T.C.

İNÖNÜ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SÜRÜ ROBOTLARIN HEDEF BULMA PROBLEMİNE SÜRÜ TABANLI OPTİMİZASYON ALGORİTMALARININ SİMÜLASYON UYGULAMASI ve KARŞILAŞTIRILMASI

Mehmet Akif FINDIKLI

YÜKSEK LİSANS TEZİ

Bilgisayar Bilimleri Anabilim Dalı

ŞUBAT 2019

Tezin Başlığı: Sürü Robotların Hedef Bulma Problemine Sürü Tabanlı Optimizasyon Algoritmalarının Simülasyon Uygulaması ve Karşılaştırılması

Tezi Hazırlayan: Mehmet Akif FINDIKLI

Sınav Tarihi: 05.02.2019

Yukarıda adı geçen tez jürimizce değerlendirilerek Bilgisayar Mühendisliği Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Tez Danışmanı: Dr. Öğr. Üyesi A. Fatih KOCAMAZ …….
İnönü Üniversitesi

Doç. Dr. Resul DAŞ …….
Fırat Üniversitesi

Dr. Öğr. Üyesi B. Baykant ALAGÖZ …….
İnönü Üniversitesi

Prof. Dr. Halil İbrahim ADIGÜZEL
Enstitü Müdürü

ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduğum "İşbirlikçi Çoklu Robotlarda Görev Paylaşımının Farklı Optimizasyon Teknikleriyle Mukayesesi" başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımca yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Mehmet Akif FINDIKLI

ÖZET

Yüksek Lisans Tezi

SÜRÜ ROBOTLARIN HEDEF BULMA PROBLEMİNE SÜRÜ TABANLI OPTİMİZASYON ALGORİTMALARININ SİMÜLASYON UYGULAMASI ve KARŞILAŞTIRILMASI

Mehmet Akif FINDIKLI

İnönü Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Bilimleri Anabilim Dalı

50+ix sayfa

2019

Danışman: Dr. Öğr. Üyesi A. Fatih KOCAMAZ

Bu tez çalışmasında, sürü robotların hedef bulma problemine sürü tabanlı optimizasyon algoritmaları uygulanmış, gerektirdiği durumlarda yeni adaptasyonlar önerilmiş ve benzetimlerinin gerçekleştirildiği bir simülasyon uygulaması geliştirilmiştir. Simülasyon uygulamasında sürü robotların tek hedefi, sinyal şiddetleri eş çoklu hedefleri, sinyal şiddetleri eş olmayan çoklu hedefleri ve konumu sabit olmayan dinamik hedefleri takip edip bulabilme başarıları incelenmiştir. Sürü tabanlı optimizasyon algoritmalarından parçacık sürüsü optimizasyon, yapay arı kolonisi, hızlı yapay arı kolonisi ve modifiye yapay arı kolonisi algoritmaları incelenmiştir ve karşılaştırılmıştır. Algoritmalara gerçek dünyada uygulanabilirliklerini arttırabilmek için değişiklikler önerilmiş ve uygulanmıştır. Önerilen değişiklikler ile beraber UNITY 3D ortamında C# dili ile sürü tabanlı metasezgisel yöntemler ile sürü robotlarda hedef bulma simülasyon uygulaması geliştirilmiştir. Aynı simülasyon

uygulaması, metasezgisel yöntemlerin karşılaştırılabilmesini grafik kütüphaneleri sayesinde daha anlaşılır hale getirmek için MATLAB ortamında da geliştirilmiştir.

Sürü tabanlı optimizasyon algoritmalarının genel becerilerini inceleyebilmek için nümerik olmayan eşleştirme problemi incelenmiştir. Çinli Postacı Probleminin eşleştirme çözümü için yapay arı kolonisi algoritması önerilerek MATLAB ortamında bir uygulama geliştirilmiştir.

ANAHTAR KELİMELER: Sürü Robotlar, Hedef Bulma, Sürü Tabanlı Optimizasyon Algoritmaları, Metasezgisel yöntemler, PSO, ABC, Çinli Postacı Problemi

ABSTRACT

Master Thesis

SIMULATION APPLICATION and COMPARISON OF SWARM BASED OPTIMIZATION ALGORITHMS on SOURCE LOCALIZATION PROBLEM of SWARM ROBOTICS

Mehmet Akif FINDIKLI

İnonu University

Graduate School of Natural and Applied Sciences

Department of Computer Science

50+ix pages

2019

Supervisor: Asst. Prof. Dr. A. Fatih KOCAMAZ

In this thesis, we applied swarm-based optimization algorithms to problem of finding source location of swarm robotics, and we suggested new adaptations to those algorithms if needed, then we developed a simulation program which can simulate those algorithms. We analyzed success of source tracking and source localization of swarm robotics in, single source localization, multiple source localization which have homogenious signal intensitiy and heterogenious signal intensity, and dynamic source localization scenarios. We analyzed and compared swarm-based optimization algorithms such as particle swarm optimization, artificial bee colony, quick artificial bee colony and modified bee colony. We also suggested and integrated modification to those algorithms in order to apply those solutions in real world applications. With suggested modifications we developed a source localization in swarm robotics simulation application with C# language in UNITY 3D. We developed same application in MATLAB in order to compare metaheuristic methods more comprehensible with figure libraries of MATLAB.

We analyzed non-numerical matching problem, in order to get better sense of swarm-based optimization algorithms' skill set. We suggested artificial bee colony algorithm and developed an application with MATLAB for matching solution of Chinese postman problem.

KEYWORDS: Swarm Robotics, Source Localization, Swarm-based Optimization Algorithms, Meta-heuristic methods, PSO, ABC, Chinese Postman Problem

TEŞEKKÜR

Tez konumun şekillenmesindeki ve çalışmalarımın ilerlemesindeki rolü için danışman hocam Sayın Dr. Öğr.Üyesi A. Fatih KOCAMAZ'a;

Tez çalışmalarım ve diğer alanlardaki çalışmalarımda desteklerini hiçbir zaman esirgemeyen değerli hocam Sayın Prof. Dr. Ali KARCI' ya;

Çalışmalarım boyunca her türlü soruma cevap veren bana yol gösteren, yardımlarını esirgemeyen değerli hocam Sayın Dr. Öğr. Üyesi B. Baykant ALAGÖZ'e;

Birlikte görev yaptığım meslektaşım, Sayın Dr. Öğr. Üyesi A. Erhan AKKAYA ve arkadaşım Sayın Eda FENDOĞLU'ya;

Aileme ve ailemizin yeni üyesi yeğenim Nil Fatma'ya;

Teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

AAC	Artificial Ant Colony
ABC	Artificial Bee Colony
APF	Artificial Potential Field
DOA	Direction of Arrival
GA	Genetic Algorithms
GAF	Gözcü Arı Fazı
İAF	İşçi Arı Fazı
KAF	Kâşif Arı Fazı
mABC	Modified Artificial Bee Colony
PFSM	Probabilistic Finite State Machines
PSO	Particle Swarm Optimization
qABC	Quick Artificial Bee Colony
RFID	Radio Frequency Identification
SN	Source Number
SSL	Sound Source Localization
TBA	Trophollaxis Based Algorithms
TS	Tabu Search
c_1	Kişisel Öğrenme Katsayısı
c_2	Sürü Geneli Öğrenme Katsayısı
i	İterasyon
w	Atalet Katsayısı
v_i	i. Çözüm
x_i	i. Birey

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Sürekli bir modeli olmayan, birçok kısıt barındıran veya karmaşık modele sahip problemlerin kesin çözümüne ulaşmak, günümüz bilgisayarlarının işlemci kapasiteleri göz önüne alındığında asırlarca vakit alabileceği çalışmalarda gösterilmektedir. Ancak kesin çözümünün imkânsız veya çok zor olduğu bu problemler için geliştirilen tekniklerden olan sezgisel yöntemler kullanabilmektedir. Sezgisel yöntemler, bir problemin kesin çözümüne tahminlerde bulunarak yaklaşmayı hedeflemektedir. Bu yöntemler, sürüce hareket ederek sorunlarının üstesinden gelebilen canlılardan esinlenerek sürü tabanlı hale gelmişlerdir. Sezgisel tekniklere göre daha etkili ve rassal bileşenler içeren bu yöntemler "daha öte" manası olan meta ön ekini alarak metasezgisel yöntemler olarak anılmaya başlamıştır. Metasezgisel yöntemler tahmin tabanlı olduğu için, türev veya diferansiyel denklem tabanlı yöntemlerde olduğu gibi evrensel kurallar barındıramamaktadır. Bu durum neredeyse her problem için özelleştirilebilen birçok metasezgisel yöntemin ortaya çıkmasına sebep olmuştur. Bu tez çalışmasında, sürü robotların otonom bir şekilde bulmaları istenen hedefe, birbirleri ile işbirliği içinde hareket ederek ulaşmaları amaçlanmıştır. Sürü robotların alanda hedef bulma davranışı problemi, metasezgisel yöntemlerin arama uzayında çözüm arama yeteneklerine benzetilmiştir. Bu kapsamda Unity 3D ortamında C# dili ile kullanılan metasezgisel yöntemlerin hedef bulma problemleri için incelenebileceği ve değerlendirilebileceği bir simülasyon uygulaması geliştirilmiştir. İkinci uygulama olarak nümerik olmayan eşleştirme problemlerinden Çinli Postacı Probleminin graflarda düğüm eşleştirme adımı için MATLAB ortamında yapay arı kolonisi algoritması kullanılarak bir uygulama geliştirilmiş ve uygulama sayesinde verilen grafta gezinme maliyetinin düşürülmesi hedeflenmiştir.

Tez kapsamıyla alakalı literatür taraması yapılmış, daha önce kullanılan teknikler ve kavramlar hakkında inceleme sonuçları tezin 2. bölümünde paylaşılmıştır. Simülasyon uygulaması için kullanılan teknikler literatürden gelen genel bilgiler doğrultusunda tezin 3. bölümünde paylaşılmıştır. Geliştirilen iki simülasyon uygulaması ve simülasyon uygulamalarının sonuçları 4. ve 5. bölümde kendi ana başlıkları altında paylaşılmıştır. Çalışmaların sonucunda elde edilen bulgular ve sonuçlar 6. bölümde incelenmiş, ve ileride yapılabilecek çalışmalar için önerilerde bulunulmuştur.

2. LİTERATÜR TARAMASI

2.1. Sürü Robotlar ve Hedef Bulma Problemi

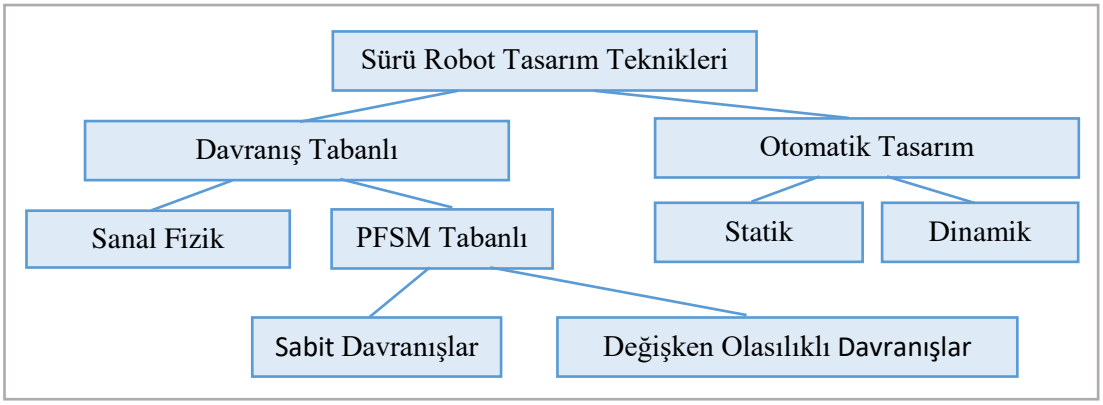
Son yıllarda sürü robot sistemleri üzerinde birçok çalışma literatüre girmektedir. Bu çalışmalarda sürü robotların tekil robotlara göre daha verimli olduğu gözlemlenmiştir [1]. Bu verimlilik sürü robotların hata toleransının daha düşük olması, verilen görevi yerine getirmedeki adaptif yapıları, dağınık halde bulunan çoklu sensörlerin avantajları ve işbirliği yapabilmelerinden kaynaklanabilmektedir [2]. Sürü robot sistemi aynı tip bireylerden oluşan homojen sürülerden oluşabileceği gibi, farklı görevlere sahip farklı bireylerden oluşan heterojen sürülerden de oluşabilir.

Sürü kavramı biyoloji alanında incelenen, balık, kuş veya koloni kurabilen böcekler gibi birçok bireyin işbirliği içerisinde bir uyumla yaşayabildiği canlıları tanımlamaktadır. Sürü robotlar da birbiri ile aynı özelliklere sahip küçük robotlardan oluşmaktadır. Bu robotlar tek başlarınayken iş kapasiteleri düşük olan ancak bir aradayken yapılması istenen işi birbirleriyle veya çevreleriyle etkileşime girerek etkili bir şekilde tamamlayan sürüleri oluşturmaktadır. Sürü robotlardan beklenen; gruplanma, karıncalar gibi istenen nesneleri toplama veya bir hedefe taşıma, alanı daha iyi tanıyıp alana daha verimli bir şekilde dağılabilmek için bölgeyi keşfetme, kuşlar ve balıklar gibi sürüce hareket etme, arıların yuva kurması gibi kümelenme işlemleri olabilir. Şekil 2.1'de biyolojik sürülere örnekler görülmektedir.

Şekil 2.1. Doğal Sürülere Örnekler *[3]*

Sürü robot yapıları tasarımı için uygulanması gereken kesin bir yöntem bulunmamaktadır [4]. Sürü robot yapısı tasarlama yöntemleri genel olarak Şekil 2.2'de görüldüğü gibi gruplandırılabilir [4]. Sürü robot yapıları genel olarak davranışsal veya otomatik metotlar ile tasarlanabilir. Davranışsal yöntem sürünün istenen işbirlikçi davranışını gerçekleştirebilmesi için tasarlanan iteratif bir metottur. Çoğu davranış tabanlı tasarımlar ya Olasılıksal Sonlu Durum Makinesi (Probabilistic Finite State

Machine - PFSM) ile ya da Sanal Fizik tabanlı tasarım teknikleri ile gerçekleştirilmektedir [4].



Şekil 2.2. Sürü Robot Tasarım Teknikleri

Olasılıksal davranış yaklaşımı literatürde ilk defa Minsky tarafından 1967 yılında önerilmiş ve bir grup sürü yapısının yönetimini göz önünde bulundurmuştur [5]. PSFM için de iki farklı yöntemden bahsedilebilir. Bunlardan biri genellikle sistemdeki robotların çözüme ulaşana kadar hepsine ortak bir şekilde uygulanan sabit davranışlar serisi şeklinde tanımlanabilirken diğeri robotların çevreden ya da birbirlerinden aldıkları girdilere göre sistemin matematiksel modelini değiştirebildikleri olasılıksal davranış modelidir. Araştırmacılar, sanal fizik tabanlı yöntemlerde ise Yapay Potansiyel Alanda (*Artificial Potential Field* - APF) bulunan sistemdeki tüm robotları, birbirlerine, çevrelerine ve çevrede bulunan engel gibi cisimlere itici veya çekici kuvvet uygulayabildikleri sanal parçacıklar olarak düşünmüşlerdir [6,7]. Örnek olarak ulaşılması gereken hedef sürüdeki robotlar tarafından çekici bir kuvvet olarak düşünülebilirken, alanda bulunan engellerin ise itici bir kuvvet uyguladığı varsayılabilir.

Otomatik tasarım metotları statik ve dinamik sahalar olarak gruplandırılmaktadır. Statik sahalar, engellerin veya hedeflerin zaman göre yer değiştirmediği sahaları temsil etmektedir. Bu tür statik sahalar için robotlar eğiticili öğrenme metotları ile ve deneme yanılma yöntemleri ile sahaya adapte olabilmektedir. Adaptasyon işlemleri genellikle yapay sinir ağları ve hızlı öğrenme teknikleri ile gerçekleştirilmektedir. Ancak eğiticili öğrenme metotları sahadaki engellerin veya hedeflerin dinamik olarak pozisyon değiştirdiği koşullarda oldukça başarısız kalmaktadırlar [8]. Dinamik sahaların verimliliğini ölçmek için sürü robotlar evrimsel

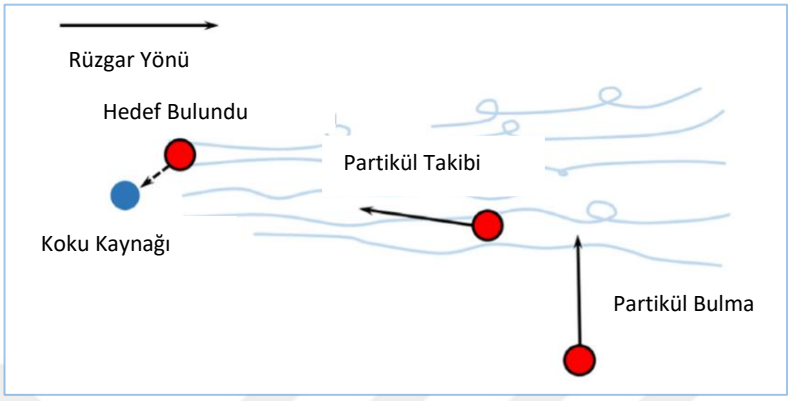
programlama yöntemleri ile tasarlanmaktadır ve genellikle incelenen sahaların verimliliğini test edebilmek için Darwin'in evrim prensiplerinden ilham alınarak modellenmektedir. Evrimsel robotların istenen gözlemleri gerçekleştirebilmeleri için robotlar, uygunluk fonksiyonu tüm sistemde geçerli olan homojen sistemlerde uygulanmaktadır.

Hedef bulma, bir nesnenin ses, ışık gibi yaydığı sinyallerin veya koku ya da sızıntı gibi kimyasal bileşenlerin takibi ile o nesnenin konumunun belirlenmesi olarak düşünülebilir. Örnek olarak yarasaların uçan böcekleri ses takibi ile bulmaları (Sound Source Localization- SSL), bakterilerin enfekte edecekleri hücrelerden yayılan kimyasallara göre yaklaşmaları (kemotaksis), köpeklerin koku duyularını kullanarak hedeflerini bulmaları (Odor Source Localization) gösterilebilir.

Chen ve Huang 2019 yılında yayınlanan çalışmalarında köpeklerin insanlara göre kokuya 108 kat daha duyarlı olduğunu savunmuşlardır [9]. Bu sebeple köpekler insanlar tarafından bazı nesnelerin ve hatta insanların bulunması için kullanılmaktadır. Ancak patlayıcı maddeler, canlılara zararlı kimyasallar veya su altı sızıntılarının kaynaklarını bulma gibi problemlerde köpek veya başka bir canlının kullanılması, kullanılan canlı ve onların yöneticisi olan insanlar için risk oluşturduğundan dolayı 1990'lı yılların başlarından itibaren bu tür tehlikeli kimyasalları bulmak için robot veya robot sürüleri tasarlanmaya başlanmıştır.

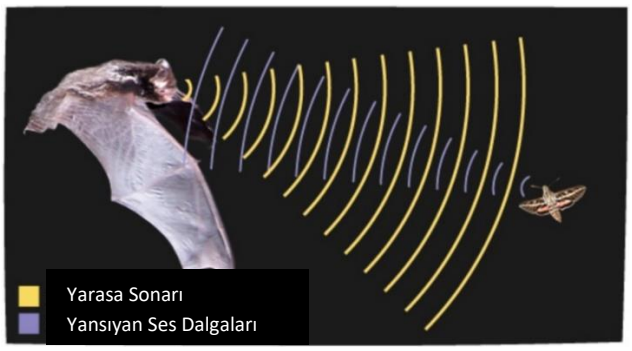
Chen ve Huang'a göre koku takibi ile hedef bulma alanında yapılan ilk çalışmalarda fazla verimli sonuçlar üretilememiştir. Robotlar, biyolojik eşleniklerine göre başarısız kalmışlardır. Bunun temel olarak iki sebebi bulunmaktadır. En önemli sebep, koku partiküllerinin havadaki türbülanslardan etkilenerek çevreye düzensiz bir şekilde dağılmalarıdır. Bir diğer sebep olarak da, koku partiküllerinin yoğunluğundaki değişimi algılayacak olan sensörlerin o yıllarda hızlı bir şekilde tepki verememesi gösterilebilir. Bu başarısızlığın üstesinden gelebilmek için, araştırmacılar, tepki süresi daha hızlı olan sensörler geliştirmek için çalışmalar yapmışlardır [9]. Ayrıca türbülanstan dolayı dağınık bir şekilde yayılan koku partiküllerini daha iyi çözümleyebilmek için tekli robotlarda sensör dizileri kullanılmış ya da sürü robotların sahada dağılabilme yeteneklerinin avantajını kullanan, dağınık sensörler kullanılmıştır [10]. Kokuyla hedef bulma işlemi Şekil 2.3'de görüldüğü gibi öncelikle koku partiküllerinin bulunduğu alanı (odor plume) bulması ile başlayıp daha sonra bu

partiküllerin kaynağına yönelip hedefi bulması ile tamamlanmaktadır. Günümüze kadar bu partiküllerin dağınıklığını göz önünde bulunduran, basit gradyan iniş algoritmalardan karmaşık ve yeni donanımların avantajlarını kullanan yüksek seviye koku ile hedef bulma algoritmaları geliştirilmiştir.



Şekil 2.3. Koku Takibi ile Hedef Bulma Adımları *[9]*

Ses ile hedef bulma, nesnenin yaydığı seslerin incelenerek o nesnenin pozisyonunun otomatik olarak tahmin edilmesi olarak tanımlanmaktadır. Ses ile hedef bulma biyolojik canlılar tarafından kaçılacak hedefin yerinin belirlenmesi, avlanacak canlının tespit edilmesi, sürünün birbiri ile iletişimi gibi hayati öneme sahip işlevleri gerçekleştirmek için kullanılmaktadır. Buna örnek olarak yarasaların hedeflerini hiçbir görsel veri olmadan ses yansımalarını kullanarak kesin bir şekilde belirleyebilmeleri işlemi (ekolokasyon) Şekil 2.4'deki gibi gösterilebilir.



Şekil 2.4. Yarasaların Ekolokasyon Davranışı [11]

Rascon ve Meza 2017 yılında yaptıkları SSL ile ilgili yaptıkları incelemede ses ile hedef bulma fonksiyonunun robotik alanında birçok durumda fayda sağladığından bahsetmişlerdir [12]. Bunlar hizmet sektöründe çalıştırılabilecek robotların, konuşan insanların yerini tespit etme, ya da görsel temasın bulunamadığı

durumlardaki arama kurtarma operasyonları, ya da bilinmeyen bir alanın akustik yöntemlerle haritalandırılması gibi fonksiyonlar olabilir. Robotların bu alandaki başarıları da robotların işitme kapasitelerinin yüksek olmasına bağlıdır. Bir robotun işitme kapasitesinin yüksek olması, ses kaynaklarının ayrılması, sınıflandırılması veya otomatik konuşma tanıma sistemleri üzerinde önemli bir etkiye sahiptir [13]. SSL'in temel olarak iki adımından bahsetmek mümkündür. Bunlar sesin yönünün tahmini ve sesin uzaklığının tahminidir.

SSL tekniklerini gerçek hayatta uygulayabilmek için, çevrede birden fazla ses kaynağı olabileceğini göz önünde bulundurmak gerekir. Dolayısıyla aynı anda gelen birden fazla sesin ayrıştırılması ve ayrı ayrı pozisyonlarının tahmin edilmesine ihtiyaç duyulmaktadır. Ayrıca robotun veya ses kaynağının hareketli olabileceği de göz önünde bulundurulduğunda, bu kaynakların takibini sürekli bir şekilde gerçekleştirebilecek kapasitede olmaları beklenmektedir. SSL teknikleri günümüze kadar robotik alanında yapılan çalışmalar ile ilerleme kaydetmektedir. Bu alandaki teknikler, geliş yönü tahmini (direction of arrival - DOA), öğrenme tabanlı yaklaşımlar, akustik ışın oluşturma yöntemleri (acoustic beamforming), altuzay metotları, kaynak kümeleme, filtreleme yöntemleri gibi alt gruplarda incelenebilir. Bu teknikler robotlara entegre edilirken, kullanılan mikrofon sayısı ve tipi, kaynakların sayısı ve mobiliteleri, gürültü veya yansımalardan etkilenmeme, kullanılacak sensörlerin dizilim geometrileri ya da entegre edilecek robot çeşitleri gibi özellikler göz önünde bulundurulmaktadır. SSL alanı, koku ile hedef bulma alanına göre daha yetkindir. Rascon ve Meza buna kanıt olarak son yıllarda yapılmış bazı çalışmaları göstermiştir. Bu çalışmalara örnek olarak stereofonik robotların işitme yeteneği ile ilgili Argentieri ve arkadaşlarının 2013'teki robotik alanındaki stereofonik sistemler başlıklı incelemesi ile 2015 yılındaki robotik alanındaki SSL tekniklerinin genişçe incelendiği çalışmaları örnek olarak gösterilebilir [14,15]. Okuno ve Nakadai ise 2015 yılındaki çalışmalarında robotik alanındaki SSL tekniklerinin yükselişini geniş bir şekilde incelemişlerdir [16].

2.2. Metasezgisel Yöntemler

Wahab v.d. 2015 yılında yaptıkları sürü tabanlı optimizasyon algoritmaları incelemelerinde günümüzde neredeyse her alanda optimizasyon işlemine ihtiyaç duyulduğunu belirtmişlerdir [17]. Bu alanlara mühendislik tasarımları, ekonomi, tatil

planlama, internet yönlendirme protokolleri tasarımı gibi alanlar örnek verilebilir. Para gibi, başka kaynaklar ve zaman da daima sınırlıdır, dolayısı ile bu kaynakların en verimli şekilde (optimal) kullanımı oldukça önemlidir. Günümüzdeki çoğu problem lineer olmayabilir ya da çoklu modeller içerebilir. Hatta bu problemler çok karmaşık kısıtlayıcılar içerebilir. Bazı problemlerin ise optimal tek bir çözümü olmayabilir [17].

Genelde optimal çözümü bulma veya yaklaşma kolay olmayabilir. Örneğin $f(x) = x^2$ probleminin minimum sonucu x, $-\infty$ $+\infty$ tanım aralığındayken $x = 0$ için $f(x) = 0$'dır. Bu tür basit problemlerde minimum veya maksimum sonucu bulmak için problemin birinci türevini sıfıra eşitleyerek çözüm noktalarını bulabiliriz. İkinci türevin kökler etrafında pozitif veya negatif değerde olması da bu çözümün minimum veya maksimum olduğunu gösterir. Ancak lineer olmayan, süreksiz, çoklu modeller ve kısıtlar içeren problemler için çözüm bulmak bu kadar basit değildir. Ayrıca metasezgisel yöntemler matematiksel modelin kesin olarak bilinmediği durumlarda etkin olarak kullanılabilir. Dolayısıyla bu tür problemlerin çözümü için daha farklı yaklaşımlara ihtiyaç duyulmaktadır. 1940'lı yıllardan 1960'lı yıllara kadar araştırmacılar bu tür problemlerin çözümü için değişik tahmini yöntemler önermişlerdir. 1966'da ise Fogel evrimsel programlamayı geliştirmiştir [18]. Holland ise yine bu dönemde genetik algoritmayı geliştirmiş ve 1975 yılında bu konudaki çalışmalarını kitaplaştırmıştır [19]. Bu zamanlarda tahmini sonuç bulma yöntemleri literatürde sezgisel yöntemler olarak anılmaktaydı. Ancak Glover 1986 yılında yaptığı seminer çalışmasında bu alanda kullanılan sezgisel kelimesi başına **"meta"** *(ötesinde)* ön ekini eklediğinden bu yana, "lokal optimumu bulmak için normalde geliştirilebilen sezgisel yöntemlerin **ötesinde** çözümler üretebilen ana stratejiler", metasezgisel yöntemler olarak adlandırılmıştır [20].

Metasezgisel yöntemleri, rassal bileşenleri bulunan yerel arama yöntemleri olarak genellemek mümkündür. Problemler için kaliteli çözümler kısa süreler içinde bu yöntemler sayesinde bulunabilir. Ancak bulunan çözümün en optimal çözüm olduğunun garantisi yoktur. Bu yöntemler çoğu problemde optimal çözüm üretebilmeleri için uyarlanabilir yöntemler olsa da bazı problemlere iyi sonuç üretemeyebilirler. Voss 2001 yılında yaptığı çalışmada neredeyse her metasezgisel yöntemin amacının global optimum noktaların bulunması olduğunu ifade etmiştir [20]. Blum ve Roli 2003 yılında metasezgisel yöntemlerin temel olarak iki ana bileşen

içerdiğinden bahsetmiştir [21]. Bu bileşenleri, yoğunlaşma (intensification) ve çeşitleme (diversification) veya sömürme (exploitation) ve keşfetme (exploration) olarak adlandırmışlardır. Çeşitleme, arama uzayının keşfi için dağınık çözümler üreterek global arama işlemini temsil etmekteyken yoğunlaşma, iyi bir çözüme yaklaşılan bölgede lokal arama işlemini temsil etmektedir. İyi bir metasezgisel yöntemin global optimuma ulaşabilmesi, yoğunlaşma ve çeşitleme bileşenlerinin dengeli bir şekilde kullanılmasına bağlıdır. Bazı metasezgisel yöntemler ise bellek bileşenini de bu iki bileşenle beraber kullanmaktadır. Bellek bileşeninin en önemli amacı, çözüm uzayında daha önceden bulunan sonuçların da yeni çözümler üretmek için kullanılmasını sağlamaktır. Bellek bileşenini kullanan literatürde önerilen ilk metasezgisel yöntem "Tabu Arama" (Tabu Search - TS) 1980'lı yıllardan itibaren Glover tarafından geliştirilmiş ve 1997 yılında Glover ve Laguna tarafından kitaplaştırılmıştır [22]. Birçok alanda kullanılan bir diğer metasezgisel yöntem olan Genetik Algoritmalar (Genetic Algorithms - GA) ise 1975 yılında Holland tarafından bilim dünyasına kazandırılmıştır [19]. Günümüze kadar bir çok optimizasyon problemi genetik algoritmalar ile çözümlenmiştir. Holland, bu algoritmayı geliştirirken popülasyonların doğal evrim süreçlerinden ilham almıştır. Ebeveynlerin genetik bilgilerinin çocuklar oluşturulurken farklı kombinasyonlar ile aktarımının gerçekleşmesi (crossing-over, çaprazlama) ve hatta bazen bu kombinasyonlar ile oluşamayacak mutasyonlar bu algoritmanın da temel yapıtaşlarını oluşturmaktadır. Algoritmanın çaprazlama safhası, yerel arama yani yoğunlaşma bileşenini, mutasyon fenomeni ise global arama yani genişletme bileşenini oluşturmaktadır.

Metasezgisel yöntemlerden biyolojideki sürü davranışlarından esinlenen belki de ilk algoritma olan Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO) 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiştir [23]. Bu algoritma özellikle kuş ve balıkların hedeflerinin etrafında sürüce hareket etmelerinden esinlenmiştir. Bu sürülerde dikkat çeken durum, sürülerin bir liderinin bulunmamasıdır. Sürüdeki her birey diğer bireylerle eşit öneme sahip olduğu halde, bu sürüler işbirlikçi bir şekilde sürü zekâlarını kullanarak bir noktaya ulaşabilmektedirler [24]. PSO algoritması da sürülerin bu etkileyici davranışını parçacıkların hareketleri olarak yorumlamıştır. PSO, sürü tabanlı yapay zekâ alanında oldukça ilgi görmüştür ve hala genişleyen bir alanda PSO ile uygulamalar geliştirilmektedir. Birçok varyantı türetilerek yeni alanlarda da etkili bir şekilde çözümler üretebildiği araştırmacılar

tarafından gösterilmiştir [25,26]. Türkiye'de de metasezgisel yöntemler alanında çalışmalar yapılmış 2005 yılında Karaboğa tarafından Yapay Arı Kolonisi (Artificial Bee Colony - ABC) algoritması literatüre kazandırılmıştır [27]. Bu algoritma, bal arılarının besin bulma davranışından esinlenmiştir. Bal arıları besinleri bulmak için kendi içlerinde görev paylaşımı yapmaktadır. Besin arama işlemi, kâşif arılar (scout bees), işçi arılar (employed bees) ve gözcü arıların (onlooker bees) ortak işbirliği ile gerçekleşmektedir.

TS, GA, ABC ve PSO algoritmalarından başka daha birçok metaezgisel yöntem geliştirilmiş ve bu yöntemler birçok alanda kullanılmıştır. Dorigo tarafından 1992 yılında önerilen karınca kolonisi algoritması [28], Yang tarafından 2008 yılında önerilen ateş böceği ve guguk kuşu algoritmaları [29-30]ve Karcı ile Canayaz'ın önerdiği ağustos böceği algoritmaları örnek olarak gösterilebilir [31]. Ayrıca mevcut olan metasezgisel yöntemler de modifiye edilerek bu metasezgisel yöntemlerin varyantları oluşturulmuştur. Bu varyantlara örnek olarak Karaboğa ve Görkemli tarafından 2013 yılında ABC için önerilen Hızlı Yapay Arı Kolonisi (quick artificial bee colony - qABC) algoritması ve Babayiğit ile Özdemir tarafından önerilen Modifiye Yapay Arı Kolonisi (modified artificial bee colony) algoritması gösterilebilir [32] [33].

Fong v.d'nin 2014 yılında yaptıkları robotikte metasezgisel yöntemler isimli inceleme çalışmalarında robotik alanında optimizasyon tekniklerine oldukça ihtiyaç duyulduğunu ortaya koymuşlardır [34]. Fong v.s.'ye göre bir robotun veya robot sisteminin sürekli değişen çevrelerine adapte olabilme yetenekleri, bu tür robotların ev temizliği, askeri alanda arama kurtarma, yabancı gezegende örnek toplama işlemlerini efektif bir şekilde gerçekleştirebilmelerini sağlayan en önemli faktördür. Robotların dinamik bir şekilde değişen çevrelere adapte olup yeni rotalar oluşturabilme yeteneği, bu konuda başarıları bir çok çalışmada gösterilen metasezgisel optimizasyon algoritmaları ile elde etmek mümkündür. Ancak sürü robotların hedef bulma yöntemleri mobil robotların tek başlarına hedef bulma yöntemlerine göre daha farklı olduğu çalışmalarda gösterilmiştir [35-37]. Hedef bulma, hedefe yaklaşma yani taksonomi yeteneğinin mobil ve sürü robotlarda farklı olmasının en önemli sebebi olarak sürü robotların dağınık halde bulunan sensörleri gösterilebilir. Bir diğer sebep olarak da sürü robotların alanda hem lokal hem de global arama yeteneklerine sahip

olmaları gösterilebilir. Fang v.d sürü robotların bu karakteristik özelliklerinden dolayı taksonomi yeteneklerinin sürü tabanlı metasezgisel optimizasyon algoritmalarının matematiksel hiperuzaydaki optimum nokta bulma davranışına oldukça benzer olduğunu belirtmiştir. İki sistemde de hedef arayan bireyler hem lokal arama hem de global arama gerçekleştirmekte ve birbirleriyle veya merkezi bir üs ile hedef noktaları hakkında elde ettikleri bilgilerini sürekli paylaşmaktadırlar [38,39]. Fogel'e göre robotik alanında sürü tabanlı optimizasyon algoritmalarının son yirmi yıldır değişik biçimlerde kullanılsa da sürü robotların ve sürü tabanlı metasezgisel yöntemlerin hedef arama davranışı benzerliği avantajını ilk defa 2010 yılında Hereford & Siebold kullanmıştır [40]. Hereford & Siebold çalışmalarında PSO algoritmasındaki bireylerin davranışını direk olarak robotlara yüklemiştir. Çalışmalarında PSO algoritmasını Trofolaksis (Canlılarda sıvı veya besin değişimi) Tabanlı Algoritmalar (Trophollaxis based Algorithms - TBA) ile birleştirerek bireylerin birbirleriyle uzaktan etkileşime girmeleri yerine, yanlızca birbirlerine yakın olan bireylerin etkileşmelerini sağlamışlardır. Böylece iyi bir pozisyon bulan bireyin fazla hareket ederek bu pozisyonu kaybetmesi engellenmek istenmiştir. Hereford & Siebold tarafından geliştirilen TBA konseptine benzer bir konsept Schmickl ve Crailsheim tarafından da incelenmiş ve çalışmalarında bu torofolaksis davranışı arıların polenleri kaynaklardan kovana taşıma sürecinde polenleri birbirlerine transferlerine benzetilmiştir [41]. Bu arı davranışından esinlenilerek tasarlanan robotlar hareket ettikçe taşıdıkları polen miktarları (çözüm değerleri) azalmaktadır. Sahada birbirlerine yaklaşan bu robotlar, üzerlerindeki polen miktarlarını paylaşarak aslında hedefe olan uzaklıkları bilgisini de paylaşmaktadır. [42]. Abidin v.d çalışmalarında, arama prensipleri PSO'ya oldukça benzer olan Sirke Sineği Algoritması'nın çözüm arama davranışını direkt olarak sürü robotların arama davranışlarına entegre etmişlerdir. Uygulamalarını Drosobot isimli su robotlarında gerçekleştirmiş ve Sirke Sineği Algoritmasının PSO algoritmasına göre robotlara uyarlanabilirliğinin daha uygun olduğunu öne sürmüşlerdir [43].

2.3. Graf Teorisi ve Çinli Postacı Problemi

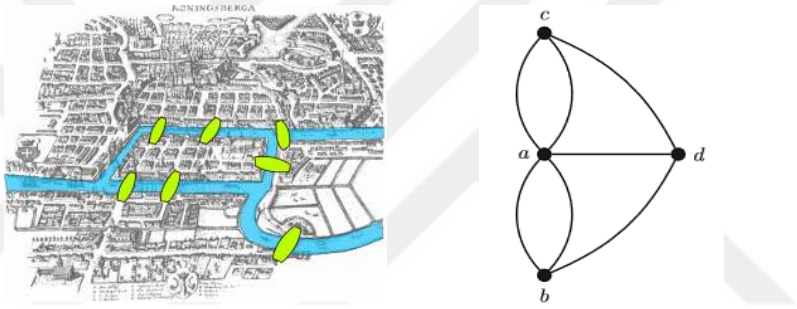
Sürü tabanlı metasezgisel yöntemlerin engelsiz, sınırsız hareket ihtimalinin olduğu sahalarda incelenmesinden önce, bu yöntemler, sınırlı hareketin olduğu graf yapıları üzerinde araştırma yapılmıştır. Metasezgisel yöntemlerin genel optimizasyon yeteneklerini kavrayabilmek için çinli postacı probleminin eşleşme adımına uygulanabilecek teknikler incelenmiştir. Çinli postacı probleminin metasezgisel yöntemler ile çözülmesi ile günlük hayatta kullanılan temizlik robotlarının gezinme maliyetinin düşürülmesi amaçlanmıştır.

Gündelik hayatta ve endüstriyel alanda ihtiyaç duyulan görevleri yerine getirmek için birçok robot ve robot uygulamaları geliştirilmiştir. Bu robotlar tamamen mobil olup otonom işlemler gerçekleştirebileceği gibi, bir platforma bağlı bir şekilde önceden kodlanmış emirleri yerine getiren bir mekanik bir uzuv da olabilir. Günlük hayatımıza dâhil olan temizlik robotu, gözlem robotu gibi amacı verilen sahayı etkili bir biçimde gezmek olan robotlar için değişik yaklaşımlar önerilmiştir. Bu yaklaşımlara sahayı önceden RFID etiketleri ile donatma, tavana veya tabana yerleştirilen çizginin takibi yöntemleri örnek verilebilir [44,45]. RFID etiketlerinin veya takip edilecek analog veya dijital çizgilerin oluşturulması, gezilecek sahanın önceden haritalanmış olması ile mümkündür. Bu haritalanmış alanda robotun gezinmesi önceden belirlenmiş çizgiler üzerinde otonom veya otonom olmayan yöntemlerle geliştirilmiş rotalar ile gerçekleşmektedir. Bu rotalar takip edilecek yollar (ayrıt) veya kavşaklar ve köşe noktalarından (düğüm) oluşmaktadır. Ayrıtlardan ve düğümlerden oluşan bu rotalar graf yapısındadır ve literatürde bulunan graf teorisi işlemleri için uygundur [46].

Graf teorisi ilk defa 1736 yılında Leonhard Euler tarafından literatüre kazandırılmıştır. Euler bu teorisini meşhur Königsberg köprü problemini çözmeye çalışması ile öne sürmüştür [46]. Şekil 2.5 (a)'da görülen Pregel nehri üzerinde yedi köprü bulunmaktadır. Eulerin iddiasına göre bu yedi köprünün oluşturduğu kapalı döngüde her bir köprüyü bir kere kullanılmak şartıyla başlanılan noktaya ulaşmak imkansızdır. Problemin Euler tarafından Şekil 2.5 (b)'deki benzetimi graf teorisinin de temellerini atmıştır . Eğer bir grafta her bir ayrıttan yalnızca bir kere geçilerek tüm ayrıtların tamamı gezilerek başlanan noktaya dönülüyorsa bu graf Eularian'dır denir. Eğer bir grafta bulunan tüm düğümler çift dereceli düğüm ise bu graf ancak ve ancak

Eularian graftır [47,48]. Çünkü bir noktanın çift dereceye sahip olması, o noktaya gidilen her bir ayrıt için çıkılacak bir ayrıt olduğunu garantiler.

Euler'in Königsberg Köprü Problemi'nin çözülmesinin imkansız olduğunu söylemesinden 256 yıl sonra matematikçi ve programcı olan Meigu Guan bu köprü problemine benzer bir problemi çözebilmiştir. Bu problem, bir postacının şehirdeki tüm sokaklara uğrayarak dağıtacağı mektupları sahiplerine ulaştırıp, başladığı noktaya en az mesafeyi yürüyerek dönmek istemesi ile ortaya çıkmıştır. Köprü problemindeki gibi her bir sokağa bir kez girilmesi yeterli ve gerekli olacağından, geçilen sokaklardan tekrar geçilmemesi istenmektedir. Ancak Euler'in de belirttiği gibi bu durum grafta tek dereceli düğümler var ise istenen bu durum mümkün olamayacağından, Guan bazı yollara sanal ek yollar eklenerek tam döngünün oluşturulabileceğini göstermiştir [49].



(a) Konigsberg Köprü Problemi (b) Köprü Probleminin Graf Yapısı

Şekil 2.5. Pregel Nehri Üzerindeki Konigsberg Köprü Probleminin Benzetimi *[50]*

Verilen sahada, robotun en düşük maliyetle gezebilmesi için bu sahada tüm ayrıtlardan en az bir kere geçmesi ve bu alanı tam olarak gezebilmesi için geçtiği ayrıtlardan tekrar geçme davranışını da minimum düzeyde gerçekleştirmesi gerekmektedir. Guan, bu probleme benzeyen Çinli Postacı Problemine önerdiği çözümde sahanın optimum maliyetle tam gezilebilmesinin, graf yapısında bulunan üzerinden tek sayıda ayrıtın çıktığı düğümleri (tek dereceli düğüm) birbirleriyle uygun bir şekilde eşleştirerek bu düğümler arasında bulunan ayrıtların tekrar edilmesi ile gerçekleşebileceğini göstermiştir [49]. Oluşabilecek tüm eşleşmeler metasezgisel olmayan yöntemler ile optimize edilmeye çalışıldığında, zaman karmaşıklığı $O(n!^2)$ olmaktadır. Bu zaman karmaşıklığı çok sayıdaki tek dereceli düğümlerin eşleşmelerinin metasezgisel olmayan yöntemler ile bulunmasını oldukça

güçleştirmektedir. Eşleşme problemi için daha önce metasezgisel yöntemlerden genetik algoritma önerilmiş ve başarılı sonuçlar elde edilmiştir [51].

Eşleşme problemi nümerik olmayan bir problemdir. Nümerik olmayan problemlerde Yapay Arı Kolonisi Algoritmasının da başarılı olduğunu gözlemlemiş [52], ve Karaboğa ve Baştürk ise yapay arı algoritmasının genetik algoritmaya göre daha başarılı sonuçlar elde ettiğini ortaya koymuştur [53].

3. MATERYAL VE METOTLAR

Literatür taramasında bahsi geçtiği gibi sürü robotların hedef bulma probleminin simülasyonu için, çözüm arama davranışlarının gerçek hayattaki arıların besin arama davranışlarına oldukça yakın modellenmesinden ötürü Yapay Arı Kolonisi algoritması, ve alanda en çok kullanılan, varyantları üretilen PSO algoritması seçilmiştir. Gezgin robotların gezinme maliyetlerinin optimizasyonu için de Guan'ın Çinli Postacı Problemine sunduğu çözüm simülasyonda uygulanmış, ve bu problem çözümünün eşleştirme adımı metasezgisel yöntemlerden olan Yapay Arı Kolonisi algoritması ile çözülmeye çalışılmıştır.

3.1. Yapay Arı Kolonisi Algoritması

Algoritmada işçi arılar, birbirlerinin komşuluklarında yeni çözüm arayan bireyleri, gözcü arılar, işçi arılardan aldıkları bilgilere göre yeni olası besin kaynaklarının yerini tahmin eden bireyleri, kaşif arılar ise hali hazırda çözüm üretmeye çalışan arıların komşuluğu dışında çözüm uzayında başka yerlerde rasgele arama yapan bireyleri temsil etmektedir. İşçi ve gözcü arıların davranışı algoritmanın lokal arama özelliğini yani yoğunlaşma bileşenini oluştururken, kaşif arıların rasgele arama davranışı ise algoritmanın global arama özelliğini yani çeşitleme bileşenini oluşturmaktadır. Günümüze kadar ABC algoritması için de değişiklikler önerilmiş ve bu değişikler ile ABC algoritmasının kullanım alanı araştırmacılar tarafından genişletilmiştir [32,33].

Arılar bölgedeki besin kaynaklarını bulmak için rasgele bir şekilde alana dağılırlar. Daha sonra bulunan kaynaklarda yiyecek miktarının azalması ile ya da diğer arılardan alınan daha kaliteli besin bölgesi bilgileri ile başka besin kaynaklarına yönelirler.

ABC algoritması ilk olarak önceden belirlenmiş popülasyon sayısına (SN) ulaşana kadar Eşitlik 3.1'e göre arama uzayında rasgele çözümler üretir [27]. Eşitlikteki x_i arıları (çözümleri), j ise çözümün parametrelerini belirtmektedir. Her bir arı için optimize edilmesi istenen problemin parametrelerine o parametrelerin minimum (x_{ij}^{min}) ve maksimum (x_j^{max}) değerleri arasında rasgele değerler atanarak arama uzayında rasgele pozisyonlarda çözümler (besin kaynakları) üretilmektedir [27].

$$x_{ij} = x_{ij}^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (3.1)$$

$$V_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (3.2)$$

İlk çözümler oluşturulduktan sonra *İşçi Arı Fazı (İAF)* başlar. Bu fazda tüm arılar bulundukları çözümün etrafında daha kaliteli bir çözümü Eşitlik 3.2 ile bulmaya çalışırlar. Denklemdeki V_{ij}, i arısının j. parametresi için yeni bir çözümü ifade eder. Yeni çözüm, o anki arının hali hazırdaki j. parametresinin sürüdeki rasgele bir arının yine j. parametresine (x_{kj}) bağlı oluşturduğu çözümdür. Buradaki φ_{ij} yeni çözümün ağırlık katsayısıdır ve bu katsayı her bir parametre için önceden tanımlanabileceği gibi sezgiselliği arttırmak adına 0-1 arasında rasgele de oluşturulabilir. Burada yeni çözüm üretilirken aynı anda rasgele seçilmiş olan j. parametre dışında başka bir parametrenin değiştirilmediğine dikkat edilmelidir.

Eşitlik 3.2 incelendiğinde, yeni çözümün iki arının j. parametrelerinin farklarının alınarak üretildiği görülmektedir. Bu durum, çözüme yaklaşıldıkça, arılar arasındaki mesafeler de küçüleceği için optimum nokta arama işleminin adaptif bir şekilde gerçekleştirileceğini göstermektedir. Bu da ABC algoritmasının yoğunlaşma bileşenini oluşturmaktadır.

Yeni çözüm üretildikten sonra bulunan çözümün uygunluğu (kalitesi) Eşitlik 3.3 ile hesaplanır [27]. Üretilen çözümün uygunluğu, önceki çözümün uygunluğundan daha iyi ise arının çözümü (pozisyonu) güncellenir (Açgözlü Seleksiyon – Greedy Selection). Ancak daha uygun bir çözüm üretilememiş ise bu arıların pozisyonları güncellenmez ve iyi çözüm üretememe sayaçları bir arttırılarak bir sonraki arı için çözüm üretme işlemine geçilir. Eğer yeni çözümün uygunluğu daha iyi ise, bu sayaç sıfırlanır.

$$fitness_i = \begin{cases} 1/(1+f_i), & f_i \geq 0 \\ 1 + abs(f_i), & f_i < 0 \end{cases} \quad (3.3)$$

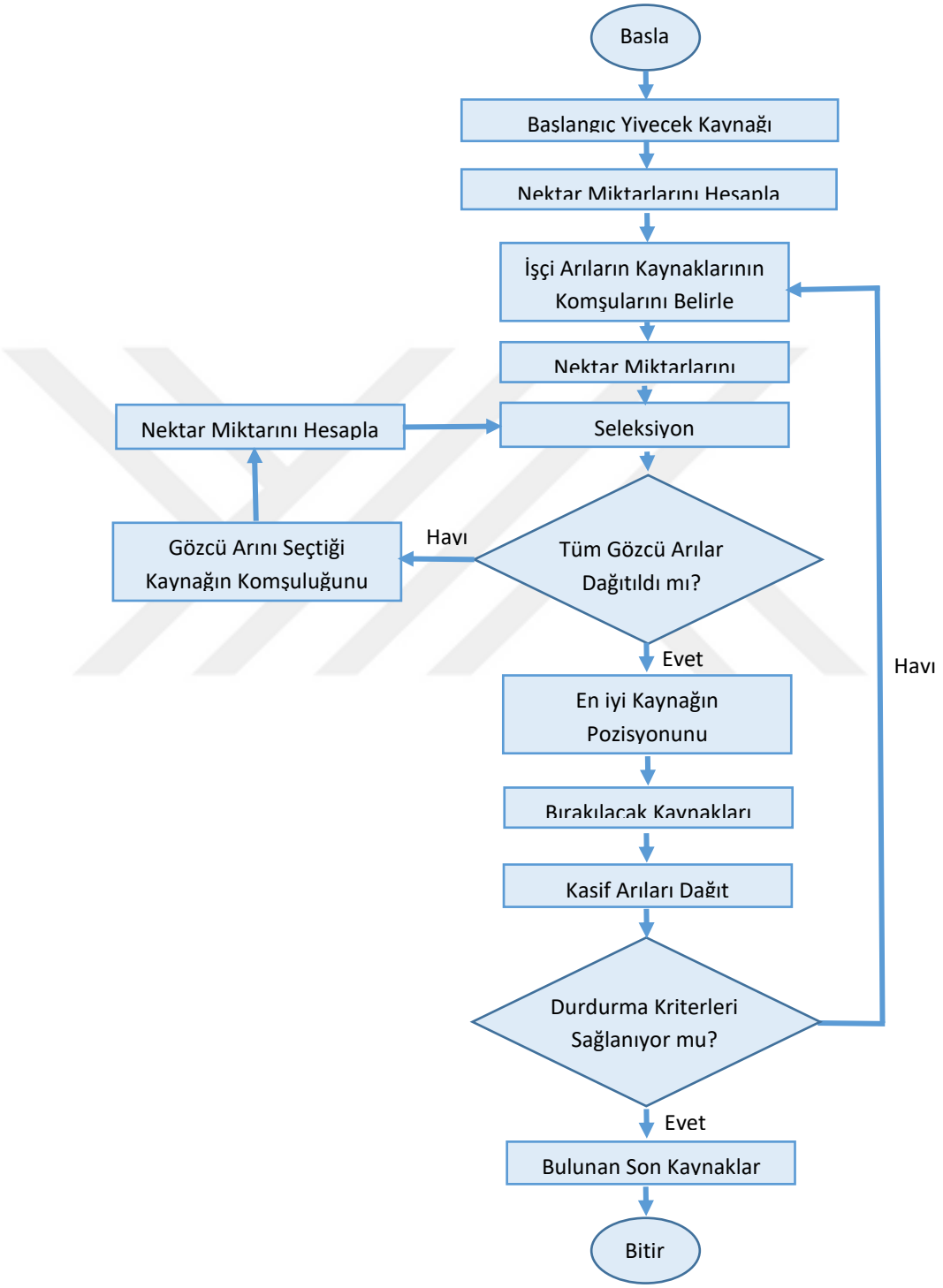
Çözümler hesaplanıp, uygunluklarına göre arıların pozisyonları güncellendikten sonra *Gözcü Arı Fazı (GAF)* başlar. Bu fazda gözcü arılar, bulunmuş olan çözümlerin uygunluk bilgilerini diğer arılara ileterek, arıların uygunluğu iyi olan çözümlerin etrafına gitmelerini, yeni çözümlerini bu arıların çözümleri etrafında üretmelerini sağlar. Bu uygun çözümün seçilme işleminde kullanılacak olan çözümün olasılık değeri (p_i) Eşitlik 3.4'deki gibi, bulunan her çözümün uygunluğunun, toplam uygunluğa bölünmesi ile hesaplanır [27].

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \tag{3.4}$$

Daha sonra arılar çözümlerin olasılık değerlerine orantılı bir olasılık ile dağılırlar. Karaboğa, bu olasılıksal seçilim için rulet tekerleği metodunu önermiştir. Rulet tekerleği yöntemine göre, tekerlekte her birey için o bireyin uygunluğuyla orantılı (Eşitlik 3.4) açıda bir dilim ayrılır. Daha sonra 0-1 arasında üretilen rasgele değişkene göre dilim seçilir. Uygunluğu yüksek olan bireyin dilim açısı daha yüksek olacağından, bu bireyin seçilme ihtimali de yüksek olmaktadır. Ardından arılar olasılıksal olarak dağıldıkları bu çözümlerin etrafında Eşitlik 3.2 ile tekrar yeni bir çözüm üretir. Bu çözümün uygunluğu daha iyi ise, arının yeni sonuç üretememe sayacı sıfırlanarak pozisyonu güncellenir. Yine daha uygun çözüm üretilemediğinde de o arının yeni çözüm üretememe sayacı bir arttırılır.

Çevrim tamamlanmadan önce her arının yeni çözüm üretememe sayacı kontrol edilir. Bir arının yeni çözüm bulma konusundaki başarısızlığı bu sayaç ile kontrol edilir. Eğer bu sayacın değeri yüksek ise o arının incelediği kaynağın etrafında yeni bir çözüm bulunmayabilir. Bu durumda o arının, komşuluğunda yeni bir çözüm bulmaya çalıştığı ama başarısız olduğu kaynağı terkedip alanda rasgele bir yerde yeni besin kaynakları keşfetmesi beklenmektedir. Bunun için yeni çözüm geliştirememe sayacına bir "limit" verilir. Çevrim sonunda herhangi bir arının yeni çözüm üretememe sayacı bu limiti aşmış ise, bu arı kaşif arı olarak görevlendirilir ve çözüm uzayında Eşitlik 3.1'e göre rasgele yeni bir pozisyonda arama işlemine devam eder. Bu faza *Kâşif Arı Fazı (KAF*) denir. Temel ABC algoritmasında her çevrim sonunda yalnızca bir kaşif arı oluşmasına müsaade edilir. Ancak optimize edilmesi istenen problem için bu parametre değiştirilebilir. Daha sonra global en iyi çözüm hafızaya alınarak tekrar İAF'ye dönülür.

Durma kriterleri sağlandığında ise algoritma global en iyi çözümü çıktı olarak verir. Algoritmanın pseudo (sözde) kodu Çizelge 3.1'deki gibi gösterilmiştir [54]. Algoritmanın akış diyagramı ise Şekil 3.1'de incelenebilir. [45].



Şekil 3.1. ABC Algoritmasının Akış Diyagramı

Çizelge 3.1 ABC Algoritmasının Pseudo Kodu

REPEAT
- *İşçi arıları besin kaynaklarına gönder ve bu kaynaklardaki nektar bilgisini ölç*
- *Gözcü arılar tarafından seçilebilmeleri için bu kaynakların olasılık değerlerini hesapla*
- *Gözcü arıları bu kaynaklara gönder*
- *Arıların nektar toplama işlemini tamamla*
- *Kâşif arıları rasgele bölgelere yeni besin kaynağı bulmaları için gönder*
- *En iyi besin kaynağını hafızaya al*

UNTIL *(gereksinimler tamamlanana kadar)*

3.1.1. Hızlı Yapay Arı Kolonisi Algoritması

ABC algoritmasında işçi ve gözcü arılar yeni çözümlerini Eşitlik 3.2 ile üretmektedirler. Ancak qABC algoritmasında gözcü arılar, işçi arılardan farklı bir şekilde Eşitlik 3.5'e göre yeni bir çözüm üretirler [32].

$$v_{N_m,i}^{best} = x_{N_m,i}^{best} + \varphi_{m,j}\left(x_{N_m,i}^{best} - x_{k,i}\right) \qquad (3.5)$$

Bu eşitlikteki $x_{N_m}^{best}$ xm komşuluğundaki en iyi çözümü ifade etmektedir. Komşuluk belirlenirken farklı algoritmalar kullanılabilir. Ancak temel qABC algoritması için Eşitlik 3.6 önerilmiştir [32].

$$md_m = \frac{\sum_{i=1}^{SN} d(m,j)}{SN-1} \qquad (3.6)$$

Eşitlik 3.6'daki j o an kontrol edilen arıyı, $d(m,$j$)$, j. arı ile gözcü arı arasındaki Öklid uzaklığını, md_m ise tüm uzaklıkların ortalamasını ifade etmektedir.

Çizelge 3.2 qABC algoritmasının Komşu Belirleme Şartı

if $d(m,j)r \times md_m$ ***then*** x_j ***is a neighbor of*** x_m, ***else not***

Eğer o an kontrol edilen arı, gözcü arıya ortalama uzaklığın önceden belirlenmiş bir menzil katsayısı (r) ile çarpımından daha yakın ise, o an kontrol edilen arı gözcü arının komşusudur denilebilir. Bu işlem Çizelge 3.2'de gösterilen pseudo code ile basit bir şekilde ifade edilebilir [32].

Komşular belirlendikten sonra da gözcü arılar yeni çözümlerini "komşularındaki" uygunluk değeri en iyi olan arıdan aldıkları bilgiye göre üretirler. Bu adımlar dışındaki tüm adımlar temel ABC algoritması ile aynıdır.

3.1.2. Modifiye Edilmiş Yapay Arı Kolonisi Algoritması

ABC algoritmasında gözcü arılar kaynakları seçme işlemini kaynakların uygunluk değeri ile doğru orantılı bir şekilde gerçekleştirmektedir. Bu durum en iyi sonucun her zaman ön planda olmasına, kâşif arıların ise öneminin düşmesine sebep olmuştur. mABC algoritmasında düşük kalitedeki kaynaklara da şans verilebilmesi için uygunluk fonksiyonu Eşitlik 3.7'deki gibi güncellenmiştir [33].

$$p_i = (1 + fit_i)\exp(-fit_i) \qquad (3.7)$$

ABC algoritmasında gözcü arılar gidilen çözümün komşuluğunda yeni çözüm bulurken tek parametre değiştirirken, mABC algoritmasında ise yeni çözümde ilk parametre Eşitlik 3.8'e göre, ilk parametreye ek olarak ikinci bir parametre de Eşitlik 3.9'a göre değiştirilmektedir. İkinci parametre değişikliğinde global minimum çözüme sahip bireyin parametresi de yeni çözüm hesabına katılmıştır. Böylece bulunan global minimum çözümden uzaklaşmadan kâşif arılardan da faydalanılması sağlanmıştır. Eşitlik 3.9'da x_{best} global minimumu, Eşitlik 3.8 ve 3.9'daki x_k ise gözcünün kendisinden farklı yeni bir arıyı temsil etmektedir [33].

$$V_{i,j} = x_{k,j} + \varphi_1(x_{k,j} - x_{m,j}) \qquad (3.8)$$

$$V_{i,n} = x_{best,n} + \varphi_1(x_{k,n} - x_{m,n}) \qquad (3.9)$$

3.2. Parçacık Sürü Optimizasyonu Algoritması

PSO algoritması istenen fonksiyonun çözümünü arama uzayında parçacık adı verilen aday çözümler ile aramaktadır. Her parçacık alanda zamana bağlı bir hız vektörü ile yörünge oluşturmaktadır. Bu parçacıkların yörüngesini etkileyen iki önemli bileşen vardır. Bunlardan ilki, sürüyü oluşturan parçacıklardan en iyi çözüme sahip olanı, ikincisi de, sürüdeki her parçacığın o ana kadar bulunduğu en iyi konumun çözümü olarak tanımlanabilir. Bir parçacık alanda rasgele gezinme işini gerçekleştirirken bu iki bileşenin etkilerine maruz kalarak hareketini

güncellemektedir. Algoritma çalışırken her anda, alanda bulunan her parçacığın kendi minimum sonuçları mevcuttur. Amaç, mevcut olan bu minimum sonuçlar arasından en iyisini bulmaktır. Eğer artık daha iyi bir çözüme sahip bir parçacığa rastlanmıyorsa veya algoritma istenen iterasyon sayısına ulaşmış ise, gezinme işi sonlandırılır ve sonuç olarak gezinme boyunca parçacıkların elde ettiği global minimum çıktı olarak elde edilir. PSO algoritmasının adımları Şekil 2.7'de özetlenmiştir.

PSO algoritması günümüze kadar bir çok problemin optimizasyonu için kullanılmış ve başka alanlarda da kullanılmak üzere bir çok varyantı türetilmiş başarılı bir optimizasyon metodudur.

PSO'da aday çözümler parçacıklar olarak temsil edilmektedir. Her parçacığın arama uzayındaki konumu, o ana kadar bulunabildiği en uygun konum (yerel en iyi çözüm) ve sahip olduğu hız bilgileri algoritmanın her anında mevcuttur. Yani popülasyondaki parçacık sayısı kadar yerel çözüm bulunmaktadır. Bu yerel çözümlerden de en iyi olanı global en iyi olarak hafızada tutulur ve sürünün bu yerel ve global en iyilere göre hareketi sağlanarak optimum sonuca ulaşmak hedeflenir.

Metodun ilk adımı parçacıkların arama uzayında ilk konumlarının ve hızlarının belirlenmesidir. Pozisyonlar ve hızlar algoritmanın başlatma aşamasında parametrelerin önceden belirlenmiş alt ve üst limitlerini aşmayacak şekilde rasgele oluşturulur. Daha sonra her parçacığın başlangıç pozisyonları, her parçacığın yerel en iyi çözümü olarak kaydedilir. Tüm bu yerel en iyi çözümlerden en uygun olanı global en iyi çözüm olarak kaydedilir. Temel PSO algoritmasında başlangıçtan sonraki her k. iterasyonda i. parçacığın hız vektörünün j. parametresi Eşitlik 3.10'a göre güncellenecektir [23].

$$v_{i,j}^{k+1} = v_{i,j}^{k} + c_1 rand_1(p_{ij}^{k} - x_{ij}^{k}) + c_2 rand_2(g_j^{k} - x_{ij}^{k}) \qquad (3.10)$$

Bu eşitlikteki $v_{i,j}^{k}$, parçacığın bir önceki iterasyondaki hız vektörünün j. parametre bileşenini ifade etmektedir. p_{ij}^{k}, hız vektörünün, güncellenecek parçacığın k. iterasyona kadar bulunmuş en uygun j. parametre değerini, g_j^{k} ise o parametrenin k. İterasyona kadar bulunmuş global minimum değerini ifade etmektedir. x_{ij}^{k} ise parçacığın bir önceki iterasyonda bulunduğu pozisyon vektörünün j. parametresini ifade etmektedir. $rand_1$ ve $rand_2$ 0-1 arasında rasgele bir değerdir. c_1 ve c_2 ise

önceden belirlenmiş öğrenme katsayılarıdır. c_1 katsayısı, parçacığın kişisel tecrübesine verdiği önemi temsil ederken c_2, parçacığın sürünün genel tecrübesine verdiği önemi temsil etmektedir ve probleme göre uyarlanabilirler.

1999 yılında Eberhart, Shi ile beraber yaptığı çalışmasında daha önce önerdiği Eşitlik 3.10'da gösterilen hız vektörü güncellemesi denklemini Eşitlik 3.11'deki gibi değiştirmiştir [55].

$$v_{i,j}^{k+1} = wv_{i,j}^{k} + c_1 rand_1(p_{ij}^{k} - x_{ij}^{k}) + c_2 rand_2(g_j^{k} - x_{ij}^{k}) \qquad (3.11)$$

Eski güncelleme metoduna ek olarak, parçacığın eski hızına w çarpanı eklenmiştir. w katsayısı, parçacığın ataletini, momentumunu temsil etmektedir ve bu katsayı parçacığın eski hızındaki değişimi kontrol etmektedir. w çarpanı büyük tutulduğunda parçacık daha çok global arama yapabilme yeteneğine kavuşurken, bu katsayının küçültülmesi, yerel aramanın daha detaylı yapılmasını sağlayacaktır. Hızlar güncellendikten sonra i. parçacığın pozisyonu Eşitlik 3.12'e göre güncellenmektedir [23].

$$x_{i,j}^{k+1} = x_{i,j}^{k} + v_{ij}^{k+1} \qquad (3.12)$$

Parçacıklar her iterasyonda kişisel ve sürüsel tecrübelerine, eski hızlarına bağlı olarak konumlarını güncellemeye devam ettikçe yerel minimum ve global minimum parametreleri de birbirine yaklaşacaktır. Bu sayede bir sonraki iterasyondaki hız vektörü güncellemesi daha minimal olacaktır. Parçacıkların çözüme yaklaştıkça toplanması, toplandıkça daha yavaş hareket etmeleri PSO algoritmasının kendi kendine bir kontrol mekanizması içerdiğini göstermektedir. Hız vektörlerinin çok büyümesi durumunda ise bir sonraki hız güncellemeleri parçacıkların alanda çok dağılmasına, sürüden ayrılmasına sebep olabilmektedir. Bu yüzden her parametre ve hız için bir alt ve üst sınır belirtilmeli, güncellemeler esnasında bu sınırların aşılmamasına dikkat edilmelidir. Aşıldığı takdirde parametreye değer öteleme, ya da momentumu kaybetmemek adına hız yönü ters çevrilerek ayna etkisi uygulanmalıdır.

PSO algoritmasının istenen problemi etkin bir şekilde optimize edebilmesi, öğrenme katsayılarının ve atalet katsayısının düzgün seçilmesi ile mümkün olacaktır. Eşitlik 3.11'de görüleceği üzere bir parçacığın hız vektörü üç farklı vektörün toplamı

ile elde edilmektedir. Vektörlerin katsayısı da o vektör parçasına verilecek önemi ifade etmektedir.

Kişisel tecrübeye verilen önemi ifade eden c_1 ve sürü tecrübesine verilen önemi ifade eden c_2 katsayılarının gereğinden küçük tutulması, parçacığın genel öğrenme yeteneğine ket vuracaktır. Gereğinden büyük tutulması ise sürünün lokal bir minimumda kilitlenmesine neden olabilir. Bu katsayılar genelde $c_1 = c_2 = 2$ olarak kullanılıp, algoritma tamamlanana kadar güncellenmemektedir [55]. Ancak farklı katsayılar kullanan çalışmalar da mevcuttur. Atalet katsayısı ise temel PSO algoritmasında bulunmamasına rağmen, 1999 yılında Eberhart tarafından önerilmesinden sonra yaygın bir şekilde kullanılmaya başlanmıştır. Bir önceki iterasyondaki hıza verilen değer, w, sürünün genel dolanma hareketini kontrol altında tutmaya yarar ve genellikle $w < 1$ olarak seçilir. Buradaki amaç program çalışmaya devam ettikçe hız değişimlerinin de küçülmesini sağlamaktır. $w > 1$ olarak da seçilebilir, ancak bu durum sürünün kararsız hale geçmesine sebep olacaktır. w katsayısı sabit bir değer olarak alınabileceği gibi diğer değişkenlere bağlı bir şekilde her iterasyonda da güncellenebilir. Demirtaş 2015 yılındaki çalışmasında w katsayısı için araştırmacılar tarafından önerilen değişik yaklaşımları incelemiştir. Bu yaklaşımlara örnek olarak Çizelge 3.3' te görülen fonksiyonlar incelenebilir.

Çizelge 3.3. Atalet Katsayısı için Önerilmiş Yaklaşımlar [56]

Strateji ismi	**Matematiksel Formül**
Sabit atalet ağırlığı	$w = c\,; c = 0.7$
Rassal atalet ağırlığı	$w = 0.5 + \frac{Rand()}{2}$
Doğrusal olarak azalan	$w_k = w_{max} - k\,\frac{w_{\max} - w_{min}}{iter_{max}}$
Global-yerel en iyi poz. kullanmak	$w_k = 1.1 - k * \frac{gbest_k}{pbest_k}$

Parçacık Sürü Optimizasyonu Algoritmasının pseudo kodu ve akış diyagramı, Çizelge 3.4 ve Şekil 3.2'de incelenebilir.

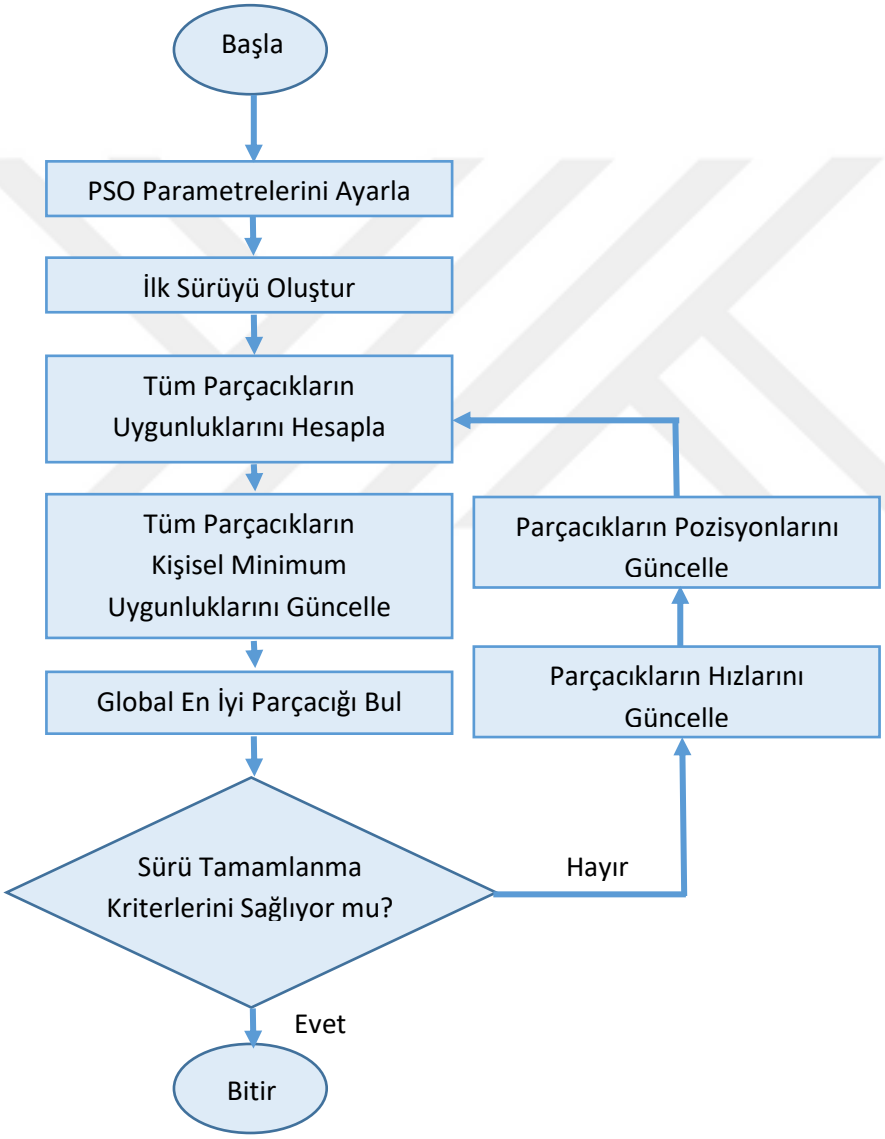
Çizelge 3.4 PSO Algoritmasının Pseudo Kodu

- *Parçacıkları Alana dağıt ve rasgele hız vektörleri ile hareketlerini başlat. En iyi çözümü global minimum olarak kaydet.*

REPEAT

- *Her parçacık için, yeni buldukları çözümler kendilerinin eski çözümlerinden daha iyi ise bu bireylerin kendi en iyi çözümlerini güncelle.*
- *Global minimumdan daha iyi bir sonuç bulunmuş ise, global minimumu güncelle.*
- *Her parçacığın hareket vektörünü, parçacığın kendi minimum çözümü ile sürüdeki en iyi parçacığın çözümüne bağlı bir şekilde güncelle ve harekete devam et.*

UNTIL *(gereksinimler tamamlanana kadar)*



Şekil 3.2. PSO Algoritmasının Akış Diyagramı

3.3. Çinli Postacı Problemi

Çinli Postacı Problemi için Guan'ın önerdiği çözümün adımları şu şekilde sıralanabilir [57];

- Öncelikle gezilecek graftaki tüm tek dereceli düğümler belirlenir.
- Tüm tek dereceli düğümlerin olası bütün eşleşmeleri (matching) yapılır.
- Her eşleşen düğüm çiftlerini birbirlerine bağlayan en kısa rotalar belirlenir.
- Tüm eşleşme serileri bulunduktan sonra, o eşleşme serisi içindeki ikililerin aralarındaki en kısa mesafelerin toplamı en az olan eşleşme seçilir.
- Bu eşleşme serisinin ikililerini birbirlerine bağlayan en kısa mesafelere yeni ayrıtlar eklenir.

Önerilen çözümün maliyeti, başlangıç grafında bulunan tüm ayrıtların uzunlukları toplamı ile yeni eklenen ayrıtların uzunlukları toplamıdır. V düğümler kümesini ve A ayrıtlar kümesini belirtmek üzere $G = (V, A)$ yönsüz grafında $e(i,j)$, $i.$ ve $j.$ düğümlerini birbirine bağlayan bir ayrıtı temsil etmektedir. x_{ij}, $e(i,j)$'den geçiş sayısını $c(i,j)$ ise bu ayrıtın uzunluğunu belirtmektedir. Problemin maliyet fonksiyonu Eşitlik 3.13'de verilmiştir. Bu denklemde amaç, postacının turunu tamamlaması için yürüyeceği $f(x)$ yolunu optimize etmektir [57].

$$f(x) = \sum_{i=0}^{n} \sum_{j=i}^{n} c_{i,j} x_{i,j} \tag{3.13}$$

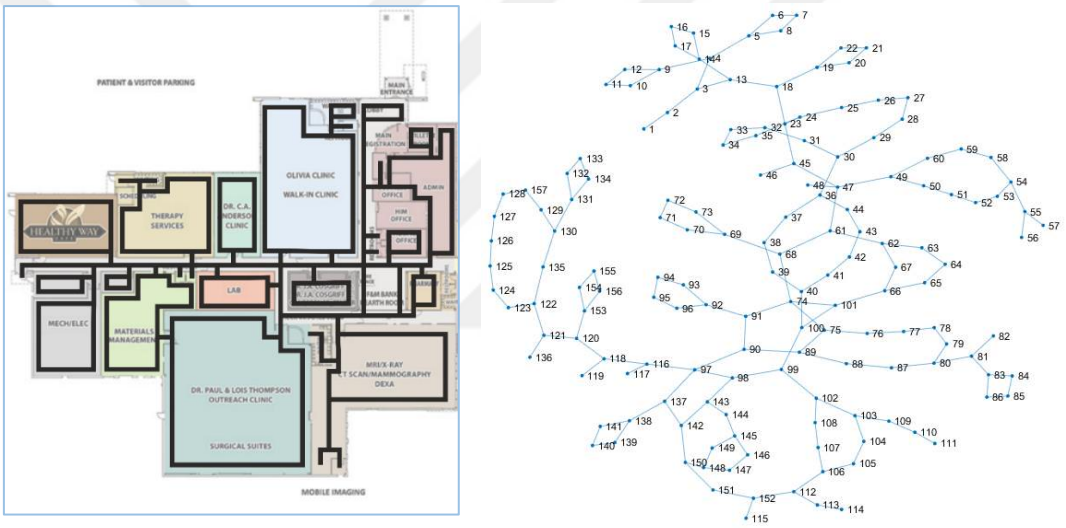
Bu problemin çözümü için geçiş sayısının doğal sayı (N) olması ve Eşitlik 3.14 ile Eşitlik 3.15'deki kısıtların sağlanmış olması gerekmektedir. Eşitlik 3.14'de $x_{i,j}$ geçiş sayısının en az bir olması gerekliliğidir. Eşitlik 3.15'deki kısıt ise başlangıç noktasından uzaklaşılan her bir hareket için başlangıç noktasına yaklaşılan bir ters hareket olması gerekliliğidir [57].

$$x_{i,j} + x_{j,i} \geq 1 \tag{3.14}$$

$$\sum_{i:(i,j)\in A} x_{i,j} = \sum_{i:(i,j)\in A} x_{j,i} \tag{3.15}$$

4. YAPAY ARI KOLONİSİ ALGORİTMASI İLE GEZGİN ROBOTUN HARİTALANMIŞ SAHADA GEZİNME MALİYETİNİN OPTİMİZASYONU UYGULAMASI

Bu uygulamada çalıştırılmak üzere Şekil 4.1 (a)'da görülen *"RC Hospital & Clinics"* binasının klinik servislerinin planı, varsayımsal gezgin temizlik robotumuzun çalışma sahası olarak düşünülmüştür. Temizlik robotunun tesisin tüm odalarını ve koridorlarını dolaşıp temizlemesi istenmektedir. Robot bir odaya girdiğinde, Şekil 1'de koyu siyah çizgiler ile ifade edildiği gibi odanın iç çeperinden bir tur attığı zaman temizliğin o oda için tamamlandığı kabul edilmiştir. Koridor temizliği ise robotun koridordan geçmesi ile tamamlanmaktadır. Çalışmaya, istenen çözümün daha sade bir şekilde uyarlanabilmesi adına, tesisin boş ve insansız olduğu kabul edilmiştir.



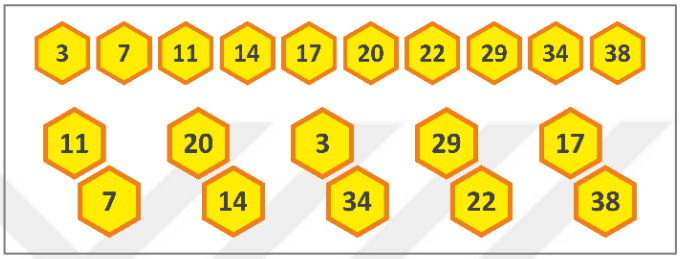
(a) Hastanenin Kat Planı (b) Planın .Graf Dönüşümü

Şekil 4.1. RC Hospital & Clinics Bina Planı ve Grafı

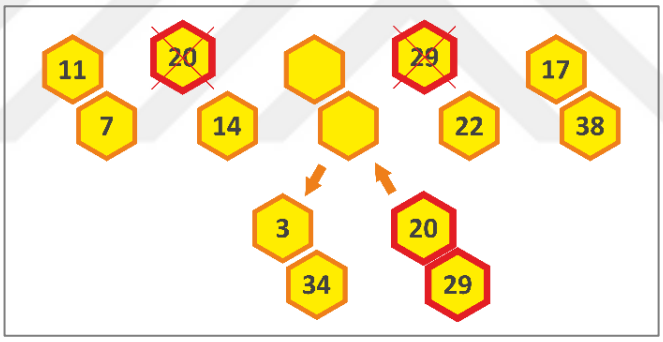
Çalışılacak olan saha, robotun gezeceği Şekil 4.1 (a)' da görüldüğü gibi tasarlanmış, Şekil 4.1 (b)' de bu tasarımın grafı paylaşılmıştır. Bu grafın komşuluk matrisi her köşeye bir id atanarak oluşturulmuştur.

Yapay Arı Kolonisi Algoritmasının Çinli Postacı Problemi üzerinde kullanılabilmesi için algoritmanın parametre değiştirme özelliği problemimize uyarlanmıştır. Arıların çözüm uzayında rasgele dağılımları, tek dereceli noktalar kalmayana kadar yapılan ikili kombinasyonlar ile ifade edilmiştir. Bu şekilde oluşturulan bir arı örneğini Şekil 4.2 (a)'da görmek mümkündür. Arının sahip olduğu

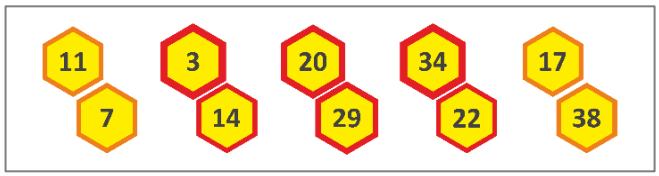
her çift ayrı bir parametreyi ifade etmektedir. Şekil 4.2 (a)'daki örnek arının 5 parametreli olduğu söylenebilir. Değişecek olan parametre Şekil 4.2 (b)'deki gibi rasgele seçilmektedir. Bu parametre yerine önerilen yeni çift, eski çift ile yer değiştirirken, diğer çiftlerin de bu değişikliklerden etkilenmesi olasıdır. Algoritmanın orjinal halinde parametre değişikliği diğer parametrelerden bağımsızdır. Ancak bizim problemimizde bir çiftin değişmesi iki farklı çiftin de değişebilmesini gerektirebilir. Güncellenecek çiftin değiştirilmesinden ötürü farklı iki çiftin daha değiştirilmesi gerektiği örnek bir durum Şekil 4.2 (c)'de gösterilmiştir.



(a) Tek Dereceli Düğümlerden Oluşmuş Bir İşçi Arı



(b) Seçilmiş Bir Çiftin Yeni Çift ile Yer Değiştirmesi



(c) Çiftler Değişiminin Diğer Çiftler Üzerimdeki Etkisi

Şekil 4.2. Eşleştirme ve Yeni Çözüm Üretme İşleminin Arılar Üzerindeki Tasarımı

Algoritmanın uygunluk değerini hesaplaması için gerekli maliyet fonksiyonu Eşitlik 4.1 ile ifade edilmiştir. Bu eşitlikte n graftaki düğüm sayısını, $s_{i,j}$ her iki düğüm arasındaki ayrıt sayısını, $d(i,j)$ ise i. ve j. düğümler arasındaki ayrıtın uzunluğunu ifade etmektedir. Eğer grafta tek dereceli en az bir çift nokta varsa maliyet kesinlikle

ilk graftaki ayrıtların uzunluğu toplamından fazla olacaktır. Temizlik robotu eşleşmelerden sonra tüm ayrıtlardan geçeceği için toplam maliyet fonksiyonuna graftaki tüm ayrıtlar eklenmiştir. Ancak çözüm için yalnızca eklenen ayrıtların uzunluğu toplamı da maliyet fonksiyonu olarak kullanılabilir.

$$f(x) = \sum_{i=1}^{n} \sum_{j=i}^{n} s_{i,j} * d(i,j) \tag{4.1}$$

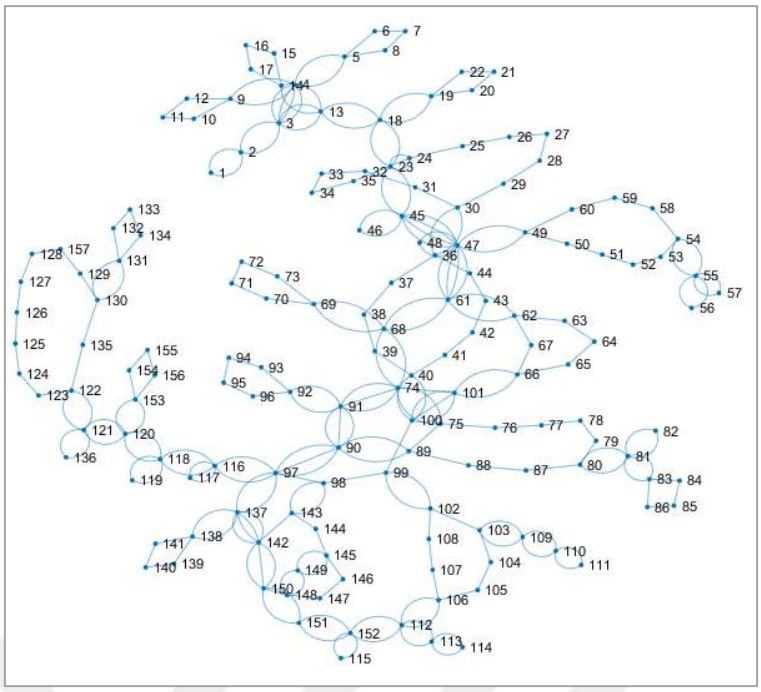
Bu değişiklikler ve aşağıdaki başlangıç değerleri ile çözüm bulunmaya çalışılmıştır.

Başlangıç Değerleri;

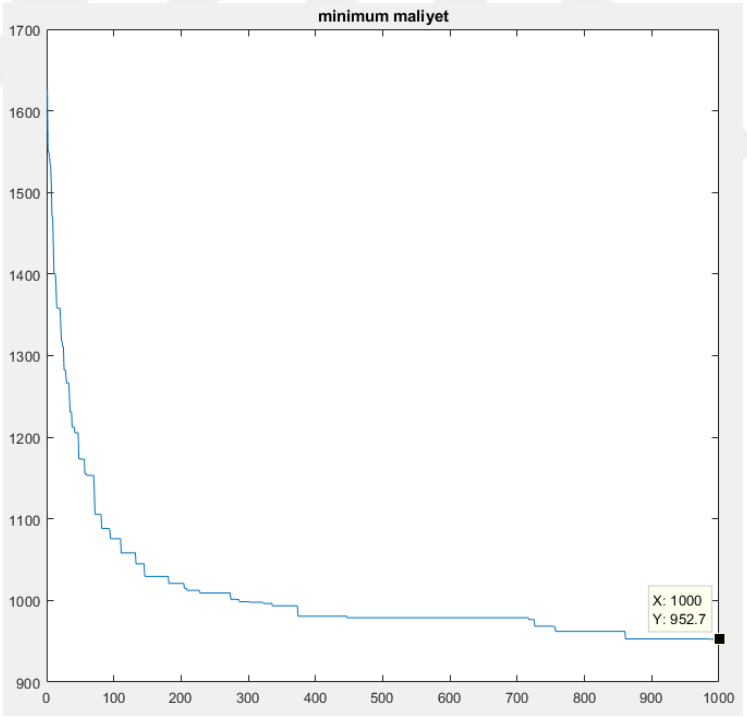
- Maksimum iterasyon: 1000
- Populasyon: 100
- Deneme limiti: 50
- Başlangıç maliyeti: 742 metre

Uygulama sonucunda eşleşecek olan düğümler ve bu düğümler arasındaki en kısa ayrıtlar çiftlenmiş şekilde sonuç grafı elde edilmiş ve Şekil 4.3'de paylaşılmıştır. Şekil 4.4'deki grafikte tam tur maliyetinin iterasyona bağlı değişimi görülmektedir. Çözüm grafında tüm düğümlerin çift olması bu grafta her ayrıttan bir kez geçilerek üretilen en az bir tur olacağını göstermektedir. Oluşabilecek her tur tüm ayrıtlardan sadece bir kere geçeceği için, oluşturulabilecek tüm turların maliyetinin aynı olması beklenmektedir.

Uygulama sonucunda bir gezgin robotun haritalanmış sahadaki tam gezinimi optimize edilmeye çalışılmış ve çıktı olarak üretilen çiftlenmiş düğümlere sahip graf incelendiğinde optimizasyon sonucunun oldukça tutarlı sonuçlar ürettiği gözlemlenmiştir. Çalışmanın devamında geliştirdiğimiz uygulama tarafından çıktı olarak verilen graf üzerinde euler turları oluşturma uygulamaları geliştirilebilir.



Şekil 4.3. Girdi Grafın Eularian Çıktısı



Şekil 4.4 Maliyetin İterasyona Bağlı Değişimi

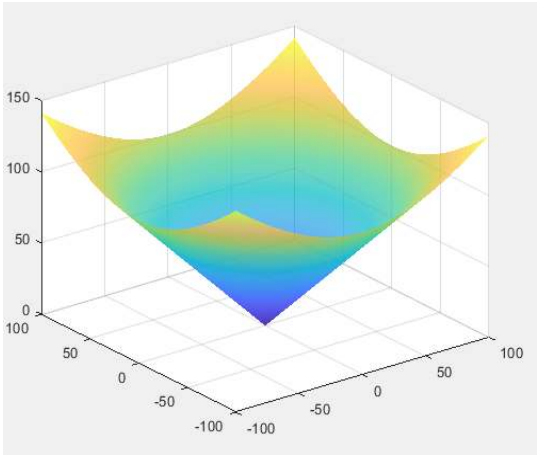
5. SÜRÜ ROBOTLARIN HEDEF BULMA PROBLEMİ İÇİN SÜRÜ TABANLI METASEZGİSEL YÖNTEMLERİN UYGULANABİLECEĞİ SİMÜLASYON PROGRAMI

Geliştirilen simülasyon uygulamasında kullanılacak olan ABC, mABC,qABC ve PSO algoritmaları, sürü robotların gerçek dünyadaki hareketlerinin daha iyi benzetiminin yapılabilmesi için modifiye edilmiştir. Bu modifikasyonlar algoritmaların çalışma prensiplerini etkilemeyen, sadece arıların fiziksel hız limitlerini ve hareket yeteneklerini göz önünde bulunduran değişikliklerdir. Örneğin Eşitlik 3.12 incelendiğinde, PSO için zaman yani t parametresinin 1 olarak alındığı görülmektedir. Popülasyondaki tüm bileşenler aynı anda hızları kadar yol gitmektedir. Ancak gerçek dünyada bu metodları uygulayabilmek için cisimlerin maksimum hızlarının da dikkate alınması gerekmektedir. Bunun için PSO'daki hız limitleri, robotlar için gerçek üstü olmayacak bir değere sabitlenmiştir.

Sürü robot uygulamalarında, ses, koku, sinyal şiddeti gibi değerler cisme yaklaştıkça artan, uzaklaştıkça azalan biçimde olduğundan basitçe benzetiminin yapılması için maliyet fonksiyonu Eşitlik 5.1'de gösterilen Öklid uzaklığı olarak kullanılmıştır. Burada b bireyleri, k ise kaynağı temsil etmektedir. Eşitlik 5.1 ve Öklid uzaklığının hedef nokta $0,0$ iken x, y düzlemindeki dağılımını gösteren Şekil 5.1'deki grafik incelendiğinde Öklid uzaklığının negatif olamayacağı görülmektedir. Bu yüzden Eşitlik 3.3'te ABC için önerilen uygunluk fonksiyonu Eşitlik 5.2'deki gibi güncellenmiştir.

$$uzaklik\ (b,k) = \sqrt{(b_x - k_x)^2 + (b_y - k_y)^2} \tag{5.1}$$

$$fitness_i = 1/(1 + fi) \tag{5.2}$$

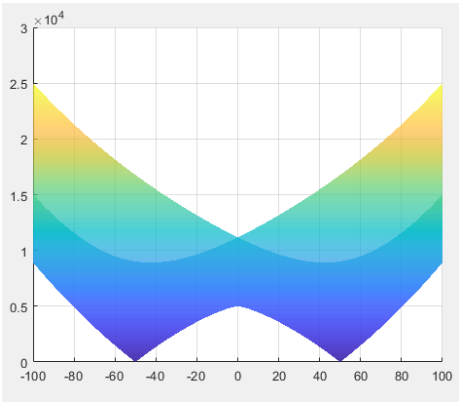


Şekil 5.1 Maliyet Fonksiyonunun X Y Parametrelerine Göre Değişimi

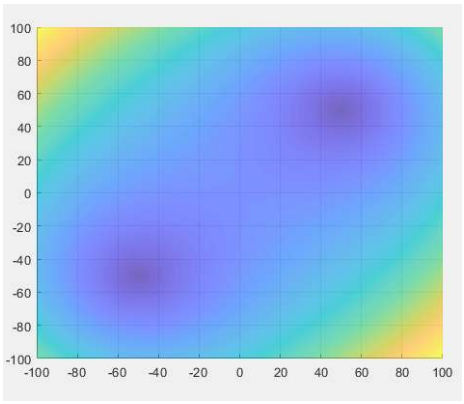
Sürü robotlar için değiştirilebilecek parametreler robotun uzaydaki konum vektörünü oluşturmaktadır. ($[x\ y\ z]$) Parametreler ortak bir özelliği temsil ettiği için, simülasyon, temel ABC algoritmasındaki tek parametre değiştirme özelliğine ek ikinci bir parametreye tercihen müsaade etmektedir. ABC algoritması arıların güncellenecek pozisyonlarına ışınlanıyor olduklarını kabul etmiştir. Bu ışınlanma, hareket fonksiyonları ile gerçekçi bir hale getirilmiştir.

$$maliyet\ (K) = \prod_{i=1}^{n} uzaklik(birey, K_i) + P_i \qquad (5.3)$$

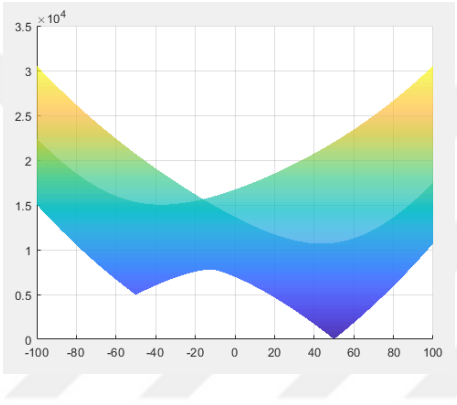
Çoklu kaynakların incelenebilmesi için, birden fazla kaynağın alana yaydığı sinyallerin benzetimi yapılmak istenmiştir. Bunun için Eşitlik 5.3'deki denklem önerilmiştir. Eşitlikteki K kaynaklar dizisini temsil etmektedir. $uzaklık$ fonksiyonu ise Eşitlik 5.1'deki öklid uzaklığıdır. Eşitlik 5.3 incelendiği zaman, bireylerin tüm hedeflere olan uzaklıkları çarpımı olduğu görülmektedir. Denklemdeki P terimi ise, arttırıldığında kaynağın yaydığı sinyali düşürmektedir. Bu da sistemde özdeş olmayan hedeflerin bulunabilirliğinin kontrolünü sağlamaktadır. p teriminin 0, hedeflerin pozisyon vektörünün $x_1 = [-50\ -50]$ ve $x_2 = [50\ 50]$ olduğu durumda maliyet fonksiyonunun alandaki dağılımı Şekil 5.2 (a) ve (b)'de gösterilmiştir. x_2 için $p = 35$ seçildiğinde ise, x_2'nin global minimum yerine lokal minimuma dönüştüğü Şekil 5.2 (c) ve (d)'de görülmektedir.



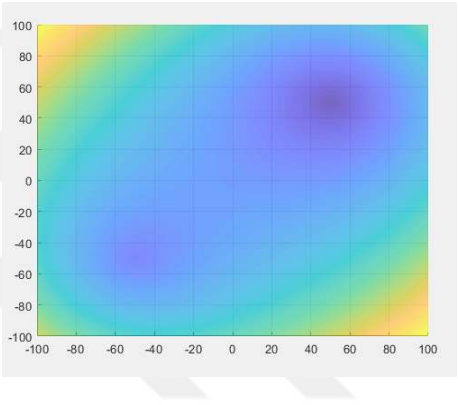
(a) Eş iki Kaynağın X-Z düzleminde Maliyet Grafiği



(b) Eş iki Kaynağın X-Y düzleminde Maliyet Grafiği



(c) Farklı İki Kaynağın X-Z düzleminde Maliyet Grafiği



(d) Farklı İki Kaynağın X-Y düzleminde Maliyet Grafiği

Şekil 5.2. Eş İki Kaynak ve Farklı İki Kaynağın Maliyetleri

5.1. Simülasyon Ortamı

Simülasyon teorik veya fiziksel bir sistemin bilgisayar ortamında modellenerek farklı şartlar altında test edilen sistemin verdiği tepkilerin incelenebilindiği yazılımlardır denilebilir. Bu yazılımlarda her bir parametrenin kontrol altında olması veya ölçülebilir olması beklenmektedir.

Sürü robotlarda hedef bulma probleminin metasezgisel yöntemler ile çözümünün benzetimi Unity 3D oyun motorunda C# dili ile yazılmıştır. Unity 3D, kendi içinde tanımlı kontrol edilebilen fizik parametreleri, fonksiyonları içeren geniş kapsamlı ve çapraz platformlu bir oyun motorudur. Bu özellikleri sayesinde, istenen platformda program geliştirilmesine olanak sağlamaktadır. İçerdiği fizik fonksiyonları sayesinde de, yer çekimi, sürtünme, çarpışma kontrolleri, kuvvetler birleşimi gibi

kodlaması ve kontrol edilmesi oldukça zor olan bileşenleri kolayca yönetilmesini sağlamaktadır. Bu paremetrelerin kontrol altında tutulması simülasyonun gerçeğe yakınlığı için çok önemlidir. Unity 3D daha önce de bilimsel çalışmalarda simülasyon amaçlı kullanılmıştır [58]. Unity 3D ortamında kullanılan nesnelere katı madde özellikleri kazandırılarak, bu nesnelerin birbirleri ile çarpışma kontrolü, çarpma noktası kontrolü, yüzeylerinin sürtünme etkileri, yer çekimi kuvvetleri gibi bir çok fiziksel olay, basit komutlar ile kontrol edilmektedir. Ayrıca cisimler için scriptler yazılarak bu cisimlere istenen yönde kuvvet uygulanabilmekte, hız, açısal hız, boyut, uzaydaki konumu gibi bir çok özelliğine erişim sağlanarak kontrol edilebilmektedir. Bu özellikler genel olarak düşünüldüğünde, simülasyon için yeterli bir ortam olduğu görülmüştür. Simülasyon sonuçları ise, MATLAB ortamında incelenmiş ve grafikler oluşturulup karşılaştırma işlemi yapılabilmiştir. Bu sonuçlar uygulama sonuçları başlığı altında detaylı bir şekilde incelenmiştir.

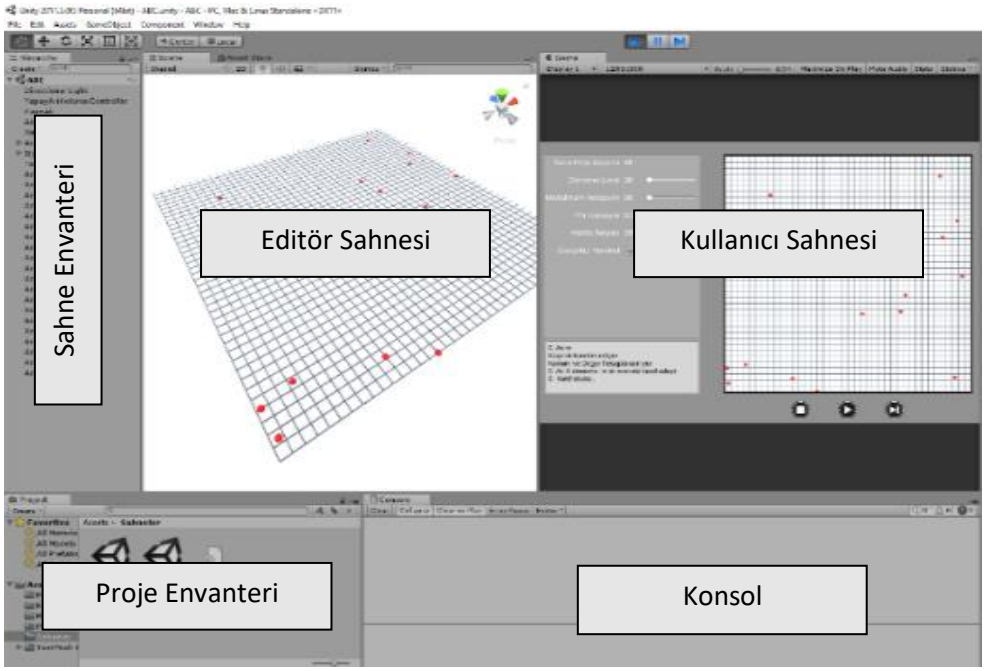
Unity 3D oyun motorunun tasarım arayüzü Şekil 5.3'de görülmektedir. Varsayılan ayarlarda, uzaydaki tüm nesneler arayüzün sol tarafındaki panelde incelenebilmekte, üst sekmelerden yeni nesneler eklenip düzenlenebilmektedir. Sol alt panelde projenin klasör hiyerarşisini görüntüleyip düzenlemek mümkündür. Sağ tarafta seçilen nesnenin özellikleri, kendisine ait scriptin tüm değişkenleri görüntülenip, erişime açık değişkenler güncellenebilmektedir. Orta panelde ise uzayı tasarımcının gözünden görmek mümkündür, orta panelin ikinci sekmesinde ise uzayın son kullanıcı gözünden nasıl görüleceğini görüntülemek mümkündür.

Arama uzayı Şekil 5.4'da görüldüğü üzere 2 boyutlu bir düzlem üzerinde sürü robotların hareketlerinin daha net görülebilmesi için ızgaralanmıştır. Robotlar kırmızı küreler, bulunması gereken hedef ise yeşil bir küre ile temsil edilmiştir. Statik olmayan kaynak ve robotlara metasezgisel yöntemlerin uygulandığı scriptler eklenmiştir. Şekil 5.4'te sağ üstte görülen pusula, tasarımcıya rehberlik etmesi için kırmızı, yeşil ve mavi renkler ile sırayla x,y ve z eksenlerinin yönünü göstermektedir.

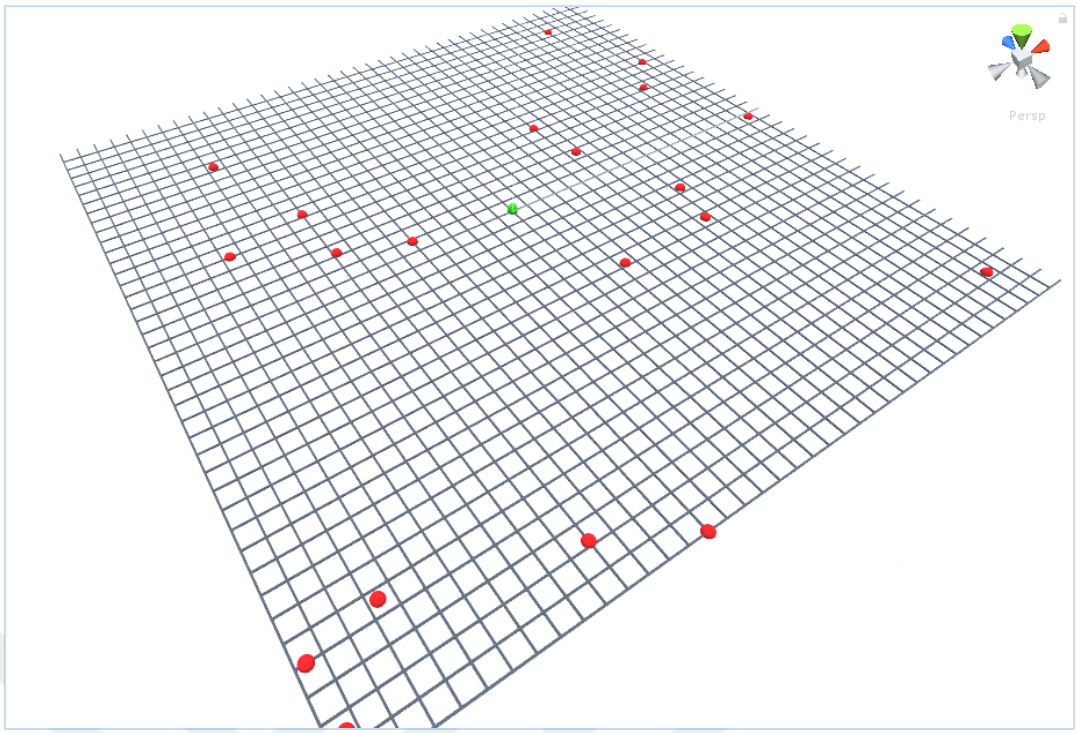
Çıktı olarak üretilen programda ise kullanıcılar seçtikleri algoritmanın simülasyonu için gereken başlangıç değerlerini Şekil 5.6, ve 5.7'de görülen arayüzün sol tarafındaki panelden ayarlayabilmektedirler. Bu panel program başladığında seçilmiş olan metasezgisel yönteme göre güncellenmektedir. Sahnenin altında bulunan

başlat, durdur ve adım adım ilerle simgeleri ile de simülasyonlarını adım adım kontrol edebilmektedirler. İstenirse simülasyon duraklatılarak o iterasyondaki konumlar ve değerler incelenebilmektedir. Simülasyonun anlık sonuçları, sol alt taraftaki konsol panelinden izlenebilmektedir. Simülasyon boyunca seçilen metasezgisel yönteme göre, önceden belirlenmiş bazı değişkenlerin aktüel olarak değiştirilmesi mümkündür. Örnek vermek gerekirse, simülasyon 1000 iterasyon ile başlatılsa dahi, bu değer 500'e düşürülebilir. Ya da öğrenme katsayıları sürünün hareketini daha rahat inceleyebilmek için, anlık olarak değiştirilebilir şekilde programlanmıştır.

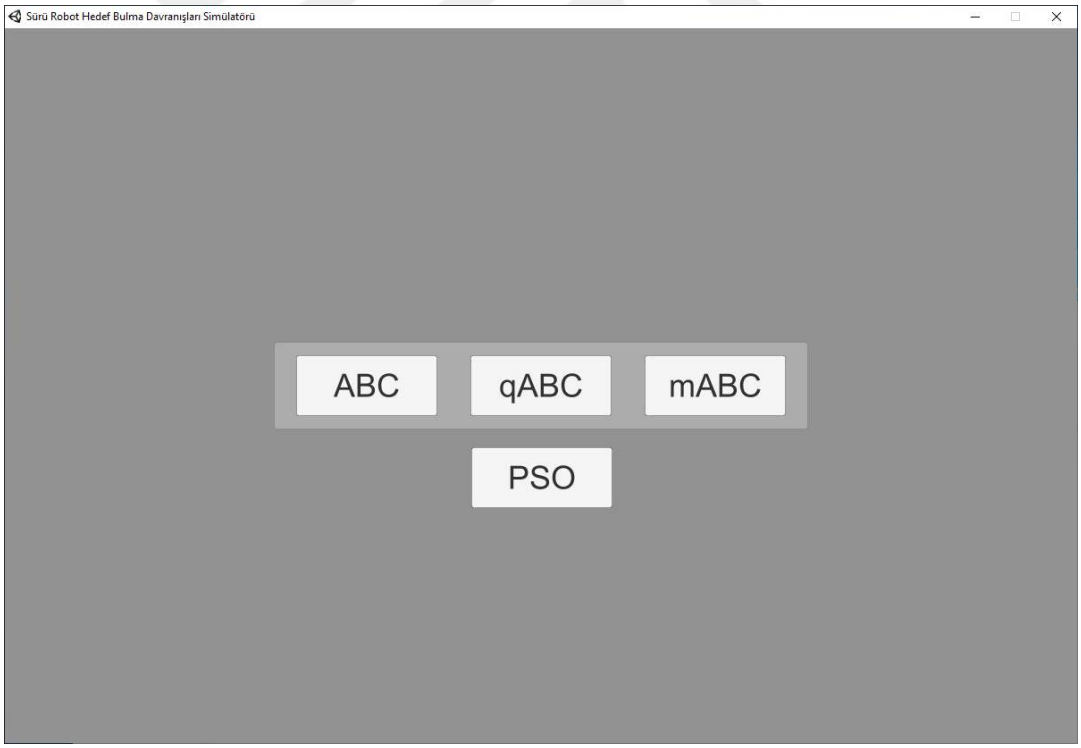
Simülasyon tamamlandığında kullanıcının, verilerini analiz edebilmesi için her iterasyondaki global minimuma sahip robotun maliyet, uygunluk, pozisyon vektörü gibi özellikleri bir text dosyasında matris olarak sunulmaktadır. Kullanıcı isterse bu matrisi MATLAB gibi matris işleme programlarında işleyerek simülasyon sonuçlarını daha detaylı bir şekilde inceleyebilir. Ayrıca kullanıcı, simülasyonu hızlandırmak için robotların gerçekçi hareketini iptal edebilme ya da robotları hızlandırma yeteneğine sahiptir. Hedefin dinamikliğini kontrol etmek isteyen kullanıcı klavyesindeki yön tuşlarını kullanarak hedefin pozisyonunu anlık olarak değiştirebilmektedir. Simülasyon programında materyal ve metotlar başlığında incelenmiş metasezgisel algoritmaların hepsinin simülasyonları ayrı ayrı gerçekleştirilebilmektedir.



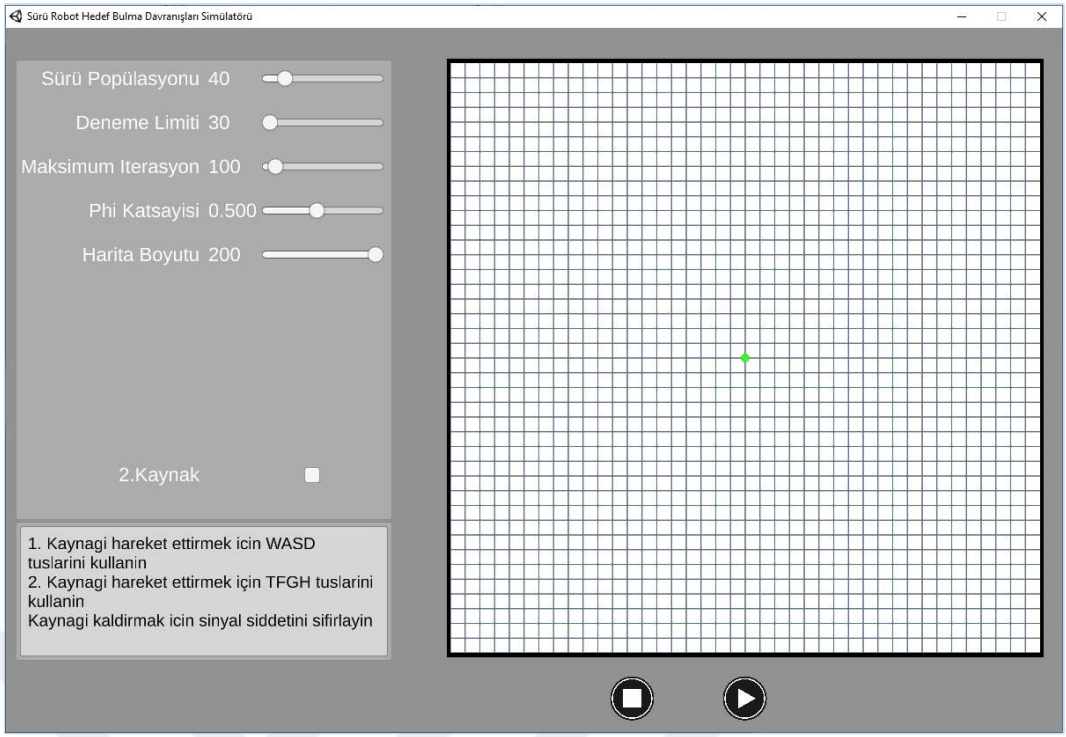
Şekil 5.3. Unity 3D Arayüzü



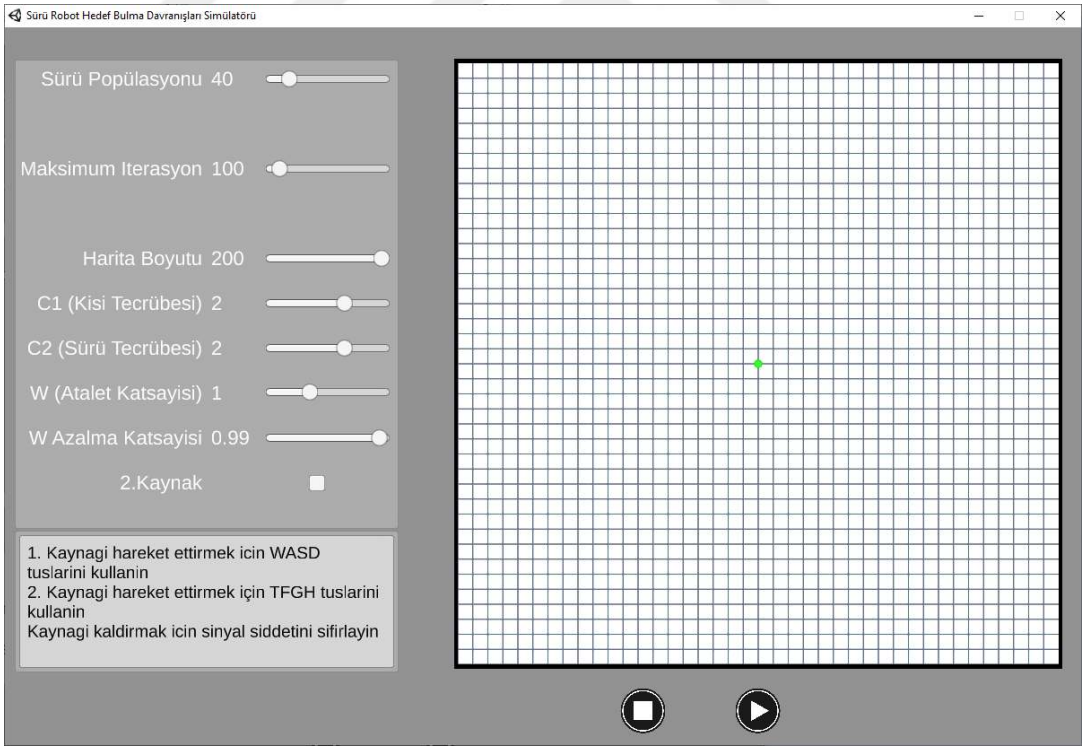
Şekil 5.4. Simülasyon Ortamı



Şekil 5.5. Simülasyon Uygulamasının Arayüzü



Şekil 5.6. ABC için Ayar Paneli ve Sahnesi



Şekil 5.7. PSO için Ayar Paneli ve Sahnesi

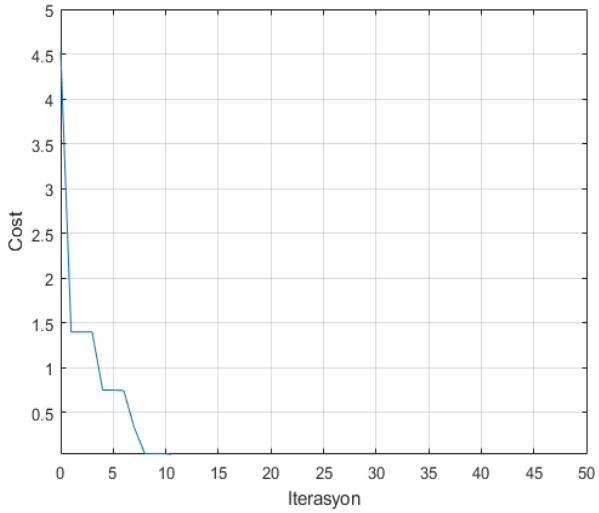
5.2. Tek Hedef Senaryosu İçin Simülasyon Sonuçları

Sürü robotlar ile kullanılabilmesi için adapte edilen ABC algoritmasının simülasyonu için aşağıdaki başlangıç değerleri kullanılmıştır.

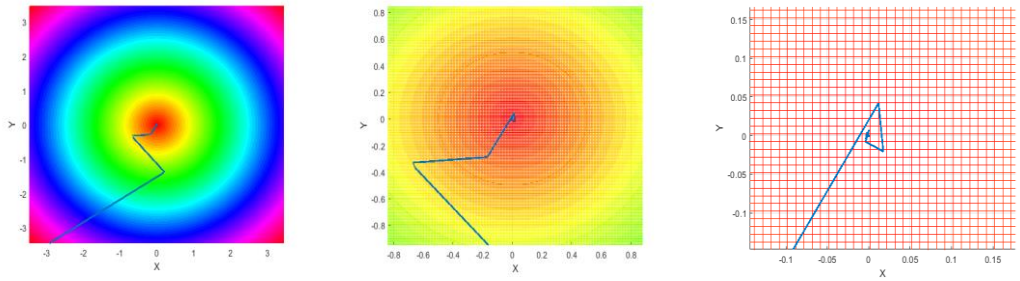
Başlangıç değerleri;

- *Sürü Popülasyonu: 40*
- *Maksimum İterasyon: 100*
- *Harita Boyutu: 200 (Parametreler [-100,100] aralığında)*
- *Deneme Sayısı: 30*
- *Phi Katsayısı: 0.5 (Adım çarpanı [-1,1] aralığında)*
- *Kaynak Konumu: [X:0, Y:0]*
- *Gerçekçi Hareket: Kapalı*
- *Arı Hızı: İnaktif*

İlk çalıştırılan simülasyonun sonuçları Şekil 5.18'de paylaşılmıştır. Grafikte her iterasyondaki popülasyonda bulunan global minimuma sahip robotun maliyet değerinin iterasyona bağlı değişimi görülmektedir. Yapılan simülasyon sonucunda robotların hedefe oldukça hızlı yaklaşabildiği gözlemlenmiştir.



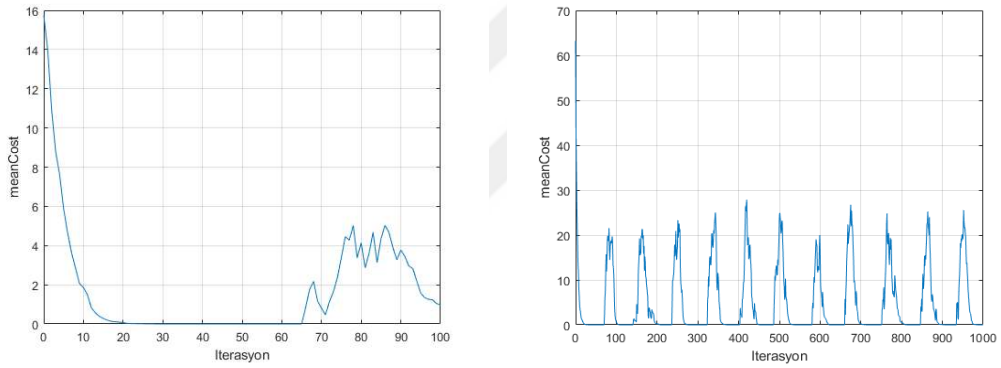
Şekil 5.8. Maliyetin İterasyona Bağlı Değişim Grafiği



(a) 1x Yaklaştırılmış (b) 4x Yaklaştırılmış (c) 20x Yaklaştırılmış

Şekil 5.9. Robotların Hedefe Yaklaşmaları

Her iterasyondaki global minimuma sahip robotların hedefe $(x = 0,\ y = 0)$ yaklaşması x ve y parametrelerinin değişimi ile Şekil 5.9'da görselleştirilmiştir. Yaklaşma hareketinin anlaşılabilmesi için grafik kademeli olarak büyütülmüştür. Grafik üzerindeki rotadaki her köşe, bir sonraki iterasyondaki global minimum çözüme sahip robotun pozisyonunu göstermektedir.



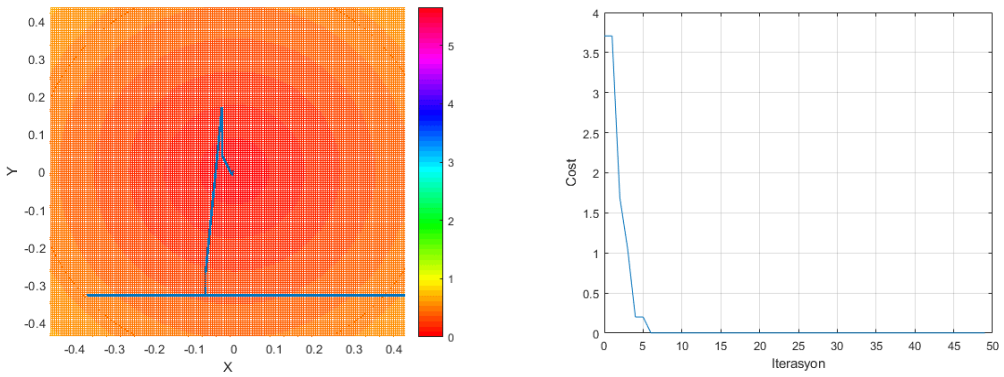
(a) Ortalama Maliyetin 100 adımdaki Değişimi (b) Ortalama Maliyetin 1000 adımdaki Değişimi

Şekil 5.10.Ortalama Maliyetin Farklı Adım Sayılarındaki Değişimi

Şekil 5.10 (a)'daki grafikte her iterasyonda sürüdeki tüm robotların maliyetlerinin ortalaması görülmektedir. 65. iterasyonda ortalama maliyet yükselmeye başlamaktadır. Bu bölgede, gözcü robotlar daha iyi sonuç bulamadıkları için kâşif robota dönüşüp alanda rasgele bir yere gitmeye başlamıştır. Bu da ortalama maliyeti arttırmıştır. Bu fenomen iterasyon sayısı Şekil 5.10 (b)'de görülen grafikteki gibi arttırıldığında daha rahat gözlemlenebilmektedir. Kaşif robotlar lokal minimumda kitlenmeyi önleyen en önemli faktördür, ve ABC algoritmasının global arama özelliğini oluşturur.

Simülasyon ortamında, hedef sabit ve her alana sinyal yayabilen bir cisim olarak tasarlandığından ABC algoritmasının kaşif robot davranışı çözüme yaklaşmada olumlu bir etki sergilememektedir. Ancak kaynaktan yayılan sinyal dalgalı, ya da başka etmenlerden kaynaklı dağınık olduğu takdirde (örneğin kokulu bir cismin yaydığı koku partiküllerinin rüzgardan etkilenmesi) kaşif robotlar sürünün lokal minimumlarda kitlenmesini engelleyecektir. Kaşif robotların tek ve sabit hedef için katkısı olmasa da, dinamik hedef ve çoklu hedeflerde kritik bir önemi vardır. Eğer robotlar hedefe çok yaklaştıkları zaman, hedef uzak bir noktaya hareket ederse, robotlar hedefin yeni konumuna yaklaşmak yerine hedefe en yakın robotun etrafında gezineceklerdir. Kâşif robotun bu olumsuz etkiyi ortadan kaldırmada önemli bir rol oynayabildiği gözlemlenmiştir. Çoklu hedef senaryosunda tek bir hedefe odaklanan robotları diğer hedeflere yönlendirme işi de kâşif robotlar sayesinde sağlanabilmektedir.

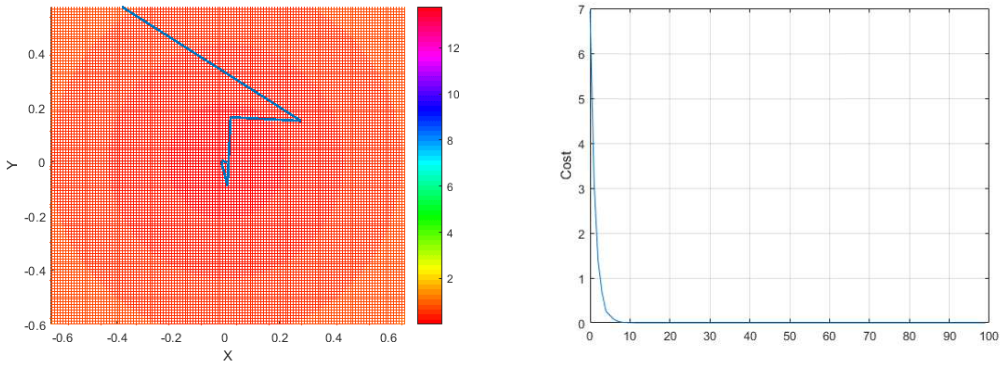
Simülasyon bu sefer qABC algoritmasındaki değişikliklerle beraber ABC algoritması simülasyonuna ek komşuluk katsayı r=1 alınarak çalıştırılmıştır. Sonuçlar Şekil 5.11'de paylaşılmıştır. Şekil 5.11 (a)'da qABC için global minimuma sahip robotların uzaydaki x ve y pozisyonlarının değişimi incelenebilmektedir. Şekil 5.11 (b)'de de bu global bireyin maliyet değerlerinin iterasyona bağlı değişimi incelenebilir.



(a) qABC X Y parametreler değişimi (b)) Maliyetin İterasyona Bağlı Değişimi

Şekil 5.11. qABC Algoritmasının Tek Kaynak Senaryosu Sonuçları

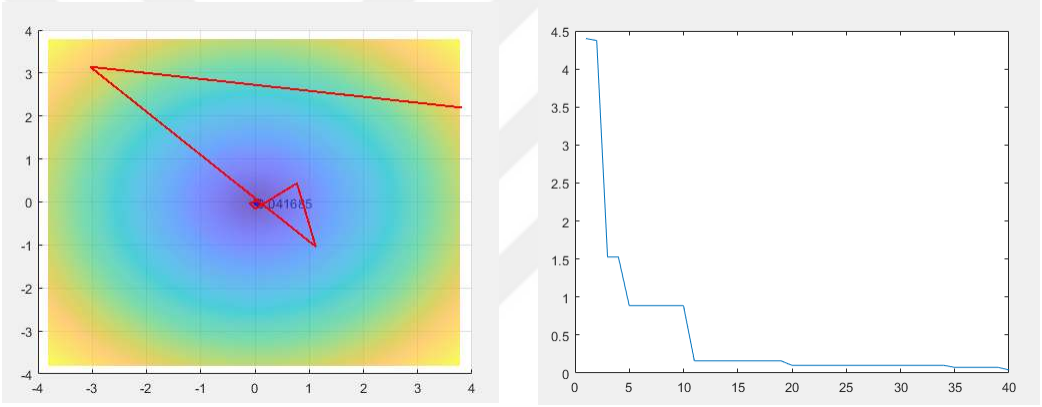
Simülasyon mABC algoritması için ABC algoritmasının simülasyonundaki başlangıç değerleri ile çalıştırılmıştır. mABC algoritmasının iki paremetre değiştirme özelliği sayesinde hareketlerin sadece bir eksende gerçekleşmesinin önüne geçilmiştir. mABC algoritmasının simülasyonunun sonuçları Şekil 5.12'de paylaşılmıştır.



(a) mABC X Y Parametrelerinin Değişimi (b) Maliyetin İterasyona Bağlı Değişimi

Şekil 5.12. mABC Algoritmasının Tek Kaynak Senaryosu Sonuçları

PSO için hız vektörü katsayıları belirlenmeye çalışılmış ve c1=c2=2 seçilmiştir. Atalet katsayısı ise 1 ile başlatılıp her iterasyonda ataletin %1'lik kayba uğratılmıştır. PSO simülasyonunun sonuçları Şekil 5.13'de paylaşılmıştır.



(a) PSO X Y Parametrelerinin Değişimi (b) Maliyetin İterasyona Bağlı Değişimi

Şekil 5.13. PSO Algoritmasının Tek Kaynak Senaryosu Sonuçları

Simülasyonda kullanılan ABC, qABC, mABC ve PSO algoritmaları 50, 100, 500 ve 1000 iterasyon boyunca 100 kere çalıştırılmış ve sonuçlar sırayla Çizelge 5.1, 5.2, 5.3 ve 5.4'te paylaşılmıştır. Her yöntem 100 iterasyon için farklı sayılarda çalıştırıldığında, bu çalışmalar boyunca minimum, maksimum ve ortalama değerleri en iyi olanlar kalın font ile belirtilmiştir. Tüm çalışmalar boyunca ABC algoritmaları için kaşif sayısı 1, yeni çözüm üretememe sayacı 30 φ katsayısı: 0.5 alınmıştır. PSO için $w = 1$'den başlayıp her iterasyonda %1'lik kayba uğratılmıştır. $c_1 = c_2 = 2$ alınmıştır. Arama uzayı $x = [-100\ 100]$, $y = [-100\ 100]$ olacak şekilde ayarlanmış, hedefin bulunduğu pozisyon $[x\ y] = [0\ 0]$ olarak belirlenmiştir.

Çizelge 5-1. Kullanılan Algoritmaların Sonuçları; İterasyon: 50 - Çalıştırma: 100

Algoritma	Minimum	Maksimum	Ortalama
ABC	**7.0594e-06**	3.2793e-04	6.4258e-05
qABC	4.8945e-09	**1.2336e-05**	6.1639e-07
mABC	4.8684e-07	6.7396e-05	**1.9508e-05**
PSO	0.0019	0.0555	0.0186

Çizelge 5-2. Kullanılan Algoritmaların Sonuçları; İterasyon: 100 - Çalıştırma: 100

Algoritma	Minimum	Maksimum	Ortalama
ABC	9.7920e-10	4.1660e-07	6.6730e-08
qABC	**1.9625e-15**	**1.3543e-10**	**3.8872e-12**
mABC	2.5020e-10	2.1456e-08	4.1689e-09
PSO	2.5306e-08	5.2324e-06	8.1668e-07

Çizelge 5-3. Kullanılan Algoritmaların Sonuçları; İterasyon: 500 - Çalıştırma: 100

Algoritma	Minimum	Maksimum	Ortalama
ABC	6.6316e-19	7.3371e-17	3.0816e-17
qABC	3.2304e-18	8.8680e-17	3.7187e-17
mABC	1.3147e-18	4.8725e-17	1.8871e-17
PSO	**2.5719e-118**	**1.6344e-113**	**9.3668e-115**

Çizelge 5-4. Kullanılan Algoritmaların Sonuçları; İterasyon: 1000 - Çalıştırma: 100

Algoritma	Minimum	Maksimum	Ortalama
ABC	2.9295e-18	7.5624e-17	2.9600e-17
qABC	2.1864e-18	7.7802e-17	3.4976e-17
mABC	1.6619e-18	5.1437e-17	1.9540e-17
PSO	**0**	**0**	**0**

Hedefin bulunacağı alan 100 metre x 100 metre olduğu, ve sürüdeki robotların 1 cm çapa sahip olduğu düşünüldüğünde, robotlardan beklenen hedefe ulaşma hareketi, robotun hedefe değmesi ile sonlanacaktır. Bu durumda da bu yöntemler çalıştırılarak hedefe değdikleri ilk iterasyon sayıları Çizelge 5.5'te paylaşılmıştır.

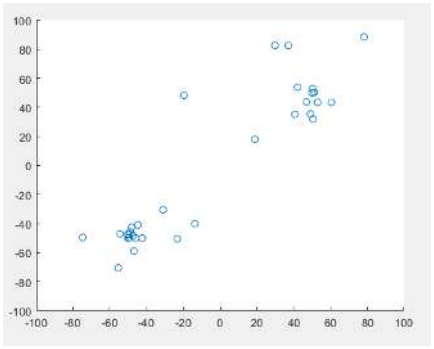
Çizelge 5-5. Kullanılan Algoritmaların Hedefe Varma İterasyonları

Algoritma	Hedefe Değdiği İterasyon
ABC	17
qABC	**15**
mABC	20
PSO	49

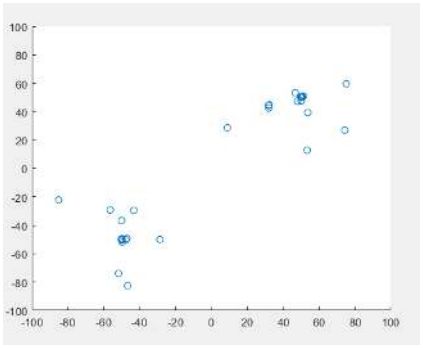
5.3. Birden Fazla Hedef Senaryosu İçin Simülasyon Sonuçları

Çoklu hedef için Eşitlik 5.3 denklemi maliyet fonksiyonu için önerilmiştir. Bu fonksiyon iki hedef için ($x_1 = [-50, -50]$, $x_2 = [50,50]$) simülasyondaki tüm uygulamalar üzerinde p terimi 0 olacak şekilde simüle edildiğindeki robotların davranışları Şekil 5.14'de gösterilmiştir. P terimi 25 olarak seçildiğinde ise robotların yeni maliyet fonksiyonuna verdikleri tepkiler Şekil 5.15'de gösterilmiştir. Çoklu hedef için maliyet fonksiyonu dışındaki parametreleri değiştirilmemiştir.

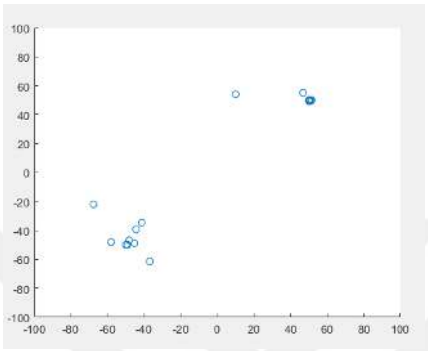
Şekil 5.14'deki robot dağılımları incelendiğinde, eş sinyal yayan hedefleri arama işleminde ABC algoritması ve varyantları robotları neredeyse eşit bir şekilde kaynaklara dağıtmış olduğu ancak PSO algoritması ise robotları, ilk yaklaşılan hedefe topladığı görülmüştür. Şekil 5.14 (d)'de görülen PSO dağılımında -50 -50 hedefinde olmayan robotlar tekil olarak alanda dağılmışken diğer tüm robotlar -50 -50 noktasında üst üste toplanmışlardır. Şekil 5.15 incelendiğinde ise, -50 -50 hedefinin sinyali düşürüldüğünden bu nokta lokal optimuma dönüşmüştür. Buna rağmen ABC algoritması ve qABC algoritmasında robotlar iki hedefte de toplanabilmişlerdir. Ancak mABC algoritmasında (Şekil 5.15 (c)) robotlar sinyal şiddeti yüksek olan hedefe daha çok toplanmışlardır. PSO algoritmasında ise (Şekil 5.15 (d)) tüm sürünün sinyal şiddeti yüksek olan hedefte toplandığı gözlemlenmiştir. Bu gözlem, ABC algoritmalarının sürü robotların çoklu hedef bulma konusunda PSO'ya göre daha başarılı olduğunu göstermekte iken, PSO algoritmasının ise Eşitlik 5.3'de verilen problemin optimizasyonunda, lokal minimumda bireyinin olmadığı göz önünde bulundurulduğunda daha başarılı olduğu söylenebilir.



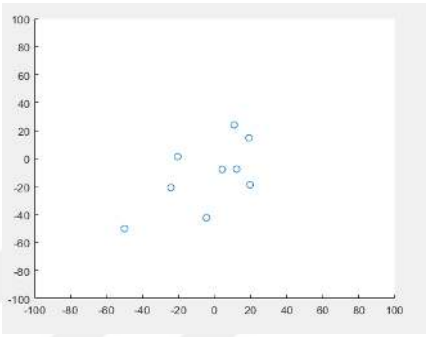
(a) ABC



(b) qABC

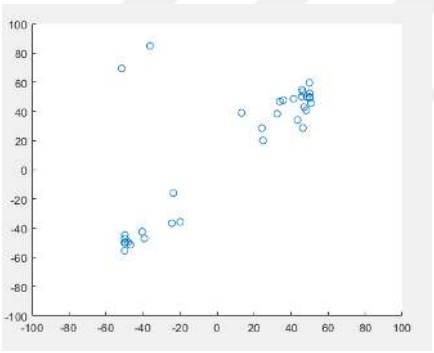


(c) mABC

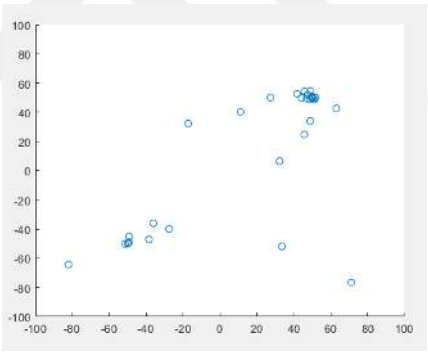


(d) PSO

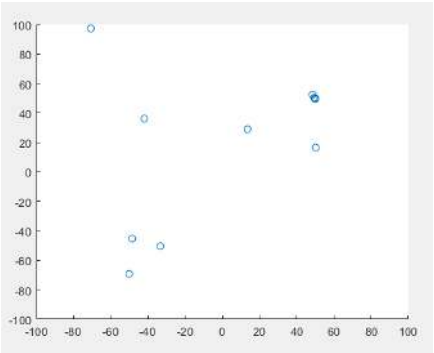
Şekil 5.14. Eş İki Kaynakta Robotların Davranışları



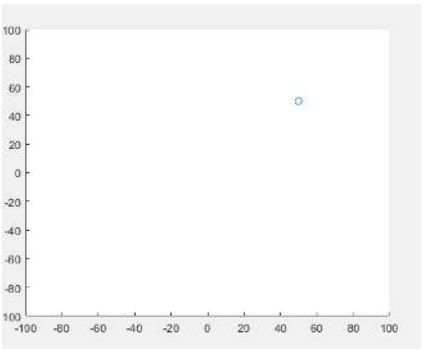
(a) ABC



(b) qABC



(c) mABC

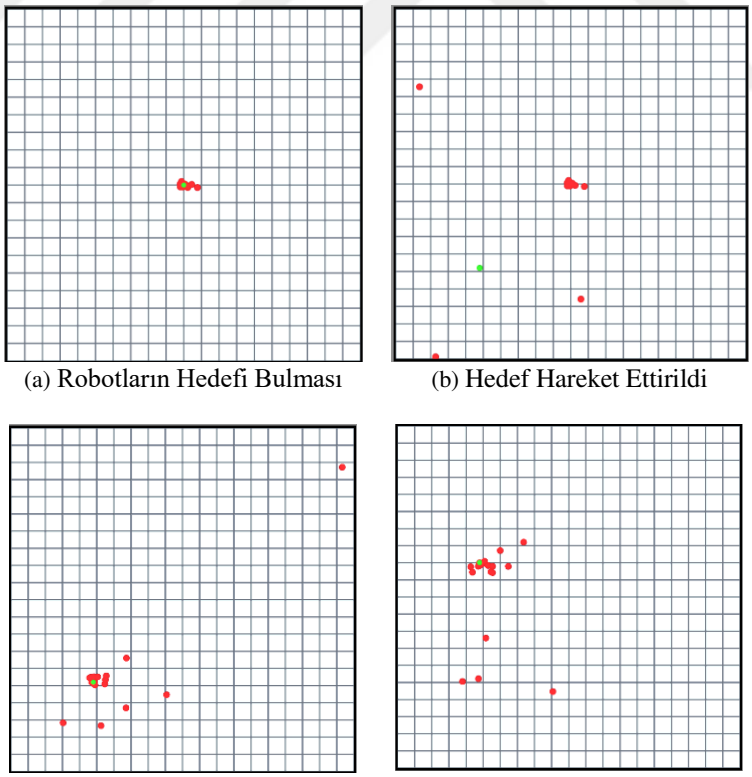


(d) PSO

Şekil 5.15. Eş Olmayan İki Kaynakta Robotların Davranışları

5.4. Dinamik Hedef Senaryosu İçin Simülasyon Sonuçları

Dinamik olarak pozisyon değiştiren hedef senaryosu geliştirilen simülasyon ortamında test edildiğinde PSO algoritmasının atalet katsayısı bileşeninden ötürü başarısız olduğu gözlemlenmiştir. ABC algoritması için ise sonuçlar Şekil 5.16'da gösterilmiştir. Hedef en başta Şekil 5.16 (a)'da görülebileceği gibi hareket ettirilmemiş ve robotların hedefe toplanması beklenmiştir. Daha sonra hedefin Şekil 5.16 (b)'deki gibi yeri değiştirilmiş, ve robotların kâşif robotlar oluşmaya başlayana kadar gruplarını terk etmedikleri gözlemlenmiştir. İlk kâşif robotlardan gelen yeni bilgilerle beraber robotlar hedefe doğru Şekil 5.16 (c)'deki gibi hareket etmeye başlamışlardır. Hedef her pozisyon değiştirdiğinde bu işlemin tekrar ettiği Şekil 5.16 (d)'de gözlemlenebilir. Bu davranış biçimi kaşif robotların dinamik kaynak takibi konusunda oldukça önemli olduğunu göstermiştir. Ancak PSO algoritmasındaki gibi Kaşif robotlar oluşturulmadığında, robotlar hedefe toplandıktan sonra hedefin yeri değiştirildiğinde, robotların hedefe o an en yakın olan robota yaklaşmaya çalışıp hedefin asıl yerini tespit edemedikleri gözlemlenmiştir.



(a) Robotların Hedefi Bulması

(b) Hedef Hareket Ettirildi

(c) Robotların Hedefi Tekrar Bulması

(d) Dinamik Kaynağı Takip Hareketi

Şekil 5.16. ABC Algoritmasının Dinamik Kaynak İçin Davranış Gözlemleri

6. SONUÇ ve ÖNERİLER

Bu tez çalışmasının sonunda sürü robotların hedef bulma davranışını metasezgizsel yöntemlerden olan ABC, qABC, mABC ile PSO algoritmaları ile benzetiminin yapılabildiği bir simülasyon uygulaması geliştirilmiştir. İncelenen ABC algoritmalarının ve PSO algoritmasının robotik alanında kullanılabilmesi için bazı adaptasyonlar önerilmiş, bu öneriler simülasyon uygulamasına başarıyla entegre edilmiştir. Daha sonra bu algoritmalar aynı başlangıç değerleri ile simülasyona tabi tutulup performansları karşılaştırılmıştır. Geliştirilen simülasyon uygulamasında, sabit hedef bulma davranışı benzetimi yapılmış ve düşük iterasyonlarda ABC algoritması ve varyantlarının PSO'ya göre hedef bulma konusunda daha başarılı olduğu, ancak yüksek iterasyonlarda PSO algoritmasının çok daha düşük hata payı ile hedefi tespit edebildiği gözlemlenmiştir. Arama uzayında birden fazla hedef olması durumu da incelenmiş ve ABC ve varyantlarının eş sinyal şiddetindeki ve farklı sinyal şiddetlerindeki çoklu hedef bulma konusunda PSO algoritmasına göre daha başarılı olduğu gözlemlenmiştir. PSO algoritmasının ise lokal minimumlara (düşük sinyalli hedeflere) takılmadan global minimumu daha iyi bir kesinlik ile bulabildiği gözlemlenmiştir. Bu gözlemler doğrultusunda ABC algoritmasının düşük iterasyonlarda daha uygun sonuçlar üretmesi, robotların daha az enerji harcamasını da sağlayacağından PSO'ya göre uyarlanabilirliğinin yüksek olduğunu göstermiştir.

Pozisyonu zamana bağlı değişebilen hedeflerin bulunma davranışı ABC algoritmaları ile incelenebilmiş ve ABC algoritmalarının dinamik hedefleri takip etme konusunda da başarılı sonuçlar ürettiği gözlemlenmiştir. PSO algoritmasının ise çalışma süresi boyunca atalet katsayısı bileşeninden ötürü adımlarını küçültmesi, algoritmanın dinamik hedefleri takip edebilme yeteneklerini düşürdüğü gözlemlenmiştir.

6.1. Öneriler

Bu tez çalışmasında tüm robotların pozisyon güncellemelerinin aynı iterasyonda olduğu kabul edilmiş, ancak fiziksel hareket esnasında çözüme yakınlıkları simüle edilen algoritmaların davranışını değiştirmemek adına dikkate alınmamıştır. Bu çalışmaya ek olarak, fiziksel hareketin sağlayabileceği bu avantaj da kullanılıp daha başarılı hedef bulma uygulamaları geliştirilebilir. Her robotun hedef noktalarına

gitmeleri beklenmeden, gidilecek hedefe ulaşan robota bir sonraki hedefi görev olarak verilerek asenkron iteratif yöntemler geliştirilebilir, böylece sürü robot sisteminin maliyeti düşürülmeye çalışılabilir. Simülasyon uygulamasına alanda kullanılan veya kullanılabilirliği olan diğer metasezgisel yöntemler de entegre edilerek hedef bulma davranışlarının karşılaştırılması daha genel bir şekilde gerçekleştirilmesi planlanmaktadır. Ayrıca literatürdeki metasezgisel yöntemlerin avantajlarını kullanıp birleştiren hibrit algoritmalar üzerinde çalışılması planlanmaktadır.

KAYNAKLAR

[1] C. G. C. ERCAN, *İnsansız Hava sistemleri Rota Planlaması Dinamik Çözüm Metotları ve Literatür Araştırması*, **Selcuk Univ. J. Eng. Sci. Tech**.,1:2, 51-72, 2013.

[2] Y. Cao, A. S. Fukunaga ve A. B. Kahng, *Cooperative Mobile Robotics: Antecedents and Directions*, **Auton. Robots**, 1997.

[3] Anonymous.(2018) http://images.google.com/ search term: natural swarms. (on-line access on: 13 12 2018).

[4] G. Seeja, A. A. Selvakumar ve B. V. Hency, *A Survey on Swarm Robotic Modeling, Analysis and Hardware Achitecture*, **Procedia Comput. Sci.**, 133, 478 - 485, 2018.

[5] M. L. Minsky, *Computation: Finite and Infinite Machines*, New Jersey: Prentice Hall Inc., 1967.

[6] J. H. Reif ve H. Wang, *Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots*, **Rob. Auton. Syst.**, 27, 171-194, 1999.

[7] M. Mataric, *Reinforcement Learning in the Multi - Robot Domain*, **Auton. Robots**., 4, 73-83, 1997.

[8] J. F. Kramer ve M. Scheutz, *Development Enviorements for Autonomous Mobile Robots: A survey*, **Auton. Robots**, 22, 101-132, 2007.

[9] X. Chen ve J. Huang, *Odor Source Localization Algorithms on Mobile Robots: A Review and Future Outlook*, **Rob. Auton. Syst.**, 112, 123-136, 2019.

[10] K. Brudzewski, S. Osowski ve J. Ulaczyk, *Differantial Electronic Nose of Two Chemo Arrays for Odor Discrimination*, **Sensors and Actuators B-chemical**, 145, 246-249, 2008.

[11] Anonymous (2019). Available: http://imahes.google.com/, search term: Bat Echolocation. (online access on 01 02 2019).

[12] C. Rascon ve I. Meza, *Localization of Sound Sources in Robotics: A Review*, **Rob. Auton. Syst.**, 96,164-210, 2017.

[13] K. Nakadai, T. Lourens, G. O. Hiroshi ve H. Kitano, *Active Audition for Humanoid*, Natl. Conf. Artif. Intell., Austin, 2000.

[14] S. Argentieri, A. Portello, M. Bernard, P. Danes ve B. Gas, *Binaural Systems in Robotics*, Technology of Binaural Listening, Berlin, 2013.

[15] S. Argentieri, P. Danes ve P. Soueres, *A Survey on Sound Source Localization in Robotics: From Binaural to Array Proccessing Methods*, **Comput. Speech. Lang**., 34:1, 87-112, 2017.

[16] H. G. Okuno ve K. Nakadai, *Robot Audition: Its Rise and Perspectives*, ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process., Brisbane, 2015.

[17] M. N. Wahab, N. Meziani ve A. Atyabi, *A Comprehensive Review of Swarm Optimization Algorithms*, **Plos One**,1-36, 2015.

[18] L. J. Fogel, A. J. Owens ve M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, Hoboken: John Wiley, 1966.

[19] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Sgart Newsl., 1975.

[20] F. Glover, *Future Paths for Integer Programming and Links to Artificial Intelligence*, Comput. Oper. Res., 1986.

[21] S. Voß, *Meta - Heuristics: The State of the Art*, Local Search for Planning and Scheduling, 2001, p. Chapter 1.

[22] F. Glover ve M. Laguna, *Tabu Search*, MA, USA: Kluwer Academic Publishers, 1997.

[23] J. Kennedy ve R. Eberhart, *Particle Swarm Optimization*, Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[24] M. Settles, An Introduction to Particle Swarm Optimization, Springer London, 2005.

[25] N. A. Aziz ve Z. Ibrahim, *Asynchronous Particle Swarm Optimization for Swarm Robotics*, **Procedia Engineering**,41, 951-957.

[26] N. Rokbani ve M. Alimi, *Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis*, **Procedia Engineering**, 64, 1602-1611, 2013.

[27] D. Karaboga. *An Idea Based On Honey Bee Swarm For Numerical Optimization,* Techical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[28] M. Dorigo ve K. Socha. *An Intoduction to Ant Colony Optimization,* Metaheuristic Procedures for Training Neural Networks, pp. 153-180

[29] X. S. Yang ve S. Deb, *Cuckoo Search Via Levy Flights*, World Congress on Nature and Biologically Inspired Computing NABIC, Caimbatore, 2009.

[30] X. S. Yang, *Firefly Algorithms for Multimodal Optimization*, **SAGA Lecture Notes in Computer Sciences**,5792, 169-178, 2009.

[31] A. Karci ve M. Canayaz, *Investigation of Cricket Behaviours as Evolutionary Computation System Design Optimization Problems*, **Measurement**, 68, 225 - 235, 2015.

[32] D. Karaboga ve B. Gorkemli, *A Quick Artificial Bee Colony (qABC) Algorithm and Its Perforrmance on Optimization Problems*, **Appl. Spft. Comput. J.**, 23, 227-238, 2014.

[33] B. Babayigit ve R. Ozdemir, *Modifiye Yapay Arı Kolonisi Algoritması ile Nümerik Fonksiyon Optimizasyonu*, ELECO 2012 Elektr. - Elektron. ve Bilgi. Mühendisliği Sempozyumu, Bursa, 2012.

[34] S. Fong, S. Deb ve A. Chaudhary, *A Review of Metaheuristics in Robotics*, **Computers and Electrical Engineering**, 43, 278-191, 2015.

[35] M. R. Akbarzadeh, K. Kumbla, E. Tunstel ve M. Jamshidi, *Soft Computing for Autonomous Robotic Systems*, **Computers & Electrical Engineering**, 26:1, pp. 5-32, 2000.

[36] A. Tuncer ve M. Yildirim, *Dynamic path planning of mobile robots with improved genetic algorithm*, **Computers & Electrical Engineering**, 38:6, pp. 1564 - 1572, 2012.

[37] M. Senanayake, I. Senthooran, J. C. Barca ve H. Chung, *Search and Tracking Algorithms for Swarms of Robots: A survey*, **Robotics and Autonomous Systems**,75, 422-434, 2016.

[38] X. S. Yang, *Nature Inspired Metaheuristic Algorithms*, Cambridge: Luniver Press, 2010.

[39] X. S. Yang, S. Deb ve S. Fong, *Metaheuristic Algorithms: Optimal Balance of Intensification and Diversification*, **Applied Mathematics & Information Sciences (AMIS) Nat Sci**,8, 2014.

[40] J. Hereford ve M. Siebold, *Bio Inspired Search Strategies for Robot Swarms*, **Intech**,1-27, 2010.

[41] T. Schmickl ve K. Crailshem, *Trophollaxis within a Robotic Swarm: Bio Inspired Strategy for Swarm Robots*, Birob 2006: Biomedical Robotics and Biomechatronics, Pisa, 2006.

[42] T. D. Ngo ve H. Schioler, *Randomized Robot Trophallaxis*, Recent Advances in Multi Robot Systems, Denmark: Intech, 2008.

[43] M. S. M. H. Z. Z. Abidin, M. R. Arshad ve U. K. Ngah, *A Calibration Framework for Swarming ASVs' System Design*, **Indian Journal of Geo-Marine Sciences**, 41:6, 581-588, 2012.

[44] L. M. Ni, Y. Liu, Y. C. Lau ve A. P. PAtil, *LANDMARC: Indoor Location Sensing Using Active RFID*, Proceedings of the First IEEE Int. Conf. on Pervasive Comp. and Comm. (PERCOM), pp. 407-415, 2003.

[45] J. Wang; Z. Luo; E. C. Wong, *RFID Enabled Tracking in Flexible Assembly Line* **The Int. Journal of Adv. Manufacturing Technology**, 46:1-4, 351-366, 2010.

[46] L. Euler, *Solutio Problematis and Geometriam Situs Perinents*, 1736.

[47] K. R. Saoub, *A Tour Through Graph Theory*, London: Chapman and Hall CRC, 2017.

[48] F. Harary, *Graphical Enumeration*, New York & London: Academic Press, 1973.

[49] M. Guan, *Graphic Programming Using Odd and Even Points*, Chienese Mathematics,1962.

[50] Anonymous (2019). Available: http://images.google.com/ search term: konigsberg problem. (online access on: 10 1 2019).

[51] H. Jiang ve K. Li-shan, *Genetic Algorithm for Chienese Postman Problems*, **Wuhan University Journal of Natural Sciences***,* 8:1,316 - 318, 2003.

[52] E. U. Kucuksille ve M. Tokmak, *Yapay Arı Kolonisi Algoritması Kullanarak Otomatik Ders Çizelgeleme*, **Suleyman Demirel Univ. Fen Bil. Enst. Dergisi***,* 15:3, 203-210, 2011.

[53] D. Karaboga ve B. Basturk, *Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems*, Found. Fuzzy Log. Soft. Comput. IFSA*,* pp. 789-798, 2007.

[54] B. Akay, Tez. *Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi.* Phd Thesis, Erciyes University Turkey 2009.

[55] Y. Shi ve C. Eberhart, *Empirical Study of Particle Swarm Optimization*, Proceedings of the 1999 Congress on Evolutionary Computation. CEC 1999, 1999.

[56] Y. E. Demirtaş, *Dinamik Araç Rotalama Problemine Parçacık Sürü Optimizasyonu Algoritması Çözüm Önerisi.* Phd Thesis, Istanbul University Turkey 2015

[57] E. Fendoglu ve H. Soyler, *Route Optimization of Malatya Metropolitan Municipality Presticide Vehicles*, **Alphanumeric J.**, 6:1, pp. 13-24, 2017.

[58] J. Polcar, P. K. P. Horejsi ve M. Latif, *Using Unity3D as an Elevator Simulation Tool*, Proceedings of the 28th DAAAM Int. Symp.*,* pp. 517-522, 2017.

ÖZGEÇMİŞ

Ad Soyad : Mehmet Akif FINDIKLI

Doğum Yeri ve Tarihi : Malatya - 02.01.1988

Adres : İnönü Üniversitesi Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

E-Posta : m.akif.findikli@inonu.edu.tr
akif.findikli@gmail.com

Lisans : İnönü Üniversitesi (2011-2015)

Mesleki Deneyim

Araştırma Görevlisi	: Mersin Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği, 2016
Araştırma Görevlisi	: İnönü Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı, 2016 – 2019
Araştırma Görevlisi	: Mersin Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği 2019 - …

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

Uluslararası Bilimsel Toplantılarda Sunulan Bildiriler

FINDIKLI M.A., KOCAMAZ A.F. "Yapay Arı Koloni Algoritmalarının Sürü Robot Hedef Bulma Problemine Uygulanması", International Symposium on Innovative Approaches in Scientific Studies. Samsun, 2018.

FINDIKLI M.A., v.d. "Yapay Arı Kolonisi Algoritması İle Gezici Robotun Haritalanmış Sahada Gezinme Maliyetinin Optimizasyonu", International Symposium on Innovative Approaches in Scientific Studies. Samsun, 2018.