

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YAPAY ARI KOLONİSİ ALGORİTMASI TABANLI BİR
YÖNTEM İLE DİNAMİK ENGELLERİN BULUNDUĞU
ORTAMLAR İÇİN BÖLGESEL YOL PLANLAMA**

**Hazırlayan
Senem Nazlı DOĞAN**

**Danışman
Dr. Öğr. Üyesi Beyza GÖRKEMLİ**

Yüksek Lisans Tezi

**Şubat 2022
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YAPAY ARI KOLONİSİ ALGORİTMASI TABANLI BİR
YÖNTEM İLE DİNAMİK ENGELLERİN BULUNDUĞU
ORTAMLAR İÇİN BÖLGESEL YOL PLANLAMA
(Yüksek Lisans Tezi)**

**Hazırlayan
Senem Nazlı DOĞAN**

**Danışman
Dr. Öğr. Üyesi Beyza GÖRKEMLİ**

**Şubat 2022
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Senem Nazlı DOĞAN

İmza

“Yapay Arı Kolonisi Algoritması Tabanlı Bir Yöntem İle Dinamik Engellerin Bulunduđu Ortamlar İin Bölgesel Yol Planlama” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Senem Nazlı DOĞAN

İmza

Danışman

Dr. Öğr. Üyesi Beyza GÖRKEMLİ

İmza

Bilgisayar Mühendisliđi ABD Başkanı

Prof. Dr. Veysel ASLANTAŞ

İmza

TEŐEKKÜR

Bana alıŐmalarım süresince her türlü yardımını ve fedakârlığı sađlayan deđerli hocam Dr. Öğr. Üyesi Beyza GÖRKEMLİ'ye, bu süreçte manevi destekleri ile her zaman yanımda olan sevgili eşim ve aileme teşekkür ederim.

Senem Nazlı DOĐAN

02 2022, KAYSERİ

YAPAY ARI KOLONİSİ ALGORİTMASI TABANLI BİR YÖNTEM İLE DİNAMİK ENGELLERİN BULUNDUĞU ORTAMLAR İÇİN BÖLGESEL YOL PLANLAMA

Senem Nazlı DOĞAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Şubat 2022
Danışman: Dr. Öğr. Üyesi Beyza GÖRKEMLİ

ÖZET

Günümüzde farklı birçok alanda otonom robot kullanımı yaygınlaşmaktadır. Otonom robotlarda yol planlama temel problemlerdendir. Bölgesel yol planlaması, robotun hareket alanı için tüm haritanın bilinmediği, dolayısıyla robotun başlangıç ve hedef nokta arasında engellere çarpmadan uygun bir yol bulabilmesi için genellikle sensörlerinden aldığı bilgileri kullanarak yolunu planladığı bir problemdir. Bu noktada, engellere çarpmadan başlangıç ve hedef noktalar arasında uygun bir yol bulmak ve bu yolun en kısa veya farklı diğer açılardan da optimal yol olması önem arz etmektedir. Bölgesel yol planlama problemi engellerin tipine göre farklı şekillerde ele alınabilmektedir. Bu tezde sabit engellerin yanında hareketli-dinamik engellerin bulunduğu ortamlar dikkate alınarak bölgesel yol planlaması üzerinde çalışılmıştır. Yapay arı kolonisi algoritmasının yeni bir versiyonu geliştirilmiş ve problemin çözümü için bu versiyon temelli bir yol planlama yaklaşımı önerilmiştir. Önerilen yöntem simülasyon ortamında test edilmiş, standart yapay arı kolonisi algoritmasına dayalı bir yol planlama yaklaşımı ile kıyaslanmıştır.

Anahtar Kelimeler: Robotik Yol Planlama, Bölgesel Yol Planlama, Hareketli Engel, Yapay Arı Kolonisi Algoritması (ABC), Değiştirilmiş Yapay Arı Kolonisi Algoritması

**LOCAL PATH PLANNING WITH AN ARTIFICIAL BEE COLONY
ALGORITHM BASED METHOD FOR THE ENVIRONMENTS HAVING
DYNAMIC OBSTACLES**

Senem Nazlı DOĞAN

Erciyes University, Graduate School of Natural and Applied Sciences

Master Thesis, February 2022

Supervisor: Dr. Beyza GÖRKEMLİ

ABSTRACT

Robot path planning has been an active area of research and many methods have been developed to overcome this problem. There are situations where each method is strong in certain aspects. In general, the main challenges for the robot-path planning problem are computational complexity, local optimum, and adaptability. Alternative and effective ways are sought to solve these problems. Regional path planning is a problem where the entire map for the robot's range of motion is unknown, so the robot usually plans its path using the information it receives from its sensors so that it can find a suitable path between the starting point and the target point without hitting obstacles. At this point, it is important to find a suitable route between the starting and target points without hitting obstacles, and that this path is the shortest or optimal in other respects. Regional road planning problem can be handled in different ways according to the type of obstacles. In this paper, regional road planning will be studied by considering the environments with dynamic obstacles. It is aimed to investigate the success of ABC based mABC (modified-ABC) algorithm, which is one of the heuristic optimization algorithms based on herd intelligence, in regional path planning problem in environments with dynamic obstacles.

Keywords: Robotic Path Planning, Artificial Bee Colony Algorithm(ABC), Dynamic Obstacles , modified-Artificial Bee Colony Algorithm(mABC)

İÇİNDEKİLER

YAPAY ARI KOLONİSİ ALGORİTMASI TABANLI BİR YÖNTEM İLE DİNAMİK ENGELLERİN BULUNDUĞU ORTAMLAR İÇİN BÖLGESEL YOL PLANLAMA

BİLİMSEL ETİĞE UYGUNLUK.....	ii
YÖNERGEYE UYGUNLUK.....	iii
KABUL VE ONAY	iv
ÖNSÖZ	v
ÖZET.....	vi
ABSTRACT.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	x
TABLolar LİSTESİ.....	xi
ŞEKİLLER LİSTESİ	xii
GİRİŞ.....	1

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Hareketli Robot Yol Planlama Problemi	4
1.2. Literatür Taraması	6

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. ABC Algoritması.....	12
2.2. mABC Algoritması.....	16
2.3. Kullanılan Yol Planlama Yaklaşımı.....	18

3. BÖLÜM

DENEYSEL ÇALIŞMALAR

3.1. Deneysel Çalışmalar.....	21
-------------------------------	----

4. BÖLÜM

TARTIŞMA, SONUÇ ve ÖNERİLER

4.1. Tartışma ve Sonuç.....	32
4.2. Öneriler.....	33
KAYNAKÇA	34
ÖZGEÇMİŞ.....	37

KISALTMALAR

ABC	: Yapay Arı Kolonisi (Artificial Bee Colony)
ACO	: Karınca Koloni Algoritması (Ant Colony Algorithm)
AMOABC	: Uyarlanabilir Çok Amaçlı ABC (Adaptive Multi-Objective Artificial Bee Colony)
EP	: Evrimsel Programlama (Evolutionary Programming)
GA	: Genetik Algoritma (Genetic Algorithm)
İHA	: İnsansız Hava Aracı
mABC	: Değiştirilmiş Yapay Arı Kolonisi (modified Artificial Bee Colony)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
qABC	: Hızlı Yapay Arı Kolonisi (quick Artificial Bee Colony)

TABLolar LİSTESİ

Tablo 3.1. ABC ve mABC için kontrol parametre değerleri	22
Tablo 3.2. Senaryo 1 için durdurma kriterinin 1000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar	24
Tablo 3.3. Senaryo 1 için durdurma kriterinin 5000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar	26
Tablo 3.4. Senaryo 2 için durdurma kriterinin 1000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar	28
Tablo 3.5. Senaryo 2 için durdurma kriterinin 5000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar	30
Tablo 3.6. Robotun engel ile ilk karşılaşma anında alınan ABC ve mABC sonuçları ...	31

ŞEKİLLER LİSTESİ

Şekil 1.1. İki sabit engel bulunan ortamda robotun hedefe ulaşırken izlediği yol	6
Şekil 2.1. mABC algoritmasının adımları.....	17
Şekil 2.2. Önerilen yol bulma yaklaşımına ait akış şeması.....	18
Şekil 3.1. Senaryo 1’de durdurma kriterinin 1000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol	23
Şekil 3.2. Senaryo 1’de durdurma kriterinin 1000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol	24
Şekil 3.3. Senaryo 1’de durdurma kriterinin 5000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol	25
Şekil 3.4. Senaryo 1’de durdurma kriterinin 5000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol	26
Şekil 3.5. Senaryo 2’de durdurma kriterinin 1000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol	27
Şekil 3.6. Senaryo 2’de durdurma kriterinin 1000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol	28
Şekil 3.7. Senaryo 2’de durdurma kriterinin 5000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol	29
Şekil 3.8. Senaryo 2’de durdurma kriterinin 5000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol	30

GİRİŞ

Robotların ve uygulama alanlarının gelişimi incelendiğinde, insan hayatında önemli bir rol oynadığı görülmektedir. Günümüzde robotların kullanım alanları oldukça geniş ve bu alanların çeşitliliği de giderek artmaktadır. Robotlar sağlık hizmetlerinde, fabrikalarda, bahçe işlerinde, hatta evlerde insanlara büyük kolaylık sağlamaktadır.

Hareketli robotlardan etkin bir şekilde yararlanabilmemiz için insan kontrolüne ihtiyaç duymadan ilerleyebilmesi gerekmektedir. Robot yolu planlama probleminde varış noktasına ortamda bulunan herhangi bir engele çarpmadan ve en kısa yoldan ulaşılması temel hedeftir. Hareketli robot yolu planlama küresel yol planlama ve bölgesel yol planlama olmak üzere iki temel başlık altında incelenebilir [1]. Küresel yol planlamada hareketli robotun ortam hakkında tam bir bilgiye sahip olması gerekmektedir [1]. Ancak bölgesel yol planlamasında robot sensörlerini kullanarak algılayabildiği bilgilere sahiptir ve çevresel değişikliklere cevap vererek yolunu planlar [1].

Robotun hareket ettiği ortamda farklı şekillerde, boyutlarda ve sayılarda, sabit veya dinamik (hareketli) engeller olabilir. Ortam ile ilgili küresel bilgilerin toparlanmasının mümkün olmadığı veya fazla maliyetli olduğu durumlarda, özellikle ortamda dinamik değişimler söz konusu ise bölgesel yol planlama çalışmaları ile robotun ilerlemesinin sağlanması önemli olmaktadır. Araştırmacılar bölgesel yol planlamada hedefe yaklaşma, engellerden kaçınma gibi farklı amaçları gözeterek robotun atacağı her adımı bir optimizasyon problemi olarak ele alıp ve evrimsel hesaplama dayalı optimizasyon algoritmaları ile bu problemleri çözmeye çalışmaktadırlar [2,3,4,5,6].

Yapay arı kolonisi (Artificial Bee Colony-ABC) algoritması, bal arılarının yiyecek arama davranışlarının modellenmesi ile 2005 yılında Karaboga tarafından literatüre kazandırılmış sürü zekasına dayalı bir optimizasyon yöntemidir [7, 8]. Algoritmanın

başarısı, kontrol parametresi sayısının az olması, esnekliği, uygulamanın kolay olması gibi hususlar çeşitli optimizasyon problemlerinde sıklıkla kullanılmasını sağlamış ve iyi sonuçlar elde edilmiştir [8,9].

Bu çalışmada dinamik engellerin bulunduğu ortamlarda bölgesel yol planlama problemi için ABC tabanlı yeni bir yöntem geliştirilmiştir. Bu yöntemde başlangıç noktasından hedefe ulaşana kadar robotun her adımında engele çarpmadan optimum noktaya varılıp varılamayacağı kontrol edilmektedir. Bu noktada bir çarpışma durumu söz konusu olacaksa [2]'deki çalışmaya benzer şekilde bir metasezgisel ile hem hedefe yakınlık hem de engellere çarpmama amaçları dikkate alınarak robotun gideceği nokta belirlenmektedir. Burada bahsedilen metasezgisel olarak tez kapsamında geliştirilen bir ABC versiyonu kullanılmıştır. Bu versiyonda, problemin gerçek zamanlı (real time) cevap isteme durumu, birçok optimizasyon problemine kıyasla problem boyutunun küçük (2 boyutlu) olması, bunlarla ilişkili olarak optimizasyon yönteminin az popülasyon büyüklüğü ile sonuç üretmeye uygun olması gibi probleme özel hususlar göz önünde bulundurularak standart ABC algoritması üzerinde iki noktada değişim yapılmış ve bu yeni algoritma çalışmada değiştirilmiş ABC (modified ABC-mABC) olarak anılmıştır. Bahsedilen değişimlerden biri gözcü arı aşmasıdır. Standart ABC'de gözcü arılar yiyecek kaynaklarını (çözümleri) kalite değerlerini dikkate alarak ve en kötü kalite değerine sahip olana bile şans vererek seçmektedirler. mABC'de ise gözcü arıların tamamı popülasyondaki en iyi çözümü seçmektedir. Bir diğer değişiklik ise kaşif arı aşması ile ilgilidir. Standart ABC algoritmasında her çevrimde en fazla bir adet kaşif arı ortaya çıkabilirken mABC'de böyle bir sınır bulunmamaktadır, eğer bir çözümün deneme sayacı değeri limiti aşarsa bütün görevli arılar kaşife dönüşmekte ve rastgele üretilen çözümlerle popülasyon yeniden başlatılmış olmaktadır. Geliştirilen yöntemin performansını test etmek için simülasyon çalışmaları yapılmıştır. Hareketli ve sabit engellerin farklı sayılarda bulunduğu çeşitli senaryolar dikkate alınmış, mABC ve standart ABC algoritmalarının işletildiği yol planlama sonuçları katedilen toplam yol ve toplam zaman gibi kriterler üzerinden kıyaslanmıştır.

Tezin birinci bölümünde hareketli robot yol planlamasından, engellerin türlerinden ve ele alınan problemin detaylarından bahsedilmiş, literatür çalışması sunulmuştur. İkinci bölümde, problemin çözümünde kullanılan yöntemin detaylı anlatımı mevcuttur. Üçüncü bölümde yapılan deneysel çalışmalara ve elde edilen sonuçlara yer verilmiştir.

Tartışma, sonuç ve ilerleyen zamanlarda yapılabilecek çalışmalar konusundaki öneriler ise dördüncü bölümde sunulmuştur.



1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Hareketli Robot Yol Planlama Problemi

Birçok farklı sektörde günlük hayatı kolaylaştıracak çeşitli robotik sistemler bulunmaktadır ve her bir sistem belirli bir göreve sahiptir [5]. Bu sistemlerden bazıları üretimde, akıllı evlerde, askeri ve savunma sanayiinde, tıp alanında, havaalanlarında, çeşitli mühendislik alanlarında, laboratuvarlarda ve hayatımızın daha birçok alanında kullanılmaktadır. Hareketli robotun ortam hakkında çeşitli şekillerde edindiği bilgiler, ulaşmaya çalıştığı bir hedef ve karşısına çıkabilecek farklı niteliklerde engeller söz konusudur. Hareketli robotlar için yol planlama probleminde temel olarak, robotun engellere çarpmadan en uygun yolu tespit ederek hedefe varması sağlanmaya çalışılır. Problem, robotun ortam hakkında bilgi edinme şekline ve engellerin yapısına göre biçimlenmektedir.

Küresel yol planlamada robot ortama dair tüm bilgilere hakimdir. Bu yaklaşımda robot yola çıktıktan sonra herhangi bir hesaplama yapmaz, engellere çarpmamak koşuluyla en başta oluşturulan rotayı takip eder ve hedefe ulaşır. Çevrimdışı olarak da bilinen bu yöntem statik bir yaklaşımdır.

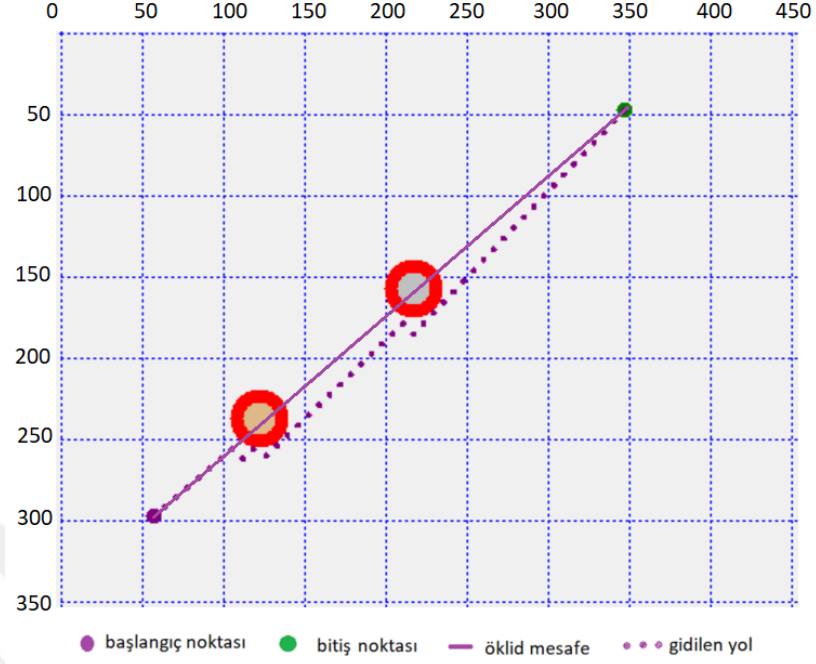
Bölgesel yol planlamada ise robot sensörleri sayesinde bilgi edinir, her bir hareketinde kendi kapsama alanı içindeki ortama dair kısmi bir bilgiye sahiptir. İlerlemeye başlamadan önce robotun ortama ve ortamda bulunan engellere dair bilgisi bulunmamaktadır. Bu sebeple robot takip edeceği yolu her hareketi için ilgili algılamayı yaptıktan sonra belirler. Çevrimiçi olarak da bilinen bu yöntem dinamik bir yaklaşımdır.

Yol planlama problemlerinde önemli bir etken de engellerin türleri ve şekilleridir. Engel şekilleri günlük hayatta olduğu gibi çok değişik şekillerde karşımıza çıkabilir. Çeşitli geometrik şekillerde olabileceği gibi, geometrik olmayan herhangi bir şekilde de

olabilir. Robotun karşılaşılabileceği engellerin geometrik şekilleri, hareketli veya durağan oluşu, hareketliyse hızı, hızlanması, yörüngesi vb. hususlar da problemi şekillendirmektedir.

Engeller, türleri bakımından hareketli ve durağan olmak üzere iki kategoride düşünülebilir. Hareketli engeller önceden belirlenmiş bir rota boyunca hareket edebilir. Bu durumda gideceği bir sonraki konum öncesinde bellidir. Ya da bir hareketli engelin gideceği yol tamamen rastgele olabilir. Sabit engellerin ise konumu değişmez. Sabit engelli problemlerde robot engelin konumunu öncesinde bilmektedir ya da bir kez öğrendikten sonra hafızasına almaktadır. Sabit engel olan ortamlardaki problemler ve bu problemlerin çözümü için uygulanan yöntemler farklı, hareketli engelli ortamdaki problemler ve bu problemlerin çözümleri farklıdır.

Bu tez çalışmasında iki boyutlu ortamda bir hareketli robot ile sabit ve hareketli engellerin olduğu varsayılmıştır. Hareketli engellerin bir sonraki konumu rastgele belirlenmektedir. Engeller daireseldir ve yarıçapları 10 birimdir. Güvenlik yarıçapı ise 8 birimdir. Engelin toplam yarıçapı 18 birim olarak belirlenmiştir. Robot noktasaldır ve 9 birimlik yarıçapa sahip dairesel bir algılama alanı vardır. Robotun sabit ve hareketli engellerin bir arada bulunduğu ortamda hedefe herhangi bir çarpışma olmadan ve en kısa yoldan ulaşması amaçlanmıştır. Bu amaçla engellerin yarıçapına bir güvenlik değeri eklenmiş, engelin yarıçapı; engelin gerçek yarıçapı ile bu güvenlik değerinin toplamı kabul edilmiştir. Robot yeni konum belirlerken tüm engellerin durduğu kabul edilmiştir. Bulunan bu yeni konum robotun algılama alanı içinde olmalıdır. Robot hedefe ulaştınca durmaktadır. Şekil 1.1’de sabit engellerden oluşan ve ABC algoritması uygulanarak başlangıçtan hedefe başarılı bir şekilde ulaşan robot için yapılan bir simülasyon işlemi gösterilmiştir.



Şekil 1.1. İki sabit engel bulunan ortamda robotun hedefe ulaşırken izlediği yol

1.2. Literatür Taraması

Robotik yol planlama alanında birçok çalışma yapılmıştır. Yapılan çalışmalar, robotun iki ya da üç boyutlu ortamlarda, sabit ya da hareketli engellerle çarpışma olmadan başlangıç noktasından hedef noktasına en optimum yoldan ulaşabilmesini amaçlamaktadır. Robotik yol planlama için kullanılan her bir algoritma belirli ortamlarda daha kullanışlıdır. Bu algoritmalar klasik ve modern olmak üzere sınıflandırılmıştır [9].

A-star, D-star gibi ızgara tabanlı algoritmalar, dijkstra, geometrik algoritmalar ve olasılıksal yol haritası (Probabilistic Road Map-PRM), hızla keşfedilen rastgele ağaç (Rapidly Randon Tree-RRT) gibi örnekleme tabanlı algoritmalar klasik algoritmalara örnektir [5]. Genetik algoritma (Genetic Algorithm-GA), ABC, karınca koloni algoritması (Ant Colony Algorithm-ACO) ve parçacık sürü optimizasyonu algoritması (Partcal Swarm Optimization-PSO) ise modern algoritmalara örnektir [5]. Klasik ve modern algoritmalar arasındaki temel fark, modern algoritmaların klasik algoritmalarından daha hızlı optimal bir çözüm elde edebilmesidir [10].

Klasik yöntemler kullanılarak yapılan yol planlama çalışmaları aşağıdaki gibidir:

Lamini ve arkadaşları, (2018) GA kullanarak statik ortamda yol planlama problemlerini çözmek için geliştirilmiş bir çapraz geçiş operatörü önermektedirler [4]. GA, engellerden kaçınırken iki konum arasında geçerli ve uygulanabilir bir yol bulmayı ve mesafe, güvenlik gibi bazı kriterleri optimize etmeyi içeren yol optimizasyonu probleminde yaygın olarak uygulanmıştır [4]. Önerilen çapraz geçiş operatörü, erken yakınsamayı önler ve ebeveynlerinden daha iyi uygunluk değerine sahip uygun yollar sunar, böylece algoritma daha hızlı yakınsar.

Hussein ve arkadaşları, mobil robot yol planlama problemine meta-sezgisel optimizasyona dayalı bir yaklaşım sunmaktadır [10]. Yörünge tabanlı meta-sezgisel optimizasyon ve popülasyon tabanlı meta-sezgisel optimizasyon arasında karşılaştırmalı bir çalışma yapılmıştır. Tabu arama, genetik algoritma ve benzetilmiş tavlama algoritması tarafından oluşturulan yolları karşılaştıran optimum çözümü bulmak için genişlik öncelikli deterministik arama kullanılmıştır. Deneysel çalışma, benzetilmiş tavlamanın hesaplama süresi açısından diğer algoritmalarından daha iyi performans gösterdiğini, tabu aramasının ise en kısa yolu verdiğini göstermektedir.

Guruji ve arkadaşları, işlem süresini en aza indirmeyi hedefleyerek engellere çarpmadan en uygun yolu bulmak için geliştirilmiş bir A-star algoritması sunmuşlardır [11]. Önerilen A* algoritması, sezgisel fonksiyonun değerini başlangıçta değil, çarpışma aşamasından hemen önce belirler ve işlem süresinde belirgin bir azalma sağlar. Yapılan çalışmada işlem süresindeki azalma sayesinde %95'e kadar bir zaman tasarrufu sağlandığı gözlenmiştir [11].

Duchon ve arkadaşları, ızgara şeklinde belirlenen bir harita üzerinden geliştirilmiş bir A-star algoritması temelli bir yol bulma yöntemi önermişlerdir [12]. Burada amaç, ızgara hücrelerinin sayısı azaltılarak çalışmayı gerçekleştirmektir. Elde edilen sonuçlara göre yol bulma hızı artmış, ancak daha uzun yollarda bu yöntem dezavantaj sağlamıştır.

Zu ve arkadaşları, çoklu insansız hava aracı (İHA) takımlarının birbirleri ile uyumlu olarak gerçek zamanlı bilinmeyen dinamik ortamda herhangi bir çarpışma olmadan hedefe ulaşmak amacıyla geliştirilmiş bir RRT yaklaşımı önermiştir [13]. RRT ile oluşan yoldaki gereksiz düğümlerin ortadan kaldırılması ve engel hızlarını belirleme üzerine odaklanılmıştır. Bu yöntem dinamik engellerin daha hızlı tespit edilmesini sağlamıştır.

Al-Dahhan ve Schmidt, çok yönlü robotlar için bir yol planlama algoritması önermektedir [14]. Amaç robot ortamındaki engellere güvenli bir mesafe bırakan kısa bir yolun hesaplanmasıdır. Önerilen algoritma, Voronoi diyagramlarının güvenlikle ilgili avantajlarını, en kısa yolları bulmak için olasılıksal yol haritası(Probabilistic Roadmap) algoritmasının faydalarıyla birleştirmektedir.

Metasezgisel yöntemler kullanılarak yapılan yol planlama çalışmaları aşağıdaki gibidir: Rashid ve arkadaşları, statik engellerden oluşan haritada hareketli robot yol planlama probleminin çözümü için ACO kullanmışlardır [15].

Raja ve Pugazhenthı, dinamik ortamlarda mobil robot yol planlama problemleri için en uygun çözümü hızlı bir şekilde belirleyebilen bir PSO uygulamışlardır [16]. Değişen hızlara sahip farklı şekillerdeki (dışbükey, içbükey ve eğimli) engeller dikkate alınmış olup yöntem, daha önceki evrimsel yaklaşımlarda olduğu gibi, geçersiz parçacıkları arama ihtiyacını ve dolayısıyla ceza fonksiyonu değerlendirme tasarımını ortadan kaldırarak, popülasyonda yalnızca geçerli parçacıkları (engelleri engellemeden hedeflemeye başlar) kullanır. Oluşturulan yollar, global optimal yolları elde etmek için PSO algoritmasına tabi tutulmuştur.

Ngah ve arkadaşları, serbest ortamda Distbug[17], TangenBug ve PointBug algoritması kullanılarak oluşturulan yörüngeleri karşılaştırmışlardır [18]. Bu makale, hedefin dönüm noktası olarak engel alanının dış çevresinde birkaç önemli noktayı arayarak bir engelin engel sınırının kullanımını en aza indirmeye çalışan PointBug [18] adlı yeni bir algoritmayı kullanan bir yaklaşımı tanıtmaktadır. Herhangi bir şekle sahip sabit engellerle dolu, bilinmeyen bir ortamdaki bir robot, başlangıç noktasından hedefe doğru hareket edeceği yeri belirler. Bir sonraki nokta, sensörden en yakın engele olan anlık uzaklık değişikliğini algılayan menzil sensörünün çıktısı ile belirlemektedir.

Sedighi ve arkadaşları, farklı engel konfigürasyonlarına sahip farklı büyüklükteki arama alanlarında GA ile yol planlama simülasyonları yapmışlardır. Bu çalışmada amaç mobil robotun başlangıç noktası ile hedef bitiş noktası arasındaki adımlarının sayısını en aza indiren yolu bulmaktır [19]. Önerilen kontrol algoritması, dört komşuluk hareketine izin vermektedir, böylece yol planlaması, düşük karmaşıklıklara sahip karmaşık arama alanlarına uyum sağlamaktadır.

Doğada var olan zeki davranışlar içeren süreçlerin gözlemlenmesi bu konularda çalışma yapan araştırmacıları yeni optimizasyon metotları geliştirmek için yönlendirmiştir. 2005 yılında Karaboğa, arıların yiyecek arama davranışını modelleyerek ABC algoritmasını literatüre tanıtmıştır [7]. ABC algoritması sürü zekasına dayalı bir algoritma olup pek çok farklı problemde optimal çözümleri bulabilmektedir [9]. Sürekli ve ayrık optimizasyon problemlerini çözmek için sıklıkla kullanılmaktadır [20]. Aşağıda ABC Algoritması kullanılarak yapılan bölgesel yol planlama çalışmalarına yer verilmiştir.

Liang ve Lee, hedef, engeller ve robotların çarpışmasından kaçınma için uygun optimum yolu seçerek birden fazla mobil robotun çevrimiçi yol planlamasını çözmek için bir ABC algoritması versiyonu önermişlerdir [2]. Önerilen algoritma, seçilen bireyleri kullanarak performansı artırır. Anlık güncelleme stratejisi en yeni çözüm bilgilerini sağlamaktadır. Bu ABC algoritması versiyonu, robotun uygulanabilir bir yol bulması için attığı her adımda çağrılmaktadır ve evrimsel programlama kullanılarak olası çarpışmalardan kaçınılmaktadır. Önerilen yaklaşımla, her robotun sonraki konumları tasarlanır. Böylelikle mobil robotlar hedeflere çarpışmadan gidebilmektedir.

Faridi ve arkadaşları, bilinmeyen dinamik bir ortamda çok etmenli (multi-agent), çok hedefli (multi-target) navigasyon problemini çözmek için bir evrim şeması önermişlerdir [3]. Genel olarak, tam yol oluşturma süreci iki aşamaya bölünmüş olup, ilk aşamada, uygun bir yol oluşturmak için ABC algoritmasını uygulamışlardır. İkinci aşamada ise yolu optimize etmek için EP uygulanmaktadır. Önerilen yeni yaklaşımın performansı GA, A* algoritması ve RRT algoritmasının performansı ile karşılaştırılmıştır.

Nyassi, 2 boyutlu uzayda bir başlangıç noktası, durağan bir hedef ve durağan elips şeklinde engellerin bulunduğu ortamlar dikkate alınarak bir hareketli robot için bölgesel yol planlama temelli yaklaşımlar ile robotun engellere çarpmadan başlangıçtan hedef noktasına ulaşmasını sağlamaya çalışmıştır [5]. Bunu yaparken ABC algoritmasına dayalı yaklaşımlardan hızlı ABC (quick ABC-qABC)'yi kullanmıştır. qABC, gözcü arıların davranışını yeniden modellemekte ve bölgesel minimuma yakınsama hızı açısından standart ABC algoritmasının performansını arttırmaktadır [6]. Bu çalışmada ilk olarak robot doğrusal olarak hedefe doğru harekete başlar. Eğer Öklid mesafesine göre hedefe giden en kısa yol üzerinde engel varsa qABC algoritması çalıştırılır, eğer

yoksa öklid mesafesine göre robotun algılama alanına giren ve hedefe en yakın olan nokta hareket edilecek nokta olarak belirlenir. Yapılan testler sonucunda qABC'nin bazı senaryolarda ABC'ye göre daha iyi sonuçlar verdiği görülmüştür.

Kumar ve arkadaşları, Evrimsel Programlama(EP)'ya dayalı yeni bir yöntem(ABC-EP) önermişlerdir. En iyi yiyecek kaynağını seçme kriteri olarak, belirlenen yiyecek kaynağının hedefe olan en kısa mesafede olmasının yanı sıra o yiyecek kaynağının en yakın engellere olan mesafesini de dikkate almışlardır. Önerilen ABC-EP'nin, yol uzunluğunda %5,75, arama maliyetinde %44,38 ve yol düzgünlüğünde %41,08 iyileştirme ile ABC-EP'ye göre daha etkin bir performansa sahip olduğu görülmüştür[21].

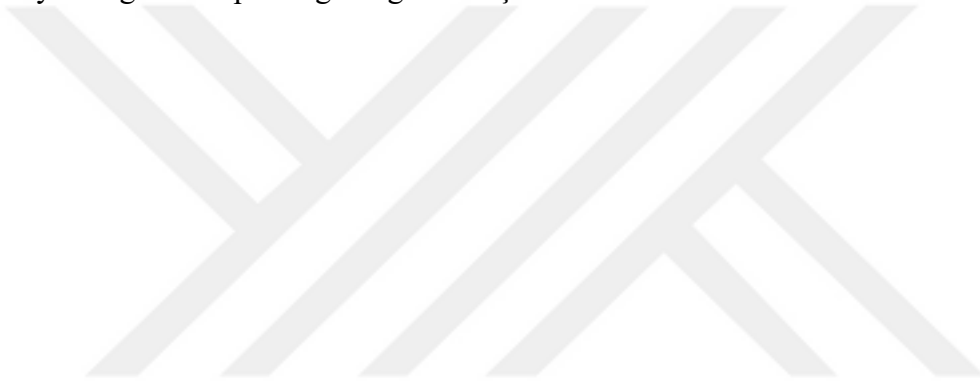
Cruz ve arkadaşları, ABC algoritmasını yerel bir arama prosedürü olarak ve evrimsel programlama algoritmasını bir dizi yerel prosedür tarafından bulunan uygun yolu iyileştirmek için birleştiren bir yöntem önermişlerdir [22].

Xu ve arkadaşları, ABC algoritmasının yavaş yakınsama hızının ve tek boyutlu arama stratejisinin dezavantajlarından kurtulmak için bir çalışma önermişlerdir. ABC'ye bir birlikte evrim çerçevesi sunmuşlar ve yakınsamasını hızını artırmak ve boyutun bağımlılığını tek tek gözlemlemek için geliştirilmiş bir yöntemle sahip Birlikte Evrim Küresel En İyi Yapay Arı Kolonisi Algoritması (Global Best Guided Artificial Bee Colony Algorithm Co-GABC)'ni tasarlamışlardır [23]. Yapılan testler sonucu önerilen algoritmanın verimli sonuçlar ürettiği görülmüştür.

Girgas ve arkadaşları, bölgesel yol planlama problemini çözmek için bir ABC algoritmasının kullanıldığı bir yaklaşım önermişlerdir [24]. Bu çalışmada ABC ve PSO algoritması tanıtılmış ve iki algoritma arasında karşılaştırma yapılmıştır. PSO algoritması dalgalı sonuçlar vermiş olup, ABC 'nin yakınsama yeteneğinin PSO'ya göre daha iyi, yakınsama hızının ise PSO'ya göre daha düşük olduğu gözlemlenmiştir. Yapılan deneysel çalışmalar sonucunda ABC'nin, PSO'dan daha iyi bir performans gösterdiği görülmüştür.

Abbas ve Abdulsahab, üç durum çalışması için bir uyarlanabilir çok amaçlı ABC (Adaptive Multi-Objective Artificial Bee Colony - AMOABC) algoritmasına dayalı bir optimal yol planlama algoritması önermişlerdir [25]. Bu çalışmada üç durum vardır. İlk

durumda iki farklı aşama söz konusudur. İlkinde iki farklı robot farklı başlangıç ve bitiş noktalarına sahiptir. İkincisinde ise robotların başlangıç noktaları farklıdır ancak hedef noktaları aynıdır. İlk durumdaki amaç, robotun başlangıç konumundan hedefe kadar ihtiyaç duyduğu minimum mesafeyi bulmaktır. İkinci durumdaki amaç ise, iki robotun yolları arasındaki maksimum mesafeyi bulmaktır. İkinci durum, üç ve dört robot için en kısa ve çarpışmanın olmadığı yolu bulmaktır. Son durum, aralarında herhangi bir çarpışma olmaksızın beş robot için en kısa yolu en sorunsuz ve en kısa sürede bulmaktır. Elde ettikleri sonuçlar, AMOABC'nin çok amaçlı bir ABC versiyonundan daha hızlı bir yakınsama ile yerel optimumlardan daha iyi bir şekilde uzaklaşma yeteneğine sahip olduğunu göstermiştir.



2. BÖLÜM

YÖNTEM VE MATERYAL

Bu bölümde, ABC algoritması, tez çalışması kapsamında geliştirilmiş bir ABC versiyonu olan mABC algoritması ve bu algoritmalara dayanan yol planlama yaklaşımı açıklanmaktadır.

2.1. ABC Algoritması

ABC algoritması, bal arılarının yiyecek arama ve birbirleri ile haberleşme şekillerinden yola çıkılarak 2005 yılında Karaboğa tarafından geliştirilmiş bir algoritmadır [7]. Popülasyon temelli bir yapısı vardır ve bir çözüm bir yiyecek kaynağı pozisyonu ile ilişkilendirilmiştir. Kolonideki iş bölümünde görevli ve görevsiz arılar şeklinde temel bir ayırım söz konusudur. Görevsiz arılar da kendi aralarında gözcü ve kaşif arı olarak ayrılmaktadır. Duruma göre görevli arı kaşif arıya dönüşebilmekte, kaşif arı da görevli arıya dönüşebilmektedir. Standart ABC algoritmasının kontrol parametreleri durdurma kriteri, limit, kaynak sayısı olarak sıralanabilir. Durdurma kriteri; algoritmanın ne zaman sonlandırılacağını belirler. Bu şekilde maksimum çevrim (iterasyon) sayısı dikkate alınabileceği gibi maksimum fonksiyon değerlendirme sayısı, saniye bazlı maksimum çalıştırma zamanı, popülasyondaki gelişim durumu vb. de durdurma kriteri olarak kullanılabilir. Limit; kaşif arı için yiyecek kaynağının tükenip tükenmediğini belirlemek için kullanılan bir değeri ifade eder. Kaynak sayısı; popülasyon büyüklüğünü gösterir. Koloni büyüklüğü; kolonideki görevli ve gözcü arıların toplam sayısıdır. Standart ABC algoritmasında görevli arı sayısı, gözcü arı sayısı ve kaynak sayısı birbirine eşittir. Yani koloni büyüklüğü yiyecek kaynağı sayısının başka bir deyişle popülasyon büyüklüğünün 2 katıdır.

Başlangıç Popülasyonunun Oluşturulması:

Algoritma arama bölgesindeki rastgele çözümlerin yani yiyecek kaynağı pozisyonlarının oluşturulmasıyla başlar. Bu durumda yiyecek kaynağı sayısı kadar kaşif arının bu kaynakları bulduğu düşünülmektedir.

Görevli Arılar:

Görevli arılar, kovandan ayrıldıklarında akıllarında belirli bir yiyecek kaynağı pozisyonu olan arılardır. Kaşif arılar yeni yiyecek kaynaklarını belirledikten sonra görevli arı olarak çalışmaya devam ederler. Her bir görevli arı, yiyecek kaynağının durumu ile ilgili bilgileri diğer arılara iletir.

Görevsiz Arılar:

Nektar toplamak için kaynak bulmaya çalışan arılardır. Görevsiz arılar, gözcü arılar ve kaşif arılardan oluşur. Görevli arılar, buldukları kaynak ile ilgili bilgileri dans etme yoluyla diğer arılara aktarır. Gözcü arılar, görevli arıların paylaştığı bilgiler doğrultusunda yeni kaynağa yönelen arılardır. Kaşif arılar diğer arılardan etkilenmeden, rastgele kaynak arayışında olan arılardır. Eğer bir görevli arının ilgilendiği kaynak tükenmiş ise görevli arı kaşif arıya dönüşür ve eski kaynağı terk ederek yeni bir kaynak aramaya başlar.

Yiyecek Kaynağı:

Yiyecek kaynağının uygunluğunun bağlı olduğu çeşitli faktörler vardır. Bunlar kaynağın nektar miktarı, türü, arıların yuvasına yakınlığı, nektarın kalitesi, nektara ulaşmanın kolaylığı olarak sıralanabilir. ABC algoritmasında bir yiyecek kaynağı pozisyonu ele alınan optimizasyon probleminin bir çözümüne karşılık gelmektedir. Bu yiyecek kaynağının kalitesi ise çözümün uygunluk değeridir.

ABC algoritmasının temel adımları aşağıdaki şekildedir;

Başlangıç aşaması

REPEAT

Görevli arı aşaması

Gözcü arı aşaması

Kaşif arı aşaması

UNTIL (durdurma kriteri sağlanana kadar)

1. Başlangıç aşaması

ABC algoritması, rastgele yiyecek kaynağı üretmekle çalışmaya başlamaktadır. Rastgele yiyecek kaynağı üretme süreci her bir karar değişkeni için alt ve üst sınırlar arasında olacak şekilde rastgele değer üretilerek Eşitlik 2.1 ile gerçekleştirilir. $x_{k,j}$ k . yiyecek kaynağının j . boyutunda yer alan değerdir. $rand(0, 1)$, 0 ile 1 arasında üretilen rastgele bir sayıdır. ub_j ve lb_j ise j . karar değişkeni için üst ve alt sınırları göstermektedir.

$$x_{k,j} = lb_j + rand(0,1) \times (ub_j - lb_j) \quad (2.1)$$

Bütün yiyecek kaynaklarına başlangıç pozisyon değerleri atandıktan sonra yani başlangıç çözümleri üretildikten sonra bu çözümler değerlendirilir ve en iyi çözüm hafızaya alınır.

2. Görevli arı aşaması

Her bir kaynaktan sorumlu bir görevli arı bulunmaktadır. Bu sebeple görevli arı sayısı ile yiyecek kaynağı sayısı birbirine eşittir. Her bir görevli arı sorumlu olduğu kaynak komşuluğunda yeni bir yiyecek kaynağı bularak bulduğu yiyecek kaynağının kalitesini değerlendirir. Eğer yeni bulunan kaynak eskisinden daha iyi ise eski kaynağı bırakarak yeni kaynak için çalışmaya başlar. Yeni kaynağın belirlenmesi işlemi Eşitlik 2.2 ile yapılmaktadır.

$$v_{k,j} = x_{k,j} + \varphi_{k,j} \times (x_{k,j} - x_{m,j}) \quad (2.2)$$

x_m popülasyondan rastgele seçilmiş bir yiyecek kaynağını, j rastgele seçilmiş bir boyutu göstermektedir. Burada m, k 'dan farklı bir değerdir. $\varphi_{k,j}$ ise $[-1,1]$ aralığında seçilmiş rastgele bir sayıyı göstermektedir. Aday çözüm v_k bulunduktan sonra bulunan çözümün uygunluğu hesaplanır ve x_k ile v_k arasında açgözlü seçim yöntemi uygulanır [7,8]. Uygunluk değeri Eşitlik 2.3 ile belirlenmektedir.

$$fit(x_k) = \begin{cases} \frac{1}{1+f(x_k)}, & \text{eğer, } f(x_k) \geq 0 \\ 1 + abs(f(x_k)), & \text{eğer, } f(x_k) < 0 \end{cases} \quad (2.3)$$

Burada $f(x_k)$, x_k çözümünün amaç fonksiyonu değeri, $fit(x_k)$ ise uygunluk değeridir.

3.Gözcü arı aşaması

Bir gözcü arı, kendisine iletilen bilgi sayesinde kaynakların nektar miktarlarına göre olasılıksal bir kaynak seçimi yapmaktadır. Olasılıksal seçme işlemi, Eşitlik 2.4'te gösterilen denklem ile yapılır. Kaynakların seçilme olasılıkları bu şekilde belirlenmektedir.

$$p_k = \frac{fit(x_k)}{\sum_{i=1}^{SN} fit(x_i)} \quad (2.4)$$

p_k k . çözümün seçilme olasılığını, SN ise popülasyondaki toplam yiyecek kaynağı sayısını ifade etmektedir. Bu olasılık değerleri dikkate alınarak her bir gözcü arı için bir kaynak (çözüm) seçilir. Daha sonra ilgili gözcü arı görevli arılara benzer şekilde, seçilen çözüm etrafında Eşitlik 2.2'yi kullanarak komşuluk araştırması yapar ve bir aday çözüm üretir. Aday çözüm mevcut çözümden daha iyi ise mevcut çözüm yerine aday çözümle devam edilir.

4.Kaşif arı aşaması

Her iterasyon sonunda kaynakların tükenme durumları değerlendirilerek bir kaşif arının ortaya çıkıp çıkmayacağı kontrol edilir. Bunun için her bir çözüm ile ilişkili olarak bir deneme sayacı tutulur. Bu sayacın değeri görevli veya gözcü arılar tarafından çözüm etrafında komşuluk araştırması yapıldığı halde daha iyi bir çözüm bulunamadığında

artırılır, bulunursa yeni çözümle devam edileceği için deneme sayacı değeri sıfırlanır. Gözcü arı aşamasından sonra en iyi çözüm hafızaya alınır ve maksimum deneme sayacı değerine sahip olan çözüm tespit edilir. Eğer bu sayaç değeri limit parametresinin değerinden fazla ise kaynak tükenmiştir ve ilgili görevli arı bu kaynağı terk ederek kaşif arıya dönüşür, Eşitlik 2.1'i kullanarak yeni yiyecek kaynağı bulur. Yeni kaynağı belirledikten sonra yine görevli arı olarak çalışmaya devam eder. Bu şekilde kaşif arı aşamasında, etrafında belirli bir miktardan fazla komşuluk araştırması yapıldığı halde iyileştirilemeyen bir çözüm popülasyondan çıkarılmakta, onun yerine rastgele üretilmiş bir çözüm koyulmaktadır.

2.2. mABC Algoritması

Bu tez çalışması kapsamında mABC algoritması ile standart ABC için önerilen değişiklikler şu şekildedir:

1. Görevli arı aşaması bittikten sonra popülasyon en iyi çözüm bulunmakta ve bütün gözcü arılar rulet yöntemini uygulamak yerine bu yiyecek kaynağını seçmektedir.
2. Kaşif arı aşamasında eğer bir çözüm için limit değeri aşılmış ise sadece mevcut çözüm için değil tüm çözümler için rastgele yeni çözümler üretilmektedir ve böylelikle tüm popülasyon sıfırdan başlatılmaktadır.

mABC algoritmasının temel adımları Şekil 2.1'de verilmiştir.

Başlangıç aşaması

Her bir yiyecek kaynağı için:

2.1'i kullanarak rastgele başlangıç pozisyon değeri ata.

Atanan rastgele başlangıç değerinin uygunluk değerini hesapla.

Deneme sayacı oluştur ve başlangıç için 0 ata.

En iyi çözümü hafızaya al.

REPEAT*Görevli Arı Aşaması*

Her bir görevli arı için:

2.2'yi kullanarak mevcut çözüm komşuluğunda yeni bir aday çözüm üret.

2.3'ü kullanarak aday çözümün uygunluk değerini belirle.

Üretilen aday çözüm eğer mevcut çözümden daha iyi ise deneme sayacını sıfırla ve mevcut çözümü aday çözüm ile güncelle. Eğer aday çözüm mevcut çözümden daha iyi değilse deneme sayacı değerini bir artır.

Popülasyonda en iyi çözümün bulunduğu indisi (b) belirle.

Gözcü arı aşaması

Her bir gözcü arı için:

Geliştirmek üzere popülasyonda (b) indisinde yer alan çözümü seç.

2.2'yi kullanarak mevcut çözüm komşuluğunda yeni bir aday çözüm üret.

2.3'ü kullanarak aday çözümün uygunluk değerini belirle.

Üretilen aday çözüm eğer mevcut çözümden daha iyi ise deneme sayacını sıfırla ve mevcut çözümü aday çözüm ile güncelle. Eğer aday çözüm mevcut çözümden daha iyi değilse deneme sayacı değerini bir artır.

Kaşif arı aşaması

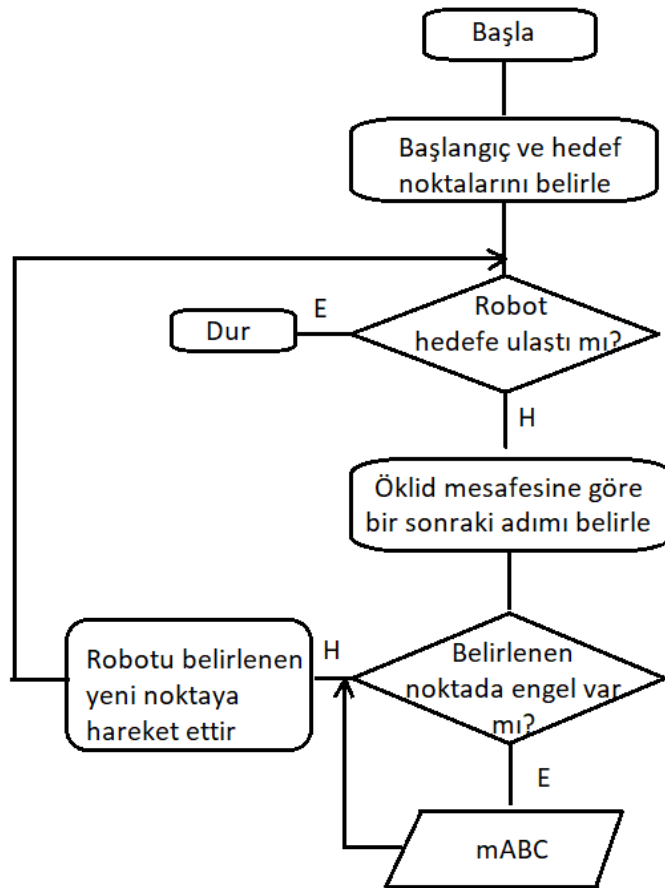
Maksimum deneme sayacı değeri limit değerinden büyükse popülasyonu yeniden başlat.

UNTIL (durdurma kriteri sağlanana kadar)

Şekil 2.1: mABC algoritmasının adımları

2.3. Kullanılan Yol Planlama Yaklaşımı

Önerilen çalışmanın temel amacı, tek bir hareketli robot için hedefe giden en kısa yolu engellere çarpmadan bulmaktır. Eğer Öklid mesafesine göre en kısa yol üzerinde herhangi bir engel bulunmuyorsa, robot algılama alanı içerisindeki bu noktaya ilerler [5]. Yani hedefle arasında Öklid mesafesinin hesaplandığı doğrultu üzerinde algılama alanı yarıçapı kadar ilerler. Ancak algılama alanı içerisinde engel bulunuyorsa mABC çalıştırılır ve robotun hareket etmesi gereken güvenli nokta belirlenerek robotun bu noktaya gitmesi sağlanır.



Şekil 2.2. Önerilen yol bulma yaklaşımına ait akış şeması

Yol planlama problemi için mABC çalıştırılırken [5]'de olduğu gibi denklem 2.6 ile çözümlerin başlangıç değerleri belirlenmiştir.

$$x_p = x_i + \varphi \times r \times \cos(\theta) \quad (2.6)$$

$$y_p = y_i + \varphi \times r \times \sin(\theta)$$

$\varphi; [0, 1]$ arasında rastgele bir değeri, r ; hareketli robotun bir sonraki konumu için arama yaptığı dairesel alanın yarıçapını, θ ; $[0, 2\pi]$ arasındaki rastgele bir değeri, x_i ve y_i ; robotun o andaki konumunu, x_p ve y_p ise robotun bir sonraki muhtemel konumunu ifade eder [2].

Yapılan çalışmada, mABC’de aday çözüm üretme aşamasında robotun algılama alanı dışında bir noktaya gitmesini engellemek için [5]’de kullanılan teknik uygulanmıştır. Bu tekniğe göre, eğer üretilen aday çözüm robotun algılama alanı dışındaysa, robotun mevcut konumu ile aday çözümün arasındaki Öklid mesafesine göre en kısa yolu veren doğru parçası bulunur. Daha sonra robotun algılama alanı sınırı ile bu doğru parçasının kesiştiği nokta belirlenir. Bu nokta yeni aday çözüm olarak kabul edilir.

Robotun temel hedefi ortamdaki engellere çarpmadan hedefe ulaşmaktır. Bunu sağlamak için [5]’de olduğu gibi ortamda bulunan engellerin yarıçaplarına bir güvenlik değeri SV eklenir. Engelin toplam yarıçapı, r_{obs} , kendi yarıçapıyla SV değerinin toplamı olarak kabul edilir (Eşitlik 2.7).

$$r_{obs} = \text{engelin kendi yarıçapı} + SV \quad (2.7)$$

Robot, hedefe giden en kısa yolda engel olup olmadığını anlamak için gitmek istediği nokta ile engelin merkez noktası arasındaki mesafeyi hesaplar. Eğer bu mesafe r_{obs} değerinden büyükse ortamda herhangi bir çarpışma tehlikesi yok demektir. Fakat mesafe r_{obs} ’den küçükse veya r_{obs} ’ye eşitse bu durumda çarpışma tehlikesi var demektir. Bu durumda mABC algoritması çalıştırılarak robot için yeni ve güvenli bir nokta bulunur.

Robotun engellere takılmadan en kısa yoldan hedefe ulaşması için mABC’de kullanılan amaç fonksiyonu [2]’deki gibidir. x_k çözümünün amaç fonksiyonu değeri $Obj(x_k)$ Eşitlik 2.8 ile hesaplanmaktadır.

$$Obj(x_k) = c_1 \times Obj_1(x_k) + c_2 \times Obj_2(x_k) + c_3 \times Obj_3(x_k) \quad (2.8)$$

$Obj(x_k)$, 3 alt amaç fonksiyonunun birlikte kullanılmasıyla oluşur. Bunlardan ilki olan $Obj_1(x_k)$, hedef noktaya en kısa mesafeyi ifade etmektedir. Eşitlik 2.9'da gösterilmiştir.

$$Obj_1(x_k) = \|x_k - Hedef\| \quad (2.9)$$

Bir diğer alt amaç fonksiyonu $Obj_2(x_k)$, robotun sabit engellerle çarpışmasını önlemektedir. Eşitlik 2.10 ve Eşitlik 2.11 kullanılarak değeri hesaplanmaktadır.

$$Obj_{2,k}(x_k) = \begin{cases} 0, & \text{eğer, } \|x_k - Obs_k\| > r_{obs_k} \\ \frac{1}{\|x_k - Obs_k\|} - \frac{1}{r_{obs_k}}, & \text{eğer, } \|x_k - Obs_k\| \leq r_{obs_k} \end{cases} \quad k = 1, 2, \dots, h \quad (2.10)$$

$$Obj_2(x_k) = \sum_{k=1}^h Obj_{2,k}(n_p) \quad (2.11)$$

$Obj_3(x_k)$ alt amaç fonksiyonu ise [2]'de ortamda bulunan diğer hareketli robotlarla çarpışmayı önlemek için kullanılmıştır. [2]'de yapılan çalışmada bu formülün hareketli engeller için de kullanılmasının uygun olduğu belirtildiğinden bu tez çalışmasında hareketli engellerle çarpışmayı önlemek için Eşitlik 2.12 ve Eşitlik 2.13 kullanılmıştır.

$$Obj_{3,i}(x_k) = \begin{cases} 0, & \text{eğer, } \|x_k - Obs_i\| > r_{obs_i} \\ \frac{1}{\|x_k - Obs_i\|} - \frac{1}{r_{obs_i}}, & \text{eğer, } \|x_k - Obs_i\| \leq r_{obs_i} \end{cases} \quad i = 1, 2, \dots, h \quad (2.12)$$

$$Obj_3(x_k) = \sum_{i=1}^h Obj_{3,i}(x_k) \quad (2.13)$$

c_1, c_2, c_3 toplamları 1'e eşit olan ve alt amaç fonksiyonlarının ağırlıklarını ayarlamak için kullanılan pozitif ağırlık değerleridir. Obs_k , sabit engeller için k. engeli ifade eder. Obs_i ise hareketli engeller için i. engeli ifade eder. h toplam engel sayısıdır. Eşitlik 2.10 için h toplam sabit engel sayısını, eşitlik 2.12 için h toplam hareketli engel sayısını ifade eder.

3. BÖLÜM

DENEYSEL ÇALIŞMALAR

Önerilen yaklaşımın performansını değerlendirmek için yapılan deneysel çalışmalara ve elde edilen sonuçlara bu bölümde yer verilmiştir.

3.1. Deneysel Çalışmalar

Bu çalışmada ızgara (grid) tabanlı ve 2 boyutlu 1200×700 birim²lik sanal bir ortam kullanılmıştır. Ortam, belirli sayıda hareketli veya durağan engel içerecek şekilde tanımlanmış bir alandan oluşmaktadır. Ortamda bir robot ve sabit noktasal bir başlangıç noktası ve sabit noktasal bir hedef noktası bulunmaktadır. Robotun dairesel ve 9 birimlik yarıçaplı bir algılama alanı mevcuttur.

Ortamda bulunan durağan ve hareketli engeller daire şeklinde olup her bir engelin yarıçapı 10 birimdir. Hareketli engeller sabit hızda hareket ederler. Hareket yönleri her bir harekette 5 birim ve her yöne olabilecek şekilde rastgele belirlenmiştir. Her bir engelin yarıçapına eklenen güvenlik değeri SV 8 birimdir.

Yapılan deneylerde temel olarak 2 farklı şekilde test yöntemi hazırlanmıştır. İlk olarak, 3 sabit engelin bulunduğu ve 4 hareketli engelin bulunduğu iki farklı senaryo belirlenmiş, bu senaryolar için metasezgiselin durdurma kriteri olarak 1000 milisaniye(ms) ve 5000 ms verilerek önerilen yol planlama yaklaşımının gösterdiği başarıyı gözlemlemek amacıyla çalışma yapılmıştır. İkinci yöntem ise önerilen yol planlama yaklaşımının genel performansından ziyade özel olarak içerisinde mABC'nin performansını incelemeye yöneliktir. Burada bir sabit engelin bulunduğu bir senaryoda robotun engelle ilk karşılaştığı yani metasezgisel ile gideceği konumu belirleyeceği aşamada metasezgisel sadece bir kez çalıştırılmak yerine 30 kez birbirinden bağımsız

bir şekilde çalıştırılarak bu aşama için kullanılan amaç fonksiyonu ve geçen süre üzerinden performans değerlendirilmesi yapılmıştır.

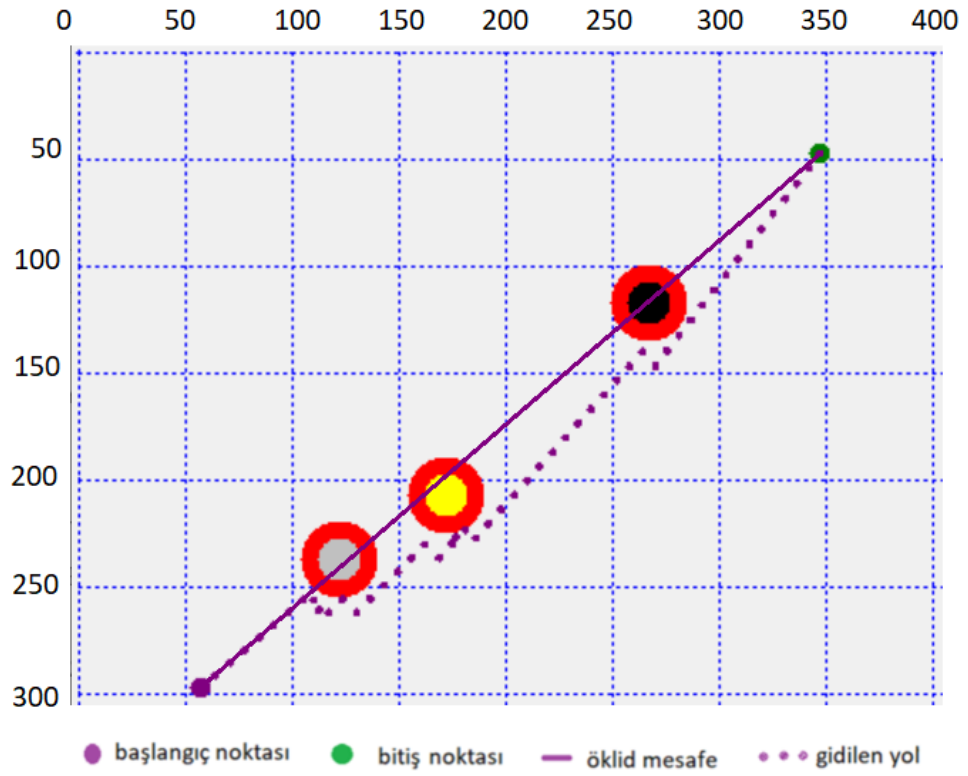
Deneysel çalışmalarda başlangıç noktası=(60,300), hedef noktası=(350,50) olarak belirlenmiştir. $c_1 = c_2 = 0.3$, $c_3 = 0.4$ olarak alınmıştır. mABC ve ABC için kullanılan parametre değerleri ise Tablo 3.1’de gösterilmiştir.

Tablo 3.1. ABC ve mABC algoritmaları için kontrol parametresi değerleri

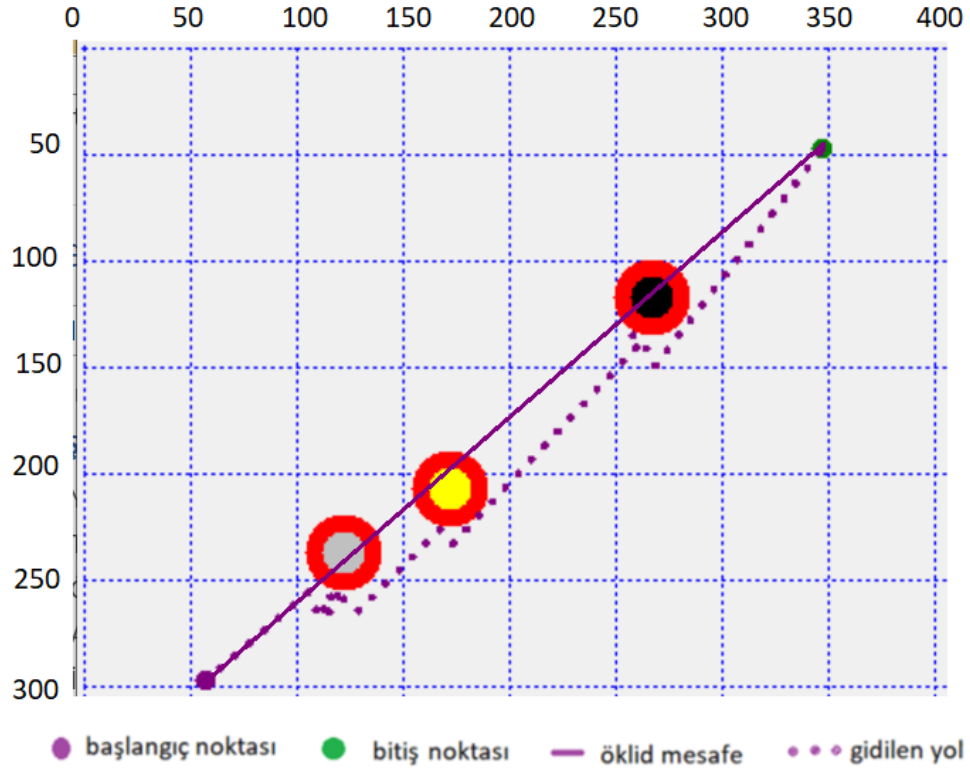
Parametre	Değer
Problemin Boyutu	2
Yiyecek Kaynağı Sayısı	5
Limit	10

Her bir senaryo-durdurma kriteri kombinasyonu için birbirinden bağımsız 30 koşma alınmış ve kat edilen toplam yol ve geçen süre için ortalama, standart sapma, en iyi ve en kötü bilgileri Tablo 3.2-Tablo 3.6’da önerilen yöntem ve önerilen yöntemde mABC yerine ABC kullanılarak elde edilen yöntem için verilmiştir. Daha iyi sonuç alınan durumlar tablolarda kalın yazı fontu ile belirtilmiştir. İlk test yöntemi ile elde edilen sonuçlar için ilgili senaryolarda her bir yöntem için robotun izlediği yol Şekil 3.1 – Şekil 3.8’de sunulmuştur. Bu şekiller 30 çalıştırmada bulunan en kötü yol üzerinden verilmiştir. Şekillerde güvenlik yarıçapı kırmızı renkli olanlar sabit engelleri, siyah renkli olanlar hareketli engelleri ifade etmektedir. Hareketli engellerin bulunduğu durumlar için içi boş daire olarak gösterilenler engelle karşılaşılan ilk durumdaki engel konumunu, içi dolu engeller ise engelle karşılaşılan son durumdaki engel konumunu göstermektedir.

İlk senaryoda 3 adet sabit engel bulunmaktadır. Bu senaryo ve metasezgiselin durdurma kriterinin 1000 ms olduğu durumda ABC algoritması için alınan en kötü yol Şekil 3.1’de, mABC algoritması için alınan en kötü yol Şekil 3.2’de gösterilmiş olup, alınan sonuçlar Tablo 3.2’de verilmiştir. ABC kullanılan yol planlama yaklaşımının, mABC kullanılan yol planlama yaklaşımına göre yol ve süre bakımından daha iyi sonuç verdiği tespit edilmiştir.



Şekil 3.1. Senaryo 1’de durdurma kriterinin 1000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol



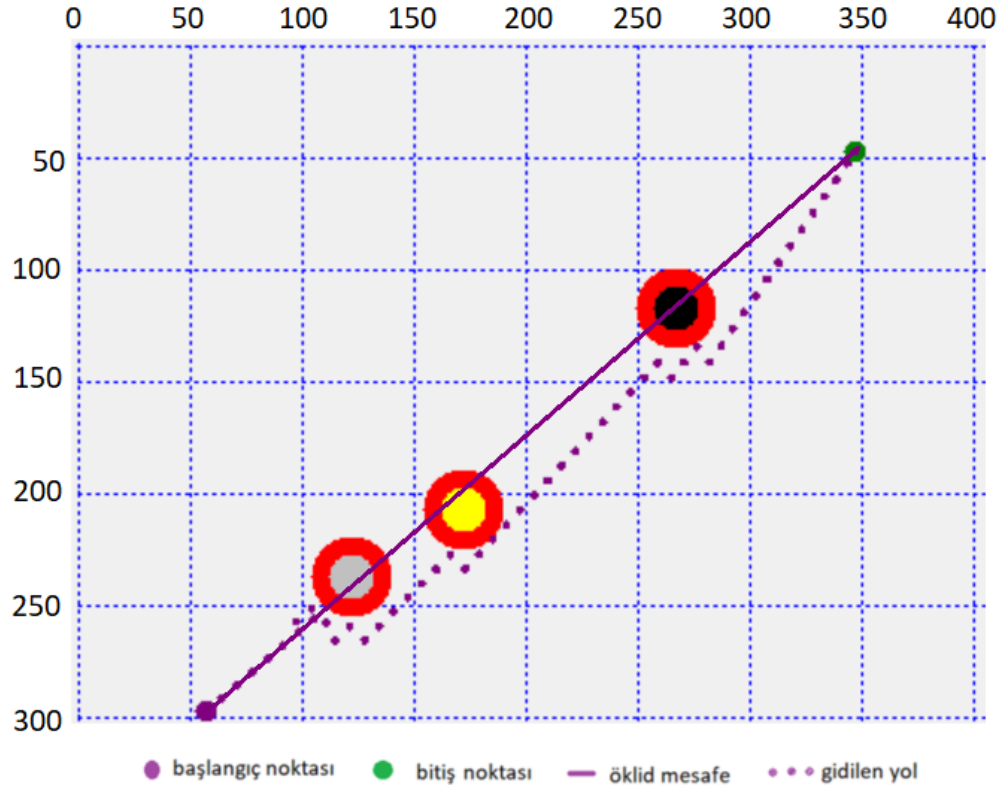
Şekil 3.2. Senaryo 1 için durdurma kriterinin 1000 ms olduğu durumda mABC uygulanarak belirlenen yol

Tablo 3.2. Senaryo 1 için durdurma kriterinin 1000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar.

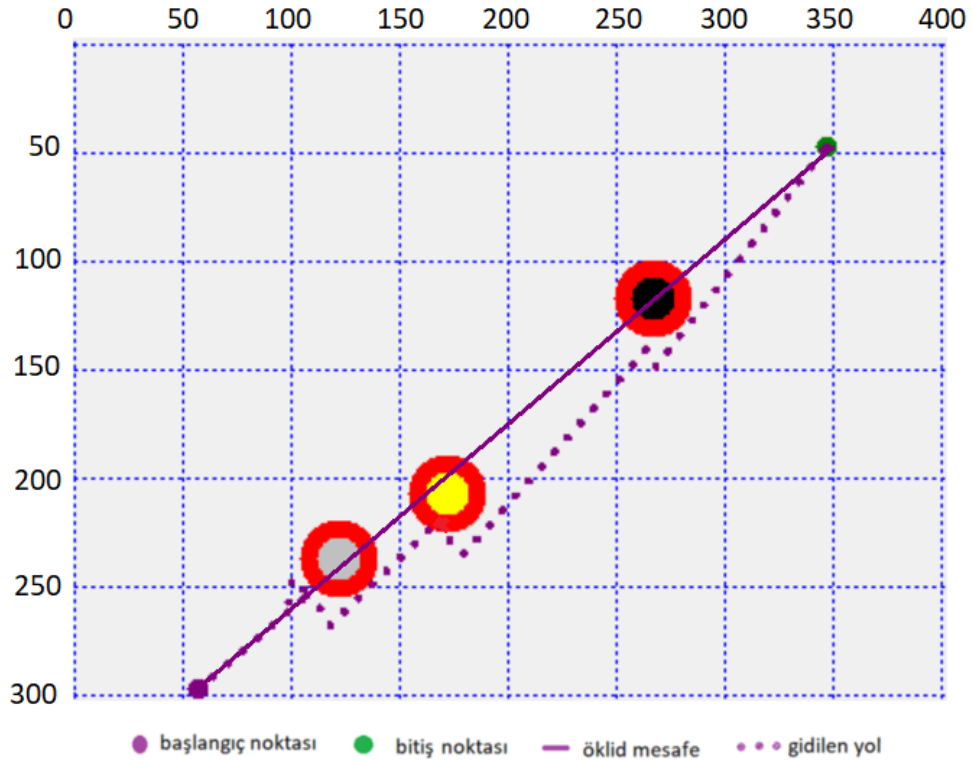
	ABC	mABC
Ortalama (Toplam Yol)	428,04	432,96
Standart Sapma (Toplam Yol)	4,77	11,81
En İyi (Toplam Yol)	418,87	418,21
En Kötü (Toplam Yol)	442,52	464,78
Ortalama (Toplam Süre)	5659,84	6269,66
Standart Sapma (Toplam Süre)	895,74	1300,06
En İyi (Toplam Süre)	5051,57	5052,38
En Kötü (Toplam Süre)	8059,65	10076,42

İlk senaryoda metasezgiselin durdurma kriterinin 5000 ms olduğu durumda ABC temelli yöntem için alınan en kötü yol Şekil 3.3'te, mABC temelli yöntem için alınan en kötü yol Şekil 3.4'te gösterilmiş olup, alınan sonuçlar Tablo 3.3'de verilmiştir. Tablo

3.3 incelendiğinde mABC yaklaşımının süre olarak ABC'den daha iyi olduğu görülmektedir.



Şekil 3.3. Senaryo 1'de durdurma kriterinin 5000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol



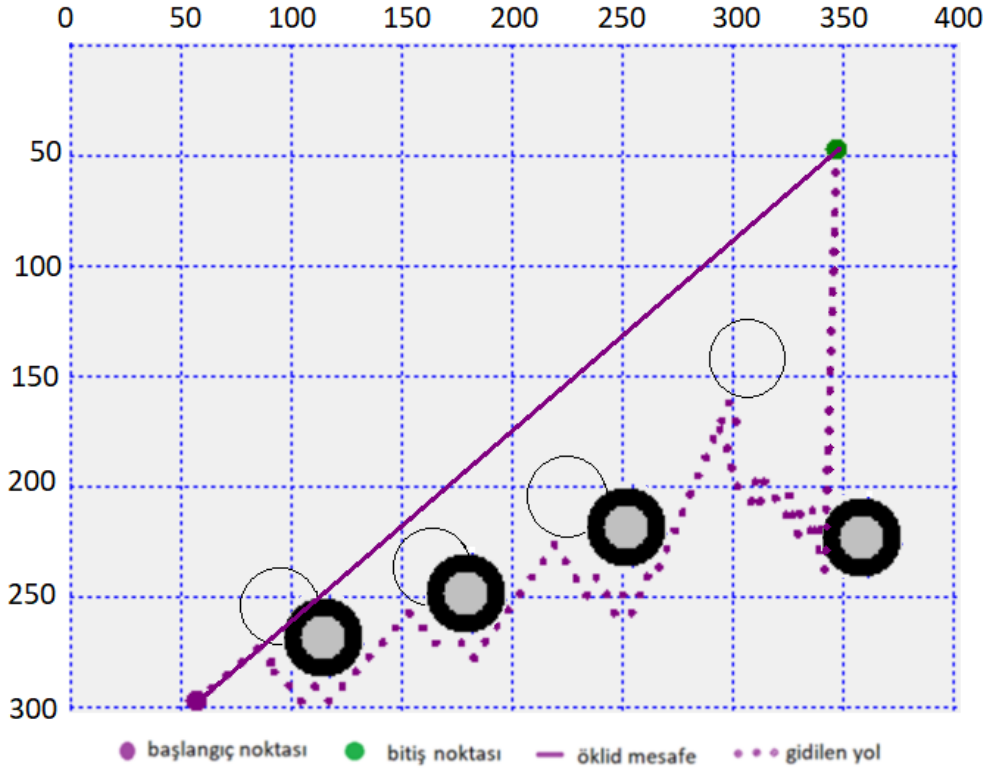
Şekil 3.4. Senaryo 1'de durdurma kriterinin 5000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol

Tablo 3.3. Senaryo 1 için durdurma kriterinin 5000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar.

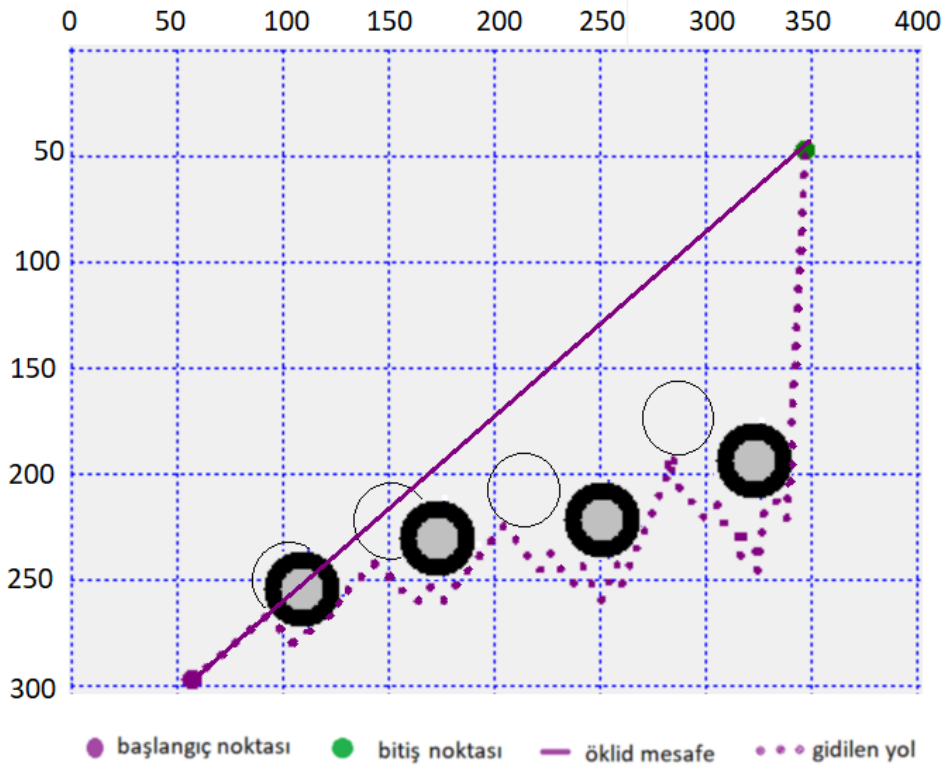
	ABC	mABC
Ortalama (Toplam Yol)	431,98	431,304
Standart Sapma (Toplam Yol)	7,52	8,76
En İyi (Toplam Yol)	424,35	423,0035
En Kötü (Toplam Yol)	454,94	461,34
Ortalama (Toplam Süre)	29535,62	28699,65
Standart Sapma (Toplam Süre)	6479,66	5241,85
En İyi (Toplam Süre)	25029,3	25028,49
En Kötü (Toplam Süre)	55035,11	45036,61

2. senaryoda dört hareketli engel bulunmaktadır. Bu senaryoda durdurma kriteri olarak 1000 ms dikkate alındığında elde edilen sonuçlar Tablo 3.4'de sunulmuştur. Şekil 3.5'de ABC algoritmasının bulduğu en kötü yol, Şekil 3.6'da ise mABC algoritmasının bulduğu en kötü yol gösterilmiştir. Bu test durumunda mABC çalıştırılan yöntem toplam

kat edilen yola göre avantaj sağlarken, ABC kullanan yöntem süre bakımından daha avantajlı görünmektedir.



Şekil 3.5. Senaryo 2’de durdurma kriterinin 1000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol

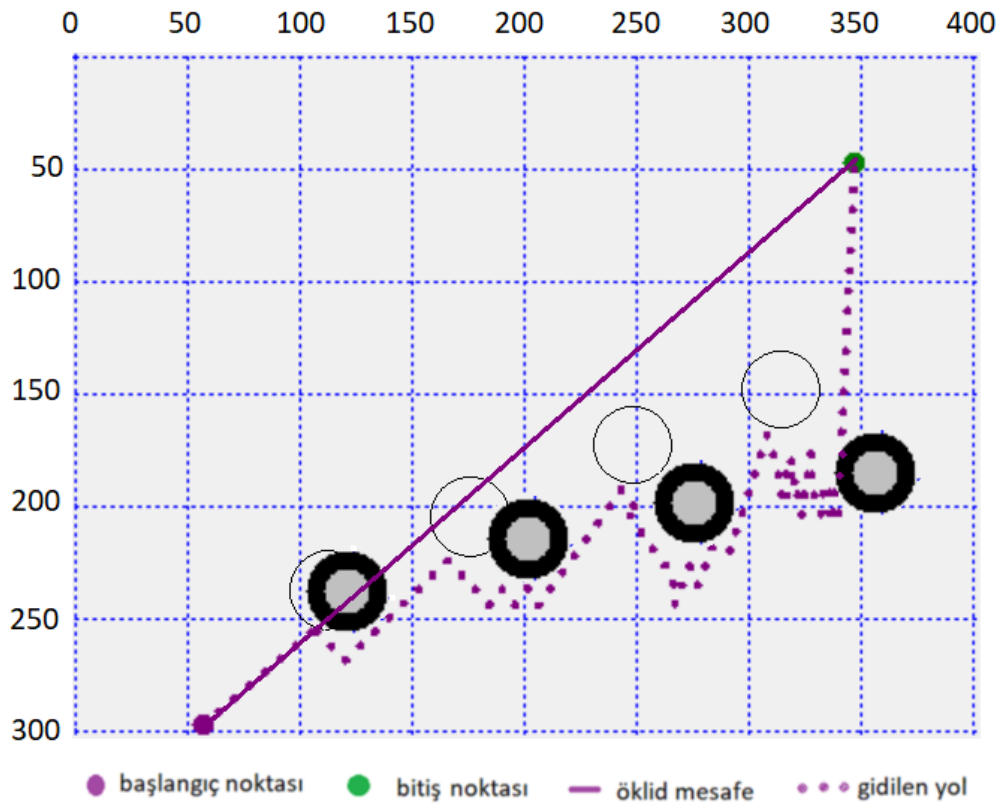


Şekil 3.6. Senaryo 2’de durdurma kriterinin 1000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol

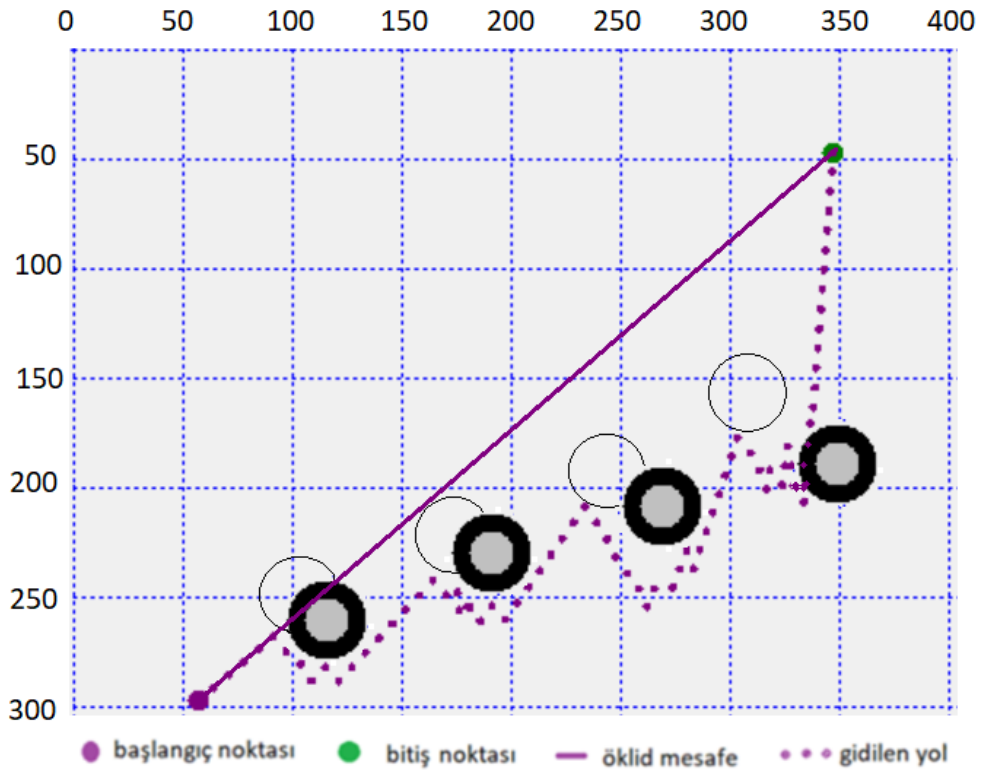
Tablo 3.4. Senaryo 2 için durdurma kriterinin 1000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar.

	ABC	mABC
Ortalama (Toplam Yol)	583,29	561,042
Standart Sapma (Toplam Yol)	90,5	60,06
En İyi (Toplam Yol)	451,099	460,39
En Kötü (Toplam Yol)	839,68	689,7
Ortalama (Toplam Süre)	22508,89	22645,13
Standart Sapma(Toplam Süre)	9479,61	10194,25
En İyi (Toplam Süre)	7059,8	10038,67
En Kötü (Toplam Süre)	43059,52	43456,88

2. senaryoda durdurma kriteri olarak 5000 ms'nin kullanıldığı durumda elde edilen sonuçlar Tablo 3.5'te verilmiştir. Şekil 3.7'de ABC algoritmasının bulunduğu en kötü yol, Şekil 3.8'de ise mABC algoritmasının bulunduğu en kötü yol gösterilmiştir. Burada mABC temelli yaklaşımın ABC temelli yaklaşıma göre yol ve süre bakımından daha iyi olduğu görülmektedir



Şekil 3.7. Senaryo 2’de durdurma kriterinin 5000 ms olduğu durum için ABC algoritması uygulanarak belirlenen yol



Şekil 3.8. Senaryo 2’de durdurma kriterinin 5000 ms olduğu durum için mABC algoritması uygulanarak belirlenen yol

Tablo 3.5. Senaryo 2 için durdurma kriterinin 5000 ms olduğu durumda mABC ve ABC uygulanarak elde edilen sonuçlar.

	ABC	mABC
Ortalama (Toplam Yol)	601,007	561,696
Standart Sapma(Toplam Yol)	90,48	57,55
En İyi(Toplam Yol)	461,73	482,01
En Kötü(Toplam Yol)	802,65	691,09
Ortalama (Toplam Süre)	116214,8	107396,8
Standart Sapma(Toplam Süre)	48318,99	38120,59
En İyi(Toplam Süre)	40029,94	55038,2
En Kötü(Toplam Süre)	235061,2	200056,4

İkinci test yönteminde algoritmalar için durdurma kriteri 100 iterasyon olarak belirlenmiş ve alınan sonuçlar Tablo 3.6’da verilmiştir. Sonuçlar incelendiğinde mABC’nin ABC’ye göre süre ve amaç fonksiyonu değerleri üzerinden sağladığı üstünlük açıkça görülmektedir.

Tablo 3.6. Robotun engel ile ilk karşılaşma anında alınan ABC ve mABC sonuçları

	ABC	mABC
Ortalama (Amaç Fonksiyonu)	155,4687	155,4555
Standart Sapma (Amaç Fonksiyonu)	0,155	0,0468
En İyi (Amaç Fonksiyonu)	155,4419	155,4419
En Kötü (Amaç Fonksiyonu)	156,073	155,6807
Ortalama (Süre)	5,73	3,47
Standart Sapma (Süre)	6,42	6,41
En İyi (Süre)	3,159	1,703
En Kötü (Süre)	38,1	37,15

4. BÖLÜM

TARTIŞMA, SONUÇ ve ÖNERİLER

4.1.Tartışma ve Sonuç

Bu tez çalışması kapsamında hareketli engellerin de bulunduğu 2 boyutlu ortamlarda bölgesel robotik yol planlama problemi için ABC temelli yeni bir yaklaşım önerilmiştir. Öncelikle gerçek zamanlı optimizasyon işlemlerinin yapısı dikkate alınarak daha kısa sürede hem sömürü (exploitation) hem de keşif (exploration) açısından etkin bir araştırma sağlamak amacıyla küçük popülasyon büyüklükleri ile çalışmaya uygun olabilecek yapıda bir ABC versiyonu olan mABC algoritması geliştirilmiştir. Önerilen yol planlama yaklaşımında Öklid mesafesine göre robotun ilerleyebileceği en iyi noktada bir engelin bulunması durumunda robot atacağı adımı mABC algoritmasını kullanarak belirlemektedir.

Yol planlama yaklaşımının performansını genel olarak değerlendirmek için biri 3 adet sabit engel, diğeri 4 adet hareketli engel içermek üzere 2 farklı senaryo dikkate alınmıştır. Gerçek zamanlı bir problem üzerinde çalışıldığı için metasezgisellerin çalıştırıldığı aşamalarda durdurma kriteri olarak süre esas alınmıştır (1000 ms ve 5000 ms).

Senaryo 1’de metasezgiselin durdurma kriteri 1000 ms olarak verildiğinde ABC kullanılan yol planlama yaklaşımında hem yol hem süre bakımından mABC kullanılan yol planlama yaklaşımına göre daha iyi sonuçlar elde edilmiştir. 5000 ms verildiğinde ise mABC temelli yöntemin süre bakımından ABC temelli yöntemle göre daha iyi olduğu görülmüştür. Senaryo 2’de her iki durumda da mABC yaklaşımı, ABC’ye yol bakımından üstünlük sağlamıştır. Süre bakımından ise durdurma kriterinin 5000 ms verildiği durumda, 1000 ms verildiği duruma göre mABC yaklaşımının ABC’den daha başarılı olduğu görülmüştür.

ABC ve mABC algoritmasının performansını net bir şekilde kıyaslayabilmek için robotun engelle ilk karşılaşma anında ABC ve mABC otuz kez çalıştırılmış ve mABC'nin daha iyi sonuçlar verdiği görülmüştür.

Bu tez çalışması ile gerçek zamanlı uygulamalarda kullanıma ve küçük popülasyon büyüklükleri ile optimizasyon yapmaya uygun olabilecek yeni bir ABC versiyonu sunulmuş ve literatüre ABC tabanlı yeni bir bölgesel yol planlama yaklaşımı tanıtılmıştır. Yapılan simülasyon çalışmalarında elde edilen sonuçlar yeni yol planlama yaklaşımının ilgili problemlerde kullanılabilir yapıda olduğunu göstermiştir. Tez çalışmasında ele alınan problemlerde ve kullandıkları aşamalarda genel olarak mABC'nin standart ABC'ye göre daha başarılı olduğu görülmektedir.

4.2.Öneriler

İleride yapılacak olan çalışmalarda mABC algoritmasının çalıştırılabileceği alternatif alanlar belirlenip algoritmanın performansı test edilebilir. Önerilen yol planlama yöntemi için kıyaslama çalışmasında test durumları, ele alınan senaryolar ve kıyaslamada kullanılacak rakip yöntemlerin sayısı artırılarak daha kapsamlı bir performans değerlendirmesi yapılabilir. Hem küresel hem de bölgesel yol planlama işlemlerinin gerçekleştirildiği problemler için önerilen yöntemin uygulamaları üzerinde çalışılabilir. Ortam, engeller, robot miktarı, hedef sayısı vb. probleme dair parametreler için farklı değerler dikkate alınarak uygun adaptasyonlar ile önerilen yaklaşımın performansı incelenebilir.

KAYNAKÇA

- [1] Pandey, A., Pandey, S., Parhi, DR., 2017. Mobile robot navigation and obstacle avoidance techniques: **A Review, International Robotics & Automation Journal, 2(3): 96-105**
- [2] Liang, J., Lee, C., 2015. Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm, **Advances in Engineering Software, 79: 47-56**
- [3] Faridi, A. Q., Sharma, S., Shukla, A., Tiwari, R., Dhar, J., 2018. Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment. **Intelligent Service Robotics, 11(2): 171-186.**
- [4] Lamini, C., Benhlina, S., Elbekri, A., 2018. Genetic algorithm based approach for autonomous mobile robot path planning. **Procedia Computer Science, 127: 180-189.**
- [5] Nyassi, O., 2019. Yapay Arı Kolonisi (ABC) Algoritması İle Robotik Yol Planlama. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri.
- [6] Karaboga, D., Gorkemli, B., 2014. A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, **Applied Soft Computing, 23: 227–238**
- [7] Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- [8] Khader, A. T., Al-betar, M. A., Mohammed, A. A., 2013. Artificial bee colony algorithm, its variants and applications: a survey.

- [9] Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. **Artificial Intelligence Review**, **42**(1): 21-57.
- [10] Hussein, A., Mostafa, H., Badrel-din, M., Sultan, O., Khamis, A., 2012. Metaheuristic optimization approach to mobile robot path planning, pp. 1-6. *International Conference On Engineering And Technology (ICET)*. IEEE
- [11] Guruji, A. K., Agarwal, H., Parsediya, D. K., 2016. Time-efficient A* algorithm for robot path planning. **Procedia Technology**, **23**: 144-149.
- [12] Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., Jurišica, L., 2014. Path planning with modified a star algorithm for a mobile robot. **Procedia Engineering**, **96**: 59-69.
- [13] Zu, W., Fan, G., Gao, Y., Ma, Y., Zhang, H., & Zeng, H., 2018. Multi-uavs cooperative path planning method based on improved rrt algorithm, pp. 1563-1567. *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE
- [14] Al-Dahhan, M. R. H., Schmidt, K. W., 2019. Path planning based on voronoi diagram and PRM for pmnidirectinal mobile robots. **DTSS2019 international conference**.
- [15] Rashid, R., Perumal, N., Elamvazuthi, I., Tageldeen, M. K., Khan, M. A., Parasuraman, S., 2016. Mobile robot path planning using ant colony optimization, pp. 1-6. *2nd IEEE International Symposium on Robotics and Manufacturing Automation IEEE*.
- [16] Raja, P., Pugazhenth, S., 2009. Path planning for mobile robots in dynamic environments using particle swarm optimization, pp. 401-405. *2009 International Conference on Advances in Recent Technologies in Communication and Computing. IEEE*.
- [17] Kamon, I., E. Rivlin, 1997. Sensory-based motion planning with global proofs. **IEE Transactions on Robotics and Automation**, **13**(6):814-822.

- [18] NGAH, W., Buniyamin, N., Mohamad, Z., 2010. Point to point sensor based path planning algorithm for autonomous mobile robots. **Proceedings Of The 9th WSEAS International Conference On System Science And Simulation in Engineering (1):** 186-191.
- [19] Sedighi, K., Ashenayi, K., Manikas, T., Wainwright, R., Tai, H., 2004. Autonomous local path-planning for a mobile robot using a genetic algorithm. **Congress On Evolutionary Computation (IEEE), 2:** 1338-1345.
- [20] Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. **Applied Soft Computing, 8:** 687–697
- [21] Kumar, S., Sikander, A. 2022. Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming. **Arabian Journal for Science and Engineering(1):** 1-21.
- [22] Cruz, M., Ramirez, V., Belmonte, U., 2015. Mobile robot path planning using artificial bee colony and evolutionary programming. **Applied Soft Computing, 30:** 319- 328.
- [23] Xu, F., Li, H., Pun, C. M., Hu, H., Li, Y., Song, Y., Gao, H., 2020. A new global best guided artificial bee colony algorithm with application in robot path planning. **Applied Soft Computing, 88:** 106037.
- [24] Gigras, Y., Jora, N., Dhull, A., 2016. Comparison between different meta-heuristic algorithms for path planning in robotics, **International Journal of Computer Applications, 142(3):**6-10
- [25] Abbas, N. H., Abdulsahab, J. A., 2017. An adaptive multi-objective artificial bee colony algorithm for multi-robot path planning. **Association of Arab Universities Journal of Engineering Sciences, 24(3),** 168-189.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Senem Nazlı DOĞAN

Uyruğu: Türkiye (T.C)

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2016
Lise	Tokat Anadolu Lisesi, Tokat	2011

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2021-Halen	Fabay A.Ş.	-
2016-2020	Uyumsoft Bilgi Sistemleri	4

YABANCI DİL

İngilizce