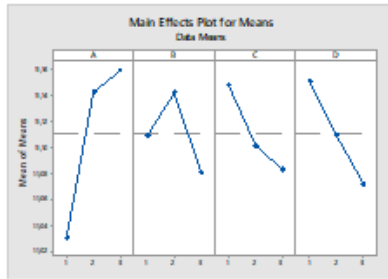
Deneyler	A	B	C	D
1	A(1)	B(1)	C(1)	D(1)
2	A(1)	B(2)	C(2)	D(2)
3	A(1)	B(3)	C(3)	D(3)
4	A(2)	B(1)	C(2)	D(3)
5	A(2)	B(2)	C(3)	D(1)
6	A(2)	B(3)	C(1)	D(2)
7	A(3)	B(1)	C(3)	D(2)
8	A(3)	B(2)	C(1)	D(3)
9	A(3)	B(3)	C(2)	D(1)

Parametre optimizasyonu için farklı boyutlarda 11 adet problem rassal olarak seçilmekte ve her bir problem için L9 dizisine göre 9 adet deneme yapılmaktadır. Problemlere ilişkin bilgiler Çizelge 3.12’de gösterilmektedir.

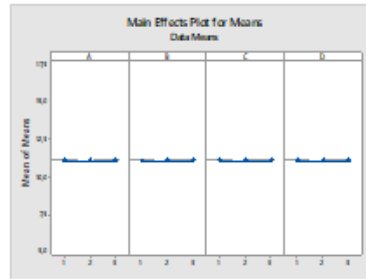
Çizelge 3.12. Parametre konfigürasyonunda kullanılan senaryolar

Senaryo Adı	Problem Boyutu	Hedef Sayısı
Senaryo 1,3	Küçük boyutlu	5
Senaryo 5,7	Küçük boyutlu	10
Senaryo 11,12	Orta boyutlu	15
Senaryo 13,14,16	Büyük boyutlu	20
Senaryo 19,20	Büyük boyutlu	25

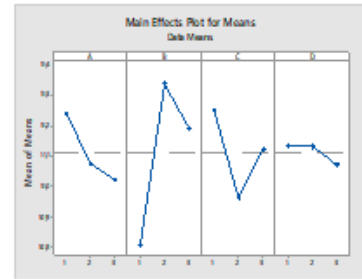
Deney çalışmaları 4 GB Ram ve 2.3 GHz işlemcili kişisel bilgisayarda, C Sharp programı kullanılarak gerçekleştirilmiştir. Taguchi metodunun kullanılmasında Minitab 18.0 programından yararlanılmıştır. Analiz çalışması sonucunda her bir problem için elde edilen değerler normalize edilmiş ve ideal parametre seti, ilgili problem boyutu içerisinde her bir parametre seti için ortalama değerlerin alınmasıyla elde edilmiştir.



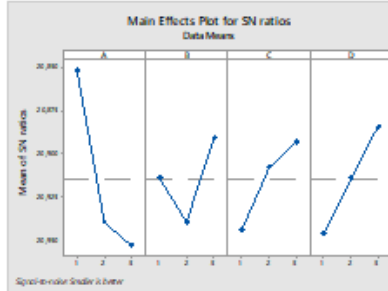
(a) Büyük boyutlu problemler için uygunluk fonksiyonuna dayalı olarak parametrelerin ortalamaya etkisi



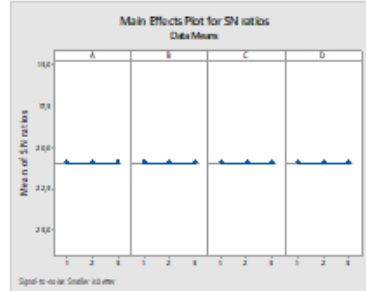
(b) Orta boyutlu problemler için uygunluk fonksiyonuna dayalı olarak parametrelerin ortalamaya etkisi



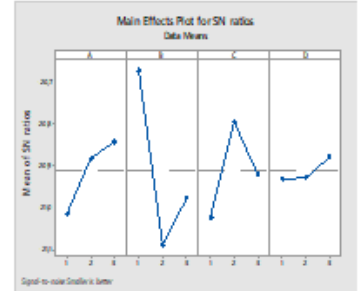
(c) Küçük boyutlu problemler için uygunluk fonksiyonuna dayalı olarak parametrelerin ortalamaya etkisi



(d) Büyük boyutlu problemler için S/N oranının ortalamaya etkisi



(e) Orta boyutlu problemler için S/N oranının ortalamaya etkisi



(f) Küçük boyutlu problemler için S/N oranının ortalamaya etkisi

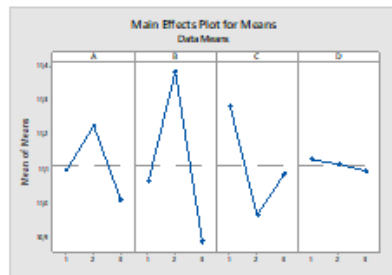
Şekil 3.6. SA-MTH algoritması için parametre analizi

Taguchi metodunda, deneysel verilerin analizi için en önemli kriter, Signal/Gürültü (S/N) oranıdır. Hem maksimizasyon hem de minimizasyon problemleri için bu oranı maksimize eden parametre değerleri ideal parametre setini oluşturmaktadır. Şekil 3.6'daki analiz sonuçlarına göre SA-MTH algoritması için ideal parametre seti Çizelge 3.13'de gösterilmektedir.

Çizelge 3.13. SA-MTH algoritması için parametre analizi

Problem Boyutu	A	B	C	D
	T0	T_f	S.C.	A
Küçük	100	0,005	90 saniye	0,9
Orta	100	1	5 dakika	0,85
Büyük	200	1	30 dakika	0,99

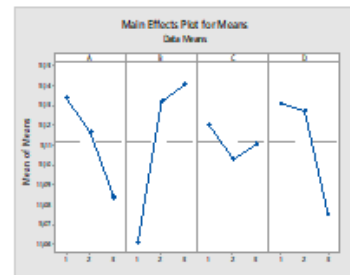
Yukarıda ifade edilen bilgiler doğrultusunda, "SA-MDH" algoritması için parametre bilgileri, Şekil 3.7 ve Çizelge 3.14'de gösterilmektedir.



(a) Büyük boyutlu problemler için uygunluk fonksiyonuna dayalı olarak parametrelerin ortalamaya etkisi



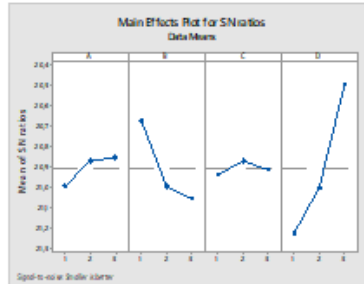
(b) Orta boyutlu problemler için uygunluk fonksiyonuna dayalı olarak parametrelerin ortalamaya etkisi



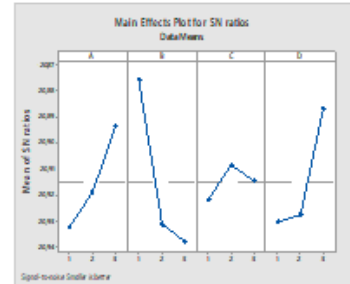
(c) Küçük boyutlu problemler için uygunluk fonksiyonuna dayalı olarak parametrelerin ortalamaya etkisi



(d) Büyük boyutlu problemler için S/N oranının ortalamaya etkisi



(e) Orta boyutlu problemler için S/N oranının ortalamaya etkisi



(f) Küçük boyutlu problemler için S/N oranının ortalamaya etkisi

Şekil 3.7. SA-MDH algoritması için parametre analizi

Çizelge 3.14. SA-MDH algoritması için parametre analizi sonuçları

Problem Boyutu	A	B	C	D
	T0	T_f	S.C.	A
Küçük	100	0,005	90 saniye	0,9
Orta	200	0,005	10 dakika	0,95
Büyük	400	1	20 dakika	0,99

Bir sonraki bölümde, belirlenen parametre değerleri kullanılarak ilgili algoritmalar problemler üzerinde test edilmiş ve sonuçlar analiz edilmiştir.

3.2.4. Deneysel sonuçlar

SA-MTH ve SA-MDH algoritmaları, tüm senaryolarda 30 kez çalıştırılmış ve elde edilen sonuçlar Çizelge 3.15’de gösterilmiştir. Çizelge, ortalama(M.), en iyi (B.), standart sapma (S.D.) ve bilinen en iyiden sapma (D.M.B.) bilgilerini içermektedir.

Çizelge 3.15. SA-MDH(A1) ve SA-MTH(A2) metotlarının karşılaştırılması

Senaryo	Hedef Sayısı	SA-MTH				SA-MDH				Minimum Zaman
		M.	S. D.	Best	D. M B. (%)	M.	S. D.	Best	D. M B. (%)	En iyi strateji
Senaryo 1	5	6,13	0,00	6,13	0	6,13	0,00	6,13	0	A1, A2
Senaryo 2	5	3,73	0,00	3,73	0	3,73	0,00	3,73	0	A1, A2
Senaryo 3	5	8,08	0,00	8,08	0	8,08	0,00	8,08	0	A1, A2
Senaryo 4	5	8,85	0,00	8,85	0	8,85	0,00	8,85	0	A1, A2
Senaryo 5	10	36,05	0,00	36,05	0	36,05	0,00	36,05	0	A1, A2
Senaryo 6	10	17,47	0,00	17,47	0	19,97	0,00	19,97	0	A1
Senaryo 7	10	30,34	0,00	30,34	0	25,97	0,00	25,97	0	A2
Senaryo 8	10	9,3	0,00	9,3	0	9,3	0,00	9,3	0	A1, A2
Senaryo 9	15	142,57	0,00	142,57	0	110,81	0,00	110,81	0	A2
Senaryo 10	15	24,79	0,00	24,79	0	24,79	0,00	24,79	0	A1, A2
Senaryo 11	15	64,54	0,00	64,54	0	64,2	0,00	64,2	0	A2
Senaryo 12	15	127,67	0,00	127,67	0	141,21	19,21	122,98	14,82	A2
Senaryo 13	20	73,27	0,00	73,27	0	93,06	20,87	73,27	27	A1, A2
Senaryo 14	20	5,95	0,00	5,95	0	5,93	0,019	5,89	0,68	A2
Senaryo 15	20	4,93	0,00	4,93	0	4,53	0,00	4,53	0	A2
Senaryo 16	20	135,57	8,99	123,5	9,77	109,96	0,00	109,96	0	A2
Senaryo 17	25	189,7	10,88	175,19	8,30	177,6	7,58	170,68	4,05	A2
Senaryo 18	25	164,12	3,25	158,43	3,59	190,12	4,35	184,91	2,81	A1
Senaryo 19	25	222,92	0,00	222,92	0	222,92	0,00	222,92	0	A1, A2
Senaryo 20	25	4,05	0,018	4,04	0,20	4,25	0,14	4,14	2,66	A1

Çizelge 3.15’deki sonuçlardan istatistiksel olarak anlamlı çıkarımlar elde etmek için, tablodaki verilere Wilcoxon Signed Ranked Testi uygulanmıştır. Analiz sonuçlarına göre, SA-MTH algoritması 20 senaryonun 8 inde, SA-MDH algoritmasından daha kötü sonuçlar

üretmiştir. SA-MDH algoritması ise 3 senaryoda, SA-MTH'dan daha kötü sonuçlara ulaşmıştır. 9 senaryo da ise iki algoritma aynı sonuçlara üretmiştir. Buna ek olarak, analize ilişkin anlamlılık değeri (p) 0,155 çıkmış, bu değer düşük bir seviyede olmasına rağmen anlamlı bir farklılık olmadığını göstermiştir.

Bu iki algoritma arasından seçim yapılması durumunda, SA-MDH algoritmasının birçok senaryoda üstün olması ve düşük ortalamaya sahip olması nedeniyle, ilgili problemlerde SA-MDH algoritmasının kullanılmasının daha doğru olacağı düşünülmektedir. Ayrıca, tablodaki sonuçlardan, SA-MDH algoritmasının, orta ve büyük boyutlu problemlerde diğer algoritmadan daha iyi sonuçlar ürettiği tespit edilmiştir. Algoritmanın sağlamlığını değerlendirmek için Çizelge 15'deki S.D. ve D.M.B. sütunları incelendiği zaman, SA-MTH algoritmasının, diğer algoritmadan biraz daha iyi olduğu anlaşılmaktadır. Buna ek olarak, tablodaki sonuçlar, iyi bir başlangıç çözümü sayesinde, daha düşük zamanlı sonuçların elde edilebileceğini göstermiştir.

3.3. Sonuç

HHGSP, savunma sektörü başta olmak üzere birçok alanda yoğun bir şekilde uygulanmakta ve teknolojiye yaşanan gelişmelerle birlikte ilginin daha fazla artması beklenmektedir. Askeri operasyonlar ise problemin temel uygulama alanlarından birini oluşturmaktadır. İnsansız hava aracı ya da dronelar ile hareketli nesnelerin yakalanması ve izlenmesi, jet vb. uçaklarla mobil birliklerin etkisiz hale getirilmesi ve okyanuslardaki deniz gözetleme operasyonları bu alanlardan bazılarına örnek olarak verilebilmektedir. Ayrıca, savunma sektöründe yaşanan teknolojik gelişmelerle birlikte, İHA ve dronelere arama kurtarma operasyonlarında sıklıkla başvurulmaktadır.

Tez çalışmasının bu bölümünde, tek vuruculu HHGSP ele alınmış ve problem Tavlama Benzetimi Algoritmasına dayalı bir çözüm yaklaşımı ile çözülmeye çalışılmıştır. Algoritmanın başlangıç çözümü için altı farklı çözüm kurucu sezgisel geliştirilmiş ve bunlar arasından en iyi algoritma, tavlama benzetimi algoritmasında başlangıç çözümü olarak kullanılmıştır. Önerilen çözüm kurucu sezgiseller, hedef sayısının 5 ile 25 arasında değiştiği, 20 farklı senaryo üzerinde test edilmiştir. Analiz sonucunda MTH ve MDH sezgisellerinin, diğer algoritmalarından daha iyi sonuçlar ürettiği tespit edilmiş ve tavlama benzetimi algoritmasında başlangıç çözümü olarak kullanılmıştır.

SA-MDH ve SA-MTH algoritmalarını etkinliđi 20 farklı problem üzerinde test edilmiř ve algoritma ierisinde parametreler iin ideal deđerlerin belirlenmesinde Taguchi metodundan yararlanılmıřtır. Buna ek olarak, algoritmanın özüm kalitesini artırmak, yerel optimaldan kaçınmak ve özüm zamanını azaltmak iin yeni özüm stratejileri geliřtirilmiřtir. Analiz sonucunda, her iki algoritmanın tüm senaryolarda kabul edilebilir özüm süreleri ierisinde uygun özümler ürettiđi belirlenmiřtir. Küük boyutlu problemlerde iki algoritmanın benzer sonuçlar ürettiđi, orta ve büyük boyutlu problemlerde SA-MDH algoritmasının, SA-MTH algoritmasından daha iyi sonuçlar ürettiđi tespit edilmiřtir. Ayrıca analiz sonuçları, algoritmanın sađlamlıđı aısından SA-MTH algoritmasının, SA-MDH algoritmasından daha üstün olduđunu göstermiřtir.

Bir sonraki bölümde, belirtilen probleme zaman penceresi kısıtlaması eklenmiř, hedeflerin belirli zaman aralıđı ierisinde sistemde bulunabileceđi ve herhangi bir t anında sistemdeki hedef sayısının deđiřebileceđi varsayılmıřtır. Problem iin önerilen özüm yaklařımı, Ege Denizi'ndeki gemilerin gözetlenmesi ile ilgili bir uygulama alıřması üzerinde test edilmiřtir.

4. ZAMAN PENCERELİ HHGSP NİN TAVLAMA BENZETİMİ ALGORİTMASI İLE ÇÖZÜMÜ: DENİZ GÖZETLEME OPERASYONU UYGULAMASI

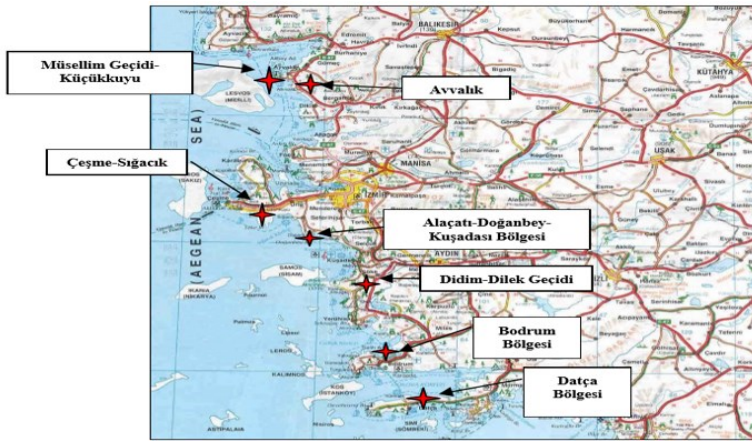
Denizler ve okyanuslar, ticaret ve turizm başta olmak üzere birçok önemli faaliyete ev sahipliği yapmaktadır. Uluslararası yolcu ve yük taşımacılığı, balıkçılık ve çeşitli askeri sevkiyatlar bu sular üzerinden gerçekleştirilmektedir. Denizlerin ya da okyanusların üzerindeki tüm faaliyetlerin güvenli bir şekilde gerçekleştirilmesi ve ulusal güvenliğin sorunsuz bir şekilde sağlanması açısından deniz gözetleme operasyonları büyük bir önem arz etmektedir.

Günümüzde denizlerde yaşanan en önemli sorunlardan birini şüphesiz mülteci gemileri oluşturmaktadır. İnsanlar çeşitli nedenlerden dolayı ülkelerinden ayrılmak zorunda kalmakta ve yeni hayatlar kurmak için yasadışı yollarla başka ülkelere göç etmektedir. Denizler bu yolların başında gelmekte ve her yıl binlerce insan denizlerde boğularak can vermektedir. Bunun yanı sıra denizler, çeşitli suçlardan aranan insanların yasadışı yollarla başka ülkelere kaçmasında sıklıkla kullandıkları yollardan biri olmaktadır.

Deniz ya da okyanuslardaki bir diğer önemli problem ise deniz korsanlarıdır. Her yıl yüzlerce gemi, deniz korsanları tarafından yağmalanmakta ve bazı zamanlar çeşitli can kayıpları meydana gelebilmektedir. Ayrıca yasadışı balık avcılığı, denizler üzerinden yapılan ülkeye sızma faaliyetleri denizlerde yaşanan problemlerin başında gelmektedir.

Belirtilen problemlerin giderilmesi ve düşman ya da tehdit unsurlarına hızlı bir şekilde müdahale edilebilmesi ancak doğru bir operasyon ve gözetleme stratejisi ile mümkün olmaktadır. Tez çalışmasının bu bölümünde, denizlerdeki hareketli unsurların hızlı bir şekilde gözetlenmesi ya da etkisiz hale getirilmesi problemi ele alınmıştır. Problem literatürde deniz gözetleme problemi başlığı altında ele alınmakta, bu çalışmada tehdit unsurlarının hareket etmesi ve hedeflerin belirli bir zaman aralığında sistemde kalması nedeniyle HHGSP-ZP (Zaman Pencereli-HHGSP) ile ilişkilendirilerek incelenmektedir. Probleme ilişkin özellikler, Bölüm 4.1. içerisinde verilmektedir.

Harita 4.2’de, Türkiye’ den Yunanistan a göçün genel şablonu, Harita 4.3’de ise Türkiye’deki kıyılardan Yunanistan’a yapılan göçün detayları gösterilmektedir. Harita 4.2’ye göre, mülteci botları ilk olarak İzmir kıyılarından ayrılıp, Lizbon, Chios ya da Samos gibi yakın Yunan adalarına geçiş yapmaktadır. Eğer bu geçişi başarılı bir şekilde gerçekleştirirlerse, kendilerini Yunan yetkililerine mülteci olarak tanıtmakta ve bu adalarda yaklaşık bir hafta durduktan sonra Atina’ ya transfer edilmektedirler.



Harita 4.3. Ege denizi kıyı bazında yasadışı göç güzergâhı [60]

Bu geçişler her zaman başarılı olamamakta, çoğu zaman Türk sahil güvenliği tarafından yakalanmakta ya da mülteci botlarının batması sonucu boğulma vakaları ile sonuçlanmaktadır. Harita 4.3’de Ege kıyılarından, Yunan adalarına yapılan göçün detaylı analizi gösterilmekte, Harita 4.4’de ise ilgili bölgelerde 2013 yılında yakalanan yasadışı göçmen sayıları gösterilmektedir.



Harita 4.4. 2013 yılında ilgili bölgelerde yakalanan yasadışı göçmen sayısı [61]

Harita 4.4'e göre en fazla yasadışı göçmenin Çeşme - Alaçatı, Seferihisar, Kuşadası üçgeninde olduğu gözlenmekte, bu nedenden dolayı uygulama çalışmasında bu kıyılardan yapılan göçlere müdahale edilmesi problemi incelenmektedir.



Harita 4.5. İlgili bölgedeki kıyılardan gerçekleşebilecek göç senaryoları [62, 63]

Harita 4.5'de, ilgili kıyılardan Yunan Adalarına yapılan göçlere ilişkin bazı güzergâhlar gösterilmiştir. Denizlerdeki mülteci ve kaçakçılık botlarının gözetlenmesi, keşfi ve ya etkisiz hale getirilmesi problemi, literatürde HHGSP ile kesişen bir problem türüdür. HHGSP'de, "n" sayıda, belirli açı ve hızlarda hareket eden hedefler bulunmakta, bu hedefler orijinden harekete başlayan sabit hızlı bir vurucu tarafından imha edilmekte, sonuç olarak tüm hedeflerin imha edilip, vurucunun orijin noktasına ulaştığı minimum zamanlı tur rotası belirlenmeye çalışılmaktadır. Karasularının belirli bir sınıra sahip olması nedeniyle mülteci botlarına ilgili sınırlar içerisinde müdahale edilmesi gerekmekte, bu nedenden dolayı ilgili problem HHGSP-ZP kapsamında incelenmektedir. Zaman penceresi ile ilgili hedeflerin, sistemde belirli bir zaman dilimi içerisinde bulunacağı ifade edilmekte, hedeflere bu zaman dilimi içerisinde müdahale edilmezse görev başarısız olmaktadır.

Harita 4.6'da ise deniz gözetleme helikopteri ya da insansız hava aracı ile gözlemlenen ve müdahale edilen hedeflere ilişkin tur rotası gösterilmektedir. Şekildeki rotada, alınan bilgiler doğrultusunda tespit edilen hedefler, orijinden harekete başlayan vurucu tarafından etkisiz hale getirilmektedir.

olarak kabul edilmektedir.

- Hedefler belirtilen zaman aralığı içerisinde sistemde bulunmakta, her bir hedefin sistemden çıkma ya da ilgili Yunan adasına varma zamanı farklı olmaktadır. Eğer bu hedef, ilgili zaman aralığı içerisinde imha edilmezse ya da bu hücum botuna müdahale edilmezse, görev başarısız sayılmaktadır.

4.2. Metodoloji

Problemin çözümünde Tavlama Benzetimi Algoritması kullanılmakta ve algoritmanın başlangıç çözümünde Bölüm 3.1.1'deki MTH algoritmasından yararlanılmaktadır. Algoritmanın adımları Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. Zaman pencereli-MTH başlangıç çözümlü-tavlama benzetimi algoritması

```

f(Sbest) = double. Maxvalue, Sbest, durdurma kriteri(d. k.)değerini gir, say = 0.
while(say < durdurma kriteri)
    Başlangıç çözümü için belirlenen algoritmaya göre başlangıç çözümünü(S0) oluştur.
    Başlangıç sıcaklığını(T), soğutma oranını( $\alpha$ ), çözüm bozma sayısı(cbs) değerlerini gir.
    S  $\leftarrow$  S0, f(S)  $\leftarrow$  f(S0)
    Sgood  $\leftarrow$  S0, f(Sgood)  $\leftarrow$  f(S0)
    bestsay = 0
    while(T  $\geq$  1)
        f(S*) = double. Maxvalue, S*
        Swap mekanizmasına göre S nin tüm komşu çözümlerini (Si*)üret
        if Si* zaman penceresi kısıdını sağlıyorsa
            if f(S*) > f(Si*) then S*  $\leftarrow$  Si* and f(S*)  $\leftarrow$  f(Si*)
            if f(Sgood) > f(Si*) then Sgood  $\leftarrow$  Si* and f(Sgood)  $\leftarrow$  f(Si*)
            if f(Sbest) > f(Si*) then Sbest  $\leftarrow$  Si* and f(Sbest)  $\leftarrow$  f(Si*)
        End
         $\Delta \leftarrow$  f(S*) - f(S)
        if  $\Delta \leq 0$  then
            S  $\leftarrow$  S*
            Vurucu ve hedeflerin görev zamanı ve lokasyon bilgilerini güncelle
        Else
            u ~ u(0,1)
            if u < e(- $\Delta$ /T) then
                S  $\leftarrow$  S*
                Vurucu ve hedeflerin görev zamanı ve lokasyon bilgilerini güncelle
            Else
                S  $\leftarrow$  S
            End
        End
        if f(S*) < f(Sgood) then bestsay ++
        if bestsay = cbs then invers mekanizmasını kullanarak S* çözümünü cbs kere
        perturbation yap ve en iyi çözümü sonraki iterasyona S başlangıç çözümü
        olarak ata, bestsay = 0 yap
    End
    T = T *  $\alpha$ 
End
say ++
End

```

4.3. Uygulama Çalışması

4.3.1. Problem tasarımı

Algoritmanın etkinliğini test etmek için geliştirilen problem setleri için örnek bir gösterim Harita 4.7 ve Çizelge 4.2 de gösterilmiştir. İlgili çizelge ve şekildeki bilgiler, gerçek verilerden elde edilmiştir. Şekil 4.8'deki görev alanı yaklaşık 9955,58 km² olup, yatayda 131 km, dikeyde ise 76 km'lik bir uzaklığa sahip olmaktadır. Şekildeki kırmızı renkli çizgiler yaklaşık olarak Türk-Yunan sınırlarını ifade etmekte, bu bilgiler gerçek harita üzerinde daha farklı olabilmektedir. Temel amaç kırmızı daire ile belirtilen orijin noktasından harekete başlayan S-70B SeaHawk Helikopteri tarafından çeşitli zamanlarda belirtilen konumlardan harekete başlayan mülteci botları ile uyuşturucu ve hazine gibi kaçakçılık yapan gemilerin, Bölüm 4.1'deki varsayımlar altında, ilgili varış noktalarına varmadan minimum zamanda imha edip başlangıç noktasına döneceği turu belirlemektir.



Harita 4.7. Uygulama çalışmasına ilişkin senaryo görseli [59]

Çizelge 4.17' de, vurucu araç ile hedeflere ilişkin bilgiler ve çözüm sonuçlarına ait imha zamanları verilmektedir. Algoritmanın çözümü 4 Gb Ram-2.3 Ghz işlemcili bilgisayarda C# programı kullanılarak gerçekleştirilmekte, belirtilen veriler doğrultusunda algoritma 10 kez çalıştırılmakta ve koşturmaların hepsinde amacı 1,99 olan 0-6-12-7-1-8-11-10-2-5-9-3-4-0 turu elde edilmektedir. İlgili analiz için ortalama çalışma zamanı ise 2 dakika olmaktadır. Ayrıca tablodaki sonuçlardan tüm hedeflerin ilgili zaman penceresi içerisinde imha edildiği anlaşılmaktadır.

Çizelge 4.2. Harita 4.7’de belirtilen probleme ilişkin bilgiler [59]

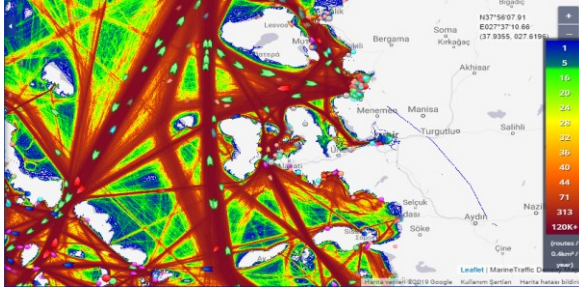
	GİRDİ BİLGİLERİ						ÇIKTI
Araç	Hız (km/sa)	Başlangıç x (km)	Başlangıç y (km)	Açı(derece)	Giriş Zamanı	Çıkış zamanı	İmha Zamanı(saat)
Helikopter	264	0	0	0	0	-----	-----
Hedef 1	60	-19	36	230	0,56	1,27	0,69288
Hedef 2	71	-21	15	145	1,06	3,69	1,14268
Hedef 3	55	-105	71	310	1,4	4,64	1,51284
Hedef 4	65	-52	43	225	1,3	2,2	1,70926
Hedef 5	75	-21	38	184	1,1	2,25	1,21627
Hedef 6	57	-61	59	268	0	0,76	0,28394
Hedef 7	80	-41	44	240	0,32	1,13	0,5867
Hedef 8	62	-61	54	233	0,5	1,48	0,8873
Hedef 9	66	-39	34	140	0,74	4,68	1,40631
Hedef 10	44	-27	35	226	0,96	2,37	1,094
Hedef 11	69	-58	54	243	0,86	1,81	0,96045
Hedef 12	42	-90	69	272	0,1	0,87	0,40171

Gerçek hayat problemlerinde, operasyon alanında daha fazla hedef bulunabilmekte ve algoritmanın etkinliğinin bu durumların dikkate alınarak belirlenmesi gerekmektedir. Tez çalışmasında, algoritmanın etkinliğinin belirlenmesinde dikkate alınan hedef sayısı aşağıda ifade edilmiştir.

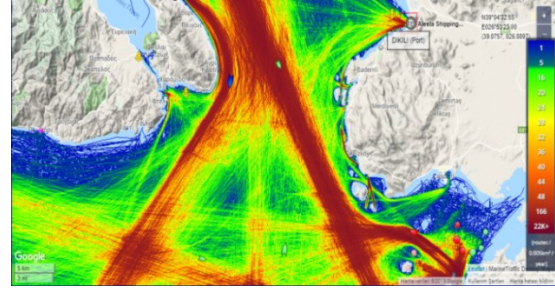
Ege Denizi, tanker, balıkçı, kargo ve yolcu gemilerinin yoğun bir şekilde kullandıkları uluslararası bir denizyoludur. Tezde kullanılan problem setlerindeki hedef sayısının tespit edilmesi için Ege Denizindeki günlük ortalama gemi sayısının belirlenmesi gerekmektedir. Bu bağlamda, günlük ortalama gemi sayısının belirlenmesinde Tıkanıklık Analizi metodundan yararlanılmaktadır. Tıkanıklık Analizi metodu, gerçek hayat uygulamalarında sistemin yoğunluğunu belirlemek ve performansını değerlendirmek için kullanılmaktadır. Yöntemin en önemli uygulama alanlarından birini karayollarındaki taşıt trafiği oluşturmaktadır. Literatürde, Tıkanıklık Analizi ile ilgili birçok çalışma bulunabilmektedir. [66-70]. Bu yöntem sayesinde, trafik ağlarındaki yoğunluk doğru bir şekilde tanımlanarak trafiği rahatlatıcı önlemler alınmasına katkıda bulunulması amaçlanmaktadır [71]. Tıkanıklık seviyesinin hesaplanmasında gerçek zamanlı veriler kullanılmakta ve bu verilere farklı yöntemler uygulanmaktadır [72].

Yukarıda ifade edilen veriler kullanılarak, denizlerdeki gemi trafiği doğru bir şekilde

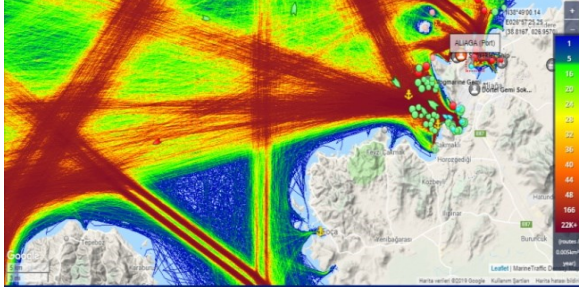
yönlendirilmekte ve deniz kazalarının önüne geçilmesi amaçlanmaktadır. Tez kapsamında, Ege Denizindeki günlük ortalama gemi sayısı, yoğunluk haritaları kullanılarak belirlenmekte ve problem setleri bu veriler yardımıyla hazırlanmaktadır. Ege Denizi ve bu denizdeki bazı limanlar için gemi trafik yoğunluğunu belirlemesinde Marinetrafic' den alınan veriler kullanılmaktadır [73].



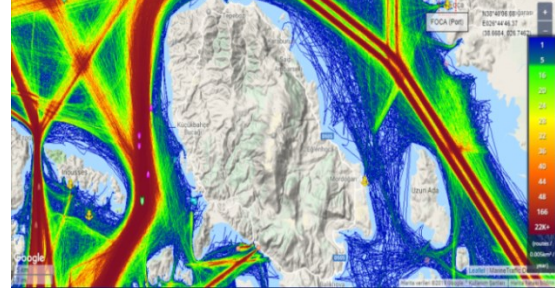
Ege denizi genel yoğunluk haritası



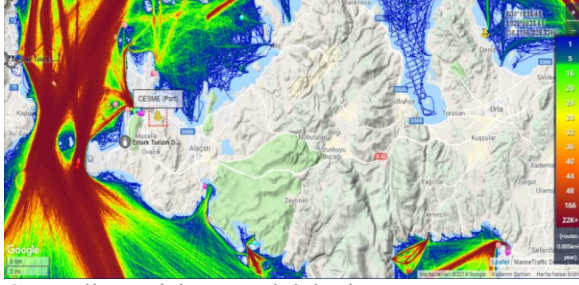
Dikili limanı için yoğunluk haritası



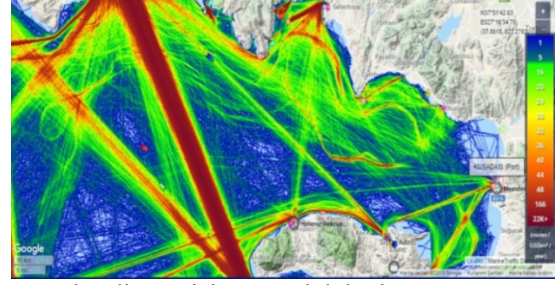
Aliaga limanı için yoğunluk haritası



Foça limanı için yoğunluk haritası



Çeşme limanı için yoğunluk haritası



Kuşadası limanı için yoğunluk haritası

Harita 4.8. Deniz-trafik verilerine göre 2017 yılında Ege denizi için yoğunluk haritası [73]

Harita 4.8'deki yoğunluk haritası analiz edilmiş ve hedef sayısının 20 ile 100 arasında değiştiği 50 farklı senaryo geliştirilmiştir. Bu senaryolarda problemler, küçük (20 hedefli ve 40 hedefli), orta (60 hedefli) ve büyük (80 hedefli ve 100 hedefli) boyutlu olmak üzere üç gruba ayrılmıştır. Problem setlerinde, belirli bir zamanda sisteme giren hedef sayısı, Poisson Dağılımına göre rassal olarak üretilmiştir. Buna ek olarak, hedeflerin sisteme varışları arasındaki zaman Üstel Dağılıma göre rastgele üretilmiştir. Poisson Dağılımına göre rassal sayı üretiminde Knuth's Algoritması ve Üstel Dağılıma göre rassal sayı üretiminde ise Ters Dönüşüm Fonksiyonu kullanılmıştır [74, 75]. Analizler sonucunda,

Poisson Dağılımı için λ parametresi 5 ve 10 olarak belirlenmiştir. Üstel Dağılımda kullanılan β parametresi ise, λ parametresine göre belirlenmiştir.

Test çalışmalarında, vurucunun ya da gözetleyici uçak için hız değerinin 200 km/sa ile 270 km/sa arasında olduğu varsayılmıştır. Buna ek olarak, hedef hızları, 20 km/sa ile 60 km/sa arasında rassal olarak üretilmiştir. Önerilen algoritma sayesinde, vurucu ya da gözetleyici tarafından orijin noktasından başlayıp, tüm hedeflerin ilgili zaman aralıkları içerisinde imha edilip, tekrar başlangıç noktasına döneceği minimum zamanlı tur rotasının belirlenmesi amaçlanmıştır.

4.3.2. Parametre konfigürasyonu

Bölüm 4.2’de belirtilen çözüm yaklaşımı içerisinde birçok parametreyi barındırmakta, çözüm kalitesi ve hızını artırmak için bu parametreler arasından ideal setin belirlenmesi gerekmektedir. Önerilen algoritmada; Çözüm Bozma Sayısı (Ç.B.S.), Soğutma Oranı (α), Her Bir Sıcaklıktaki Komşu Çözüm Sayısı (N), Başlangıç (T0) ve Son Sıcaklık (Tf) olmak üzere beş farklı parametre bulunmaktadır. Bu parametrelerden Ç.B.S., kaç iterasyon sonra Perturbasyon mekanizmasının kullanılması gerektiğini ifade etmektedir. N parametresi, swap ve insert mekanizması ile elde edilen tüm çözümlerin sayısını ifade etmekte ve deterministik olarak belirlenmektedir. Diğer dört parametre için ise parametre konfigürasyonu yapılması gerekmektedir. Konfigürasyonda kullanılan parametreler ve onların seviyeleri Çizelge 4.3’de gösterilmektedir.

Çizelge 4.3. Konfigürasyonda kullanılan parametreler ve onların seviyeleri

Parameter Level	A	B	C	D
	T0	Tf	Ç.B.S.	α
1	50	0,005	5	0,85
2	100	0,5	15	0,90
3	150	1	30	0,95

Çizelge 4.3’deki bilgiler, deneysel testler yoluyla analiz edilmiş ve ideal parametre seti belirlenmeye çalışılmıştır. Analiz sonucunda parametreler için ideal değerler, T0 için 50, Tf için 1, Ç.B.S. için 5 ve α için 0,9 olarak belirlenmiştir.

4.3.3. Deneysel sonuçlar

Analiz çalışması, 4 GB Ram ve 2.3 GHz işlemcili bilgisayarda C# programı kullanılarak gerçekleştirilmiştir. Algoritma, her bir senaryo için 10 kez çalıştırılmış ve elde edilen sonuçlar Çizelge 4.4 ve 4.5’de gösterilmiştir. Çizelge içerisinde ortalama, standart sapma ve en iyi değerin ortalamadan sapmasına ilişkin bilgiler verilmiştir.

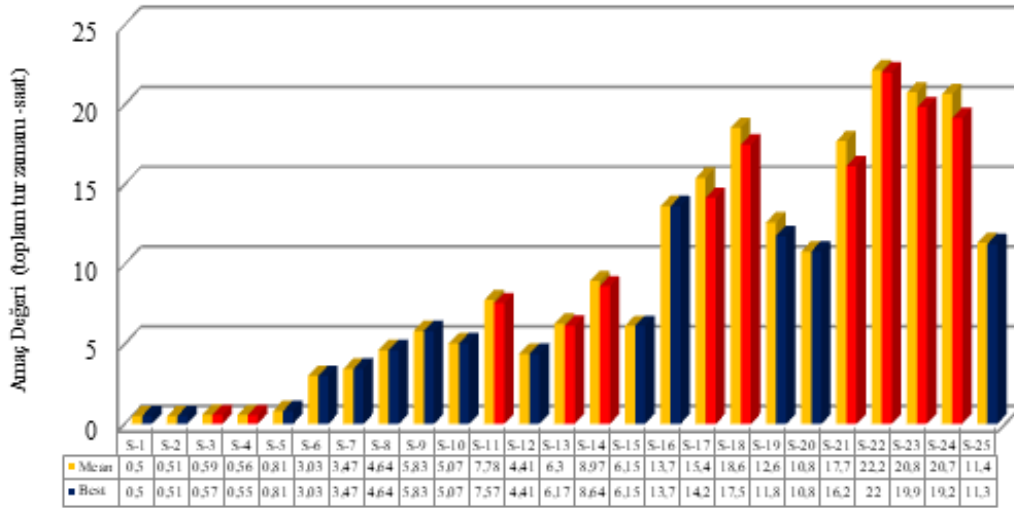
Çizelge 4.4. $\lambda=5$ için analiz sonuçları

Senaryo	Hedef Sayısı	Ortalama	Standart Sapma	En iyi	En iyiden sapma (%)	Ortalama bilgisayar işlemci zamanı (h / dk / sn)
Senaryo 1	20	0,5	0	0,5	0	3 sn
Senaryo 2	20	0,51	0	0,51	0	3 sn
Senaryo 3	20	0,5883	0,012	0,57	3,21	3 sn
Senaryo 4	20	0,5633	0,012	0,55	2,42	3 sn
Senaryo 5	20	0,81	0	0,81	0	2 sn
Senaryo 6	40	3,03	0	3,03	0	25 sn
Senaryo 7	40	3,47	0	3,47	0	26 sn
Senaryo 8	40	4,64	0	4,64	0	26 sn
Senaryo 9	40	5,83	0	5,83	0	25 sn
Senaryo 10	40	5,07	0	5,07	0	25 sn
Senaryo 11	60	7,7783	0,16	7,57	2,75	1 dk 47 sn
Senaryo 12	60	4,41	0	4,41	0	1 dk 45 sn
Senaryo 13	60	6,30	0,06	6,17	2,03	1 dk 48 sn
Senaryo 14	60	8,965	0,16	8,64	3,76	1 dk 52 sn
Senaryo 15	60	6,15	0	6,15	0	2 dk 34 sn
Senaryo 16	80	13,66	0	13,66	0	5 dk 15 sn
Senaryo 17	80	15,43	0,74	14,19	8,74	5 dk 23 sn
Senaryo 18	80	18,5583	1,05	17,52	5,93	5 dk 5 sn
Senaryo 19	80	12,6467	0,48	11,78	7,36	5 dk 8 sn
Senaryo 20	80	10,82	0	10,82	0	6 dk 17 sn
Senaryo 21	100	17,7433	0,78	16,16	9,80	12 dk 44 sn
Senaryo 22	100	22,1817	0,069	22,04	0,64	15 dk 15 sn
Senaryo 23	100	20,7883	0,93	19,86	4,67	12 dk 55 sn
Senaryo 24	100	20,6867	1,17	19,18	7,86	12 dk 42 sn
Senaryo 25	100	11,355	0,037	11,28	0,66	11 dk 35 sn

$\lambda=5$ için analiz sonuçları Şekil 4.9’da gösterilmiştir. Bu sonuçlara göre, her bir senaryoda minimum görev zamanı içerisinde tüm hedeflerin gözetlendiği/kurtarıldığı/imha edildiği belirlenmiştir. Buna ek olarak, 14 senaryoda ortalama ve bilinen en iyi değerler arasında bir fark olmadığı, diğer senaryolarda ise aradaki farkın oldukça düşük olduğu belirlenmiştir. Çizelge 4.4’deki standart sapma sütununa bakıldığında, sapma miktarının

düşük olduğu tespit edilmiştir. Bu sonuçlar, $\lambda=5$ için algoritmanın etkin olduğunu ve farklı problemler için uygulanabileceğini göstermektedir.

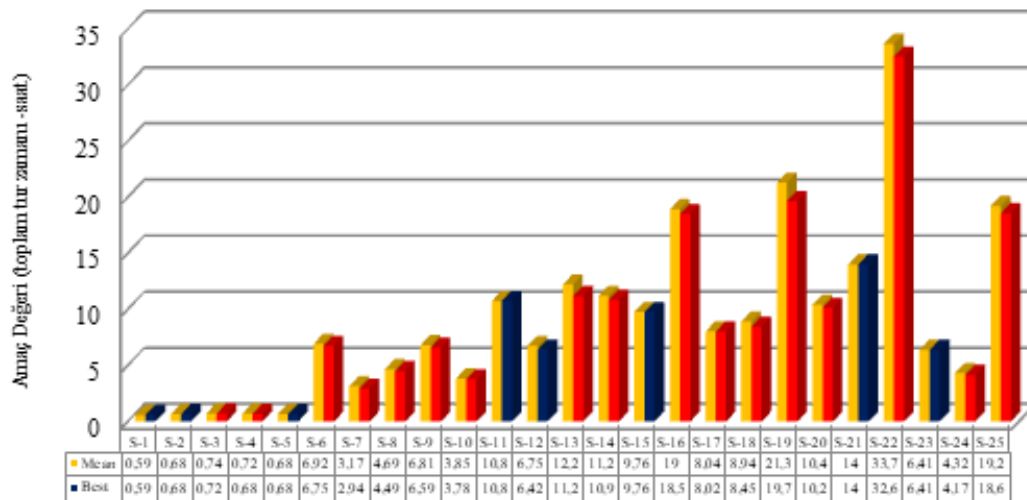
If mean == best then mean = yellow bar & best = blue bar
Else mean = yellow bar & best = red bar



Şekil 4.1. $\lambda=5$ için amaç değerleri açısından analiz sonuçları

Şekil 4.1’de, $\lambda = 10$ için amaç değerleri açısından analiz sonuçları gösterilmiştir. Bu şekil ve Çizelge 4.5’deki sonuçlardan, algoritmanın etkin olduğu ve tüm senaryolarda bilinen en iyi değer ve ortalama değer arasındaki farkın %10’dan az olduğu tespit edilmiştir. Buna ek olarak, Çizelge 4.5’deki standart sapma sütununa göre her bir senaryoda ortalamaya yakın sonuçlar üretildiği tespit edilmiştir.

If mean == best then mean = yellow bar & best = blue bar
Else mean = yellow bar & best = red bar

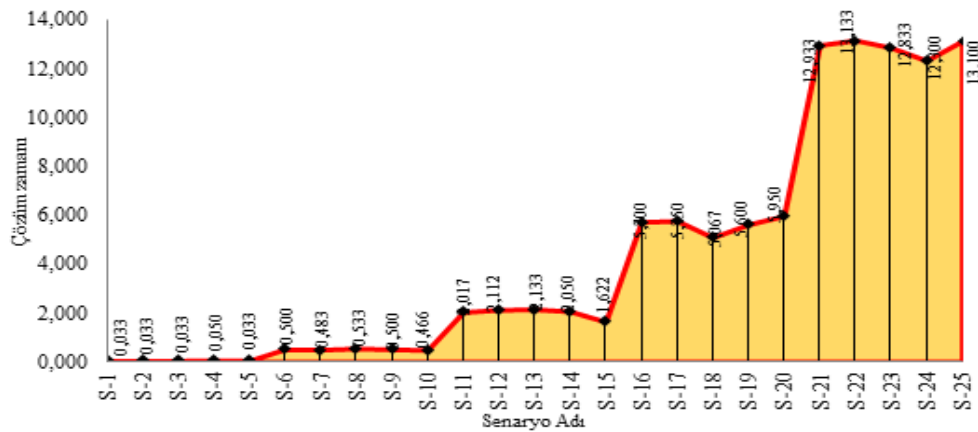


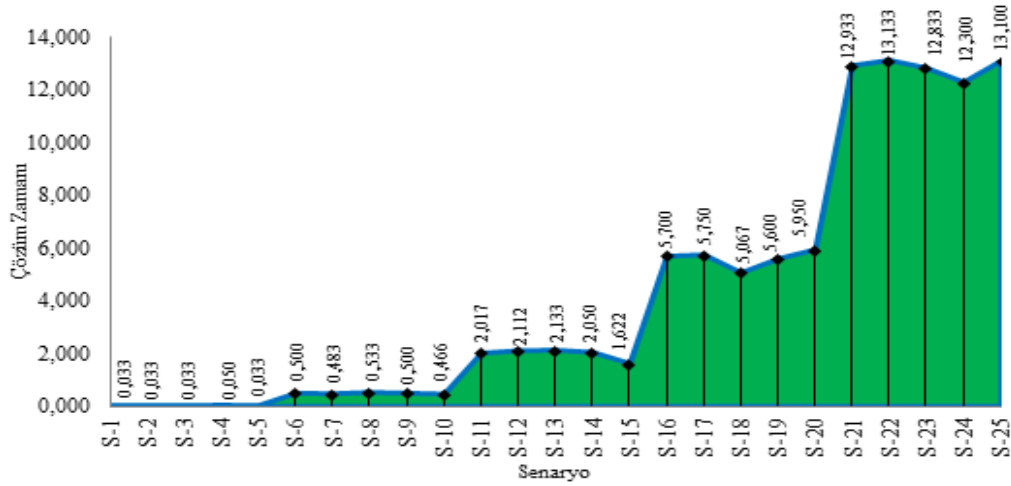
Şekil 4.2. $\lambda=10$ için amaç değerleri açısından analiz sonuçları

Çizelge 4.5. $\lambda=10$ için analiz sonuçları

Senaryo	Hedef Sayısı	Ortalama	Standart Sapma	En iyi	En iyiden sapma (%)	Ortalama bilgisayar işlemci zamanı (h / dk / sn)
Senaryo 1	20	0,59	0	0,59	0	2 sn
Senaryo 2	20	0,68	0	0,68	0	2 sn
Senaryo 3	20	0,738	0,02	0,72	2,5	2 sn
Senaryo 4	20	0,722	0,026	0,68	6,176	3 sn
Senaryo 5	20	0,684	0,055	0,68	0,588	2 sn
Senaryo 6	40	6,922	0,104	6,75	2,548	30 sn
Senaryo 7	40	3,166	0,190	2,94	7,687	29 sn
Senaryo 8	40	4,686	0,224	4,49	4,365	32 sn
Senaryo 9	40	6,812	0,203	6,59	3,369	30 sn
Senaryo 10	40	3,852	0,044	3,78	1,905	28 sn
Senaryo 11	60	10,75	0	10,75	0	2 dk 1 sn
Senaryo 12	60	6,748	0,183	6,42	5,109	2 dk 7 sn
Senaryo 13	60	12,242	0,725	11,19	9,401	2 dk 8 sn
Senaryo 14	60	11,206	0,210	10,86	3,186	2 dk 3 sn
Senaryo 15	60	9,76	0	9,76	0	1 dk 59 sn
Senaryo 16	80	18,956	0,255	18,5	2,465	5 dk 42 sn
Senaryo 17	80	8,044	0,013	8,02	0,299	5 dk 45 sn
Senaryo 18	80	8,938	0,273	8,45	5,775	5 dk 4 sn
Senaryo 19	80	21,288	1,205	19,65	8,336	5 dk 36 sn
Senaryo 20	80	10,414	0,148	10,15	2,60	5 dk 57 sn
Senaryo 21	100	14,03	0	14,03	0	12 dk 56 sn
Senaryo 22	100	33,678	0,631	32,55	3,465	13 dk 8 sn
Senaryo 23	100	6,41	0	6,41	0	12 dk 53 sn
Senaryo 24	100	4,322	0,097	4,17	3,645	12 dk 18 sn
Senaryo 25	100	19,238	0,392	18,58	3,541	13 dk 6 sn

Şekil, $\lambda = 5$ ve $\lambda = 10$ için çözüm zamanı cinsinden algoritmanın performansı Şekil 4.12 ve 4.13’de gösterilmiştir.

Şekil 4.3. $\lambda=5$ için her bir senaryo için ortalama çözüm zamanı



Şekil 4.4. $\lambda=10$ için her bir senaryo için ortalama çözüm zamanı

Şekil 4.3 ve 4.4'deki sonuçlardan, tüm senaryoların kabul edilebilir çözüm zamanı içerisinde çözüldüğü, küçük ve orta boyutlu problemlere kısa sürelerde uygun çözümler üretildiği ve hedef sayısı arttıkça çözüm zamanının arttığı tespit edilmiştir. Problemin kompleks ve dinamik bir yapıda olması, problem içerisindeki parametrelerin sürekli değişmesi ve sistem içerisindeki hedef sayısının anlık olarak artıp-azalabilmesi, çözüm süresindeki artışın temel nedenleri arasında görülmektedir. Buna ek olarak, çizelge ve tablodaki sonuçlardan algoritmanın parametrelerdeki değişmelere hızlı bir şekilde tepki verdiği, birim zamanda sisteme giren hedef sayısı arttığında ve hedeflerin varışlararası zaman sıklaştığında algoritmanın etkili çözümler ürettiği tespit edilmiştir.

4.3. Sonuç

Dünya üzerinde birçok ülkenin denizlere ya da okyanuslara sınırları bulunmakta, bu sular turizm başta olmak üzere lojistik, balıkçılık gibi birçok faaliyete ev sahipliği yapmaktadır. Bu karasuları aynı zamanda birçok ülke için önemli bir tehdit oluşturmakta, ülkeler ulusal ve uluslararası güvenliğin sağlanması adına büyük bir çaba sarf etmektedir. Günümüz dünyasında birçok ülkede iç karışıklık ya da savaş olmakta, bu nedenden dolayı birçok insan kendi ülkelerinden, başka ülkelere göç etmek zorunda kalmaktadır. Bu göç işlemi genelde sahil güvenlik güçleri tarafından engellenmekte, bazı zamanlarda ise bu yasadışı göç süreci, mülteci gemilerinin batması sonucu ölümle sonuçlanmaktadır. Bunun dışında kaçakçılık ve yasal olmayan balık avcılığı denizler üzerinde gerçekleşmekte, okyanus gibi daha büyük su alanlarında ise korsancılık faaliyeti yürütülmektedir. Tüm bu problemler

için ortak amaç ise tüm hedeflerin ya da gemilerin minimum sürelerde taranması, izlenmesi ve gerekliyse imha edilmesidir. Tez çalışmasının bu aşamasında, belirtilen problemlerin önüne geçmek ve ulusal güvenlik güçlerine yardımcı olmak adına S-70B SeaHawk Helikopteri ile kısa sürelerde hedeflerin tamamının gözlenebileceği/etkisiz hale getirilebileceği uygun tur rotasının belirlenebilmesi için Tavlama Benzetimi Algoritması'na dayalı bir çözüm yaklaşımı geliştirilmiştir. Ele alınan problem HHGSP-ZP problemi ile ilişkilendirilmiş, önerilen algoritmanın etkinliği gerçek verilere dayalı olarak 50 farklı senaryo üzerinde analiz edilmiştir. Ege Denizi'ndeki gemi trafik yoğunluğunun belirlenmesinde tıkanıklık analizinden yararlanılmış, geliştirilen senaryolarda belirli bir zamanda sisteme giren hedef sayısı Poisson Dağılımına göre, hedeflerin sisteme varışları arasındaki zaman ise Üstel Dağılıma göre rassal olarak üretilmiştir. Sonuç olarak tüm senaryolarda kısa sürelerde tüm hedeflerin belirtilen zaman penceresi içerisinde imha edildiği tespit edilmiş, analiz sonuçlarından hareketle önerilen çözüm yaklaşımının etkin bir algoritma olduğu belirlenmiştir. Önerilen çözüm yaklaşımının, denizlerdeki ve okyanuslardaki gemi trafiğini yönlendirilmesi ve kontrol edilmesi problemlerinde kullanılabileceği düşünülmektedir. Bu sayede, denizlerdeki trafik kazalarının önlenilebileceği ve kaza riskinin azalabileceği tahmin edilmektedir. Ayrıca önerilen yöntemin, balık toplama cihazları için uygun rota planının belirlenmesi problemlerinde de kullanılabileceği öngörülmektedir.

Bir sonraki bölümde, bu bölümde belirtilen problemin çok vuruculu hali dikkate alınmış ve problemdeki vurucuların belirli bir uçuş kapasitesine sahip oldukları varsayılmıştır. Buna ek olarak sonraki bölümde ele alınan problemde minimum görev zamanı amacına ek olarak minimum görev maliyeti amacı eklenmiş ve bu amaçları eş zamanlı olarak minimize edecek ideal tur rotası tavlama benzetimi algoritmasına ek olarak, yığın tabanlı (genetik algoritma ve NSGA-II) ve çözüm kurucu sezgisel algoritmalar ile çözülmeye çalışılmıştır.

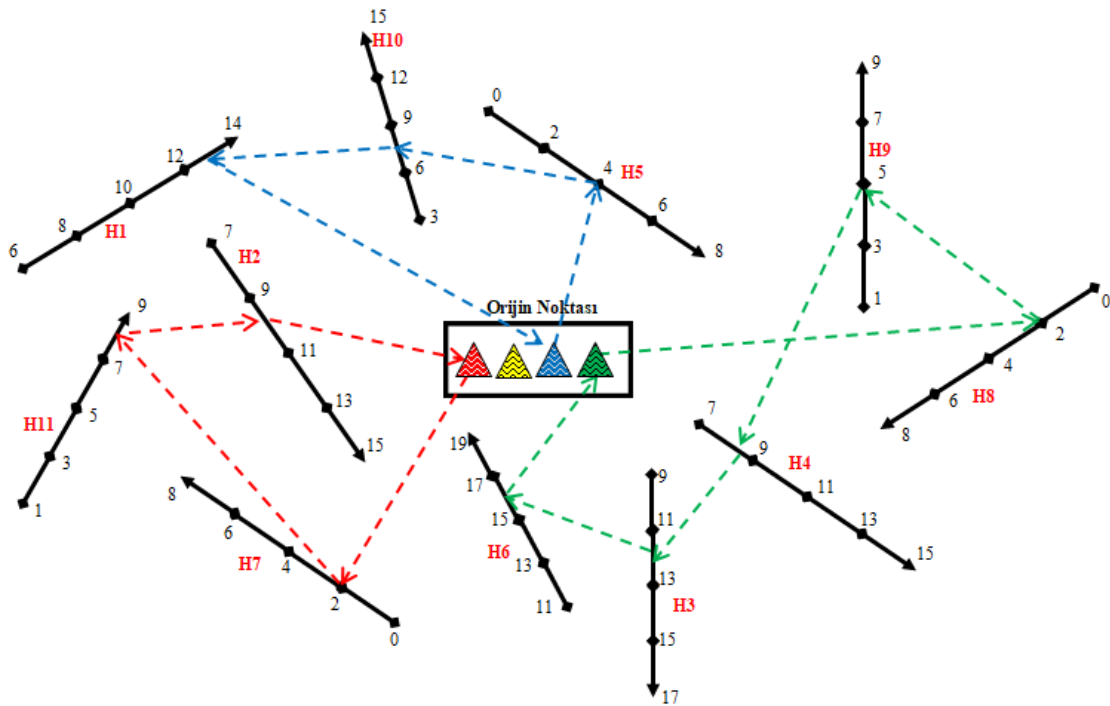
5. UÇUŞ KAPASİTESİ VE ZAMAN PENCERESİ KISITLARI ALTINDA ÇOK AMAÇLI-HETEROJEN FİLOLU-HAREKETLİ MÜŞTERİLİ ARAÇ ROTALAMA PROBLEMİNİN METASEZGİSEL ALGORİTMALARLA ÇÖZÜMÜ

Teknoloji dünyasında yaşanan gelişmeler sayesinde yapay zekaya dayalı robotik sistemler, sağlık, savunma ve ulaşım sektörleri başta olmak üzere birçok alanda yoğun olarak uygulanmaktadır. İHA'lar ve dronelar, yapay zekanın yoğun kullanıldıkları sistemler olup, askeri operasyonlar, lojistik ve gözetleme aktivitelerinde bu sistemlerden yaygın bir şekilde faydalanılmaktadır. Filo halinde gerçekleştirilen operasyonlarda ya da görevlerde, İHA ve dronelerin, araç uçuş kapasitesi ve hedef zaman penceresi kısıtları altında minimum operasyon zamanı veya operasyon maliyeti ile rotalanması gereksinimi ise Hareketli Müşterili Araç Rotalama Problemini (HM-ARP) ortaya çıkarmaktadır. Problem, Dinamik Araç Rotalama Problemi ve Hareketli Hedefli Gezgin Satıcı probleminin hibritlenmesiyle elde edilmekte ve ilgili problemlere ilişkin literatür, Bölüm 2 içerisinde verilmektedir.

HM-ARP' de, $H = \{h_1, h_2, \dots, h_n\}$ hedeflerin ya da müşterilerin kümesini ifade etmekte, her bir hedef (h_i), $w_i [w_{i,x}, w_{i,y}]$ başlangıç koordinatına, m_i hedefi etkisiz hale getirmek için gerekli mühimmat sayına, $z_i [z_{i,g}, z_{i,\zeta}]$ zaman penceresine, sabit bir a_i açısına ve zaman içerisinde değişmeyen bir r_i hızına sahip olmaktadır. $V = \{v_1, v_2, \dots, v_n\}$ seti vuruculara ait kümeyi belirtmekte, her bir v_i vurucusu, zaman içerisinde değişmeyen q_i hızına, $c_i [c_{i,x}, c_{i,y}]$ başlangıç koordinatına, u_i uçuş, yakıt ya da mühimmat kapasitesine, n_i başlangıç görev maliyetine ve f_i uçuş görev maliyetine sahip olmaktadır. Buna ek olarak, vuruculara ait hızların, hedeflere ait hızlardan ($q_i > r_i$) daha fazla olduğu varsayılmaktadır. Bu varsayımlar ve kısıtlamalar altında, vurucuların orijinden harekete başlayıp, görev alanındaki hedefleri ilgili zaman aralığında imha edip, tekrardan başlangıç noktasına döneceği, minimum görev zamanlı ve minimum görev maliyetli tur rotasının tespit edilmesi problemi, HM-ARP olarak isimlendirilmektedir. Problemin klasik gösterimi Şekil 5.1'de gösterilmektedir.

Şekil 5.1'de, farklı hıza, doğrultuya ve zaman penceresine sahip 11 farklı hedef bulunmaktadır. Hedeflerin üzerindeki rakamlar, ilgili hedefin hangi zaman aralığında sistem girdiğini ve hangi zaman aralığında sistemden çıktığını göstermektedir. Örneğin 10

numaralı hedef (H10), $t=3$ anında, ilgili başlangıç noktasında sisteme girmekte ve ilgili doğrultuyu takip ederek $t=15$ anında sistemden ayrılmaktadır. Orijin noktasında ise hedeflerin etkisiz hale getirilmesinde kullanılacak farklı özelliklerde (hız, uçuş kapasitesi, başlangıç ve uçuş maliyeti) dört adet vurucu bulunmaktadır. Problemde zaman birimi sürekli olarak dikkate alınmakta, hedeflere ait zaman penceresi ve araç yakıt kapasite kısıtları altında, vurucuların orijinden harekete başlayıp, tüm hedefleri imha edip, tekrar orijin noktasına döneceği, minimum görev zamanlı ve görev maliyetli tur rotası belirlenmeye çalışılmaktadır. Şekil 5.1'deki sonuçlara göre, operasyonda ilgili amaçları sağlamak için vurucuların tamamı kullanılmamakta, üç vurucu ile hedefler etkisiz hale getirilmekte ve bu sayede daha az İHA kullanarak, vurucu araca ait görev ve bakım onarım maliyetleri azaltılmakta ve araç kullanım ömrü artırılmaktadır. Ayrıca operasyon alanlarında az sayıda İHA kullanılmasıyla düşman unsurlarının dikkatinin çekilmemesi amaçlanmaktadır [76].



Şekil 5.1. Hareketli müşteri araç rotalama problemi

Şekildeki bilgilere göre ve orijin noktasının dikdörtgen kutu ile gösterildiği varsayımı altında, kırmızı renkli vurucu için tur rotası 0-H7-H11-H2-0 olarak belirlenmiştir. Buna ek olarak, mavi renkli vurucu için tur rotası 0-H5-H10-H1-0 ve yeşil renkli vurucu için tur rotası 0-H8-H9-H4-H3-0 olarak belirlenmiştir.

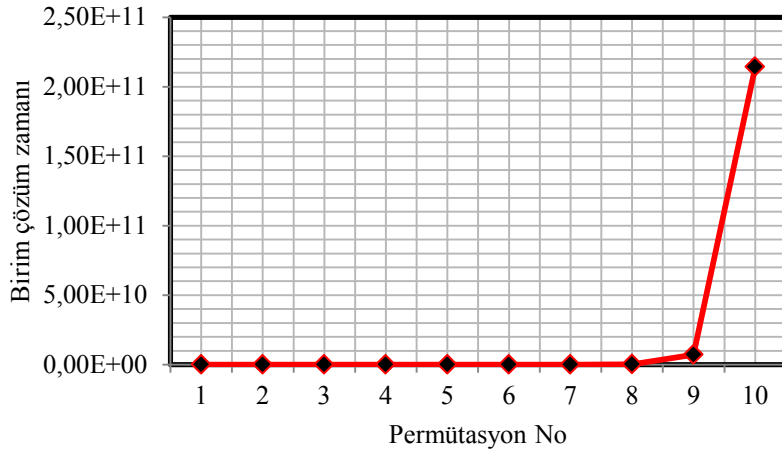
HM-ARP, bazı nedenlerden dolayı D-ARP ve HHGSP'den ayrılmaktadır. HM-ARP'de, HHGSP'den farklı olarak, operasyon alanında çok sayıda vurucu bulunmakta, bunlara ait yakıt ya da mühimmat kapasiteleri tanımlanmakta, herhangi bir hedefin imha edilebilmesi için birden fazla mühimmata ihtiyaç duyulabileceği varsayılmakta, vurucular farklı özelliklere (uçuş hızı, havada kalma süresi, mühimmat kapasitesi, görev maliyeti vb.) sahip olabilmekte ve az sayıda vurucu kullanılarak hedefler etkisiz hale getirilmeye çalışılmaktadır. Buna ek olarak, sistem içerisindeki düğümlerin ya da müşterilerin farklı açı ve hızlarla hareket etmesi, HM-ARP'yi, D-ARP'den ayrılmaktadır. GSP ve ARP' nin NP-Zor problem sınıfında yer almasından dolayı, HM-ARP'nin de NP-Zor problem sınıfında olduğu düşünülmektedir[77, 78, 79, 80].

Vurucu sayısının “n”, hedef sayısının “r” olduğu bir problem için olası çözüm sayısı $((n^r) * r!)$ olarak hesaplanmakta, vurucu ve hedef sayısı arttıkça, çözüm zamanı ve çözüm sayısı önemli ölçüde artmaktadır. Örneğin, 3 vuruculu bir problem için hedef sayısı arttıkça, problemin çözüm sayısı ve biri çözüm zamanındaki artış Çizelge 5.1 ve Şekil 5.2'de gösterilmektedir.

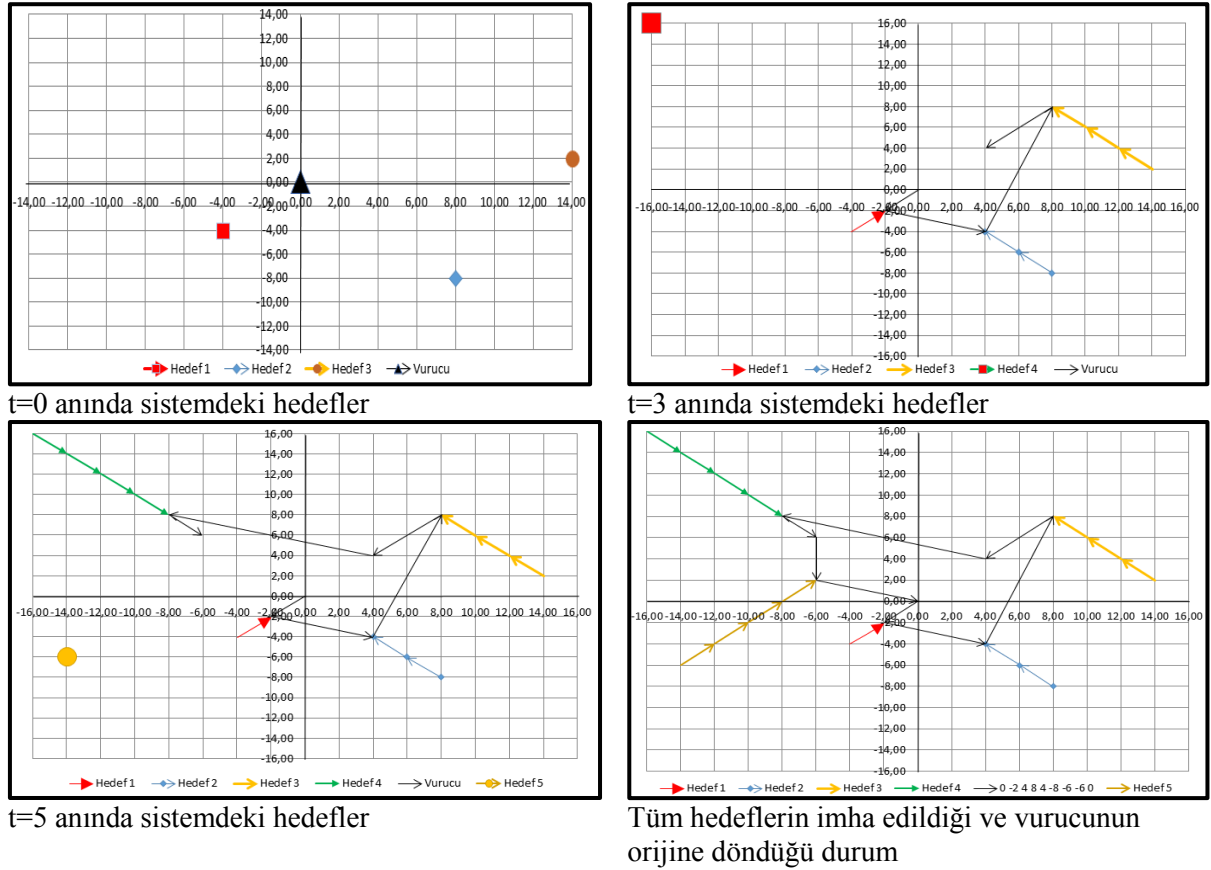
Çizelge 5.1. 3 vuruculu HM-ARP için hedef sayısı arttıkça çözüm sayısında meydana gelen artış

Permütasyon Sıra No	Vurucu Sayısı	Hedef Sayısı	Çözüm sayısı
1	3	1	3
2	3	2	18
3	3	3	162
4	3	4	1944
5	3	5	29160
6	3	6	524880
7	3	7	11.022.480
8	3	8	264.539.520
9	3	9	7.142.567.040
10	3	10	214.277.011.200

Tez çalışmasında, HHGSP literatüründen farklı olarak, sistemde hedef bulunmadığı durumda İHA'nın ya da vurucunun nasıl bir yol izleyeceği planlanmış ve tur rotasının hesaplanmasına entegre edilmiştir.



Şekil 5.2. HM-ARP için Çizelge 5.1’deki denemelere ilişkin birim çözüm zaman grafiği



Şekil 5.3. Hareketli Sistemde hedef bulunmadığı durum için vurucuya ait rota planı

Önceki çalışmalarda vurucu, sistemde hedef bulunmadığı durumda orijine yönlendirilmiş ve başka bir zaman diliminde operasyon alanına hedeflerin girmesi durumunda vurucunun nasıl bir davranış sergileyeceği dikkate alınmamıştır. Bu çalışmada, vurucunun havada kalma kapasitesi yeterli olduğu süre boyunca sistem içerisinde kalması sağlanıp, farklı

zamanlarda sisteme girecek hedeflerin etkisiz hale getirilmesini sağlayacak alternatif bir çözüm mekanizması geliştirilmiştir. Bu sayede, gereksiz vurucu kullanımı ve bunlardan kaynaklı maliyetlerin (uçuş görev maliyeti, bakım onarım maliyeti, yakıt maliyeti vb.) önüne geçilmesi amaçlanmıştır, araç kullanım ömrü artırılmaya çalışılmıştır. İlgili sisteme ilişkin bilgiler şekil 5.3’de gösterilmektedir.

Şekil 5.3’de, 5 tane hedefin tek bir vurucu tarafından ilgili algoritmaya göre imha edilmesi anlatılmaktadır. Şekil 5.3.a’da $t=0$ anında yani başlangıç anında sistemde 3 tane hedef bulunmaktadır. Şekil 5.3.b’de vurucu 3 hedefi imha etmiş ve sistemde hedef bulunmadığı için orijine yöneldiği esnada yani $t=3$ anında 4 numaralı hedef sisteme dâhil olmuştur. Şekil 5.3.c’de ise 4 numaralı hedef imha edildikten sonra vurucu sistemde hedef bulamadığı için orijine yönelmiş ve $t=5$ esnasında sisteme 5 numaralı hedefin dâhil olduğunu tespit etmiştir. Daha sonra vurucu Şekil 5.3.d’deki rotayı izleyerek 5 numaralı hedefi $[-6,2]$ noktasında imha ettikten sonra orijine dönmüştür. Algoritma tüm bu işlemleri kapasite ve zaman penceresi kısıtlarını dikkate alarak gerçekleştirmiştir.

Literatürde, HM-ARP ile yakın özelliklere sahip olduğu düşünülen çalışma Groba ve ark. ait olmaktadır [81]. Groba ve ark., balıkçı gemilerinin, Hint okyanusunda balık avlamada kullandıkları balık toplama cihazlarını, toplam görev zamanını minimize edecek şekilde toplanması problemini incelemiştir.

Çizelge 5.2. Bu tez çalışması ve Groba ve ark., çalışmaları arasındaki fark

Groba ve ark., dikkate aldığı problem	Tezde dikkate alınan problem
➤ Zaman kavramının kesikli olduğu varsayılmıştır. ➤ Balık toplama cihazları ya da hedefler için zaman penceresi kısıdı tanımlanmamıştır. ➤ Sistem içerisindeki balık toplama cihazı ya da hedef sayısı zaman içerisinde değişmemektedir. ➤ Göreve çıkan balıkçı gemileri için herhangi bir kapasite kısıdı tanımlanmamıştır. ➤ Filodaki araçların tamamı kullanılmıştır. ➤ Amaç fonksiyonu içerisinde dikkate alınan herhangi bir amaç bulunmamaktadır.	➤ Zaman kavramının sürekli olduğu varsayılmıştır. ➤ Sistem içerisindeki hedeflerin belirli bir zaman aralığı içerisinde imha edilmesi gerekmektedir. ➤ Herhangi bir “t” anında sistem içerisindeki hedef sayısı artabilmekte ya da azalabilmektedir. ➤ Sistemdeki vurucuların için havada kalma kapasitesi (uçuş kapasitesi) tanımlanmıştır. ➤ Filodaki araçlar, araçların spesifik özellikleri ve maliyet kriterleri dikkate alınarak ve fazla sayıda araç kullanılmaması amacı doğrultusunda rotalanmaya çalışılmıştır. ➤ Toplam görev süresi ve görev maliyetinin minimize edildiği çok amaçlı optimizasyon gerçekleştirilmiştir.

Problemde balıkçı gemilerinin ve balık toplama cihazlarının belirli bir hız ile hareket ettiğini ifade etmişlerdir. HM-ARP ile Groba ve ark., ele aldıkları problem benzer özellikler taşımakla birlikte, birçok açıdan birbirinden ayrılmaktadır. İkili çalışma arasındaki fark Çizelge 5.2’de gösterilmektedir.

HM-ARP, geniş bir uygulama alanına sahip olmakta, farklı amaçlar ve kısıtlamalar altında incelenebilmektedir. Tez kapsamında dikkate alınan probleme ilişkin varsayımlar ve kısıtlamalar aşağıda ifade edilmektedir.

- Filodaki vurucuların hızları, hedeflerin hızlarından daha fazla olmalıdır.
- Vurucuların, sistemdeki hedeflerin tamamını etkisiz hale getirdikten sonra orijin noktasına dönmesi gerekmektedir.
- Hedefler, koordinat eksenin $[x,y]$ dört farklı bölgesinde, herhangi bir açı ve hız ile hareket edebilirler.
- Vuruculara ve hedefler, zaman içerisinde değişmeyen bir hızla sistem içerisinde hareket etmektedir.
- Vurucular farklı hızlara, havada kalma kapasitelerine ve görev maliyetlerine sahip olmaktadır.
- Hedeflere ait açılarda zaman içerisinde herhangi bir değişme olmamaktadır.
- Hedeflere ait bir sisteme giriş ve çıkış zamanı tanımlanmakta, hedefler bu zaman aralığı içerisinde operasyon alanında bulunmaktadır.
- Vurucuların havada kalma kapasitesi sınırlı kalmakta, mühimmat ya da cephanelik için herhangi bir sınırlama bulunmamaktadır.
- Operasyon alanı için herhangi bir sınırlama bulunmamaktadır.
- Her bir hedefin bir adet mühimmat ile etkisiz hale getirilebileceği varsayılmaktadır.
- Hedeflere ilişkin başlangıç pozisyonunun, hızın, açının ve sisteme giriş-çıkış zamanlarının başlangıçta bilindiği varsayılmaktadır.
- Vuruculara ait hız, kapasite, başlangıç ve birim uçuş maliyet bilgilerinin başlangıçta bilindiği varsayılmaktadır.
- Hedeflere ait hızların 30 km/sa ile 400 km/sa arasında, vuruculara ait hızların 401 km/sa ile 700 km/sa arasında olduğu varsayılmaktadır.
- Vurucu ve hedeflere ait açıların 0 ile 360 derece arasında olabileceği, hedeflerin başlangıç koordinatlarının ise $[\pm 400 \text{ km}, \pm 400 \text{ km}]$ arasında olduğu varsayılmaktadır.
- Problemde temel amaç, minimum operasyon zamanlı ve minimum görev maliyetli turun belirlenmesidir.

Bir sonraki bölümde ilgili problemin çözümünde kullanılan çözüm metodolojileri takdim edilmektedir.

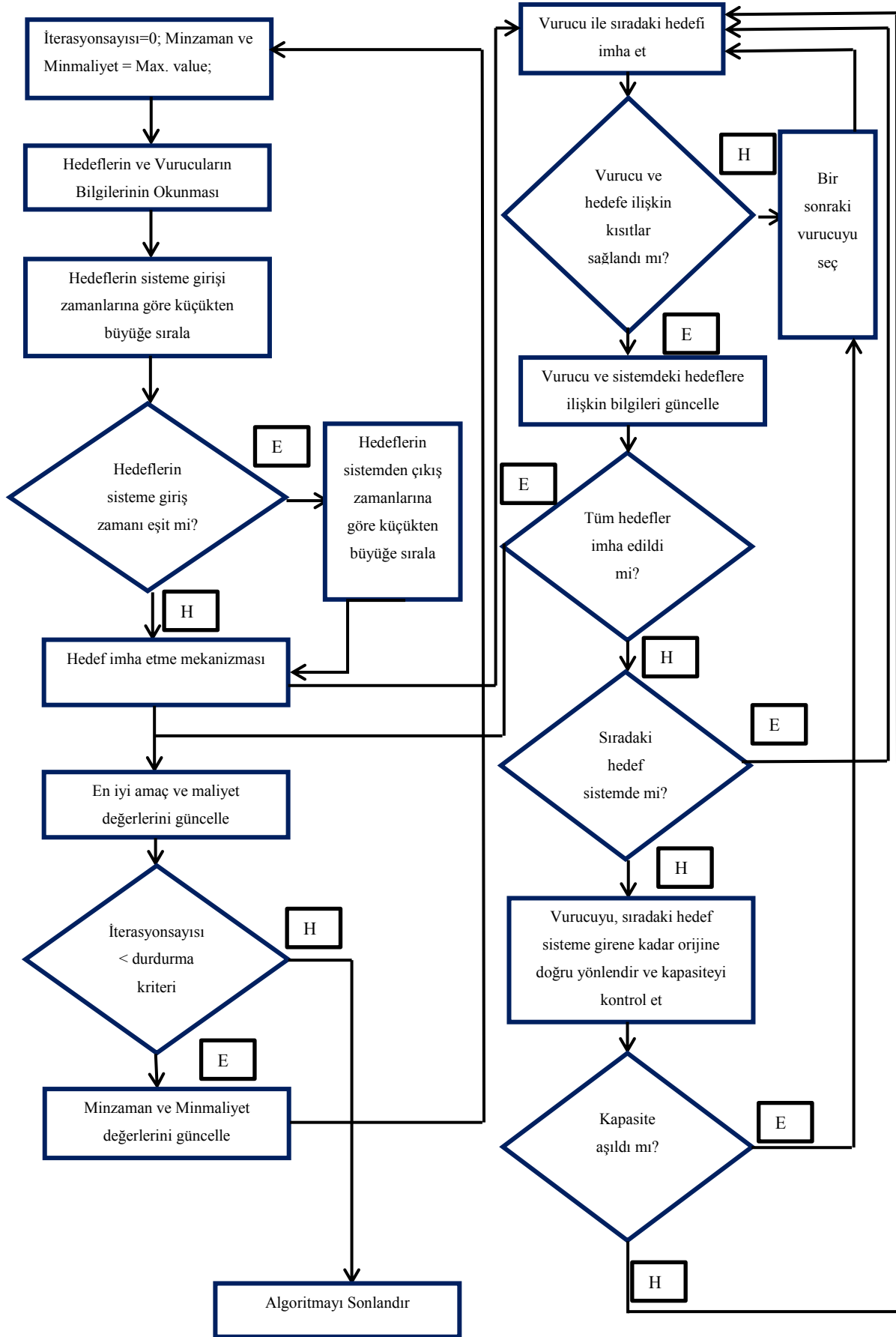
5.1. HM-ARP İçin Önerilen Çözüm Metodolojileri

HM-ARP içerisinde hedef sayısının herhangi bir “t” anında artabilmesi ya da azalabilmesi, vurucu ve hedeflere ait koordinatların sürekli hesaplanması ve vuruculara ait koordinat bilgilerinin sürekli kontrol edilmesi nedenlerinden dolayı problem dinamik bir yapıya sahip olmaktadır. Bu dinamik yapı, problemin matematiksel modelleme ile çözülmesini zorlaştırmakta, matematiksel modellemenin kullanıldığı birimlerde hedef ve vurucu sayısı az alınıp, zaman biriminin kesikli olduğu varsayılmaktadır. Gerçek hayat uygulamalarında zamanın sürekli olması nedeniyle geliştirilen matematiksel modelleme yaklaşımları yetersiz kalabilmektedir. Buna ek olarak, vurucu ve hedef sayısı arttıkça, probleme ilişkin çözüm zamanı önemli ölçüde artmaktadır. Bu nedenlerden dolayı, literatürdeki makalelerde belirtilen problemlerin çözülebilmesi için probleme özgü sezgisel algoritmalar geliştirilmekte ve metasezgisel algoritmalar kullanılmaktadır. Tez çalışmasında, problemin minimum görev zamanı ve minimum görev maliyeti amacı doğrultusunda çözülebilmesi için sezgisel ve metasezgisel algoritmalar kullanılmaktadır. İlgili problemin çözümünde kullanılan çözüm yaklaşımları aşağıda verilmektedir.

5.1.1. Çok hedefli çok vuruculu rotalama algoritması

Çok Hedefli-Çok Vuruculu Rotalama Algoritması (ÇARA), HM-ARP için geliştirilmiş çözüm kurucu sezgisel bir algoritma olup, literatürde bilindiği kadarıyla ilk kez kullanıldığı düşünülmektedir. Önerilen algoritmada (Şekil 5.4), hedefler operasyon alanına giriş zamanlarına ve operasyon alanından ayrılma zamanlarına göre önceliklendirilmekte ve filodaki vurucular bu öncelik değerlerine göre, vurucuların havada kalma süreleri de dikkate alınarak hedefleri etkisiz hale getirmeye çalışmaktadır. Algoritmada 4 adet karar noktası bulunmakta, bu noktalarda verilen kararlar aşağıda ifade edilmektedir.

Hedeflerin sisteme giriş zamanı eşit mi? Bu karar mekanizmasında, sisteme giriş zamanlarına göre hedefler küçükten büyüğe doğru sıralanır. Eğer sistemi giriş zamanları aynı olan hedefler varsa o zaman çıkış zamanı en küçük yani en yakın olan hedef öncelik kazanır ve diğer hedeften daha önce etkisiz hale getirilir.



Şekil 5.4. Çok hedefli çok vuruculu rotalama algoritması

Vurucu ve hedefe ilişkin kısıtlar sağlandı mı? Sıradaki hedef, ilgili vurucu tarafından imha edildikten sonra hedef ve vurucu için bazı durumlar kontrol edilir. Bu durumlar; hedef, vurucunun havada kalma kapasitesi aşılmadan ve orijine dönebilecek kadar enerji bulunduğu durum dikkate alınarak imha edilip edilmediği ve hedefin, ilgili zaman aralığı içerisinde sistemden çıkmadan etkisiz hale getirilip getirilmediğidir.

Tüm hedefler imha edildi mi? Sistemdeki tüm hedefler imha edildiyse algoritma sonlanmakta, eğer tüm hedefler imha edilmemişse sıradaki hedefe geçilmektedir.

Sıradaki hedef sistemde mi? Vurucu sistemde bulunduğu durumda, sıradaki hedef sisteme girmemiş olabilir. Bu durumda vurucu orijine yönlendirilmekte, eğer vurucu uçuş zaman kapasitesi sınırları içerisinde hedefi sistemde tespit ederse, ilgili hedefe doğru yönelmektedir.

5.1.2. Rassal hedef imha algoritması

Rassal Hedef İmha Algoritmasında (RSA), her bir iterasyonda, filodaki vurucular kendi aralarında ve hedefler kendi aralarında rassal bir şekilde sıralanmakta, daha sonra ilk sıradaki vurucudan başlanarak sistemdeki hedefler, vurucu uçuş kapasiteleri dikkate alınarak imha edilmeye çalışılmaktadır. Eğer vurucunun kapasitesi sıradaki hedefi imha etmeye yetmiyor ise o vurucu için tur sonlandırılmakta ve sıradaki hedef, filodaki sırada bekleyen bir diğer vurucu tarafından imha edilmektedir. Operasyon alanındaki tüm hedefler imha ettikten sonra tur sonlanmakta ve bu tur bir çözüm olarak hafızada tutulmaktadır. Algoritma, belirtilen durdurma kriterine ulaşıncaya sonlandırılmakta ve hafızadaki çözümler arasından en iyisi çıktı olarak kullanıcıya sunulmaktadır.

5.1.3. Genetik algoritma

Genetik Algoritma Holland tarafından geliştirilmiş olup, çözüm yığınının evrimsel gelişiminin simüle edildiği, gerçek hayat problemlerinde sıklıkla kullanılan metasezgisel bir çözüm algoritmasıdır [82,83]. Genetik Algoritmada, belirli bir sayıda çözümden oluşan bir başlangıç yığına iteratif adımlar uygulanarak ve çözüm kalitesini geliştirmek için farklı çözüm mekanizmaları kullanılarak problem için optimal bir çözüme ulaşılmaya çalışılmaktadır. Algoritmada, çözüm gösterimi, yeni yığının seçilmesi, çaprazlama ve

mutasyon operatörleri gibi çözüm hızına ve kalitesine etki eden birçok parametre bulunmaktadır. Tezde kullanılan parametreler ve uygunluk fonksiyonuna ilişkin bilgiler aşağıda ifade edilmektedir.

Çözüm gösterimi

Çözüm gösterimi, incelenen problemin özelliklerine göre farklılık gösterebilmekte, bu çalışmada dikkate alınan çözüm gösterimi Şekil 5.5’de gösterilmektedir.

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

Şekil 5.5. 10 hedeften oluşan bir kromozom için çözüm gösterimi

Şekil 5.5’de, tek bir vurucu tarafından sistemdeki hedeflerin imha edilmesine ilişkin örnek bir çözüm gösterimi gösterilmektedir. İlgili gösterimde bir kromozom, 10 tane genden oluşmakta ve her bir gen bir hedefe karşı gelmektedir. Bu bağlamda ilgili kromozoma göre vurucunun izleyeceği tur rotası “0-10-1-4-7-9-3-8-6-5-2-0” şeklinde olmakta, “0” vurucuya ait başlangıç ya da orijin noktasını göstermektedir.

Vurucu sayısının iki ve üzerinde olduğu durumda, kromozom yapısı, vurucu sayısı kadar eşit parçaya bölünmekte ve her bir parça bir vurucu için ilgili turu ifade etmektedir. Örneğin Şekil 5.5’deki örnek, 2 vurucu kullanılarak imha edilmiş olsaydı; birinci vurucu için tur rotası “0-10-1-4-7-9-0” ve ikinci vurucu için tur rotası “0-3-8-6-5-2-0” şeklinde olacaktı. Vurucu sayısının üç olması durumunda ise birinci vurucu “0-10-1-4-7-0”, ikinci vurucu “0-9-3-8-0” ve üçüncü vurucu “0-6-5-2-0” sırasıyla hedefleri imha etmektedir.

Başlangıç popülasyonunun üretilmesi

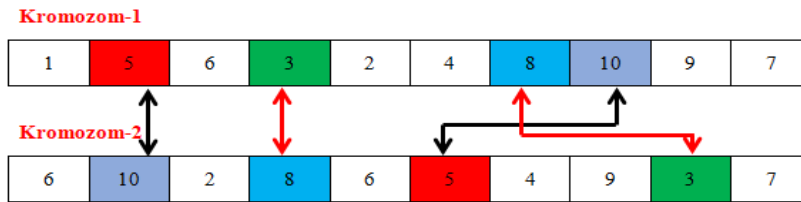
Önerilen algoritma için başlangıç yığını rassal bir şekilde belirlenmiş ve yeni yığının oluşturulmasında bu popülasyondan yararlanılmıştır.

Yeni yığın seçim mekanizması

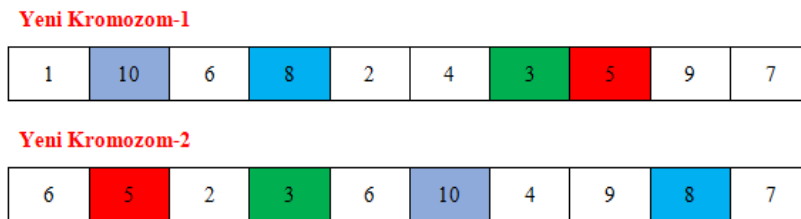
Yeni yığın seçim mekanizması popülasyon içerisindeki güçlü bireylerin sonraki iterasyonlara aktararak daha kaliteli çözümler oluşturulmasını sağlayan bir seçim mekanizmasıdır. Bu çalışmada yeni yığının seçilmesinde turnuva mekanizmasından yararlanılmakta ve n tane kromozomdan oluşan bir popülasyonda, yeni yığındaki her bir kromozom için popülasyondan rasgele seçilen m çözüm karşılaştırılmakta ve bunlar arasından en iyi çözüme sahip kromozom ilgili yığının yeni bireyi olmaktadır [81]. Bu sayede güçlü bireylerin sonraki nesillere aktarılması amaçlanmaktadır.

Çaprazlama

Çaprazlama operatörleri, farklı kromozomların üretilmesiyle çözüm uzayının farklı noktalarında arama yapılmasını sağlayan ve bu sayede çözümün yerel optimuma takılmasını engelleyen genetik algoritma adımlarından bir tanesidir. Literatürde çaprazlama operatörleri için geliştirilmiş birçok algoritma bulunmakta, uygulama çalışmada çaprazlama işlemi için PMX (Partially Mapping Crossover) operatöründen yararlanılmaktadır. PMX operatöründe, bir kromozom içerisindeki rassal olarak n adet gen belirlenmekte ve seçilen genler, başka bir kromozomun genleriyle yer değiştirilerek farklı bireyleri oluşturulması amaçlanmaktadır. İlgili mekanizmaya ilişkin bilgiler Şekil 5.6 ve 5.7’de gösterilmektedir.



Şekil 5.6. PMX operatörüne göre çaprazlamadan önce kromozomların durumu



Şekil 5.7. PMX operatörüne göre çaprazlamadan sonra kromozomların durumu

Şekil 5.6’da “5 ile 10” ve “3 ile 8” hedeflerinin yer değiştirdiği varsayılmış, Şekil 5.7’de ise oluşan iki yeni kromozomdaki hedef sıraları gösterilmiştir. Tek vuruculu durumda kromozomdaki tüm genler, tek vurucuya ait tur rotasını oluşturmaktadır. İki vuruculu durumda ise kromozom ikiye bölünerek uygunluk değeri hesaplanmaktadır. Örneğin Şekil 5.7’de oluşan bir numaralı kromozomda 0-1-10-6-8-4-3-5-9-7-0 turu elde edilmekte, birinci vurucu = 1-10-6-8-2 ve ikinci vurucu = 4-3-5-7-9 hedeflerini imha etmektedir.

Mutasyon

Kromozomlarda bulunan genlerin birinde ya da birkaçında önceden belirlenmiş bir kabul olasılık değerine göre yer değişikliği yapılarak çözüm uzayında yoğunlaşma yapılmasını sağlayan genetik operatör türüne mutasyon operatörü denilmektedir. Literatürde, ikili yer değiştirme, araya ekleme veya tersine çevirme gibi çeşitli mutasyon operatörleri bulunabilmektedir. Bu çalışmada, mutasyon operatörünün kullanılmasında ikili yer değiştirme metodundan yararlanılmaktadır.

Uygunluk fonksiyonu

Genetik algoritmada, kromozomların her biri bir çözüme karşılık gelmektedir. Algoritmada, bu çözümlerin problem kısıtlarına uygunluğunun değerlendirilmesi ve amaca olan katkısının belirlenmesi ise uygunluk fonksiyonunun içerisinde gerçekleştirilmektedir [84]. Tez çalışmasında, uygunluk fonksiyonu içerisinde toplam operasyon zamanı ve toplam operasyon maliyeti minimize edilmeye çalışılmaktadır. Uygunluk fonksiyonuna ilişkin bilgiler Çizelge 5.3’de verilmektedir.

Çizelge 5.3’e bakıldığı zaman iki amaç eş zamanlı minimize edilmeye çalışılmakta ve çok amaçlı optimizasyon işlemi gerçekleştirilmektedir. Buradaki önemli noktalardan biri, iki amacın birbirine olan etkisinin belirlenmesi ve çok amaçlı optimizasyona ihtiyaca olup olmadığının tespit edilmesidir. Bu bağlamda, rassal oluşturulmuş bir senaryo üzerinde algoritma her bir amaç için ayrı ayrı çalıştırılmıştır. İlgili senaryoya ilişkin bilgiler ve her bir amaç için algoritma sonuçları Çizelge 5.4, 5.5 ve 5.6’da gösterilmektedir.

Çizelge 5.3. HM-ARP için uygunluk fonksiyonuna ilişkin bilgiler

İndis $i = \text{vurucu indisi}, i \in I = \{1, 2, \dots, n\}$
Parametre $b_i = i. \text{ vurucu için başlangıç maliyeti}$ $c_i = i. \text{ vurucu için uçuş maliyeti}$
Değişken $x_i = i. \text{ vurucu için toplam havada kalma süresi}$
Amaç Amaç 1 = $Z_{\text{görev zamanı}} = \sum_{i=1}^n x_i$ Amaç 2 = $Z_{\text{görev maliyeti}} = \sum_{i=1}^n (b_i + x_i * c_i)$ $Z_{\min} = (\text{Min}(\text{Amaç 1})) \& (\text{Min}(\text{Amaç 2}))$

Çizelge 5.4. Amaçların analizinde kullanılan senaryolar

Vurucu Adı	Başlangıç Maliyeti	Uçuş Maliyeti	Hız
Vurucu-1	16000	3	644
Vurucu-2	2000	1410	601
Vurucu-3	8000	86	574
Vurucu-4	1500	164	566
Vurucu-5	500	560	697
Vurucu-6	27000	146	633
Vurucu-7	10000	1310	598
Vurucu-8	14000	135	589
Vurucu-9	5000	28	431
Vurucu-10	600	2960	662

İki sonucu karşılaştırdığımızda, minimum zaman amacının gözetildiği durumda toplam görev zamanı “36,62”, toplam görev maliyeti ise “92366,87” çıkmıştır. Minimum maliyet amacı gözetildiğinde ise toplam görev zamanı “42,07”, toplam görev maliyeti ise “60922,04” birim çıkmıştır.

Çizelge 5.5. Amaç 1 için algoritma sonuçları

Aktif olan vurucu sayısı = 7
Minimum Time = 36,62
Minimum görev maliyeti = 92366,87
Minimum vurucu görev sırası = [0-vurucu6] - [1-vurucu8] - [2-vurucu2] - [3-vurucu3] - [4-vurucu7] - [5-vurucu5] - [6-vurucu10]
İterasyon sayısı: 5000
Popülasyon = 50, çaprazlama 0,9, mutasyon: 0,9, turnuvada yer değişen 8, rakip 35
Çözüm zamanı: 4 dakika 12 saniye

Çizelge 5.6. Amaç 2 için algoritma sonuçları

Aktif olan vurucu sayısı = 6
Minimum Time = 42,07
Minimum görev maliyeti = 60922,04
Minimum vurucu görev sırası = [vurucu3]- [vurucu10] - [vurucu4] - [vurucu5]- [vurucu2]-[vurucu7]
İterasyon sayısı: 5000
Popülasyon = 50, çaprazlama 0,9, mutasyon: 09, turnuvada yer değişen 8, rakip 35
Çözüm zamanı: 4 dakika 10 saniye

Algoritma, zaman amacını gözetmediği durumda maliyet kriterini dikkate almayıp, filodaki hızlı araçları dikkate almış ve daha fazla sayıda araçla operasyon işlemini gerçekleştirmiştir. Maliyet amacı gözetildiği durumda ise, filodaki düşük maliyetli araçlarla ve daha az sayıda araç ile görev gerçekleştirilmiştir. Özellikle 6 ve 8 numaralı vurucuların, amaç 1 deki çözüm de kullanılıp, amaç 2 deki çözümde kullanılmayışı bu durumun en büyük örneğidir. Buna ek olarak, düşük maliyete sahip olan vurucu 4'ün, hızının düşük olmasına rağmen minimum görev zamanı amacı içerisinde kullanılmayıp, görev maliyeti içerisinde kullanılması amaç fonksiyonlarının farklı çalıştığına bir göstergesidir.

Algoritmada kullanılan parametreler ise Çizelge 5.7'de, algoritmanın diğer senaryolarda gösterdiği performans Çizelge 5.8'de, gösterilmektedir. Çizelge 5.8'de "V.S." ile vurucu sayısı kastedilmiş ve her bir "Run" 5000 iterasyon için çalıştırılmıştır. Çizelge 5.8'deki sonuçlar, algoritmanın sadece Amaç-1 ve Amaç-2 için ayrı ayrı çalıştırılmasından elde edilmiştir.

Bu sonuçlardan hareketle, Amaç-1 ve Amaç-2'nin çözüm üzerindeki etkisini belirlemek için istatistiksel açıdan analiz edilmesi gerekmektedir.

Çizelge 5.7. HM-ARP için amaçların karşılaştırılmasında dikkate alınan algoritma parametreleri

Vurucu Sayısı	Hedef Sayısı	Popülasyon	Turnuva rakip sayısı	Çaprazlama	Mutasyon	Çapr. Yer. Değ. El. Say.
5	10	30	18	0,9	0,1	6
6	15	30	18	0,9	0,1	6
7	20	40	27	0,9	0,1	7
8	25	40	27	0,9	0,1	7
9	30	50	35	0,9	0,1	8
10	35	50	35	0,9	0,1	8

Çizelge 5.8. Amaçların karşılaştırılmasına ilişkin deneysel analiz sonuçları

Senaryo	Amaç 1				Amaç 2				Zaman	Maliyet
	V.S.	Zaman	Maliyet	Ç. Zmn	V.S.	Zaman	Maliyet	Ç. Zmn	Fark	Fark
V5_H10_E1	2	12,39	11910,18	29sn	2	20,73	8251,10	27sn	%67,29	%44,35
V5_H10_E2	2	12,07	22976,24	28sn	2	17,33	7758,70	28sn	%43,63	%196,14
V5_H10_E3	2	12,47	21590,78	29sn	2	17,83	13536,02	27sn	%42,91	%59,51
V5_H10_E4	1,5	11,90	21293,18	28sn	1	12,01	2569,39	24sn	%0,97	%728,73
V5_H10_E5	2	11,94	19408,83	27sn	2	14,16	3103,18	27sn	%18,64	%525,45
V6_H15_E1	3,1	27,27	26092,75	49sn	3	30,18	19061,74	48sn	%10,67	%36,89
V6_H15_E2	2	10,27	16149,46	46sn	2	15,42	7832,60	49sn	%50,17	%106,18
V6_H15_E3	2,4	19,97	58448,20	45sn	2,8	27,73	35582,04	47sn	%38,82	%64,26
V6_H15_E4	2	12,43	20270,04	53sn	2	17,33	8436,05	46sn	%39,35	%140,28
V6_H15_E5	2,3	18,86	34499	46sn	2,7	26,25	14627,40	51sn	%39,18	%135,85
V7_H20_E1	3,3	26,76	33075,27	1dk35sn	3,10	31,73	22095,71	1dk30sn	%18,55	%49,69
V7_H20_E2	3	22,5	31535,45	1dk29sn	3	27,76	18519,25	1dk29sn	%23,40	%70,28
V7_H20_E3	2,9	18,89	31885,69	1dk33sn	2,8	22,58	18194,16	1dk29sn	%19,53	%75,25
V7_H20_E4	3	25,11	50296,04	1dk33sn	3,3	35,98	24860,74	1dk27sn	%43,29	%102,31
V7_H20_E5	3	25,37	34911,20	1dk30sn	3,10	31,02	25664,72	1dk34sn	%22,29	%36,03
V8_H25_E1	3,3	26,03	37141,49	2dk2sn	3,8	33,14	26810,06	2dk11sn	%27,30	%38,54
V8_H25_E2	4	32,71	55557,15	2dk8sn	4,5	40,33	43818,46	2dk5sn	%23,28	%26,79
V8_H25_E3	4,9	44,55	44541,02	2dk19sn	4,3	41,58	36262,64	2dk4sn	-%7	%22,83
V8_H25_E4	5,6	51,26	97381,22	2dk6sn	5,2	48,66	72467,51	2dk6sn	-%5,34	%34,38
V8_H25_E5	4,3	36,94	90087,33	2dk10sn	5	47,37	77548,18	2dk5sn	%28,22	%16,17
V9_H30_E1	4,6	31,49	57739,72	3dk19sn	4,2	38,92	23725,47	3dk24sn	%23,58	%143,37
V9_H30_E2	5,2	39,10	47978,74	3dk29sn	5,4	51,07	34112,26	3dk29sn	%30,62	%40,65
V9_H30_E3	4,5	29,90	43126,53	3dk27sn	3,9	43,63	21074,11	3dk23sn	%45,88	%104,64
V9_H30_E4	7,7	68,34	950294,28	3dk23sn	7,5	77,48	932634,55	3dk25sn	%13,37	%1,89
V9_H30_E5	6,5	44,12	78573,72	3dk18sn	5,4	50,95	51593,44	3dk24sn	%15,49	%52,29
V10_H35_E1	7,2	35,8	919761,71	5dk30sn	6,3	50,44	669199,35	4dk10sn	%40,89	%37,44
V10_H35_E2	8,10	61,23	94985,53	4dk7sn	7,5	67,81	69429,18	4dk30sn	%10,75	%36,81
V10_H35_E3	7,9	52,16	81629,19	4dk8sn	7,3	59,23	64003,44	4dk13sn	%13,56	%27,54
V10_H35_E4	8,10	56,45	87367,71	4dk8sn	7,3	64,09	75028,06	2dk19sn	%13,54	%16,45
V10_H35_E5	8,3	45,93	1285654,61	4dk29sn	6,4	59,74	941956,75	6dk1sn	%30,07	%36,49

Bu bağlamda, her bir amaç için ayrı ayrı yapılan analizler sonucunda, Amaç-1 ve Amaç-2 için görev zamanı ve görev maliyeti ortalamaları arasında anlamlı bir farklılık olup olmadığının belirlenmesinde Wilcoxon Signed Ranked Test yönteminden yararlanılmıştır. Analiz çalışması SPSS 15.0 programında gerçekleştirilmiş ve elde edilen sonuçlar aşağıda ifade edilmiştir. Amaç-1 ve Amaç-2 için, görev zamanları arasındaki farklılığın değerlendirilmesi için aşağıdaki hipotez testi kurulmuş ve elde edilen sonuçlar Resim 5.1’de gösterilmiştir.

H0: Amaç-1 ve Amaç-2 için elde edilen görev zamanı ortalamaları arasında anlamlı bir farklılık yoktur.

H1: Amaç-1 ve Amaç-2 için elde edilen görev zamanı ortalamaları arasında anlamlı bir farklılık vardır.

Ranks

		N	Mean Rank	Sum of Ranks
Amac2 - Amac1	Negative Ranks	2(a)	4,00	8,00
	Positive Ranks	28(b)	16,32	457,00
	Ties	0(c)		
	Total	30		

a Amac2 < Amac1
b Amac2 > Amac1
c Amac2 = Amac1

Test Statistics(b)

	Amac2 - Amac1
Z	-4,618(a)
Asymp. Sig. (2-tailed)	,000

a Based on negative ranks.
b Wilcoxon Signed Ranks Test

Resim 5.1. Amaç-1 ve Amaç-2 için görev zamanları arasındaki anlamlı farklılığın analizi

Bu şekle göre anlamlılık değeri 0,05 den küçük çıkmakta(“Asymo. Sig = 0”), bu durum iki ortalama arasında anlamlı bir farklılık olduğunu göstermektedir. Ayrıca toplam 30 senaryonun 28’ inde görev zamanının, Amaç-2’de elde edilen sonuçlardan daha küçük çıktığı, yalnızca 2 senaryoda aksi bir durumun gerçekleştiği görülmektedir. Bu senaryolarda ise iki ortalama arasındaki fark oldukça düşük olduğu tespit edilmektedir.

Senaryoların, Amaç-1 ve Amaç-2 için ayrı ayrı çalıştırıldığında görev maliyetleri arasındaki farklılığın değerlendirilmesi için kurulan hipotez testi aşağıda verilmiş ve elde edilen sonuçlar Resim 5.2’de ifade edilmiştir.

H0:Amaç-1 ve Amaç-2 için elde edilen görev maliyet ortalamaları arasında anlamlı bir farklılık yoktur.

H1: Amaç-1 ve Amaç-2 için elde edilen görev maliyet ortalamaları arasında anlamlı bir farklılık vardır.

Ranks

		N	Mean Rank	Sum of Ranks
Amac2 - Amac1	Negative Ranks	30(a)	15,50	465,00
	Positive Ranks	0(b)	,00	,00
	Ties	0(c)		
	Total	30		

a Amac2 < Amac1
b Amac2 > Amac1
c Amac2 = Amac1

Test Statistics(b)

	Amac2 - Amac1
Z	-4,782(a)
Asymp. Sig. (2-tailed)	,000

a Based on positive ranks.
b Wilcoxon Signed Ranks Test

Resim 5.2. Amaç-1 ve Amaç-2 için görev maliyetleri arasındaki anlamlı farklılığın analizi

Resim 5.2’ye göre anlamlılık değeri 0,05 den küçük çıkmakta(“Asymo. Sig = 0”), bu durum iki ortalama arasında anlamlı bir farklılık olduğunu göstermektedir. Bu şekle göre

30 senaryonun tamamında, minimum maliyet amacının gözetildiği zaman, Amaç-2 bazlı çalıştırılan algoritmaların, Amaç-1 bazlı çalıştırılan algoritmalarından daha düşük değerlere sahip olduğu anlaşılmaktadır.

Çizelge 5.9. HM-ARP için önerilen genetik algoritma

```

Adım 1. Başla,  $t = 0$ , vurucusayı = 1, degisimsay = 0; artismiktari = belirli iterasyonsayı;
 $S_{best} \leftarrow 0$ ,  $f(S_{best}) = \text{Max. value}$ 
Adım 2.  $P_t$  başlangıç yığını oluştur ve  $f(P_t)$ yi hesapla.
    If zaman penceresi ve kapasite kısıtları sağlandıysa then
        If  $f(S_{best}) > f(P_t)$  then  $f(S_{best}) \leftarrow f(P_t)$  ve  $S_{best} \leftarrow P_t$ 
Adım3. While not {durdurma zamanı} do
     $t = t + 1$  yap ve  $P_{t-1}$  den belirtilen yığın seçim mekanizmasına göre  $P_t$  yi seç ve  $f(P_t)$ yi hesapla
    If ilgili yığında herhangi bir kromozom için zaman penceresi ve kapasite kısıtları
    sağlandıysa then
        If  $f(S_{best}) > f(P_t)$  then  $f(S_{best}) \leftarrow f(P_t)$  ve  $S_{best} \leftarrow P_t$ 
    Else
        degisimsay = degisimsay + 1
    Seçilen yeni yığına çağrazlama mekanizmasını uygula ve  $f(P_t)$ yi hesapla.
    If ilgili yığında herhangi bir kromozom için zaman penceresi ve kapasite kısıtları
    sağlandıysa then
        If  $f(S_{best}) > f(P_t)$  then  $f(S_{best}) \leftarrow f(P_t)$  ve  $S_{best} \leftarrow P_t$ 
    Else
        degisimsay = degisimsay + 1
    Seçilen yeni yığına mutasyon mekanizmasını uygula ve  $f(P_t)$ yi hesapla.
    If ilgili yığında herhangi bir kromozom için zaman penceresi ve kapasite kısıtları
    sağlandıysa then
        If  $f(S_{best}) > f(P_t)$  then  $f(S_{best}) \leftarrow f(P_t)$  ve  $S_{best} \leftarrow P_t$ 
    Else
        degisimsay = degisimsay + 1
    If degisimsay < vurucusayı * artismiktari
        vurucusayı = vurucusayı + 1

    If degisimsay < vurucusırasıdegis
        vurucuların baslangic sırasını degistir ve vurucu sayısı = 0 yap.

Adım 4. Minimum araç sayılı ve görev maliyetli tur rotası ve vuruş bilgilerini raporla.
Adım 5. Bitiş

```

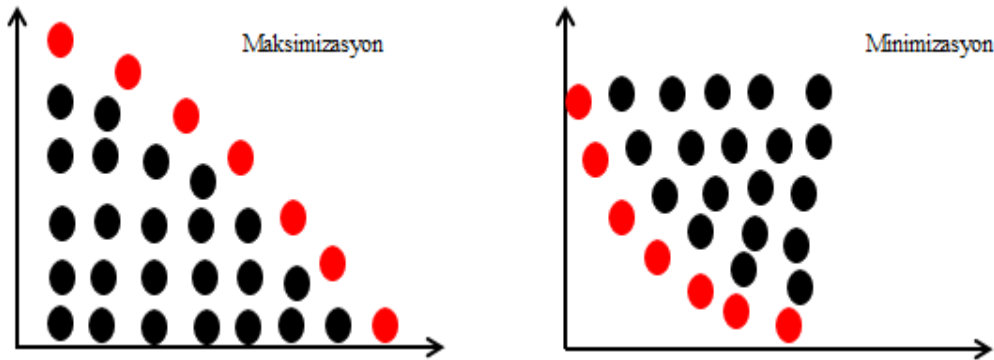
Resim 5.1 ve 5.2'deki sonuçlardan, amaçların birbiri ile çeliştiği, toplam görev zamanının minimize edildiği durumda toplam görev maliyetinin arttığı, toplam görev maliyetinin minimize edildiği durumda ise toplam görev zamanının arttığı anlaşılmaktadır. Bu durum, uygunluk fonksiyonunda iki amacın da dikkate alınmasının doğru olduğunu ve problemin çok amaçlı bir optimizasyon problemi olması gerektiğini göstermektedir. Bu bilgiler doğrultusunda geliştirilen genetik algoritma Çizelge 5.9'da gösterilmektedir.

5.1.4. Pareto optimalite kavramı ve problem için geliştirilen NSGA II algoritmasına dayalı çözüm yaklaşımı

Pareto Verimliliği ya da Pareto Optimal kavramı Vilfredo Pareto tarafından bireylerin maksimum refaha ulaşması için kaynakların dengeli bir şekilde dağıtılmasını esas alan bir ekonomi teorisidir. Bu teoriye göre toplumda bulunan bireylerden en az bir bireyin refah

düzeyini azaltmadan başka bir bireyin refah düzeyini artırma ihtimali bulunmuyorsa bu durumda toplum optimum refah düzeyine ulaşmış denilmektedir. Pareto optimal düzeyine ulaşıldığı durumda toplumdaki bireylerin faydalı mal ve hizmetlerden en iyi derecede faydalanmakta, Pareto optimalitenin sağlanması için üretim ve tüketimde eş zamanlı etkinliğin sağlanması gerekmektedir [85,86].

Pareto Optimalite kavramındaki ödünleşme, zamanla çok amaçlı optimizasyon problemlerine uygulanmış ve ilgili kavram kullanılarak evrimsel algoritmalar önerilmiştir. Algoritma içerisinde herhangi bir popülasyondaki çözüm, popülasyondaki diğer çözümlere kıyasla daha iyi, daha kötü veya aynı olabilir. Buradaki iyi çözüm amaçların herhangi birisinde kötüleşme yapmadan en az bir amacı iyileştiren çözümdür. Optimal çözüm ya da global en iyi çözüm ise çözüm uzayı içerisindeki diğer çözümler ya da bireylerden tarafından bastırılmayan çözümü ifade etmektedir. En iyi çözümlerden ya da diğer çözümler tarafından bastırılmayan çözümlerin oluşturduğu kümeye ise Pareto Optimal Küme denilmektedir. Bastırılmamış çözümler dışında olan çözümlere ise bastırılmış çözümler olarak nitelendirilmekte ve bu çözümler ilgili problem için uygun çözüm olarak kullanılabilmektedir. Maksimizasyon ve minimizasyon problemleri için bastırılmamış (kırmızı nokta) ve bastırılmış çözümler (siyah nokta) Şekil 5.8’de gösterilmektedir [86].



Şekil 5.8. Maksimizasyon ve minimizasyon problemleri için bastırılmış ve bastırılmamış çözümler

Çok Amaçlı Optimizasyon Problemleri NP-Zor özelliğe sahip olmakta, bu nedenden dolayı herhangi bir problem için tüm bastırılmamış çözümler ya da baskın noktalar enumerasyon (birelerle) yöntemi kullanılarak elde edilebilmektedir [87]. Ancak tüm bastırılmamış çözümlerin bulunması zor olacağından ve oldukça zaman alacağından, belirli bir kalite ya da performans düzeyinde tüm bastırılmamış çözümler kümesine en yakın bir alt çözümler

kümesinin oluşturulması Çok Amaçlı Optimizasyon Problemleri için büyük bir öneme sahip olmaktadır [88].

Bu çalışmada, diğer çözüm metodolojilerine kıyasla bağımsız amaçlardan oluşan problemlerde çok amaçlı optimizasyonu etkin bir şekilde yapabilmesi nedeniyle Çok Amaçlı Genetik Algoritma yöntemi kullanılmaktadır. Çok Amaçlı Genetik Algoritmalar ise Pareto Temelli Yaklaşımlar ve Pareto Temelli Olmayan Yaklaşımlar olmak üzere ikiye ayrılmaktadır. Pareto Temelli Olmayan Yaklaşımlarda, Pareto çözüm kümesinin önemli bir kısmının değerlendirilmeden bırakılma ihtimali olabilmekte ve dağılım etkin bir şekilde yapılamamaktadır. Pareto Temelli Yaklaşımlarda ise baskın olmayan çözümleri sıralama yöntemi ile öne sürülmektedir [89]. Pareto Temelli Yaklaşımın bu avantajı ve Pareto Temelli Olmayan Yaklaşımın yukarıda belirtilen dezavantajından dolayı çalışmada ilgili problemin çözümü için Pareto Temelli Çok Amaçlı Genetik Algoritma metodolojilerinden NSGA-II (Non-dominated Sorting Genetic Algorithm) yöntemi kullanılmıştır.

NSGA-II Algoritması 2002 yılında, Deb ve arkadaşları tarafından geliştirilmiştir. NSGA algoritmasında, popülasyon boyutu büyüdükçe meydana gelen hesaplama zorluğu ve elitizm stratejisinin algoritma içerisinde dikkate alınmaması problemlerini gidermek için geliştirilmiş olan NSGA-II algoritması, genetik algoritma temelli çalışmaktadır. Algoritma üç adımdan oluşmakta, bu adımlar aşağıda verilmektedir [90,91].

- Hızlı Baskın Olmayan Sıralama Yaklaşımı (Fast Non-dominated Sorting)
- Yığılma Uzaklığı Sıralaması (Crowding Distance)
- Ana Döngü

Hızlı baskın olmayan sıralama yaklaşımı

Hızlı Baskın Olmayan Sıralama Yaklaşımı, popülasyon ya da yığın içerisinde çözümlerin birbirlerine baskınlıklarını belirlenmek için kullanılır. İlgili yöntemde, yığındaki bireylerin uygunluk fonksiyonundaki her bir amaç kendi aralarında karşılaştırılarak baskın olan bireylerin belirlenmesi amaçlanmaktadır. Algoritmanın adımları Şekil 5.10'da gösterilmektedir. Hızlı Baskın Olmayan Sıralama Yaklaşımında, başlangıçta iki değer hesaplanmakta ve bu değerler aşağıda ifade edilmektedir [91].

- 1) Baskınlık sayısı (n_p): p çözümüne ya da bireyine baskın çözümlerin sayısı
- 2) S_p : her bir p bireyinin ya da çözümünün baskıladığı çözümlerin seti.

Baskınlık sayısı hesaplanırken, ilk baskılanmamış yüzeydeki tüm çözümlerin baskınlık sayısı, $n_p = 0$ olarak tanımlanır. Daha sonra $n_p = 0$ olan her bir p çözümü ya da bireyin için S_p kümesindeki her bir q bireyi ziyaret edilir ve n_p yani baskın çözümlerin sayısı bir azaltılır. Bu esnada eğer herhangi bir q bireyi ya da çözümü için baskınlık sayısı 0 olur ise q çözümü, Q listesine aktarılır. Q listesinin üyeleri ise ikinci baskılanmamış yüzeyi(cephesi) oluşturur. Yukarıda ifade edilen adımlar Q setinin tüm üyeleri için tekrarlanarak üçüncü baskılanmamış yüzey oluşturulur. Bu proses tüm bireyler ya da çözümlerin yüzeylere atanması ve tüm yüzeylerin ya da cephelerin belirlenmesine kadar devam eder [91].

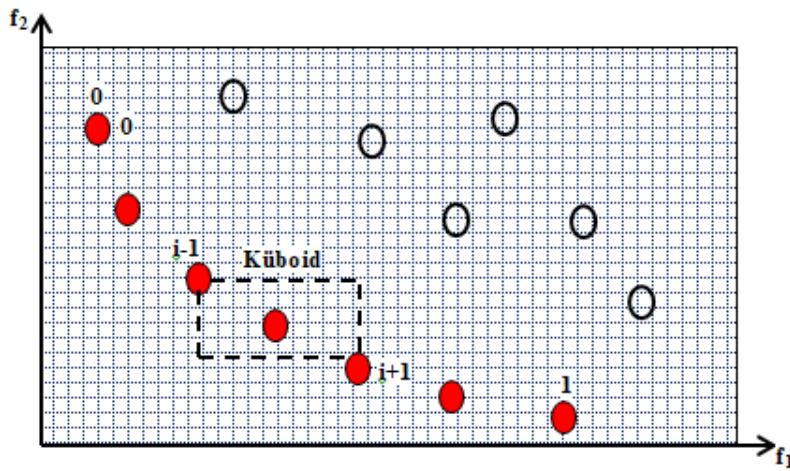
İkinci ve sonraki bastırılmamış cephelerde her bir p bireyi ya da çözümü için baskınlık sayısı (n_p) en fazla $N-1$ (N = popülasyon sayısı) değeri kadar olabilir. Herhangi bir p çözümü için baskınlık sayısı sıfır olduğunda ilgili birey baskılanmamış cepheye atanır ve o birey için süreç tamamlanmış olur.

N : yığın büyüklüğü, n_p : baskınlık sayısı,
 S_p : p bireyinin baskın olduğu bireylerin kümesi
 her $p \in P$ için $S_p = \{ \}$
 $n_p = 0$
 her $q \in P$ için
 eğer $(p < q)$ ise $S_p = S_p \cup \{q\}$
 eğer $(q < p)$ ise $n_p = n_p + 1$
 eğer $n_p = 0$ ise $p_{derece} = 1$ ve $F_1 = F_1 \cup \{p\}$
 $i = 1$
 $F_i \neq \{ \}$ iken
 $Q = \{ \}$
 her $p \in F_i$ için
 her $q \in S_p$ için
 $n_q = n_q - 1$
 eğer $n_q = 0$ ise
 $q_{derece} = i + 1$
 $Q = Q \cup \{q\}$
 $i = i + 1$ $F_i = Q$

Şekil 5.9. Hızlı baskın olmayan sıralama yaklaşımı [91]

Yığılma uzaklığı sıralaması

Yığılma uzaklığı sıralaması, popülasyondaki çözümler cephelere ayrıldıktan sonra, her bir cephe içerisinde kalan bastırılmamış çözümlerin ya da bireylerin, ilgili yığılma uzaklığı değerlerine göre kendi içlerinde sıralanmasını sağlar. Sıralama yapılırken, ilgili cephedeki çözüm ile komşu çözümleri arasındaki yakınlık belirlenir. Bu mekanizma sayesinde, oluşturulacak yeni yığın daha kaliteli çözümlerden oluşturulması amaçlanmaktadır [78, 79, 80].



Şekil 5.10. Yığılma uzaklığı hesaplaması. (kırmızı noktalar aynı cephedeki bastırılmamış çözümleri göstermektedir) [91]

Şekil 5.10'daki çözüm uzayında aynı cephede olan(kırmızı noktalar) ve olmayan(diğer noktalar) bireyler bulunmaktadır. Bu çözüm uzayından hareketle, yığındaki herhangi bir çözümün yığılma uzaklığını belirlemek için, problemdeki amaçların her biri için ilgili çözümün(i) etrafındaki iki çözümün ortalama uzaklık değeri hesaplanır. Hesaplanan uzaklık değerine göre, çözümün bulunduğu cephedeki her bir çözüm, ilgili amaç fonksiyonu değeri için artan derecede sıralanır. Amaç fonksiyonlarının her birinde uç noktalarda kalan bireylere sonsuz uzaklık değeri olarak minimum ve maksimum değerler verilir. Uç noktalar arasında kalan bireyler için uzaklık değeri, komşu iki çözümün amaç değerlerinin mutlak normalize edilmiş fark değerleridir. Bu hesaplama dikkate alınan değer amaçlar için de yapılmakta ve her bir amaç bazında elde edilen uzaklık değeri toplanarak toplam yığılma uzaklığı hesaplanmaktadır [85,90,91]. Yığılma uzaklığı hesaplama algoritmasına ilişkin bilgiler Şekil 5.11'de verilmektedir [91].

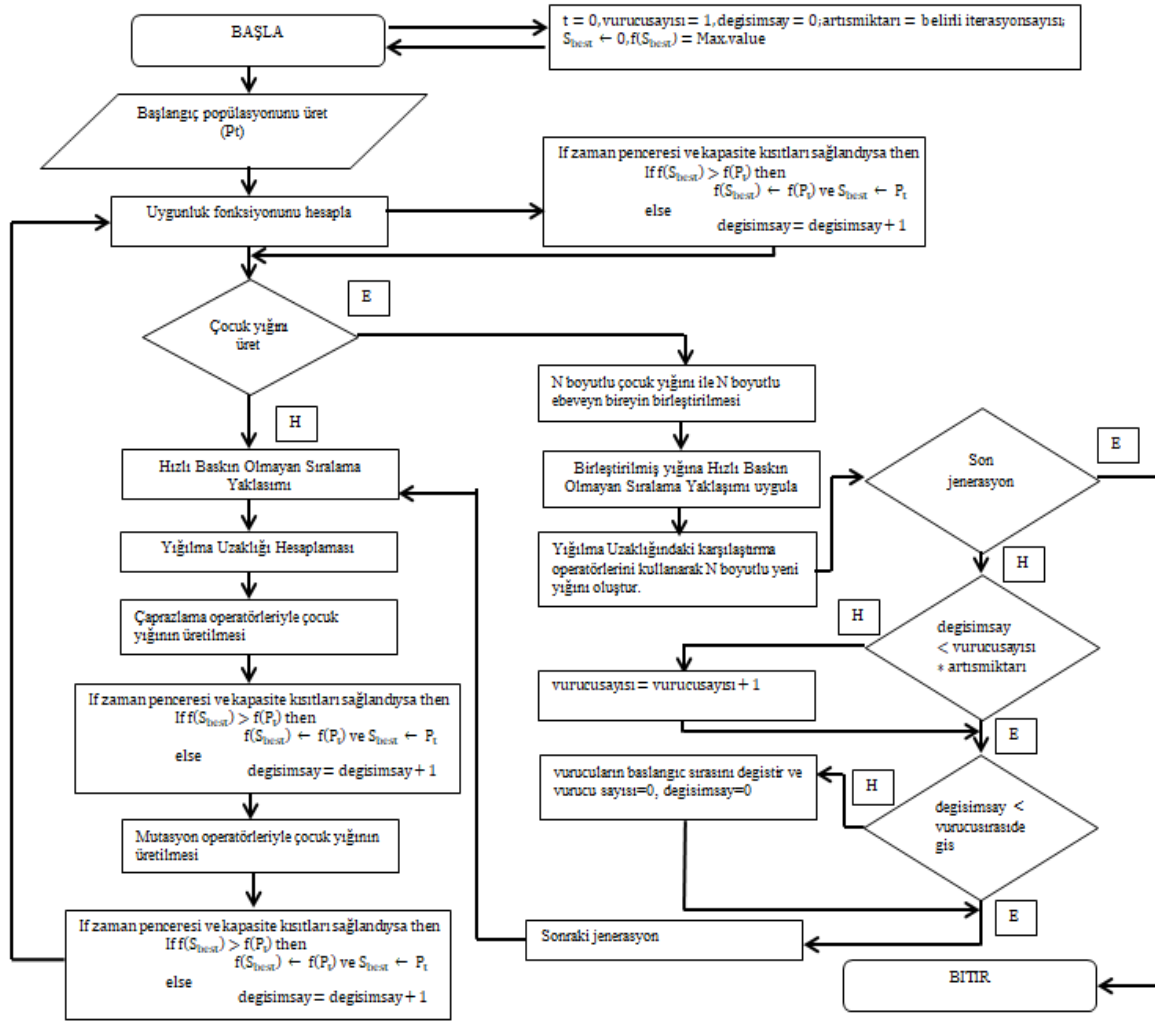
$l = I $	I içerisindeki çözümlerin sayısı
for each i , set $I[i]_{distance} = 0$	başlangıç uzaklığı
for each objective m	
$I = \text{sort}(I, m)$	her bir amaç değerinin kullanılarak sıralanması
$I[l]_{distance} = I[l]_{distance} = \infty$	sınır noktalar sonsuz olarak seçildi
for $i = 2$ to $(l - 1)$	tüm diğer noktalar için
$I[i]_{distance} = I[i]_{distance} + \frac{(I[i+1].m - I[i-1].m)}{(f_m^{mac} - f_m^{min})}$	

Şekil 5.11. Yığılma uzaklığı (crowding-distance) hesaplama algoritması [91]

Hesaplanan yığılma uzaklığı değerleri yoğunluk karşılaştırma operatörünün kullanılmasıyla herhangi iki çözüm arasından ideal olanın belirlenmesi için kullanılır. Eğer farklı baskın olmayan cephelerdeki iki çözüm arasından seçim yapılıyorsa iki çözüm içerisinde yoğunluk karşılaştırma operatörü değeri (a_n) düşük olan çözüm seçilir. Eğer bu iki çözüm aynı cephede yer alıyorsa bu durumda yığılma uzaklığı değeri büyük olan çözüm seçilir. Yoğunluk karşılaştırma operatörü değeri hesaplanırken popülasyondaki her “i” çözümü baskın olmama sırası (i_{rank}) ve Yığılma uzaklığı ($i_{distance}$) özelliklerine sahip olmalıdır. İlgili özellik: $i a_n j$ if ($i_{rank} < j_{rank}$) or (($i_{rank} = j_{rank}$) and ($i_{distance} > j_{distance}$))

Ana döngü

Başlangıçta P_0 ana popülasyonu rassal olarak oluşturulur. Popülasyon içerisindeki çözümler baskın olmama derecelerine göre sıralanır. Bu aşamada her bir birey ya da çözüm baskın olmama derecesine göre bir cepheye ya da yüzeye atanır. En iyi çözümler birinci cephede, sonraki çözümler ikinci ve devamı cephelerde yerleşecek şekilde atama işlemi gerçekleştirilir. Genetik algoritmanın adımlarında olduğu üzere turnuva seçimi, yeniden üretim operatörü, çaprazlama ve mutasyon işlemleri uygulanarak N büyüklüğünde Q_0 çocuk popülasyonuna ait bireyler üretilir. NSGA-II yönteminde elitizm seçim stratejisi sadece ana popülasyona yani ilk yığına uygulanmakta, sonraki yığınlarda bu işlem en iyi baskın olmayan çözümlerle karşılaştırılması metodolojisiyle yapıldığı için klasik genetik algoritma yönteminden ayrılmaktadır. Önerilen algoritmanın adımları Şekil 5.12’de gösterilmektedir.



Şekil 5.12. NSGA-II algoritmasına dayalı çözüm yaklaşımı

Önerilen algoritmada, başlangıçta vurucu sayısı 1 olarak belirlenir. Eğer çözümde belirli bir iterasyon boyunca iyileşme meydana gelmezse ve ya ilgili kısıtları sağlayan herhangi bir çözüm üretilemezse, vurucu sayısı bir artırılır. Algoritmada vurucu sayısı maksimum değere ulaşmışsa ve çözümde ilgili iterasyon sayısı sonucunda iyileşme meydana gelmemişse, bu durumda filodaki vurucuların sırası rassal bir şekilde değiştirilir ve vurucu sayısı 1 yapılarak algoritmanın adımları yeniden tekrarlanır. İlgili değişiklik sayesinde çözüm uzayının farklı noktalarının taranması ve yerel optimalin önüne geçilmesi amaçlanmaktadır.

5.1.5. Tavlama benzetimi

Tavlama Benzetimi algoritmasının tanımı ve özellikleri Bölüm 3.1.7 içerisinde tanımlanmıştır. Bir önceki bölüm içerisinde ifade edilen başlangıçta vurucu sayısının 1

kabul edilip, belirli iterasyon sonucunda çözümde iyileşme meydana gelmezse vurucu sayısının 1 artırıldığı ve maksimum vurucu sayısına ulaşıp belirli iterasyon boyunca çözümde iyileşme meydana gelmezse vurucu sırasının rassal bir şekilde değiştirilip vurucu sayısının 1 yapılması işlemi, önerilen tavlama benzetimi algoritması içerisinde de kullanılmıştır. Algoritmadaki parametrelerin belirlenmesinde Taguchi yönteminden yararlanılmış ve ilgili parametre konfigürasyonuna ilişkin bilgiler Bölüm 5.2.2 içerisinde tartışılmıştır.

5.2. Uygulama çalışması

5.2.1. Problem tasarımı

Tezde incelenen problem, literatürde ilk kez ele alınmakta ve bu nedenden dolayı geliştirilen algoritmaların etkinliğini test edilebilecek bir problem seti bulunamamaktadır. Bu bağlamda, geliştirilen algoritmaların performansının belirlenmesi için farklı hedef ve vurucu sayısına sahip 30 farklı problem seti üretilmiştir. Problem setleri, hedef ve vurucu sayılarına göre sınıflandırılmıştır. Bu sınıflandırma aşağıda ifade edilmiştir.

- 5 vurucu ve 10 hedeften oluşan Tip-1 Problemler
- 6 vurucu ve 15 hedeften oluşan Tip-2 Problemler
- 7 vurucu ve 20 hedeften oluşan Tip-3 Problemler
- 8 vurucu ve 25 hedeften oluşan Tip-4 Problemler
- 9 vurucu ve 30 hedeften oluşan Tip-5 Problemler
- 10 vurucu ve 35 hedeften oluşan Tip-6 Problemler

İlgili sınıflandırma sayesinde, hedef ve vurucu sayısındaki artışın, algoritma performansı üzerindeki etkisi belirlenmeye çalışılmıştır.

5.2.2. Parametre konfigürasyonu

Tavlama Benzetimi, Genetik Algoritma ve NSGA-II algoritması içerisinde birçok parametreye sahip olmakta, bu parametreler çözüm süresine ve kalitesine önemli ölçüde etki etmektedir. Buradaki önemli noktalardan biri ise parametreler için ideal değerlerin belirlenmesidir. Klasik parametre konfigürasyon yöntemleri, kompleks bir yapıya sahip olmakta ve uygulanabilirlik konusunda bazı zorluklar ortaya çıkmaktadır.

Konfigürasyonda parametre sayısı arttığı zaman ideal parametre değerlerini belirlemek için çok sayıda denemeye ihtiyaç duyulmaktadır. Tez çalışmasında, ideal parametre setinin belirlenmesinde Taguchi yönteminden yararlanılmakta ve literatürde yöntem ile ilgili birçok çalışma bulunabilmektedir. Taguchi yöntemi sayesinde, az sayıda deneme yaparak parametre uzayının detaylı bir şekilde taranması için özel bir şekilde tasarlanmış ortogonal dizilerden yararlanılmaktadır. Böylelikle, daha az deney ile ideal parametre setlerine ulaşılabilir.

Çizelge 5.10. Tavlama benzetimi, genetik algoritma ve NSGA-II algoritmaları için dikkate alınan parametreler

Tavlama Benzetimi		Genetik Algoritma ve NSGA-II	
D.K.	Durdurma kriteri	P.B.S.	Popülasyondaki birey sayısı
A	Soğutma oranı	M.O.	Mutasyon oranı
N	Her bir sıcaklıkta aranacak komşu sayısı (swap ve insert mekanizması ile elde edilen tüm çözümler)	Ç.O.	Çaprazlama oranı
T0	Başlangıç sıcaklığı	Ç.Y.S.	Çaprazlamada yer değişecek eleman sayısı
Tf	Son sıcaklık	T.K.S.	Turnuvada karşılaştırılacak eleman sayısı – $0,6 * P.B.S$
		T.İ.S	Durdurma kriteri olarak kullanılan toplam iterasyon sayısı
V.S.M.	Vurucu sayısının kaç iterasyon sonra artırılacağını belirleyen vurucu sayısı artış miktarı	V.S.M.	Vurucu sayısının kaç iterasyon sonra artırılacağını belirleyen vurucu sayısı artış miktarı

Tavlama Benzetimi, Genetik Algoritma ve NSGA-II yöntemleri için Taguchi yöntemi ile ideal parametre setinin belirlenmesinde dikkate alınan parametreler Çizelge 5.10’da gösterilmiştir.

Algoritmaların çözüm performanslarının aynı zaman dilimi içerisinde değerlendirilebilmesi için Durdurma Kriteri (D.K.) olarak Genetik Algoritmanın çözüm zamanları dikkate alınmış ve Tavlama Benzetimi Algoritması için ilgili konfigürasyona bu parametre değeri dahil edilmemiştir. Buna ek olarak Tavlama Benzetimi Algoritması parametresi olan “N” değeri deterministik olarak belirlenmektedir. Geriye kalan dört parametre ve NSGA-II ile Genetik Algoritma için parametre konfigürasyonunda dikkate alınan değerler Çizelge 5.11’de verilmektedir.

Çizelge 5.11. HM-ARP için konfigürasyonda kullanılan parametreler ve seviyeleri

Problem Tipi	Level	Genetik Algoritma ve NSGA-II						Tavlama Benzetimi Algoritması				
		A	B	C	D	E	F	A	B	C	D	E
		P.B.S	Ç.O	M.O.	Ç.Y.S	T.İ.S	V.S.M	T0	Tf	D.K.	α	V.S.M.
Tip 1	1	10	0,80	0,01	3	4000	400	100	0,5	25	0,85	400
	2	20	0,90	0,05	6	6000	600	200	0,75	25	0,9	600
	3	30	0,99	0,1	9	8000	800	300	1	25	0,95	800
Tip 2	1	10	0,80	0,01	6	4000	400	100	0,5	1,5	0,85	400
	2	20	0,90	0,05	9	6000	600	200	0,75	1,5	0,9	600
	3	30	0,99	0,1	12	8000	800	300	1	1,5	0,95	800
Tip 3	1	40	0,80	0,01	9	6000	600	200	0,5	2,5	0,85	600
	2	50	0,90	0,05	12	8000	800	300	0,75	2,5	0,9	800
	3	60	0,99	0,1	15	10000	1000	400	1	2,5	0,95	1000
Tip 4	1	40	0,80	0,01	12	6000	600	200	0,5	4	0,85	600
	2	50	0,90	0,05	15	8000	800	300	0,75	4	0,9	800
	3	60	0,99	0,1	18	10000	1000	400	1	4	0,95	1000
Tip 5	1	70	0,80	0,01	15	8000	800	300	0,5	12	0,85	800
	2	80	0,90	0,05	18	10000	1000	400	0,75	12	0,9	1000
	3	90	0,99	0,1	21	12000	1200	500	1	12	0,95	1200
Tip 6	1	70	0,80	0,01	18	8000	800	300	0,5	19	0,85	800
	2	80	0,90	0,05	21	10000	1000	400	0,75	19	0,90	1000
	3	90	0,99	0,1	24	12000	1200	500	1	19	0,95	1200

Taguchi yönteminde parametre sayısına ve onların seviyelerine göre yapılacak deney sayısı değişebilmektedir. Tez çalışmasında, Taguchi yöntemi uygulanmasına Minitab 17.0 programı kullanılmıştır. Genetik Algoritma ve NSGA-II Algoritmasında 6 parametre bulunduğu ve her bir parametre 3 seviyeye sahip olduğu için, ideal deney setinin belirlenmesinde 27 adet deneyin yapılması gerekmekte ve bu deneyler için L27 ortogonal diziliminin kullanılması önerilmektedir. Buna ek olarak, Tavlama Benzetimi Algoritmasında 4 tane parametrenin dikkate alınması ve her bir parametrenin 3 seviyeli olması nedeniyle her bir örneklem setinde 9 adet deneyi yapılması ve bu deneylerde L9 ortogonal diziliminin kullanılması önerilmektedir. Bu bölümde ve bir sonraki uygulamalar Intel(R) Core(TM) i5-6200@ 2.3 GHz işlemcili ve 4 GB RAM'li bilgisayarlarda C# programı üzerinde gerçekleştirilmiştir.

Taguchi yönteminde maksimizasyon ve minimizasyon problemleri için önemli kriterlerden birini Signal/Noise(S/N) değeri oluşturmakta, bu değeri maksimize eden parametrelere ait değer uygun parametre kümesini oluşturmaktadır. Bu bilgiler doğrultusunda, Tavlama Benzetimi Algoritması için parametre değerleri Çizelge 5.12, Genetik Algoritma için parametre değerleri Çizelge 5.13'de gösterilmektedir.

Çizelge 5.12. Tavlama benzetimi algoritması için ideal parametre değerleri

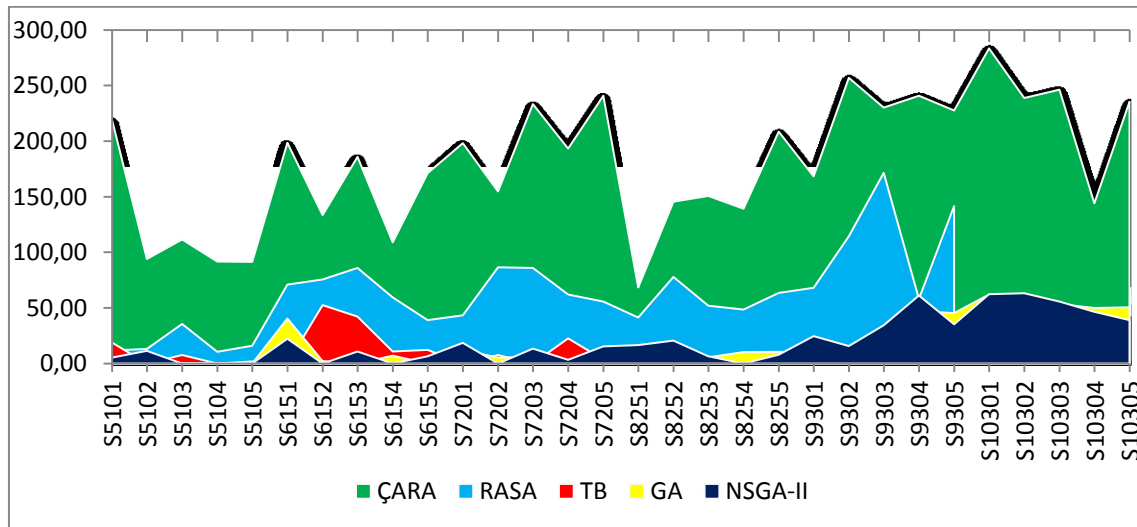
Problem Tipi	T0	Tf	A	V.SM.
Tip 1	300	0,75	0,9	600
Tip 2	200	1	0,9	600
Tip 3	400	1	0,9	800
Tip 4	200	0,75	0,9	800
Tip 5	500	0,5	0,95	800
Tip 6	400	0,75	0,95	800

Çizelge 5.13. NSGA-II ve genetik algoritma için ideal parametre değerleri

Problem Tipi	A	B	C	D	E	F
Tip 1	20	0,9	0,1	9	6000	400
Tip 2	30	0,9	0,05	6	8000	800
Tip 3	40	0,8	0,05	9	10000	600
Tip 4	50	0,9	0,05	15	8000	1000
Tip 5	90	0,9	0,1	15	10000	800
Tip 6	90	0,9	0,1	18	12000	800

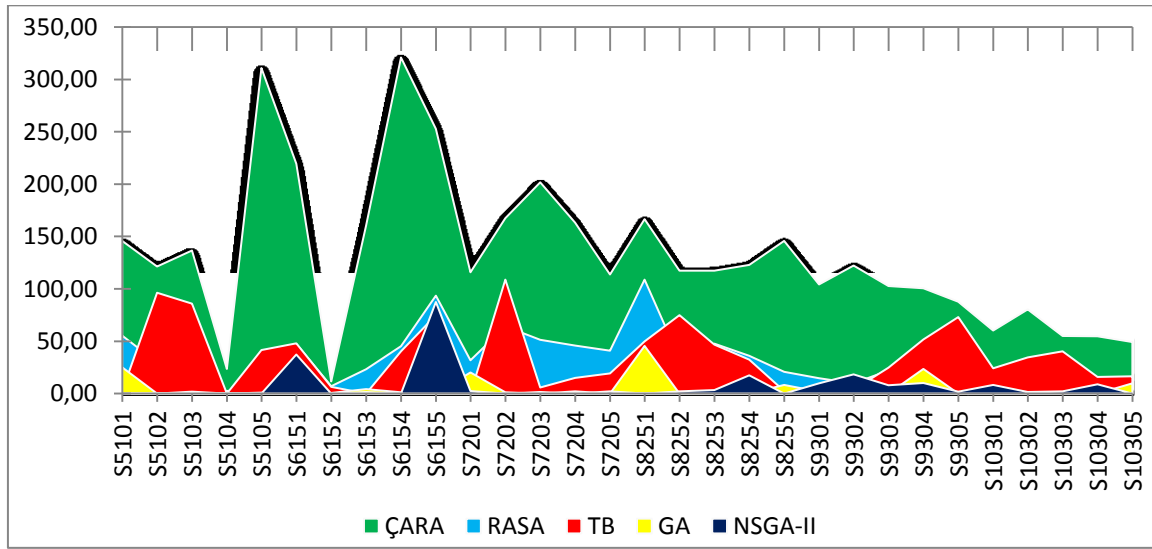
5.2.3. Uygulama çalışması

Uygulama çalışmasında, HM-ARP için geliştirilen algoritmalar, her bir problem üzerinde 10 kez çalıştırılmış ve analiz sonuçlarına ilişkin bilgiler Şekil 5.14 ve 5.15’de gösterilmiştir.



Şekil 5.13. Algoritmaların, amaç-1 için en iyi değerden sapma miktarları(%)

Şekil 5.14 ve 5.15'deki sonuçlardan, Amaç-1'de Tavlama Benzetimi Algoritmasının, Amaç 2'de ise Genetik Algoritma ve NSGA-II algoritmalarının daha iyi olduğu anlaşılmaktadır. Algoritmaların, iki amacı eş zamanlı minimize etmede gösterdikleri performansın belirlenmesi için ise istatistiksel yöntemlere ihtiyaç duyulmaktadır. Bu bağlamda, şekillerdeki sonuçların analiz edilmesi ve algoritmaların performansları karşılaştırılması için Friedman Anova Testinden yararlanılmaktadır.



Şekil 5.14. Algoritmaların, amaç-2 için en iyi değerden sapma miktarları(%)

Dikkate alınan verilerin normal dağılıma uygunluk göstermemesi, karşılaştırılacak algoritma sayısının üç veya daha fazla olması ve tüm algoritmalarda aynı verilerin kullanılması nedeniyle Friedman Anova testine başvurulmuştur. İlgili test için kurulan hipotez aşağıda verilmiştir.

H0:Çözüm kurucu sezgisellerin ortalamaları arasında anlamlı bir farklılık yoktur.

H1: Çözüm kurucu sezgisellerin ortalamaları arasında anlamlı bir farklılık vardır.

Descriptive Statistics						Ranks		Test Statistics(a)	
	N	Mean	Std. Deviation	Min.	Max.		Mean Rank	N	25
ÇARA	30	57,0820	27,84227	17,90	112,26	ÇARA	5,00	Chi-Square	78,8
RASA	30	16692,0347	37893,36587	10,32	100000,00	RASA	3,92	Df	02
TB	30	20,4883	7,28878	9,34	32,95	TB	1,92	Asymp. Sig.	,000
GA	30	24,2210	12,58813	8,80	48,48	GA	1,88	a. Friedman Test	
NSGA_II	30	24,6383	12,84155	8,61	53,07	NSGA_II	2,28		

Resim 5.3. HM-ARP'de amaç-1 için friedman anova testi

SPSS programında yapılan Friedman Anova Testi sonucunda elde edilen sonuçlar Resim 5.3’de gösterilmiştir. Bu şekle göre, algoritmaların performansları arasında anlamlı bir farklılık olduğu söylenmektedir. Şekle göre TB, GA ve NSGA-II algoritmaların ortalamaları birbirine yakın olmaktadır. TB algoritması, diğer algoritmalarla göre en düşük ortalamalara sahip olmakta, sıralamaların ortalaması incelendiğinde GA’ nın daha iyi olduğunu anlaşılmaktadır. Bu farklılığın kaynağının belirlenmesi ve algoritmaların ikili olarak karşılaştırılabilmesi için Wilcoxon Testinden yararlanılmaktadır. Test çalışması için kurulan hipotez aşağıda ifade edilmekte ve SPSS programı kullanılarak yapılan analiz sonuçları Çizelge 5.14’de gösterilmektedir.

H0:Sezgisel-1 ve Sezgisel-2 ortalamaları arasında anlamlı bir farklılık yoktur.

H1:Sezgisel-1 ve Sezgisel-2 ortalamaları arasında anlamlı bir farklılık vardır.

Çizelge 5.14. HM-ARP’de amaç-1 için wilcoxon signed ranked test analiz sonuçları

Sezgisel-1	Sezgisel-2	Sezgisel1>Sezgisel2	Sezgisel2>Sezgisel1	Sezgisel1=Sezgisel2	Asymp. Sig.
ÇARA	RASA	25	5	0	0,00
ÇARA	TB	30	0	0	0,00
ÇARA	GA	30	0	0	0,00
ÇARA	NSGA-II	30	0	0	0,00
RASA	TB	29	1	0	0,00
RASA	GA	30	0	0	0,00
RASA	NSGA-II	29	1	0	0,00
TB	GA	11	18	1	0,013
TB	NSGA-II	10	19	1	0,04
GA	NSGA-II	9	18	3	0,171

Çizelge 5.14’deki sonuçlar, Tavlama Benzetimi Algoritmasının, önerilen diğer algoritmalarla daha üstün olduğunu ve diğer algoritmalarla arasındaki farkın anlamlı olduğunu göstermektedir. Ayrıca çizelgedeki sonuçlardan GA’nın, NSGA-II’den daha iyi bir performans sergilediği anlaşılmaktadır.

Maliyet açısından sonuçlar istatistiksel açıdan analiz edilmiş ve elde edilen değerler aşağıda belirtilmiştir. Yukarıda belirtildiği gibi, grupların ortalamaları arasında anlamlı bir farklılığın olup olmadığının belirlenmesinde Friedman Anova Testinden yararlanılmıştır.

Descriptive Statistics						Ranks		Test Statistics(a)	
	N	Mean	Std. Deviation	Min.	Max.		Mean Rank	N	
ÇARA	30	4167,61 47	2113,80516	755,97	8389,78	ÇARA	4,83	30	
RASA	30	1666683 88,8591	379048238,43 639	403,75	1000000 000,00	RASA	3,77	Chi-Square	86,837
TB	30	2655,06 77	1780,55076	540,16	6254,25	TB	2,93	Df	4
GA	30	2118,27 60	1395,69025	379,01	5093,45	GA	1,65	Asymp. Sig.	,000
NSGA_II	30	2162,72 77	1410,37698	384,40	4826,78	NSGA_II	1,82	a. Friedman Test	

Resim 5.4. HM-ARP’de amaç-2 için friedman anova testi

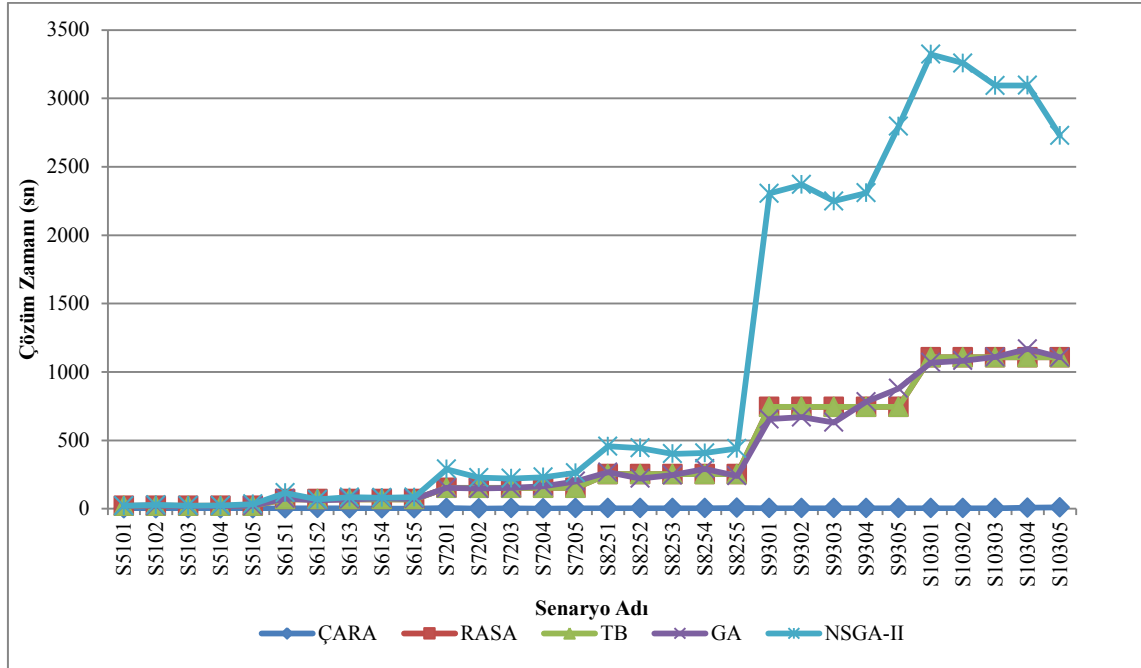
Resim 5.4’deki sonuçlara göre, diğer algoritmalarla kıyasla GA’ nın daha iyi bir performans sergilediği ve NSGA-II ile GA nın ilgili problem setleri üzerinde benzer birbirine yakın sonuçlar ürettiği belirlenmiştir. Algoritma performanslarının ikili karşılaştırılmasının yapılmasında ve aralarında anlamlı bir farklılık olup olmadığının belirlenmesinde Wilcoxon testinden yararlanılmış ve test sonuçları Çizelge 5.15’de gösterilmiştir.

Çizelge 5.15. HM-ARP’de amaç-2 için wilcoxon signed ranked test analiz sonuçları

Sezgisel-1	Sezgisel-2	Sezgisel1>Sezgisel2	Sezgisel2>Sezgisel1	Sezgisel1=Sezgisel2	Asymp. Sig.
ÇARA	RASA	25	5	0	0,057
ÇARA	TB	30	0	0	0,00
ÇARA	GA	30	0	0	0,00
ÇARA	NSGA-II	30	0	0	0,00
RASA	TB	21	9	0	0,045
RASA	GA	29	1	0	0,00
RASA	NSGA-II	28	2	0	0,00
TB	GA	24	5	1	0,00
TB	NSGA-II	24	5	1	0,00
GA	NSGA-II	12	16	2	0,219

Çizelge 5.15’deki sonuçlar, GA’nın, diğer algoritmalarla kıyasla daha üstün bir performans sergilediğini göstermiştir. Bu algoritmayı sırasıyla NSGA-II ve TB algoritmalarının takip ettiği Çizelge 5.15’deki sonuçlardan anlaşılmıştır.

Yukarıdaki bilgilere ek olarak algoritmaların çözüm zamanlarına ilişkin performansları Şekil 5.15’de gösterilmektedir.



Şekil 5.15. HM-ARP’de algoritmaların çözüm zamanlarına ilişkin bilgiler

Şekil 5.15’deki sonuçlardan, ÇARA çözüm kurucu sezgisel algoritmasının problemi saniyeler içerisinde çözdüğü, RASA, GA ve TB algoritmalarının ise çözüm zamanı açısından birbirine yakın performanslar sergilediği anlaşılmaktadır. Buna ek olarak NSGA-II algoritmasının vurucu ve hedef sayısı arttıkça, çözüm zamanının diğer algoritmalarla kıyasla önemli ölçüde arttığı tespit edilmektedir.

Yukarıdaki çizelgeler, şekiller ve istatistiksel analiz sonuçlarından hareketle her iki amacın eş zamanlı minimize edilmesi ve çözüm süreleri dikkate alındığında, en etkili algoritmanın Genetik Algoritma olduğu, Tavlama Benzetimi ve NSGA-II algoritmalarının bu algoritmayı takip ettiği belirlenmiştir. HM-ARP, askeri operasyonlar ve deniz gözetleme operasyonları gibi kısa sürede etkin çözümler üretilmesi gereken problemlerde uygulanmakta, NSGA-II algoritmasının çözüm süresinin yüksek olmasının bu bağlamda bir dezavantaj yaratacağı öngörülmektedir. Tavlama Benzetimi algoritmasının tek bir amaç üzerinde daha etkili olmakta ve iki amacın eş zamanlı minimizasyonunda daha düşük bir performans sergilemektedir. Sonraki çalışmalarda ilgili algoritmanın eksik yönlerinin güçlendirilmesiyle daha iyi sonuçlar üretilebileceği tahmin edilmektedir. ÇARA, çözüm kurucu sezgisel algoritmasının tüm senaryolarda saniyeler içerisinde uygun çözümler üretmesi nedeniyle hem gerçek hayat uygulamalarında hem de geliştirilen metasezgisel algoritmaların başlangıç çözümünde başarılı bir şekilde kullanılabileceği düşünülmektedir.

RASA algoritmasının, küçük ve orta boyutlu problemlerde başarılı çözümler ürettiği, 10 vuruculu-35 hedefli problem setinde belirlenen çözüm süresi içerisinde uygun çözümler üretilmediği tespit edilmiştir. Buna ek olarak geliştirilen algoritmalar için tüm senaryolarda, ortalama ile en iyi değer aralarındaki sapma miktarının %10'dan daha düşük olduğu belirlenmiştir. Bu durum, algoritmaların sağlam olduğunu göstermekte ve farklı problem tipleri üzerinde de başarılı bir şekilde uygulanabileceğini göstermektedir.

6. SONUÇ VE ÖNERİLER

Savunma sanayisinde yaşanan teknolojik gelişmelerle birlikte ülkeler, yapay zekaya dayalı savunma araçlarını askeri operasyonlarda yoğun bir şekilde kullanmakta, bu sayede zorlu coğrafik koşullarda askeri personel kaybının önüne geçilmesini ve düşman unsurlarına maksimum zararın verilmesini amaçlamaktadır. Bu operasyonlarda yoğun bir şekilde kullanılan araçların başında İHA ve SİHA'lar gelmekte, bu araçlar için ideal tur rotalarının kısa çözüm süreleri içerisinde belirlenmesi, ulusal güvenliğin sağlanması açısından büyük bir önem arz etmektedir. HM-ARP, İHA ve SİHA'ların filo halinde gerçekleştirilen operasyonlarda, hareketli nesnelerin gözetlenmesi, takibi ve etkisiz hale getirilmesi için minimum görev zamanlı ve görev maliyetli tur rotalarının hızlı ve etkili bir şekilde belirlenmesi gereksiniminden ortaya çıkan dinamik yapıli bir problem türüdür. Problemde, filodaki vurucuların heterojen özelliklere (farklı hız, havada kalma kapasitesi ve görev maliyeti) sahip oldukları varsayılmakta ve ideal sayıda vurucu kullanılarak operasyon alanındaki hareketli hedefler, ilgili zaman aralıkları içerisinde etkisiz hale getirilmeye çalışılmaktadır. Problem, birçok açıdan literatürdeki diğer çalışmalardan ayrılmaktadır. Literatürde HM-ARP ve HHGSP için vurucu kapasite kısıtları ve hareketli hedefler için zaman penceresi kısıtları altında, çok amaçlı optimizasyonun gerçekleştirildiği ve zaman biriminin sürekli olarak dikkate alındığı herhangi bir çalışma bulunmamaktadır. “n” adet vurucu ve “r” adet hedeften oluşan bir problem için $(n^r) * r!$ çözüm bulunmakta, hedef ve vurucu sayısı arttıkça, çözüm sayısı ve çözüm zamanı önemli ölçüde artmaktadır. Buna ek olarak sistem içerisindeki hedeflerin ya da düğümlerin sürekli olarak hareket etmesi ve herhangi bir “t” anında düğüm sayısının artabilmesi ya da azalabilmesi, problemin çözümünü oldukça karmaşıktır. Bu dinamik durum, problemin matematiksel modelleme yaklaşımlarıyla çözülmesini zorlaştırmakta, literatürdeki çalışmalarda problemin çözümü için probleme özgü sezgisel algoritmalar geliştirilmekte ya da metasezgisel algoritmalar kullanılmaktadır. Buna ek olarak, matematiksel modelin kullanıldığı çalışmalarda zaman birimi kesikli olarak dikkate alınmakta ve hedef sayısı arttıkça ilgili yöntemin yetersiz kaldığı ifade edilmektedir. Bu tez çalışmasında, yukarıda belirtilen nedenlerden dolayı problem üç aşamada çözülmekte, ilgili aşamalar aşağıda ifade edilmektedir.

Birinci aşamada, bir vurucu ve “n” tane hareketli hedeften oluşan klasik HHGSP ele alınmakta ve problemin çözümü için tavlama benzetimi algoritmasına dayalı bir çözüm yaklaşımı geliştirilmiştir. Algoritmanın başlangıç çözümü için 3 tanesi literatürde ilk kez kullanılan 6 farklı sezgisel algoritma geliştirilmiş ve önerilen algoritmalar çeşitli senaryolar üzerinde test edilerek ideal başlangıç çözümü belirlenmeye çalışılmıştır. Test çalışması sonucunda MDH ve MTH sezgisellerinin en iyi iki sezgisel olduğu belirlenmiş ve algoritmada başlangıç çözümü olarak kullanılmıştır. Buna ek olarak algoritmanın çözüm kalitesini artırmak ve çözüm süresini kısaltmak için ikisinin literatürde ilk kez kullanıldığı düşünülen üç farklı çözüm strateji geliştirilmiş ve tavlama benzetimi algoritmasına entegre edilmiştir. Yapılan test çalışmaları sonucunda algoritmanın tüm test problemlerinde kabul edilebilir çözüm süreleri içerisinde uygun çözümler ürettiği belirlenmiş ve algoritmaların sağlam olduğu tespit edilmiştir. Buna ek olarak, iyi bir başlangıç çözümünün algoritmanın çözüm performansına olumlu yönde etki ettiği ve SA-MDH algoritmasının, SA-MTH algoritmasından daha başarılı olduğu sonuçlara ulaştığı analiz sonuçlarından anlaşılmıştır.

Tez çalışmasının ikinci aşamasında, klasik HHGSP problemi zaman penceresi kısıtlaması altında incelenmiş ve problemin çözümünde tavlama benzetimi algoritmasından faydalanılmıştır. Önerilen algoritmanın, Zaman Pencere-HHGSP’de bilindiği kadarıyla ilk kez kullanıldığı düşünülmektedir. Geliştirilen algoritma, hedef sayısının 20 ile 200 arasında değiştiği 50 farklı senaryo üzerinde test edilmiş ve algoritmanın başlangıç çözümünde MTH sezgiselinden yararlanılmıştır. Analiz sonucunda, tüm senaryolarda kabul edilebilir çözüm süreleri içerisinde tüm hedeflerin imha edildiği tespit edilmiştir. Önerilen metodun, deniz gözetleme operasyonlarının daha hızlı ve etkili bir şekilde gerçekleşmesine katkı yapacağı düşünülmektedir.

Üçüncü ve son aşamada ise tezin odak noktasını oluşturan HM-ARP ele alınmakta ve problemin çözümünde sezgisel ve metasezgisel algoritmalarından yararlanılmaktadır. Problemin çok amaçlı olması nedeniyle, ilk olarak amaçların birbirine etkisi tespit edilmiş ve görev zamanı minimize edilmeye çalışıldığında görev maliyetinin arttığı, görev maliyetinin minimize edilmeye çalışıldığında ise görev zamanının arttığı belirlenmiştir. Bu nedenden dolayı problemin çözümünde Genetik Algoritma, Tavlama Benzetimi Algoritması, ÇARA ve RASA Algoritmalarına ek olarak pareto temelli bir çözüm yaklaşımı olan NSGA-II Algoritması da kullanılmıştır. HM-ARP ve HHGSP literatüründe,

bilindiği kadarıyla pareto temelli çözüm yaklaşımlarının kullanıldığı başka bir çalışma bulunmamaktadır. Ayrıca ÇARA algoritması problem için geliştirilmiş bir çözüm kurucu sezgisel olup, bilindiği kadarıyla literatürde ilk kez kullanılmaktadır. Önerilen algoritmalar vurucu sayısının 5 ile 10 ve hedef sayısının 10 ile 35 arasında değiştiği 30 farklı problem üzerinde test edilmiş, analiz sonucunda hem çözüm süresi açısından hem de iki amacın eş zamanlı minimizasyonu açısından Genetik Algoritmanın daha başarılı olduğu tespit edilmiştir. ÇARA, GA, NSGA-II ve TB' nin tüm senaryolarda kabul edilebilir sürelerde hedefleri etkisiz hale getirdiği, RASA Algoritmasının vurucu sayısının 10 hedef sayısının 35 olduğu problemlerde belirtilen çözüm zamanı içerisinde uygun çözümler üretilmediği analiz edilmiştir. Test çalışmaları sonucunda hedef ve vurucu sayısı arttıkça çözüm süresinin önemli ölçüde arttığı belirlenmiş, popülasyon tabanlı çözümlerin, diğer çözüm yaklaşımlarından daha başarılı olduğu tespit edilmiştir. Buna ek olarak, önerilen algoritmaların dinamik yapı ve büyük sayıda çözüm içeren problem tiplerini kabul edilebilir sürelerde çözmesi ve jenerik bir yapıya sahip olması nedeniyle HM-ARP ve HHGSP'ye ek olarak, İHA Rotalama Problemi, Dinamik ARP, Dinamik GSP ve Robotik Sistemlerin Rotalanması problemlerinde de kullanılabileceği düşünülmektedir.

Tez çalışmasında ele alınan problem, literatürde az çalışılan bir problem olup, teknolojiye yaşanan gelişmelerle birlikte probleme olan ilginin önemli ölçüde artacağı düşünülmektedir. Özellikle İHA ve SİHA sistemlerinin mühimmatlarla donatılması ve bu araçların akıllı bir şekilde yönlendirilmesi nedeniyle ileriki yıllarda suikast ve toplu saldırı gibi olaylarda bu araçların kullanılabileceği öngörülmekte, bu saldırıların engellenmesinin ise aynı tipteki dost araçların akıllı bir şekilde rotdlanması ile mümkün olacağı düşünülmektedir. Gelecek çalışmalarda araştırmacıların bu konuya yönelebileceği ve geliştirilecek yöntemlerle daha başarılı çözümler üreteceği öngörülmektedir.

Arama-kurtarma faaliyetlerinin gelecek yıllarda problemin en önemli uygulama alanlarından bir olacağı, herhangi bir afet anında hareketli nesnelerin kurtarılması ve taşınmasında belirtilen çözüm yaklaşımlarından yararlanılabileceği düşünülmektedir. Buna ek olarak, güvenlik kameralarının rotdlanması, uyduların konumlandırılması ve uydular için yakıt ikmal rotalarının belirlenmesi, deniz gözetleme operasyonları, dronelar ile kara trafiğindeki taşıtların yönlendirilmesi vb. alanlardaki problemlerin ilgili problem kapsamında düşünülebileceği ve bu problemler için alternatif çözüm yöntemlerinin geliştirilebileceği tahmin edilmektedir.

Üretim sektöründe robotik sistemlerin yoğun bir şekilde kullanılmasıyla birlikte, üretim hattından hızla akan ürünlerin, robotik kollar tarafından hızlı ve etkili bir şekilde kavranıp ilgili noktalara akıllı bir şekilde yerleştirilmesi veya robotik kol için birim zamanda üretim bandından maksimum sayıda ürünü toplayacak rotanın belirlenmesinin, HHGSP ve HM-ARP kapsamında incelenebileceği düşünülmektedir.

Günümüzde dronelar lojistik faaliyetlerinde yoğun bir şekilde kullanılmakta; kargo, ilaç, gıda vb. ürünlerin bir noktadan başka bir noktaya nakledilmesinde bu araçlardan yararlanılmaktadır. Drone teknolojisinde yaşanan gelişmeler ile birlikte dronelerin ürün taşıma kapasitelerinin artacağı, oluşturulacak ara depo ve bayilerle, kara taşıt trafiğine takılmadan ürünlerin hızlı ve etkili bir şekilde ilgili hareketli ya da hareketsiz müşterilere ulaşılması için ideal tur rotalarının belirlenmesinin, HM-ARP kapsamında incelenebileceği öngörülmektedir. Örneğin herhangi bir afet durumunda, farklı noktalardan hastanelere hareket eden ambulanslarda bulunan yaralıları, dronelar yardımıyla kan merkezinden hızlı bir şekilde kan dağıtılması için uygun rotaların belirlenmesinde önerilen çözüm yaklaşımlarından yararlanılabileceği tahmin edilmekte ve bu sayede ambulansların hastaneye varmasına gerek kalmadan yaralıları ilgili kanın ulaştırılabileceği düşünülmektedir. Buna ek olarak, operasyon alanında hareket halinde bulunan askerlere ya da askeri araçlara, karargâh noktasından dronelar yardımıyla mühimmat ya da ilaç gönderilmesinin problem kapsamında incelenebileceği tahmin edilmektedir.

Yukarıdaki bilgilere ek olarak, HM-ARP içerisinde aşağıdaki durumların dikkate alınabileceği ve bunlara yönelik çözüm stratejilerinin geliştirilebileceği öngörülmektedir.

- Hedef sayısı ve yörüngelerin belli olmaması gibi stokastik durumların dikkate alınması
- Vurucu ve hedeflerin üç boyutlu düzlemlerde hareket etmesi
- Vuruculara ait hızların, hedef hız ve konumlarına göre zaman içerisinde değişebilmesi
- Hedeflere ilişkin mühimmat gereksiniminin ya da müşterilere ait taleplerin birden fazla olabilmesi
- Vurucuların ve hedeflerin doğrusal olmayan yörüngelerde hareket edebilmesi
- Vuruculara yakıt ya da mühimmat ikmali yapıp tekrardan göreve çıkması gereksinimi ve bu gereksinim için ilgili hazırlık sürelerinin probleme eklenmesi

KAYNAKLAR

1. Helvig, C. S., Robins, G. and Zelikovsky, A. (1998). *Moving-Target TSP and Related Problems*. In European Symposium on Algorithms on 453-464, Italy.
2. İnternet: The multiple traveling salesmen with moving targets, RL:https://www.researchgate.net/publication/264121337_The_Multiple_Traveling_Salesmen_Problem_with_Moving_Targets , Son Erişim Tarihi: 17.09.2019.
3. Stieber, A., Fügenschuh, A., Epp, M., Knapp, M. and Rothe, H. (2015). The multiple traveling salesmen problem with moving targets. *Optimization Letters*, 9(8), 1569-1583.
4. Stieber, A. and Fügenschuh, A. (2016). Variants in modeling time aspects for the multiple traveling salesmen problem with moving targets. *Angewandte Mathematik und Optimierung Schriftenreihe (AMOS)*, 1-21.
5. Jiang, Q., Sarker, R. and Abbass, H. (2005). Tracking moving targets and the non-stationary traveling salesman problem. *Complexity International*, 11(2005), 171-179.
6. Jindal, P. and Kumar, A. (2011). Multiple target intercepting traveling salesman problem. *International Journal of Computer Science and Technology*, 2(2), 327-331.
7. Englot, B., Sahai, T. and Cohen, I. (2013). *Efficient Tracking and Pursuit of Moving Targets by Heuristic Solution of the Traveling Salesman Problem*. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on 3433-3438, Italy.
8. Khosravi, M. and Aghdam, A. G. (2014). *Cooperative Receding Horizon Control for Multi-Target Interception in Uncertain Environments*. In Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on 4497-4502, United States of America.
9. Hammar, M. and Nilsson, B. J. (1999). *Approximation Results for Kinetic Variants of TSP*. In International Colloquium on Automata, Languages, and Programming on 392-401, Czech Republic.
10. Zhou, A., Kang, L. and Yan, Z. (2003). *Solving Dynamic TSP with Evolutionary Approach in Real Time*. The 2003 Congress on Evolutionary Computation on 951-957, Australia.
11. Choubey, N. S. (2013). Moving target travelling salesman problem using genetic algorithm. *International Journal of Computer Applications*, 70(2), 30-34.
12. Lee, Z. J., Lee, C. Y. and Su, S. F. (2002). An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem. *Applied Soft Computing*, 2(1), 39-47.
13. Agharkar, P. and Bullo, F. (2014). *Vehicle Routing Algorithms To Intercept Escaping Targets*. In American Control Conference (ACC), 2014 on 952-957, United States of America.

14. Bengt, J. (2002). Approximation results for kinetic variants of tsp. *Discrete & Computational Geometry*, 27(4), 635-651.
15. Knapp, M. and Rothe, H. (2012). *Concept for Simulating Engagement Strategies for C-RAM Systems Using Laser Weapons*. Conference on Defense and Military Modeling & Simulation on 49-56, United States of America.
16. Bourjolly, J. M., Gurtuna, O. and Lyngvi, A. (2006). On- orbit servicing: a time-dependent, moving- target traveling salesman problem. *International Transactions in Operational Research*, 13(5), 461-481.
17. Blough, O. P., Farrington, T. K. and Hudson, J.(2016). *Trojan Asteroid Mission Design: Target Selection and Sequencing Optimization*. Conference: 26th AAS/AIAA Space Flight Mechanics Meeting, At Napa, California, United States of America.
18. Mei, G., Ran, X., Fang, D. and Zhang, C. (2015). *Improved Satellite Scheduling Algorithm for Moving Target*. In Proceedings of the Fourth International Conference on Information Science and Cloud Computing (ISCC2015), Guangzhou, China.
19. Groba, C., Sartal, A. and Vázquez, X. H. (2015). Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices. *Computers & Operations Research*, 56, 22-32.
20. Mercer, G., Barry, S. I., Marlow, D. O. and Kilby, P. (2008). Investigating the effect of detection and classification range and aircraft dynamics on a. *ANZIAM Journal*, 49, 475-492.
21. Kilby, P., Tobin, P., Luscombe, R., Barry, S. I. and Hickson, R. (2007). *The Maritime Surveillance Problem*. Proceedings of the Mathematics in Industry Study Group on 32-55, Australia.
22. Marlow, D. O., Kilby, P. and Mercer, G. N. (2007). *The Travelling Salesman Problem in Maritime Surveillance–Techniques, Algorithms and Analysis*. In Proceedings of the International Congress on Modelling and Simulation on 684-690, New Zealand.
23. Fang, F., Jiang, A. X. and Tambe, M. (2013). Protecting moving targets with multiple mobile resources. *Journal of Artificial Intelligence Research*, 48, 583-634.
24. Cross, M., Marlow, D. and Looker, J. (2007). Application of the non-stationary travelling salesman problem to maritime surveillance. *Proceedings of Multidisciplinary Innovation for Sustainability and Growth*, 1-4.
25. Del Bimbo, A. and Pernici, F. (2005). *Distant Targets Identification as an On-Line Dynamic Vehicle Routing Problem Using an Active-Zooming Camera*. In Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on 97-104, Switzerland.
26. Bimbo, A. D. and Pernici, F. (2005). *Saccades Planning with Kinetic TSP for Distant Targets Identification*. In Imaging for Crime Detection and Prevention, 2005. ICDP 2005. The IEE International Symposium on 145-149, United Kingdom.

27. Ilavarasi, K. and Joseph, K. S. (2014). *Variants of Travelling Salesman Problem: A Survey*. In Information Communication and Embedded Systems (ICICES), 2014 International Conference on 1-7, India.
28. Asahiro, Y., Horiyama, T., Makino, K., Ono, H., Sakuma, T. and Yamashita, M. (2004). How to collect balls moving in the euclidean plane. *Electronic Notes in Theoretical Computer Science*, 91, 229-245.
29. Asahiro, Y., Miyano, E. and Shimoirisa, S. (2008). Grasp and delivery for moving objects on broken lines. *Theory of Computing Systems*, 42(3), 289-305.
30. Chalasani, P., Motwani, R. and Rao, A. N. I. L. (1996). *Algorithms for Robot Grasp and Delivery*. In 2nd International Workshop on Algorithmic Foundations of Robotics, France.
31. Uçar, U. and İşleyen, S . (2019). Hareketli hedefli - heterojen filolu iha rotalama problemi için yeni bir çözüm yaklaşımı. *Politeknik Dergisi*, 22(4) , 999-1016.
32. Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80-91.
33. Uslu, A. (2016). *Afet sonrası insani yardım lojistiğinde stokastik talepli çok depolu araç rotalama problemi: Ankara ili örneği*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
34. Psaraftis, H. N. (1995). Dynamic vehicle routing: status and prospects. *Annals of Operations Research*, 61(1), 143-164.
35. Bertsimas, D. J. and Van Ryzin, G. (1991). A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39(4), 601-615.
36. Bertsimas, D. J. and Van Ryzin, G. (1993). Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. *Operations Research*, 41(1), 60-76.
37. Liao, T. Y. (2017). On-line vehicle routing problems for carbon emissions reduction. *Computer-Aided Civil And Infrastructure Engineering*, 32(12), 1047-1063.
38. Haghani, A. and Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32(11), 2959-2986, 2005.
39. Novoa, C. and Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands, *European Journal of Operational Research*, 196(2), 509-515.
40. Haghani, A. And Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32(11), 2959-2986.
41. Potvin, J. Y. and Xu, Y., Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 33(4), 1129-1137.

42. Van Woensel, T., Kerbache, L., Peremans, H. and Vandaele, N. (2008). Vehicle routing with dynamic travel times: a queueing approach. *European Journal of Operational Research*, 186(3), 990-1007.
43. Taniguchi, E. and Shimamoto, H. (2004). Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transportation Research Part C: Emerging Technologies*, 12(3-4), 235-250.
44. Ning, T., Guo, C. and Chen, R. (2015). A novel method for dynamic vehicle routing problem. *The Open Cybernetics & Systemics Journal*, 9(1), 2254-2258.
45. Chen, S., Chen, R. and Gao, J. (2017). A monarch butterfly optimization for the dynamic vehicle routing problem. *Algorithms*, 10(3), 107.
46. İnternet: Dynamic vrps: a study of scenarios, URL:https://pdfs.semanticscholar.org/19df/8728e4a9b273b6e0a6f84b7f39ff4543e565.pdf?_ga=2.74300882.1114155529.1571680186-2010751376.1523632581, Son Erişim Tarihi: 17.09.2019.
47. Kilby, P., Prosser, P. and Shaw, P. (1998). Dynamic VRPs: A study of scenarios. *University of Strathclyde Technical Report*, 1-11.
48. Ritzinger, U., Puchinger, J. and Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1), 215-231.
49. Shuttleworth, R., Golden, B. L., Smith, S. and Wasil, E. (2008). *Advances In Meter Reading: Heuristic Solution Of The Close Enough Traveling Salesman Problem Over A Street Network*. In *The Vehicle Routing Problem: Latest Advances and New Challenges* on 487-501, United States of America.
50. Mafarja, M. M. and Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260, 302-312.
51. Ingber, L. (1993). Simulated annealing: practice versus theory. *Mathematical and Computer Modelling*, 18(11), 29-57.
52. Talbi, El-Ghazali. (2009). *Metaheuristics: from Design to Implementation*. United States of America: John Wiley & Sons, 126-138.
53. Nasab, S. M. T. I. and Kaveh, M. (2016). A parallel simulated annealing (PSA) for solving project scheduling problem with discounted cash flow policy in pricing strategy of the project suppliers. *Tehnicki Vjesnik-Technical Gazette*, 23(6), 1555-1564.
54. Ehsan, M. and Jelodar Mostafa, J. (2011). *A Hybrid Response Surface Methodology and Simulated Annealing Algorithm*. International Conference on Computer Communication and Management on 570-576, Singapore.

55. Maadi, M., Javidnia, M. and Jamshidi, R. (2017). Two strategies based on meta-heuristic algorithms for parallel row ordering problem (PROP). *Iranian Journal of Management Studies*, 10(2), 467-498.
56. Che, Z. H. (2012). Clustering and selecting suppliers based on simulated annealing algorithms. *Computers & Mathematics with Applications*, 63(1), 228-238.
57. Yıldız, A. R. (2008). Hybrid Taguchi-harmony search algorithm for solving engineering optimization problems. *International Journal of Industrial Engineering*, 15(3), 286-293.
58. Raj, A. (2013). A simulated annealing algorithm for the optimization of surface finish in dry turning of ss 420 materials. *International Journal of Engineering Research & Technology (IJERT)*, 2(6), 482-495.
59. İşleyen, S.K., Uçar, U. and Balo, F. (2019). A new solution approach for maritime surveillance operation: the case of aegean sea. *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2019/8575219>.
60. Bodur, M. Z. (2010). Ege'de denizden yapılan yasa dışı göç ve göçmen profilleri, göçmenlerin geleceğe yönelik beklentileri ve öngörüler. *Güvenlik Stratejileri Dergisi*, 12(12), 103-136.
61. Ege Kıyılarında(İzmir-Aydın) Gerçekleşen Yasa Dışı Göç Hakkında İnceleme Raporu. (2014). Türkiye Büyük Millet Meclisi İnsan Haklarını İnceleme Komisyonu. URL:https://www.tbmm.gov.tr/komisyon/insanhaklari/docs/2014/ege_kiyilarinda_yas_adisigoc_hakkinda_inceleme_raporu_27052014.pdf, Son Erişim Tarihi:16.10.2017.
62. İnternet: Ege denizi, URL:<https://www.google.com/maps/place/%C3%87e%C5%9Fme,+Ovac%C4%B1k,+%C3%87e%C5%9Fme%2F%C4%B0zmir/@37.9938478,26.6706833,10z/data=!4m5!3m4!1s0x14bb79b752d35675:0x61287b379ef50511!8m2!3d38.3228017!4d26.3027607>, Son Erişim Tarihi: 17.09.2019.
63. İnternet: Gemi ve yat resimleri, URL:<https://www.nisanboard.com/konu/png-gemi-ve-yat-resimleri-png-uzant%C4%B1%C4%B1-gemiler-png-uzant%C4%B1%C4%B1-yatlar.166894/>, Son Erişim Tarihi: 17.09.2019.
64. İnternet: S-70-B2_Seahawk helikopteri hakkında bilgi, URL:https://webcache.googleusercontent.com/search?q=cache:YosbDmJ4geAJ:https://tr.wikipedia.org/wiki/S-70-B2_Seahawk+&cd=1&hl=tr&ct=clnk&gl=tr, Son Erişim Tarihi: 16.10.2017.
65. İnternet: S-70-B2 Seahawk helikopter görseli, URL:https://www.google.com.tr/search?q=S-70-B2+Seahawk&rlz=1C1NDCM_trTR717TR717&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiT3NP57_TWAhXFVxoKHauhCHMQ_AUICigB&biw=1366&bih=613#imgrc=_lZtt26tmnRmOM:, Son Erişim Tarihi:16.10.2017.

66. Tsubota, T., Bhaskar, A., Chung, E. and Billot, R. (2011). Arterial traffic congestion analysis using bluetooth duration data. *Australasian Transport Research Forum 2011 Proceedings*, 1-14.
67. Kan, Z., Tang, L., Kwan, M. P., Ren, C., Liu, D. and Li, Q. (2019). Traffic congestion analysis at the turn level using taxis' gps trajectory data. *Computers, Environment and Urban Systems*, 74, 229-243.
68. Susilo, B. H. and Imanuel, I. (2018). Traffic congestion analysis using travel time ratio and degree of saturation on road sections in Palembang, Bandung, Yogyakarta, and Surakarta. In *MATEC Web of Conferences*, 181(2018), 1-10.
69. Wang, Q., Wan, J. and Yuan, Y. (2018). Locality constraint distance metric learning for traffic congestion detection. *Pattern Recognition*, 75, 272-281.
70. He, S. (2012). Quantitative problem of road traffic congestion simulation and extension information analysis. *International Journal of Emerging Technology and Advanced Engineering*, 2(2), 51-55.
71. He, F., Yan, X., Liu, Y. and Ma, L. (2016). A traffic congestion assessment method for urban road networks based on speed performance index. *Procedia Engineering*, 137, 425-433.
72. Petrovska, N. and Stevanovic, A. (2015). *Traffic Congestion Analysis Visualisation Tool*. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems on 1489-1494, Spain.
73. İnternet: Deniz trafiğine ilişkin marinetrafficten alınan trafik yoğunluk bilgileri, URL:<https://www.marinetraffic.com/en/ais/home/centerx:25.6/centery:38.4/zoom:8>, Son Erişim tarihi: 29.04.2019.
74. İnternet: Poisson dağılımı, URL:https://tr.wikipedia.org/wiki/Poisson_da%C4%9F%C4%B1l%C4%B1m%C4%B1, Son Erişim Tarihi: 07.05.2019
75. İnternet: Ters dönüşüm fonksiyonu, URL:https://en.wikipedia.org/wiki/Inverse_transform_sampling, Son Erişim Tarihi: 07.05.2019.
76. İnternet: En az sayıda insansız hava aracı kullanarak sabit hedeflerin gözetlenmesi, URL:http://user.atilim.edu.tr/~murat.karakaya/publications/TOK2013_En_Az_IHA.pdf, Son Erişim Tarihi: 20.10.2019
77. Karp, R. M. (1972). *Complexity of Computer Computations*. New York: Plenum Press, 85-73.
78. Keskindürk, T., Kiremitçi, B. and Kiremitçi, S. (2016). 2-opt algoritması ve başlangıç çözümünün algoritma sonuçları üzerindeki etkisi. *Journal of Industrial Engineering*, 27(3), 2-12.

79. Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics*, 54(8), 811–819.
80. Çolak, S. and Güler, H. (2009). Dağıtım rotaları optimizasyonu için meta sezgisel bir yaklaşım. *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 11(2), 171-190.
81. Groba, C., Sartal, A. and Vázquez, X. H. (2018). Integrating forecasting in metaheuristic methods to solve dynamic routing problems: Evidence from the logistic processes of tuna vessels. *Engineering Applications of Artificial Intelligence*, 76, 55-66.
82. Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, And Artificial Intelligence*. Oxford, England: U Michigan Press.
83. Şahin, M. and Kellegöz, T. (2017). An efficient grouping genetic algorithm for U-shaped assembly line balancing problems with maximizing production rate. *Memetic Computing*, 9(3), 213-229.
84. Gözütok, S. and Özdemir, O. N. (2004). Genetik algoritma yöntemi ile su şebekelerinde hidrolik kalibrasyonun geliştirilmesi. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 19(2), 125-130.
85. Sağ, Tahir. (2008). *Çok kriterli optimizasyon için genetik algoritma yaklaşımları*, Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
86. İnternet: Pareto verimliliği, URL:https://webcache.googleusercontent.com/search?q=cache:R_SYBO9q9YYJ:https://tr.wikipedia.org/wiki/Pareto_verimlili%25C4%259Fi+&cd=1&hl=tr&ct=clnk&gl=tr, Son Erişim Tarihi: 27.01.2019.
87. Temiz, İ. ve Erol, S. (2007). Bulanık akış tipi çizelgeleme problemi için çok amaçlı genetik algoritma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 22(4), 855-862.
88. Lokman, B. (2017). Çok amaçlı tamsayı programlama problemleri için temsili çözüm üreten yaklaşımların ve kalite ölçülerinin incelenmesi. *Journal of Industrial Engineering (Turkish Chamber of Mechanical Engineers)*, 28(1), 19-39.
89. Atlas, M. (2008). Çok amaçlı programlama çözüm tekniklerinin sınıflandırılması. *Anadolu Üniversitesi Sosyal Bilimler Dergisi*, 8(1), 47-68.
90. Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.
91. Durmaz, E. D. (2016). *Çok amaçlı tek sıra tesis düzenleme probleminin çözümü için nsga ii ve hedef programlama yaklaşımı*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : UÇAR, Ukbe Üsame
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 11.10.1991, Elazığ
 Medeni hali : Bekâr
 Telefon : 0 (424) 237 00 00 - 5647
 e-mail : uuucar@firat.edu.tr



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Endüstri Mühendisliği	Devam ediyor
Yüksek lisans	Gazi Üniversitesi / Endüstri Mühendisliği	2015
Lisans	Atatürk Üniversitesi / Endüstri Mühendisliği	2013
Lise	Mehmet Akif Ersoy Lisesi	2009

İş Deneyimi

Yıl	Yer	Görev
2013-2014	Fırat Üniversitesi	Araştırma Görevlisi
2014-2017	Gazi Üniversitesi	Araştırma Görevlisi
2017-Halen	Fırat Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

- 1- İşleyen, S.K., Uçar, U. and Balo. F. (2019). A new solution approach for maritime surveillance operation: the case of aegean sea. *Mathematical Problems in Engineering*, 2019.
- 2- Uçar, U. and İşleyen, S.K. (2019). A meta-heuristic solution approach for the destruction of moving targets through air operations. *International Journal of*

Industrial Engineering: Theory, Applications and Practice. (Accepted: 08.08.2019).

- 3- Uçar, U.Ü. ve İşleyen, S.K. (2016). Telif dersi çizelgeleme probleminin tdcp matematiksel modelle çözümü gerçek bir uygulama. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 31(2), 331-346.
- 4- Uçar, U. ve İşleyen, S. (2019). Hareketli hedefli - heterojen filolu iha rotalama problemi için yeni bir çözüm yaklaşımı. *Politeknik Dergisi*, 22(4) , 999-1016.
- 5- Uçar, U.Ü. and İşleyen, S.K. (2019). Analysis based on graduation opinions of industrial engineering education in Turkey. *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi*, 11(1),109-123.
- 6- Eraslan, Ü.G., Balo, F., Uçar, U.Ü. ve Çetin, B., (2018). Çok kriterli karar verme metodu çerçevesinde elektrik arıza analizinin değerlendirilmesi. *Uluslararası İktisadi ve İdari İncelemeler Dergisi*, 135-146.
- 7- Uçar, U.Ü. and Balo, F. (2018). Determining of the convenient metal dental implant material in terms of strength properties. *Organic and Medicinal Chemistry International Journal*, 6(2),1-3.
- 8- Uçar, U.Ü., Balo, F., Eraslan, G. and Çetin, B. (2017). A decision support system study based on data mining for ideal animal culture (ISIMSIT 2017). *Gaziosmanpaşa Journal of Scientific Research*, 6.
- 9-Uçar, U.Ü., İşleyen, S.K. ve Demir, Y. (2015). Üniversite ders çizelgeleme probleminin bulanık ahp ve çok amaçlı karışık tam sayılı matematiksel modelle çözümü. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 3(3),513-523.
- 10-Uçar, U., and İşleyen, S.K. (Baskıda). *A Survey of Moving Target Traveling Salesman Problem*. Human-Computer Interaction(Chapter 5). Nova Science Publishers.
- 11-Ertekin, Ü.Z., Uçar, U.Ü. ve İşleyen, S.K. (2018). *Üniversite Ders Çizelgeleme Probleminin Gelişimi Ve Yönelimlerine İlişkin Literatür Araştırması*. Mühendislik Alanında Akademik Çalışmalar, Türkiye: Gece Kitaplığı, 99-122.
- 12- Uçar, U.Ü. and İşleyen, S.K. (2018). *Evaluation of Unmanned Combat Aerial Vehicles with AHP Method*. 12th NCMCONFERENCES International Conference on New Challenges in Industrial Engineering and Operations Management, Ankara.
- 13- Uçar, U.Ü. ve Balo, F. (2018). *Enerji Açısından Balıkesir Şehri İlçelerinin Tarımsal Potansiyelin Değerlendirilmesi*. International Symposium of Bandırma and Its Surroundings, Balıkesir.
- 14- Uçar, U.Ü. ve Balo, F. (2018). *Balıkesir Şehri İlçeleri İçin Turistik Gezi Yerlerinin AHP Yöntemi İle Değerlendirilmesi*. International Symposium Of Bandırma And Its Surroundings, Balıkesir.
- 15- Uçar, U.Ü. and Balo, F. (2018, 1-3 November). *Ideal Hologram Paper Selection With Multi Criteria Decision Making Method*. 6 th International Printing Technologies,

İstanbul.

- 16- Uçar, U.Ü. and Balo, F. (2018, 1-3 November). *Ideal Paper Selection with AHP Method for Inject Printers*. 6 th International Printing Technologies Symposium, İstanbul.
- 17- Uçar, U.Ü. and Balo, F. (2018). *A Data Solution Approach Based on Data Mining for Vegetable Cultivation in terms of Ecological*. International Symposium on Artificial Intelligence and Data Processing, Malatya.
- 18- Uçar, U.Ü. and Balo, F. (2018). *A Decision Support System Study on Prioritization of Touristic Spaces in Gaziantep*. 2nd International History And Culture Congress, Gaziantep.
- 19- Uçar, U.Ü. and Balo, F. (2018). *Analysis Based on Customer Views of Gaziantep Food Places*. 2nd International History And Culture Congress, Gaziantep.
- 20- Uçar, U.Ü. and Balo, F. (2018). *Ergonomic Design of Vocational High School Workshop in Terms of Minimum Industrial Accident Risk*. 1.Uluslararası Güvenli Okullar, Çalışan Sağlığı ve İş Güvenliği Sempozyumu, Bolu.
- 21- Uçar, U.Ü. and Balo, F. (2018). *Configuring the Ergonomic Classroom in The University*. 1.Uluslararası Güvenli Okullar, Çalışan Sağlığı ve İş Güvenliği Sempozyumu, Bolu.
- 22- Uçar, U.Ü. and İşleyen, S.K. (2017). *A Solution Approach Based on Simulated Annealing for the Destruction Of Moving Targets In Time Window By Air Operations*. 8th International Advanced Technologies Symposium(IATS), Elazığ.
- 23- Hadık, İ.E., Uçar, U.Ü., Atak, M. ve İşleyen, S.K. (2017). *A Decision Support System for Fish Farming Based on Data Mining: An Application in Turkey*. 8th International Advanced Technologies Symposium(IATS), Elazığ.
- 24- Uçar, U.Ü., İşleyen, S.K., Balcıoğlu, M., Kocaman, H. ve Çakılcı, H. (2017). *Türkiye'deki Endüstri Mühendisliği Bölümlerinin İstatistiksel Açından Analizi ve Karşılaştırılması*. EYİ 2017XVIII. Uluslararası Ekonometri Yöneylem Araştırması ve İstatistik Sempozyumu, Trabzon.
- 25- Çetin, B., Balo, F., Uçar, U.Ü. and Eraslan, G. (2017). *Estimation of Immigration Level in Turkey with ANN from Demographic, Economic and Cultural Perspectives*. I. International Demirci Symposium, Manisa.
- 26- Çetin, B., Balo, F., Uçar, U.Ü. ve Eraslan, G. (2017). *Estimated of Tourist Number with ANN Based Analysis*. I. International Demirci Symposium, Manisa.
- 27- Eraslan, G., Balo, F., Uçar, U.Ü. ve Çetin, B. (2017). *Arıza-Bakım-Onarım Personellerine Ait Doluluk Oranının Tespiti İçin İstatistiksel Bir Çözüm Yaklaşımı*. 18. International Symposium on Econometrics, Operations Research and Statistics EYİ'xx17, Trabzon.

- 28- Eraslan, .G., Balo, F., Çetin, B. ve Uçar, U.Ü. (2017). *Estimation Of Electricity Consumption By Regression And Artificial Neural Networks Approaches*. 1st International Symposium On Multidiciplinary Studies And Innovative Technologies, Tokat.
- 29- Uçar, U.Ü., Balo, F., Çetin, B. ve Eraslan, G. (2017). *Meyve Yetiştiriciliği İçin Veri Madenciliğine Dayalı Bir Karar Destek Sistemi Çalışması*. International Conference on Artificial Intelligence and Data Processing, Malatya.
- 30- Çetin, B., Balo, F., Uçar, U.Ü. and Eraslan, G. (2017). *Rüzgâr Enerjisi Potansiyelinin Sürdürülebilirliğinin Değerlendirilmesinde Yapay Sinir Ağları Yaklaşımı*. 11th International Computer and Instructional Technologies Symposium, Malatya.
- 31- Çetin, B., Balo, F., Eraslan, G. and Uçar, U.Ü. (2017). *Artificial Neural Networks Model Of Wind Speed Values For Wind Farm Feasibility*. Uluslararası Mühendislik Araştırmaları Sempozyum, Düzce.
- 32-Çetin, B., Balo, F., Eraslan, G. ve Uçar, U.Ü. (2017). *Solar Radyasyon Tahminlemesinde Meteorolojik Veri Temelli YSA Modeli*. Uluslararası Mühendislik Araştırmaları Sempozyumu, Düzce.
- 33-Uçar, U.Ü., Balo, F., Çetin, B. ve Eraslan, G. (2017). *Binalardaki Yalıtım Malzemesi Seçimine Veri Madenciliğine Dayalı Bir Karar Destek Yaklaşımı*. 11th International Computer and Instructional Technologies Symposium, Malatya.
- 34-Çetin, B., Balo, F., Uçar, U.Ü. ve Eraslan, G. (2017). *Solar Sistemler İçin Bölgesel Güneş Radyasyonunun Tahmininde Yapay Sinir Ağlarının Kullanılması*. 11th International Computer and Instructional Technologies Symposium, Malatya.
- 35-Uçar, U.Ü. and İşleyen, S.K. (2016). *A Mathematical Modelling Approach for Exam Task Assignment Problem Considering Assistant Preferences: A Case Study*. International Conference on Research in Education & Science, Muğla.
- 36-Uçar, U.Ü., and İşleyen, S.K. (2016). *Mathematical Modeling Solution of University Course Timetabling Problem Under Instructor Preference Constraint A Case Study*. International Conference on Research in Education & Science, Muğla.
- 37-Uslu, A., İşleyen, S.K., Uçar, U.Ü. ve Göldoğan, K. (2015). *Afet Sonrası İnsani Yardım Lojistiğinde Araç Rotalama Problemi*. Yöneylem Araştırması ve Endüstri Mühendisliği 35. Ulusal Kongresi, Ankara.
- 38-Uçar, U.Ü., İşleyen, S.K. ve Uslu, A. (2015). *Üniversite Ders Çizelgeleme Problemi İçin Hiyerarşik Çözüm Yaklaşımı*. 15. Uluslararası Katılımlı Üretim Araştırmaları Sempozyumu, İzmir.
- 39-Uçar, U.Ü., İşleyen, S.K., Uslu, A. ve Göldoğan, K. (2015). *Öğretim Elemanı Tercihlerine Dayalı Üniversite Ders Çizelgeleme Probleminin Kısıt Programlama ile Hiyerarşik Çözümü: Gerçek Bir Uygulama*. Yöneylem Araştırması ve Endüstri Mühendisliği (YAEM) 35. Ulusal Kongresi, Ankara.
- 40-Balo, F. ve Uçar, U.Ü. (2015). *Ömür Maliyet Analiziyle Entegre Edilmiş Tavlama*

Benzetimi Meta Sezgiseli Yöntemiyle Yalıtım Kalınlığının Optimizasyonu. 12. Ulusal Tesisat Mühendisliği Kongresi, İzmir.

Hobiler

Futbol, Müzik, Doğa Yürüyüşü, Organik Tarım



GAZİ GELECEKTİR..