

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**MOBİL ROBOTLARIN YOL PLANMASI İÇİN META-  
SEZGİSEL HİBRİT ALGORİTMALAR  
GELİŞTİRİLMESİ VE UYGULANMASI**

**DOKTORA TEZİ**

**Zeynep GARİP**

**Enstitü Anabilim Dalı : MEKATRONİK MÜHENDİSLİĞİ**

**Tez Danışmanı : Prof. Dr. Durmuş KARAYEL**

**Ağustos 2018**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL ROBOTLARIN YOL PLANMASI İÇİN META-  
SEZGİSEL HİBRİT ALGORİTMALAR  
GELİŞTİRİLMESİ VE UYGULANMASI

DOKTORA TEZİ

Zeynep GARİP

Enstitü Anabilim Dalı : MEKATRONİK MÜHENDİSLİĞİ

Bu tez 02 / 08 /2018 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.



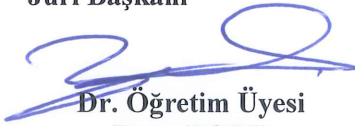
Prof. Dr.  
Durmuş KARAYEL  
Jüri Başkanı



Prof. Dr.  
Sinan Serdar ÖZKAN  
Üye



Doç. Dr.  
Cüneyt BAYILMIŞ  
Üye



Dr. Öğretim Üyesi  
Barış BORU  
Üye



Prof. Dr.  
Recep KOZAN  
Üye

## **BEYAN**

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Zeynep GARİP

20.07.2018

## ÖNSÖZ

Günümüzde teknolojik gelişmelere paralel olarak, robot konusundaki çalışmalar çoğalmakta ve robotların kullanım alanları askeri alandan bilimsel alanlara, endüstriden günlük yaşamlarımızın içindedir. Yol planlama, çoklu mobil robot uygulamalarında önemli bir rol oynakta ve uygulamalarda büyük bir problemlerden biridir. Yol planlama problemleri için son zamanlarda, meta-sezgisel algoritmalar sıklıkla kullanılmaktadır. Bu çalışmada meta-sezgisel algoritmaların verimliliğini kullanarak yol planlamada mobil robotların yolu minimize edilerek belirlenen zaman, maliyet gibi en önemli performans kriterlerini geliştirilmiştir.

Tez çalışmasına maddi açıdan destek sağlanan Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Koordinatörlüğüne (Proje No: 2015-50-02-001) teşekkür ederim.

Tüm çalışmalarında her türlü desteğini esirgemeyen danışmanım Prof. Dr. Durmuş KARAYEL ve tezin ortaya çıkmasına katkı sağlayan herkese teşekkür ederim. Sonsuz sabrı ve desteği için eşim Yigit GARİP'e, bugünlere gelmemde en büyük emeğe sahip annem Müzeyyen BATIK, babam Ali BATIK'a şükranlarımı sunarım. Desteklerini her zaman arkamda hissettiğim kardeşlerime teşekkür ederim.

# İÇİNDEKİLER

ÖNSÖZ .....	i
İÇİNDEKİLER .....	ii
SİMGELER VE KISALTMALAR LİSTESİ .....	vi
ŞEKİLLER LİSTESİ .....	viii
TABLolar LİSTESİ .....	xiii
ÖZET .....	xiv
SUMMARY .....	xv

## BÖLÜM 1.

GİRİŞ .....	1
1.1. Mobil Robotlar Yol Planlama Algoritmaları .....	5
1.1.1. Geleneksel algoritmalar .....	5
1.1.2. Meta-sezgisel algoritmalar .....	8
1.2. Mobil Robotun Sistem Modellemesi .....	15
1.3. Literatür Araştırması .....	19
1.4. Tezin Amacı, Kapsamı ve Hedefler.....	31

## BÖLÜM 2.

MOBİL ROBOTLARIN YOL PLANLAMASINDA AMAÇ FONKSİYONUN BELİRLENMESİ .....	34
2.1. Giriş .....	34
2.2. Mobil Robot Yol Planlamada Uygunluk Fonksiyonun Oluşturulması ..	35

### BÖLÜM 3.

ÇOKLU MOBİL ROBOTLAR İÇİN PARÇACIK SÜRÜ OPTİMİZASYON (PSO) ALGORİTMASI TABANLI GLOBAL YOL PLANLAMA .....	40
3.1. Giriş .....	40
3.2. PSO Algoritması İçin Matematiksel Modelleme .....	43
3.2.1. PSO algoritmasının temel parametreleri.....	46
3.2.2. PSO algoritmasının çeşitleri .....	49
3.2.3. PSO algoritmasının avantaj ve dezavantajları .....	50
3.3. Mobil Robotlar İçin PSO Algoritma Tabanlı Yol Planlama .....	51
3.3.1. PSO tabanlı uygunluk fonksiyonun oluşturulması .....	51
3.3.2. Geliştirilen PSO tabanlı mobil robotlarda global yol planlama ...	52
3.4. Simülasyon ve Deneysel Sonuçlar .....	53
3.5. Geliştirilen PSO Algoritmasının Diğer Algoritmalarla Karşılaştırılması ve Performans Analizi .....	58

### BÖLÜM 4.

ÇOKLU MOBİL ROBOTLAR İÇİN ATEŞ BÖCEĞİ ALGORİTMASI (FA) TABANLI GLOBAL YOL PLANLAMA .....	62
4.1. Giriş .....	62
4.2. FA için Matematiksel Modelleme .....	64
4.2.1. FA'nın temel parametreleri .....	67
4.2.2. FA'nın çalışma prensibi .....	69
4.2.3. FA'nın türevleri .....	71
4.2.4. FA'nın avantaj ve dezavantajları .....	72
4.3. Mobil Robotlar için Tarafımızdan Geliştirilen FA Tabanlı Yol Planlama .....	72
4.3.1. FA tabanlı uygunluk fonksiyonun oluşturulması .....	73
4.3.2. Tarafımızdan geliştirilen FA tabanlı mobil robotlarda global yol planlama .....	73
4.4. Simülasyon ve Deneysel Sonuçlar .....	75
4.5. Diğer Yapılan Çalışmalarla Karşılaştırma .....	80

## BÖLÜM 5.

### ÇOKLU MOBİL ROBOTLAR İÇİN GUGUK KUŞU (CS) ALGORİTMASI

TABANLI GLOBAL YOL PLANLAMA .....	84
5.1. Giriş .....	84
5.2. CS Algoritması için Matematiksel Modelleme .....	86
5.2.1. CS algoritmasının çeşitleri .....	90
5.2.2. CS algoritmasının avantaj ve dezavantajları .....	90
5.3. Mobil Robotlar İçin Geliştirilen CS Algoritma Tabanlı Yol Planlama	91
5.3.1. Geliştirilen CS algoritma tabanlı uygunluk fonksiyonun oluşturulması .....	91
5.3.2. Geliştirilen FA tabanlı mobil robotlarda global yol planlama ....	92
5.4. Simülasyon ve Deneysel Sonuçlar .....	93
5.5. Diğer Yapılan Çalışmalarla Karşılaştırma .....	97

## BÖLÜM 6.

### ÇOKLU MOBİL ROBOTLAR İÇİN GELİŞTİRİLEN HİBRİD

ALGORİTMALAR TABANLI GLOBAL YOL PLANLAMA .....	101
6.1. Giriş .....	101
6.2. CS ve PSO Tabanlı Geliştirilen Hibrit Algoritmanın Matematiksel Modellenmesi .....	102
6.3. CS ve FA Tabanlı Geliştirilen Hibrit Algoritmanın Matematiksel Modellenmesi .....	104
6.4. CS, PSO ve FA Tabanlı Geliştirilen Hibrit Algoritmanın Matematiksel Modellenmesi .....	105
6.5. CS, PSO ve FA tabanlı Bu Tez Çalışmasına Özgü Geliştirilen Yeni Bir Hibrit Algoritma ile Global Yol Planlama .....	107
6.5.1. Yol planlama için gerçekleştirilen planlama sistemi .....	109
6.5.2. Yerel yol planlama için kullanılan planlama sistemi .....	112
6.5.3. Robot ve algoritmalar için MATLAB ve ROS tabanlı kullanıcı arayüz tasarımı .....	115
6.6. Simülasyon ve Deneysel Sonuçlar .....	118

6.7. Geliştirilen Özgün Hibrit Algoritmaların Diğer Hibrit Algoritmalarla Karşılaştırılması ve Performans Analizi .....	126
BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER .....	134
7.1. Öneriler .....	136
KAYNAKLAR .....	138
EKLER .....	145
ÖZGEÇMİŞ .....	149



## SİMGELER VE KISALTMALAR LİSTESİ

ABC	: Yapay Arı Koloni Algoritması
ACO	: Karınca Koloni Optimizasyonu
AG	: Agoraphilic algoritması
ANFIS	: Uyarlamalı nöron bulanık çıkarım sistemini
APF	: Yapay potansiyel Alan
BFOA	: Bakteriye Besin Arama Optimizasyon Algoritması
BYSA	: Benzetim Yapay Sinir Ağı
CSA	: Guguk Kuşu Arama Algoritması
EA	: Evrimsel Programlama
FA	: Ateş Böceği Algoritması
FL	: Bulanık Mantık
FPA	: Çiçek Tozlaşma Algoritmaları
GA	: Genetik Algoritma
gbest	: Sürü içerisindeki en iyi pozisyon
IGSA	: Geliştirilmiş Yerçekimi Araştırması Algoritması
IPSO	: Geliştirilmiş Parçacık Sürüsü Optimizasyonu
LSE	: En küçük kareleme
NN, YSA	: Yapay Sinir Ağı
p	: Konum
pbest	: En iyi pozisyon
PRM	: Klasik Olasılıklı Yol Haritası Metodu
PSO	: Parçacık Sürü Optimizasyonu
r	: Uzaklık
RBPF	: Rao-Blackwellized Parçacık Filtre
RMSE	: Kök ortalama karesi hatası
SA	: Benzetimli Tavlama

SBPL	: Arama Tabanlı planlama Kütüphanesi
T1-SFSL	: Tek tip 1 bulanık mantık sistem
$v$	: Hız
WDO	: Rüzgara dayalı optimizasyon
$V_c$	: Mobil robotun lineer hızı [m/s]
$V_r$	: Sağ tekerlek lineer hızı [m/s]
$V_l$	: Sol tekerlek lineer hızı [m/s]
$L$	: Ön iki tekerleğin ortalarının arasındaki mesafe [m]
$x, y$	: Koordinat sistemi.
$W_c$	: Mobil robotun açısal hızı [rad/s].
$\varphi_R$	: Sağ tekerlek açısal hızı
$\varphi_L$	: Sol tekerlek açısal hızı
$\theta$	: Robotun dönme açısı [rad]
$I_{cc}$	: Anlık eğrilik merkezi
$R$	: Tekerleğin dinamik yarıçapı [m].
$\tau_R, \tau_L$	: Sağ ve sol motorun torku
$I$	: Işık yoğunluğu
$I_0$	: Başlangıç ışık yoğunluğuna
$\gamma$	: Gamma

## ŞEKİLLER LİSTESİ

Şekil 1.1. Robot navigasyon yapısı(Siegwart ve ark., 2004) .....	1
Şekil 1.2. Yol Planlamanın Sınıflandırılması .....	2
Şekil 1.3. Yapay Potansiyel Alan yaklaşımı ile mobil robot navigasyonu .....	8
Şekil 1.4. Literatürde Bulunan Meta-sezgisel Algoritmalar .....	9
Şekil 1.5. Kobuki Robot Kinematik Model Parametreleri .....	15
Şekil 2.1. Heuristik Fonksiyonlar .....	37
Şekil 2.2. Global Yol Planlamada Heuristik Fonksiyonların Etkisi .....	38
Şekil 3.1. PSO algoritması sözde kodu .....	44
Şekil 3.2. PSO temel davranışları (Del Valle ve ark., 2008) .....	45
Şekil 3.3. İterasyon sayısına göre PSO algoritmasının hareketi .....	45
Şekil 3.4. Global en iyi yakınsamanın ana yapısı .....	46
Şekil 3.5. PSO algoritmasının akış diyagramı (Tozan ve ark., 2008) .....	49
Şekil 3.6. PSO algoritmasının çeşitleri .....	50
Şekil 3.7. Parçacıkların hedefe doğru ilerlemesi .....	52
Şekil 3.8. Geliştirilen PSO tabanlı Global Yol Planlamanın Akış Diyagramı ....	53
Şekil 3.9. Statik engelli bir ortamda mobil robot için PSO tabanlı global yol planlama .....	54
Şekil 3.10. PSO tabanlı yol planlayıcının gerçek ortamda uygulanması .....	55
Şekil 3.11. Statik engelli bir ortamda çoklu mobil robot için PSO tabanlı global yol planlama .....	56
Şekil 3.12. Tek robot için PFL tabanlı mobil robot navigasyonu (Patle, 2016) ....	58
Şekil 3.13. Tarafımızdan geliştirilen PSO tabanlı global yol planlama .....	59
Şekil 3.14. Çoklu robot için PFL tabanlı mobil robot navigasyonu (Patle, 2016)	59
Şekil 3.15. Çoklu Mobil Robot için tarafımızdan geliştirilen PSO tabanlı global yol planlama .....	60
Şekil 4.1. FA algoritması sözde kodu (Yang, 2014) .....	66

Şekil 4.2. FA akış diyagramı (Yang, 2014) .....	69
Şekil 4.3. Ateş Böceklerinin hareketlerinin aşamaları (Raja ve ark., 2013) .....	70
Şekil 4.4. FA türevleri .....	71
Şekil 4.5. Global Yol Planlama için Geliştirilen FA tabanlı Akış Diyagramı .....	74
Şekil 4.6. Statik engelli bir ortamda mobil robot için FA tabanlı global yol planlama .....	76
Şekil 4.7. Gerçek ortamda yerel yol planlayıcının uygulanması .....	77
Şekil 4.8. Statik engelli bir ortamda çoklu mobil robot için FA tabanlı global yol planlama .....	79
Şekil 4.9. Diğer FA tabanlı global yol planlaması (Patle ve ark., 2016) .....	81
Şekil 4.10. Geliştirilen FA tabanlı global yol planlaması .....	81
Şekil 4.11. Diğer FA tabanlı çoklu mobil robotlar için global yol planlaması (Patle ve ark., 2016) .....	82
Şekil 4.12. Bu çalışmada tarafımızdan geliştirilen FA tabanlı çoklu mobil robotlar için global yol planlaması .....	82
Şekil 5.1. Levy uçuş Dağılımı .....	87
Şekil 5.2. CS algoritma sözde kodu .....	88
Şekil 5.3. CS algoritmasının akış diyagramı(Yang, 2009) .....	89
Şekil 5.4. CSA çeşitleri (Fister ve ark., 2014) .....	90
Şekil 5.5. Global Yol Planlama için Geliştirilen CS tabanlı Akış Diyagramı .....	92
Şekil 5.6. Tekli mobil robot için CS tabanlı global yol planlama .....	94
Şekil 5.7. Gerçek ortamda yerel yol planlayıcının uygulanması .....	94
Şekil 5.8. Statik engelli bir ortamda çoklu mobil robot için CS tabanlı global yol planlama .....	96
Şekil 5.9. GA algoritma tabanlı yol planlama (Wang ve ark., 2010) .....	98
Şekil 5.10. PSO algoritma tabanlı yol planlama (Mohammed ve ark., 2011) .....	98
Şekil 5.11. CS algoritma tabanlı yol planlama (Mohantyve ark., 2017) .....	99
Şekil 5.12. Tarafımızdan Geliştirilen CS algoritma tabanlı yol planlama .....	99
Şekil 6.1. Geliştirilen CS-PSO algoritmasının sözde kodu .....	103
Şekil 6.2. Geliştirilen CS-FA algoritmasının sözde kodu .....	105
Şekil 6.3. Geliştirilen CS-PSO-FA hibrit algoritmanın sözde kodu .....	106

Şekil 6.4. Çoklu mobil robotların yol planlaması için geliştirilen CS-PSO-FA hibrit algoritmanın akış diyagramı .....	110
Şekil 6.5. Statik çevrede çoklu mobil robotlarının yol planlaması için tasarlanacak sistem .....	110
Şekil 6.6. Laboratuvar Ortamının Modeli .....	110
Şekil 6.7. Görüntü işleme fonksiyonu blok şeması .....	111
Şekil 6.8. Statik çevrenin tanımlanması .....	111
Şekil 6.9. Önerilen Yazılımın Taslağı .....	109
Şekil 6.10. Pure Pursuit algoritma yapısı .....	113
Şekil 6.11. Küçük Look Ahead noktasının seçilmesi .....	114
Şekil 6.12. Büyük Look Ahead noktasının seçilmesi .....	115
Şekil 6.13. Pure Pursuit başlangıç tanımlamaları .....	115
Şekil 6.14. Pure Pursuit konum tanımlamaları .....	115
Şekil 6.15. Pure Pursuit konum tanımlamaları .....	115
Şekil 6.16. Açısal ve doğrusal hızların hesaplanması .....	115
Şekil 6.17. Açısal ve doğrusal hızların robota gönderilmesi .....	115
Şekil 6.18. Tasarlanan Ara yüz .....	117
Şekil 6.19. Kameradan alınan çalışma alanı görüntüsü ve haritaya dönüştürülmesi	117
Şekil 6.20. Tekli mobil robot için geliştirilen hibrit algoritma tabanlı global yol planlama .....	120
Şekil 6.21. Tekli mobil robot için özgün CS-PSO-FA hibrit algoritmasının deneysel analizi .....	120
Şekil 6.22. Çoklu mobil robot için hibrit algoritmalar tabanlı global yol planlama .....	124
Şekil 6.23. Çoklu mobil robotlar için geliştirilen özgün CS-PSO-FA hibrit algoritmasının deneysel analizi .....	124
Şekil 6.24. FA-PFL tabanlı navigasyon .....	127
Şekil 6.25. CS-PSO tabanlı navigasyon .....	127
Şekil 6.26. FA-MGA tabanlı navigasyon .....	127
Şekil 6.27. CS-FA tabanlı navigasyon .....	127
Şekil 6.28. FA-PFL-MGA tabanlı navigasyon .....	127
Şekil 6.29. CS-PSO-FA tabanlı navigasyon .....	127

Şekil 6.30. FA-PFL tabanlı navigasyon .....	128
Şekil 6.31. CS-PSO tabanlı navigasyon .....	128
Şekil 6.32. FA-MGA tabanlı navigasyon .....	128
Şekil 6.33. CS-FA tabanlı navigasyon .....	128
Şekil 6.34. FA-PFL-MGA tabanlı navigasyon .....	129
Şekil 6.35. CS-PSO-FA tabanlı navigasyon .....	129
Şekil 6.36. FA-PFL tabanlı navigasyon (Patle ve ark., 2016) .....	130
Şekil 6.37. CS-PSO tabanlı navigasyon .....	130
Şekil 6.38. FA-MGA tabanlı navigasyon .....	130
Şekil 6.39. CS-FA tabanlı navigasyon .....	130
Şekil 6.40. FA-PFL-MGA tabanlı navigasyon .....	131
Şekil 6.41. CS-PSO-FA tabanlı navigasyon .....	131
Şekil 6.42. Statik bir ortamda tekli mobil robot için yapılan yol planlama .....	132
Şekil 6.43. Statik bir ortamda tekli mobil robot için yol planlama .....	132
Şekil 6.44. Statik bir ortamda çoklu mobil robot için yol planlama .....	133

## TABLolar LİSTESİ

Tablo 1.1. Denklem 1.3-17’de verilen kısaltmaların açıklamaları .....	16
Tablo 1.2. Mobil robot yol planlamada Hibrit algoritmaların uygulanması .....	30
Tablo 2.1. Heuristik Uzaklık Fonksiyonu .....	36
Tablo 2.2. Heuristik uzaklık fonksiyonların algoritmalar üzerinde etkisi .....	39
Tablo 3.1. Mobil robot yol planlamada PSO algoritmasının uygulanması .....	43
Tablo 3.2. Şekil 3.9. ve 3.10.'da gösterilen PSO tabanlı yol planlayıcı tabanlı mobil robot için simülasyon ve deneysel çalışmadaki yol uzunluklarının karşılaştırılması .....	56
Tablo 3.3. Şekil 3.11. gösterilen PSO tabanlı yol planlayıcı tabanlı çoklu mobil robotlar için simülasyon ve deneysel çalışmadaki yol uzunluklarının karşılaştırılması .....	57
Tablo 3.4. Simülasyon sonuçlarının karşılaştırılması .....	60
Tablo 3.5. Simülasyon sonuçlarının karşılaştırılması .....	60
Tablo 4.1. Mobil robot yol planlamada FA uygulaması .....	64
Tablo 4.2. FA Kontrol Parametreleri .....	75
Tablo 4.3. Şekil 4.6. ve 4.7.’da gösterilen mobil robot için FA tabanlı yol planlayıcının simülasyon ve deneysel çalışmalarındaki yol uzunluklarının karşılaştırılması .....	78
Tablo 4.4. FA tabanlı yol planlayıcı kullanılarak gerçekleştirilen simülasyon (Şekil 4.8.) ve deneysel sonuçlardan elde edilen yol uzunluklarının karşılaştırılması .....	79
Tablo 4.5. Tekli mobil robot için simülasyon sonuçlarının yol uzunluğuna göre karşılaştırılması .....	82
Tablo 4.6. Çoklu mobil robotlar için simülasyon sonuçlarının yol uzunluğuna göre karşılaştırılması .....	83
Tablo 5.1. Mobil robot yol planlamada CS algoritmasının uygulanması .....	85

Tablo 5.2. CS algoritmasının parametreleri (Shehaba ve ark., 2017) .....	88
Tablo 5.3. Şekil 5.8.'de gösterilen çoklu mobil robotlar için CS tabanlı yol planlayıcının bulunduğu simülasyon analiz ve deneysel çalışmalardan elde edilen yol uzunluklarının karşılaştırılması .....	95
Tablo 5.4. Şekil 5.6. ve 5.7.'de gösterilen mobil robotun CS tabanlı global yol planlayıcı için simülasyon çalışması ve deneysel çalışmada elde edilen yol uzunluklarının karşılaştırılması .....	96
Tablo 5.5. Tekli mobil robot için farklı algoritmalarla gerçekleştirilen simülasyon sonuçlarının yol uzunluğuna göre karşılaştırılması(Şekil 5.6.-9.) .....	99
Tablo 6.1. Tekli mobil robot için özgün hibrit algoritmaların simülasyon ve deneysel analizinden elde edilen yol uzunluklarının karşılaştırılması	122
Tablo 6.2. Çoklu mobil robotlar için özgün hibrit algoritmaların simülasyon ve deneysel analizinden elde edilen yol uzunluklarının karşılaştırılması	125
Tablo 6.3. Çoklu mobil robotlarda hibrit algoritmaların simülasyon ve deneysel çalışmalar arasındaki ortalama hata oranları .....	125
Tablo 6.4. Statik bir ortamda yol uzunluğuna göre kıyaslanması (Şekil 6.24.-6.29.) .....	128
Tablo 6.5. Statik bir ortamda tekli mobil robot için yol uzunluğuna göre karşılaştırma (Şekil 6.30.-6.35.) .....	129
Tablo 6.6. Statik bir ortamda çoklu mobil robot için yol uzunluk değerlerini karşılaştırma .....	131
Tablo 6.7. Statik bir ortamda yol uzunluk değerlerine göre kıyaslama (Şekil 7.1.)	132
Tablo 6.8. Statik bir ortamda tekli mobil robot için yol uzunluğuna göre karşılaştırma (Şekil 7.2.) .....	133
Tablo 6.9. Statik bir ortamda çoklu mobil robot için yol uzunluğuna göre karşılaştırma (Şekil 7.3(a-c).) .....	133



## ÖZET

Anahtar kelimeler: Meta-sezgisel algoritmalar, Mobil Robotlar, Yol planlama, Matlab, ROS (Robot Operating System)

Günümüzde teknolojik gelişmelere paralel olarak, robot teknolojisinin tüm sektörlerde yaygın olarak kullanıldığı görülmektedir. Endüstriyel ve teknik uygulamalarda mobil robotlar güvenilirlik, erişilebilirlik ve maliyet gibi sahip olduğu üstün özelliklerle ön plana çıkmaktadır. Mobil robotlar hareket halinde olduklarından diğer robotlara göre enerji tüketimi daha fazla önem kazanmaktadır. Dolayısıyla mobil robotlar için hedefe ulaşabilecek en kısa yolun belirlenmesi önemli bir çalışma alanı olmuştur. Bu kapsamda, mobil robotlarda yol planlama algoritmalarının geliştirilmesi üzerine çalışmalara odaklanılmış olup konu hala güncelliğini korumaktadır. Bu tez çalışmasının amacı, meta-sezgisel algoritma tabanlı mobil robot yol planlaması için özgün bir hibrit algoritma geliştirmektir. İlk olarak meta-sezgisel tabanlı yol planlama algoritmaları incelenerek daha etkili ve özgün bir hibrit algoritmanın geliştirilmesi için zemin hazırlandı. Kullanılan algoritmaların amacı meta-sezgisel algoritmaların verimliliğini ve yol planlamada yolun minimize edilerek belirlenen zaman, maliyet gibi en önemli performans kriterlerini geliştirmektir. Bu çalışmada öncelikle tek bir tepe kamerası kullanılarak görüntü düzleminden alınan görüntü işlenerek mobil robotların ve engellerin koordinatları alınmaktadır. Daha sonra üzerinde engeller bulunan statik bir ortamda tekli ve çoklu mobil robotların yol planlaması için özgün hibrit bir algoritma geliştirildi ve uygulandı. PSO(Parçacık Sürü Optimizasyonu), FA (Ateş Böceği Algoritması), CS (Guguk Kuşu Algoritması) gibi meta-sezgisel algoritmalarından her biri ayrı ayrı kullanılarak elde edilen sonuçlar benzer çalışmalarla karşılaştırıldı. Daha sonra aynı ortam şartlarında robotlar için yol planlamalar yapmak üzere çalışmaya özgü CS-PSO(Guguk Kuşu Algoritması-Parçacık Sürü Optimizasyonu), CS-FA(Guguk Kuşu Algoritması-Ateş Böceği Algoritması), CS-PSO-FA (Guguk Kuşu Algoritması-Parçacık Sürü Optimizasyonu-Ateş Böceği Algoritması) gibi hibrit algoritmalar geliştirildi. Kameradan alınan görüntüyü işleyerek harita oluşturma, algoritmalar ile yol bulma, robot ile haberleşme ve robotun algoritmalar ile belirleyeceği yol bilgisine göre ilerleyebilmesi gibi tüm faaliyetleri kolaylıkla gerçekleştirebilmek amacıyla MATLAB-GUI (Graphical User Interface) tabanlı bir ara yüz tasarlanmıştır. Ayrıca geliştirilen algoritmaları doğrulamak üzere fiziksel ortamda çeşitli uygulamalar gerçekleştirilmekte ve elde edilen sonuçların karşılaştırılması yapılmaktadır. Son olarak geliştirilen CS-PSO-FA hibrit algoritmasıyla elde edilen yolun diğer algoritmalara göre daha kısa olduğu ve böylece daha yüksek bir performansa sahip olduğu kanıtlanmıştır.

# **DEVELOPMENT AND APPLICATION OF META-HEURISTIC HYBRID ALGORITHMS FOR PATH PLANNING OF MOBIL ROBOTS**

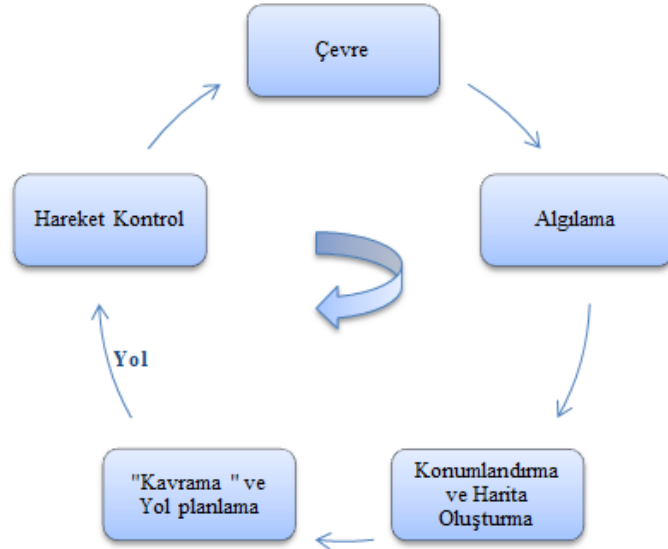
## **SUMMARY**

Keywords: Meta-heuristic algorithm, Mobil Robots, Path Planning, Matlab, ROS (Robot Operating System)

Today, parallel to technological developments, robot technology seems to be widely used in all sectors. In industrial and technical applications, mobile robots come into prominence with superior features such as reliability, accessibility and cost. Energy consumption is more important for mobile robots than other robots because mobile robots are in motion. Therefore, determining the shortest path to reach the target for mobile robots has become an important field of study. In this context, in mobile robots the development of road planning algorithms has been focused and the issue keep up-to-date. The purpose of this thesis is to develop an authentic hybrid algorithm for meta-heuristic algorithm based mobile robot path planning. Firstly, meta-heuristic-based path planning algorithms were examined and the basis for developing a more efficient and authentic hybrid algorithm was prepared. The purpose of the algorithms used is to improve the efficiency of meta-heuristic algorithms and the most important performance criteria such as time, cost, etc., determined by minimizing the path in the path planning. In this study, coordinates of mobile robots and obstacles are taken by processing the image taken from the image plane using a single top camera. In a static environment with obstacles on it, an authentic hybrid algorithm for path planning of single and multiple mobile robots has been developed and implemented. The results were obtained using every meta-heuristic algorithm such as PSO (Particle Swarm Optimization), FA (Firefly Algorithm) and CS (Cuckoo Search Algorithm) separately and these results were compared with similar studies. Then, hybrid algorithms specific to study such as CS-PSO (Cuckoo Search Algorithm - Particle Swarm Optimization), CS-FA (Cuckoo Search Algorithm - Firefly Algorithm) and CS-PSO-FA (Cuckoo Search Algorithm - Particle Swarm Optimization - Firefly Algorithm) were developed to make path planning for robots in the same environment conditions. A MATLAB-GUI (Graphical User Interface) based interface was designed in order to easily perform all activities such as generating maps by processing the images taken from the camera, finding paths with algorithms, communicating with robots, and navigating according to path information that the robot determines with algorithms. In addition, various applications are performed in the physical environment to verify the developed algorithms and the obtained results are compared. In conclusion, it is proved that the path obtained with the developed CS-PSO-FA hybrid algorithm is shorter than the other algorithms and thus has a higher performance.

## BÖLÜM 1. GİRİŞ

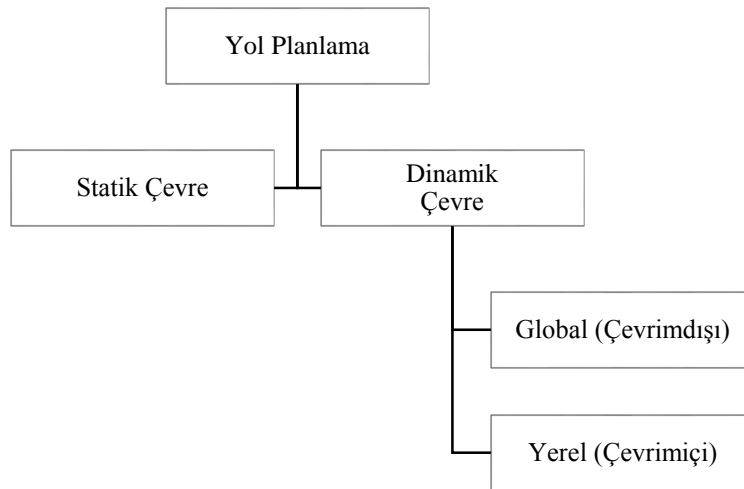
Robotik konusu uzun yıllar araştırmacıların ilgisini çekmiş ve güncelliğini koruyan bir çalışma alanı olmuştur. Son gelişmeler göz önüne alındığında robot teknolojisinin tüm sektörlerde yaygın olarak kullanılacağı anlaşılmaktadır. Mobil robotlar da endüstriyel ve teknik uygulamalarda özellikle güvenilirlik, erişilebilirlik ve maliyet gibi özelliklerle ön plana çıkmaktadır. Günümüzde mobil robotlar endüstri, askeri, sağlık gibi bilim ve mühendislik alanlarının tümünde yaygın olarak kullanılmaktadır. Mobil robot navigasyonu engellere çarpmadan hedef konuma doğru ilerleyen bir süreçtir. Bu süreç Şekil 1.1.'de gösterildiği gibi dört temel bileşenden oluşmaktadır. (i) Algılama, robot sensör aracılığıyla bilgi toplar. (ii) Konumlandırma, robot çalışma alanındaki konumunu belirler. (iii) Kavrama ve yol planlama, robot hedefe ulaşmak için nasıl yönlendirileceğine karar verir. (iv) Hareket kontrol, robot istenen yolu gerçekleştirmek için hareketini düzenler.



Şekil 1.1. Robot navigasyon yapısı (Siegwart ve ark., 2004)

Mobil robotların yol planlaması ve konumlandırılması oldukça karmaşık ve zor bir problemdir. Yol planlamada başlangıç ve hedef noktaları arasındaki optimum yolun

bulunması temel problemlerden biridir. Yolun planlandığı ortamda herhangi bir engel var ise çarpışma olmaksızın robotun hedefe ulaşması da diğer bir problemidir. Yol planlama problemlerini çözebilmek için iki önemli hususa dikkat edilmesi gerekir. Bunlar yol planlama çevresinin tipi ve yol planlama algoritmalarıdır (Şekil 1.2.). Statik çevrede mobil robotun dışında her şey sabittir ve çözüm mutlaka uygulamaya başlamadan önce bulunmalıdır. Dinamik veya kısmen gözlenebilir (yeniden planlanabilir) çevrelerde ise robot ve robotun dışındaki her şey hareketli olabilir. Bundan dolayı robot ve çevre hakkında daha fazla planlama güncellemesi yapılması gereklidir. Yol planlama algoritmaları global ve yerel yol planlama olmak üzere de sınıflandırılabilirler. Global yol planlama algoritması, arama çevresi hakkında tam bilgi gerektirir ve arama çevresi statik yapıda olmalıdır. Bu yaklaşımda, algoritma, araç harekete başlamadan önce, başlangıç noktasından hedef noktaya kadar olan bütün yolu üretir. Öte yandan, yerel yol planlama da ise yol bulma işlemi araç hareket halinde iken gerçekleştiriliyor demektir; diğer bir ifadeyle, algoritma çevresel değişimlere yanıt olarak yeni bir yol üretmeye yeteneklidir (Mnubi, 2016).



Şekil 1.2. Yol planlamanın sınıflandırılması

Optimal yolun planlanması için geleneksel (Dijkstra, A\*, yapay potansiyel alan metodu, görünürlük grafiği yöntemi, voroni diyagramlar) ve meta-sezgisel yöntemler (Genetik Algoritma (GA), Yapay Sinir Ağı (NN), Bulanık Mantık (FL), Bakteriyel Besin Arama Optimizasyon Algoritması (BFOA), Karınca Koloni Optimizasyonu (ACO), Guguk Kuşu Arama Algoritması (CSA), Parçacık Sürü

Optimizasyonu (PSO), Yapay Arı Koloni Algoritması (ABC), Ateş Böceği Algoritması (FA), Benzetimli Tavlama (SA), ve yukarıdaki algoritmaların birleştirilmesinden oluşan hibrit algoritmalar (Hybrid algorithm) uygulanmıştır.

Mobil robotlarda diğer bir önemli konu; konumlandırmanın hızlı, doğru ve düşük maliyetli olarak yapılmasının gerekliliğidir. Konumlandırmanın sağlanması için enkoder, görüntü işleme, WLAN, jiroskop, GPS, lazer aralık bulucu, manyetik sensör, kinect sensör gibi farklı çözüm metotları kullanılmaktadır.

Literatürde mobil robot üzerine yapılan çalışmalar konumlandırma, çevredeki nesnelere algılama, gezinme, engellere çarpmaksızın kaçınım hedefe ulaşma ve yol planlama şeklinde olduğu görülmektedir. Mobil robot çalışma konularını çevrenin belirlenmesi, yol planlama ve konumlandırma olarak sınıflandırmak mümkündür. Bu tez çalışmasının amacı, mobil robotların yol planlama problemine ilişkin çözümler üretmek olduğundan, tezde özellikle yol planlaması üzerinde durulmuştur.

Mobil robotların çevrelerini algılamak ve konumlarını belirlemek için pahalı sensörler kullanılması yerine genellikle görüntü işleme ile birlikte kamera aygıtlarının kullanılması tercih edilmektedir. Wen ve arkadaşları gözetleme kameralarından elde ettikleri gerçek zamanlı görüntü ile kör robotların, izci robotlar tarafından belirlenen yörüngeyi takip etmelerini sağlamışlardır. Harita oluşturulurken izci robot tarafından VSLAM algoritması kullanılarak kamera ile 3D haritası elde edilmiştir. 3D harita 2D haritaya çevrilerek navigasyon planlamasını doğru bir şekilde gerçekleştirmişlerdir. Gerçekleştirdikleri sistemin doğruluğunu simülasyon ve deneysel çalışmalarla göstermişlerdir (Chang ve ark., 2014). Jing ve arkadaşları görsel navigasyon problemini çözmek için dinamik bir çevrede kinect kamera ile bir yol planlama sistemi tasarlamışlardır. Bu çalışmada harita yapay kinect 2D lazer sensör ile çıkarılmakta ve mobil robotun yol planlaması RPF (Rao-Blackwellized Parçacık Filtre) algoritması ile gerçekleştirilmektedir. Global yol planlamada SBPL (Arama Tabanlı planlama Kütüphanesi) ve ARA\* algoritmasını birlikte kullanılmıştır. Yerel yol planlamada ise mobil robotun optimal hareket kontrolü için DWA algoritmasını kullanılmıştır. Belirlenen bu özellikler ile daha düz yollar

üretilmesi amaçlanmıştır. Deneysel sonuçların, diğer tasarlanan navigasyon sistemlerine göre daha başarılı sonuçlar verdiği gösterilmiştir (Jing ve ark., 2016). Ra ve arkadaşları gerçek zamanlı çalışarak, statik bir çevrede mobil robotun hareketi için yol planlama yapmışlardır. Yapılan çalışmada engellerin konumları kamera ve lazer kaynağı kullanılarak bulunmuştur. Kamera ile çevrenin resmi alınarak sisteme aktarılmıştır. Yeni bir yöntem olan lazer ışığı, engeller üzerine yayılarak engellerin konumunu ve robotun engele olan uzaklığını belirlemektedir (Ra ve ark. 2015). Joshi ve arkadaşları mobil robot için en kısa yolu bulmak amacıyla çalışma ortamının resmini kamera yardımıyla elde etmişlerdir. Elde edilen resmin görüntüsü ile görüntü işleme tekniklerinden biri olan gri ölçeklendirme yöntemi kullanılarak engellerin yeri tespit edilmiştir. Yapılan başka bir çalışmada mobil robot için en kısa yol planlaması A\* algoritması ile belirlenmiştir (Joshi ve ark., 2015).

Mobil robotların yol planlamasında en kısa yolu bulmak için birçok algoritma ve yöntem kullanılmaktadır. Bu algoritmalar hesaplama karmaşıklığını azaltarak en kısa yolun bulunmasında yardımcı olmaktadır. Fakat bu tekniklerin adaptasyon eksiklikleri yüzünden ortaya çıkan performans düşüşünü önlemek için meta-sezgisel yöntemler geliştirilmiştir. Meta-sezgisel algoritmaların çoğu biyolojik, fiziksel veya sosyolojik sistemlerin başarılı özelliklerinden ilham alınarak taklit edilmiştir. Bu algoritmalar optimal bir çözüm için arama yeteneği ve etkinliği sayesinde kabul görmüştür. Ayrıca bu yöntemler rastgele değişkenler kullanan ve problemin türüne göre parametreler belirleyen yapıdadır. Bu özelliklerinden dolayı genel arama çözümlerini iyileştirmek için bireysel arama stratejilerinden faydalanacak şekilde çalışırlar. Meta-sezgisel algoritmalar, sezgisel olarak işlem yaptıkları için daha genel, sağlam ve basit yapıda olma gibi özelliklere sahiptirler (Tu ve ark., 2003). Bu özelliklere ek olarak kolay uygulanma, yakınsama özellikleri ve yerel optimumdan uzaklaşma yeteneği gibi avantajlara sahiptirler (Hayat ve ark., 2015). Ayrıca klasik tekniklerin dezavantajlarından (doğrusal programlama ve doğrusal olmayan programlama) da etkilenmezler. Meta-sezgisel algoritmaların güçlü ve zayıf yönleri bulunmaktadır. Zayıf yönlerini iyileştirmek için diğer algoritmaların güçlü yönleri ile birleştirilmektedirler. Bu birleşim hibrit algoritmaları meydana getirmektedir. Bir optimizasyon problemini çözmek için hibrit algoritmalar ile algoritmanın bileşenleri

karşılaştırıldığında, hibrit algoritmaların daha kısa sürede daha iyi çözüm sağlayarak performans artışı sağladığı görülmektedir. Bu sebepten dolayı bu çalışmada çoklu mobil robotların yol planlama probleminde optimale yakın ve daha kısa sürede çözüm elde etmek için yeni özgün hibrit algoritmalar geliştirilmiştir.

## 1.1. Mobil Robotlar Yol Planlama Algoritmaları

Tez çalışmasında statik ortamda tekli ve çoklu mobil robotlar için zeki hibrit yaklaşımlar kullanılarak etkili yol planlamasının geliştirilmesi amaçlanmıştır. Mobil robotun navigasyonunda kullanılan yöntemler geleneksel ve meta-sezgisel yaklaşımlar olarak ayrılmıştır. Meta-sezgisel yaklaşımların çoğu biyolojik davranışlardan ilham alınmış yapay sinir ağı ve bulanık mantıktan gibi algoritmalarından oluşmaktadır.

### 1.1.1. Geleneksel algoritmalar

Mobil robotların navigasyon kontrolü ve yol planlaması için kullanılan geleneksel yaklaşımlar aşağıda incelenmiştir.

#### a. A\* algoritma

Yol planlamasında kullanılan A\* algoritması başlangıç düğümü ile hedef düğümü arasındaki en az kısa ve en az maliyetli yolu bulmak için kullanılır. En kısa yolun hesaplanmasında (Denklem 1.1) kullanılmaktadır.

$$f(n) = g(n) + h(n) \quad (1.1)$$

Burada,

$f(n)$  = Toplam maliyet fonksiyonu

$g(n)$  = Uzaklık fonksiyonu

$h(n)$  = Heuristik fonksiyon olarak verilmiştir.

Uzaklık fonksiyonu  $g(n)$ , başlangıç düğümü ile üzerinde bulunan mevcut düğüm arasındaki geline yol uzunluğu olarak tanımlanmaktadır.  $h(n)$  heuristik fonksiyonu ise mevcut düğüm ile bitiş düğümü arasındaki düz bir çizginin uzunluğunu temsil

eder. Bu, mevcut düğüm ile hedef arasındaki kuş bakışı mesafe olarak da adlandırılmaktadır. Algoritma başlangıç düğümünden başlayarak, en düşük  $f(n)$  değerli düğüme doğru yayılır.

Ankit ve arkadaşları tarafından yapılan çalışmada kısmen bilinen bir çevrede geleneksel A\* algoritması modifiye edilerek en kısa yol bulunmuştur. A\* algoritmasının yapısına yeni bir heuristik fonksiyonla robotun ilerlemesi sağlanarak açılabilir hız hesaplanmıştır. Yapılan çalışmada geleneksel A\* algoritmasına göre önerilen algoritma %8 daha etkili olmuştur (Ankit ve ark., 2014). ZAR ve arkadaşları tarafından yapılan çalışmada A \* algoritması Yangon Şehir alanında otobüs güzergâhları için optimal yolun hesaplanmasında kullanılmıştır (Zar ve ark., 2015).

#### b. D\* algoritma

1994'te Anthony Stenz tarafından kısmen veya tamamen bilinmeyen bir dinamik ortam için D\* Algoritması veya Stenz algoritması olarak da bilinen Dinamik A\* Algoritması veya dinamik Dijkstra algoritması geliştirildi (Stenz, 1994). Algoritma bilinmeyen bir ortamın haritasını güncelleme yeteneğine sahiptir ve yol yeni bir engel bulunduğu zaman maliyet fonksiyonunu yeniden hesaplanarak tekrardan planlama yapabilir. D\* Lite algoritması, ters arama metodu ile D\* algoritmasından farklılık göstermektedir. A\* algoritmasına benzer olarak çalışma alanı mantıksal olarak  $n \times n$ 'lik bir gride bölünür. Bu alanda engellerin bulunması veya bulunmamasına bağlı olarak gridin durumu değişebilir. Örneğin, başlangıçta hiçbir engel olmadığını varsayar. Bu ve benzeri varsayımlar altında, mevcut koordinattan hedef koordinata en kısa yolu bulmaya çalışır. Araç, önceden bilinmeyen engelleri artık biliniyor olması gibi yeni bir harita bilgisi keşfettiğinde, bu bilgiyi kendi haritasına ekleyerek güncelleme yapar ve gerekli ise mevcut konumundan hedef konuma en yakın yolu yeniden planlar. Hedef konuma ulaşana kadar bu işlemleri tekrarlar veya hedef konuma ulaşamayacağını belirler.

D\* algoritmasındaki maliyet hesaplaması heuristik fonksiyon içermez. Hedef hücreden geriye doğru işlem yapılır. Bitişik hücrelerden hedef hücreye hareket etme maliyeti hesaplanır ve minimum maliyet seçilir. Daha sonra başlangıç hücresine



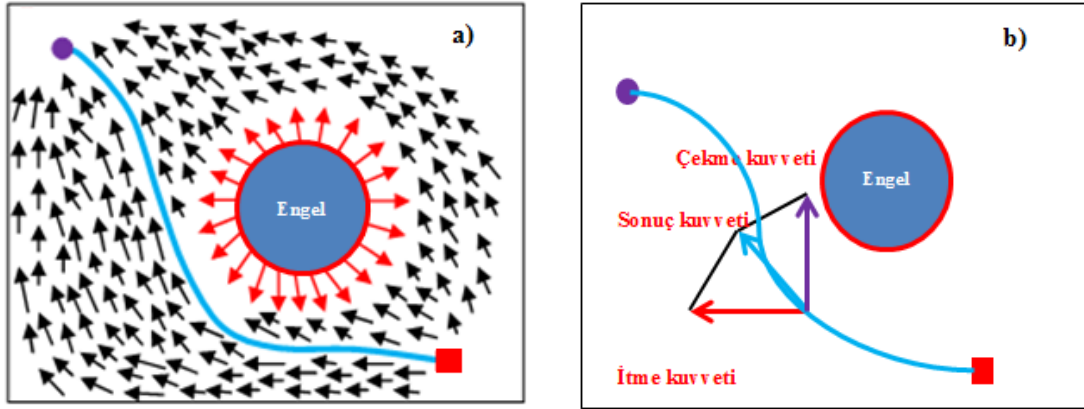
dođru yayılır. Griddeki her X hücre için, maliyet (Y, X) Y'den X'e geçiř maliyetini gösterir ve  $h(X, G)$  ise X'den G'ye geçiř ile ilgili olan toplam maliyeti ifade eder. Burada hedef G ile temsil edilmektedir. Y'den G'ye hareket etmenin toplam maliyeti Denklem 1.2 ile ifade edilmektedir.

$$\text{Maliyet (Y,G)} = \text{Maliyet (Y,X)} + h(X,G) \quad (1.2)$$

Otonom rotalarda D \* algoritma tabanlı çeřitli yol planlama algoritmaları uygulanır. Soh Chin Yun ve ark. (2010) tarafından yürütölen bir arařtırmada D \* Lite algoritması geliřtirilmiřtir. MATLAB yardımıyla geliřtirdikleri algoritmayı kullanarak bařlangıç ve hedef noktaları arasındaki en kısa yol bulmaya çalıřmıřlarıdır. Bu çalıřmada mevcutlara göre daha etkin sonuçlar elde edildiđi görölmüřtür (Yun ve ark., 2010).

### c. Yapay potansiyel alan

Khatib tarafından mobil robot yol planlaması için Yapay potansiyel Alan (APF) yöntemi geliřtirilmiř olup yöntem günümüzde kullanılmaktadır (Khatib, 1986). Potansiyel alan metodunda hedef mobil robot üzerinde çekme etkisi oluřtururken engeller itme etkisi oluřturmaktadır. Bu iki etkinin sonucu olarak robot engellerden kaçınırken hedefe yönlenmektedir. C-uzayına yerleřtirilen bir nokta mobil robot olduđunda engeller arasındaki çekim kuvvetine sahip hedefe dođru ilerler (Vadakepat ve ark., 2001). Mobil robot, algoritmasının yapısında eřit sayıda itici ve çekici kuvvet olması durumunda veya itici kuvvetin hedef noktasının yakınında bulunmadıđı durumlarda karar mekanizmasını geliřtirmesi gerekmektedir (řekil 1.3.). Bu durumda çözümler genellikle yerel minimum noktasına takılmıř olduđu anlamına gelmektedir (Hamani ve ark., 2013). Yerel minimum noktasına takılma sorununun üstesinden gelmek amacıyla robot ve hedef arasında göreceli konum, hız ve ivmelenme gibi parametreleri kullanarak yeni bir form geliřtirilmiřtir. Geliřtirilen algoritma yöröngenin kalitesini artırmak ve robotun hedef, engel yakınında kararsızlıklarının azaltılmasını sađlamak için kullanılmıřtır (Yin ve ark., 2009).

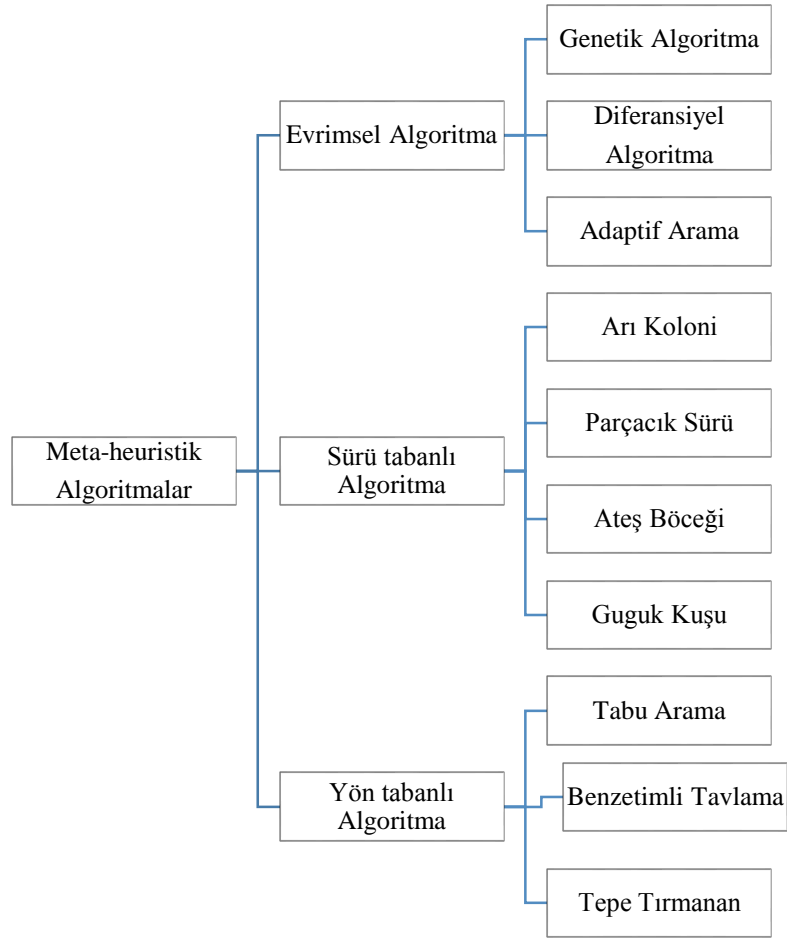


Şekil 1.3. Yapay potansiyel alan yaklaşımı ile mobil robot navigasyonu

### 1.1.2. Meta-sezgisel algoritmalar

Meta-sezgisel algoritmalar, mobil robotlarda navigasyon sisteminin modellenmesi için klasik yaklaşımlara alternatif olması amacıyla yaygın olarak kullanılmaktadır. Şekil 1.4.'te literatürde karşılaşılan meta-sezgisel algoritmalar sistematik bir şemayla verilmektedir.

Tez çalışmasında, özellikle kısa yol bulma konusu üzerinde durulduğu için, literatürde en iyi sonuç veren meta-sezgisel algoritmalarından aşağıda bir kaçını incelemiştir.



Şekil 1.4. Literatürde bulunan meta-sezgisel algoritmalar

#### a. Genetik algoritma

Genetik Algoritmalar (GA) doğal evrim sürecini taklit eden stokastik bir global arama yöntemidir. Genetik algoritma optimizasyona doğru çözümü bilmeden başlar ve tamamen çevresinden gelen cevaplara bağlı olarak süreci tamamlar (Thomaz ve ark., 2006). En iyi çözüme ulaşmak için seçim, mutasyon ve çaprazlama gibi operatörleri kullanmaktadır. Algoritma çeşitli bağımsız noktalarda paralel arama yaparak yerel minimumdan kaçınır ve en uyguna yakın çözüme yakınsama gerçekleştirir. GA kullanılarak elde edilen yolların uygunluk değerleri, engellerden kaçınarak ve hedefe bağlı pozisyona göre güvenli bir şekilde değerlendirilir.

GA'ya dayalı mobil robot yol planlamasıyla ilgili birçok çalışma bulunmaktadır. Samadi ve arkadaşları, GA ile bir mobil robot için yol planlamasının optimizasyonu

üzerine çalışmışlardır. Engelli bir çevrede en uygun yolun bulunmasında GA'yı kullanmışlardır. Yolun uzunluğu ve maliyeti optimize edilerek yol planlama algoritmasını geliştirmişlerdir. Buna ek olarak GA'nın seçim ve çaprazlama operatörlerini değiştirerek hesaplama süresini azaltmışlardır (Samadi ve ark., 2013). Jigolo ve arkadaşları ise GA' başlangıç popülasyonu oluştururken olasılığa dayanan hızlı bir rastgele arama yöntemi geliştirerek algoritmanın hesaplama karmaşıklığını ve yakınsama hızını artırmışlardır. Geliştirilen bu özellikler sayesinde mobil robot yol planlamada hızlı ve etkin bir şekilde en kısa yolun üretilmesini sağlamıştır (Jianguo ve ark., 2011).

#### b. Karınca koloni optimizasyonu

Gerçek karınca koloni davranışlarının matematiksel modelleri üzerine dayalı bir algoritmadır. Karınca çevre şartlarına göre besin kaynağı ile evi arasında gidebileceği yolları belirlemektedir. Belirlenen yollardan birinden ilk geçen karınca yola feromon adında bir koku bırakmaktadır. Eğer yol kısa ise bu koku daha yoğun olmaktadır ve diğer karıncalar da aynı şekilde yolda devam etmektedirler. İki yolun kesiştiği noktada karınca hangi yola gideceğini belirlemektedir. Hangi yolu seçeceğine ilk önce koku miktarının yoğunluğuna göre ikinci olarak ise gelişigüzel bir ölçüte göre karar vermektedir. Bu gelişigüzel seçimin nedeni ise bütün karıncaların aynı yolda gitmesini engelleyerek yeni ve daha kısa yolları keşfetmektir (Karaboğa, 2011).

Hiang ve arkadaşları tarafından yapılan bir çalışmada Karınca Koloni Optimizasyonu (ACO) kullanılarak mobil robotların yol planlaması gerçekleştirilmiştir. ACO algoritmasıyla mobil robotlar için yol planlama sağlanırken karıncaların, başlangıç konumundan (yuva) hedef konumuna (besin) kadar tüm olası hareket yolları belirlenir. Bu yollar arasından seçilecek olan yol karıncaların bıraktığı feromon miktarına bağlı olarak belirlenir. Sonuç olarak en kısa yolu feromon izinin yoğunluğuna göre seçmişlerdir (Chia ve ark., 2010). Zhangqi ve arkadaşları ise ACO temelli mobil robot yol planlamada kararlılık eksikliği, erken yakınsama hızı gibi sorunlarla karşılaşmışlardır. Bu sorunu çözmek için ACO algoritmasının parametreleri GA ile optimize edilerek kullanılmıştır. Simülasyon sonuçlarına göre geliştirilen ACO ile daha güçlü arama optimizasyon kabiliyeti, daha fazla kararlılık

ve daha kapsamlı performans gösteren yol planlama yapılmasını sağlamışlardır (Zhangqi ve ark., 2011).

### c. Bakteriyel besin arama optimizasyon algoritması

Bakteriyel Besin Arama Algoritması, E. coli bakterisinin beslenme davranışından esinlenerek karmaşık mühendislik problemlerini çözmek için geliştirilmiş bir hesaplama tekniğidir. Bakteriler, karmaşık yaşam formlarındaki diğer canlılara göre çok daha basit yapıdadırlar. Sınırlı algı ve hareket kabiliyetlerini kullanarak optimum düzeyde enerji harcıyıp beslenme faaliyetlerini gerçekleştirmeleri gerekmektedir. Diğer yaşam formlarına nazaran modellenebilmeleri daha kolaydır. Bu türden canlılardan biri olan E. coli bakterisi, yapısı ve çalışma şekli en iyi anlaşılan mikroorganizmalardan birisidir. E. coli bakterisi besin maddesine ulaştığında diğer bakterileri uyarıcı etkiye sahip kimyasal bir madde salgılamaktadır. Bu madde, diğer E. Coli bakterilerinin besini bulan bakterinin bulunduğu yere doğru hareket etmesini sağlamaktadır. Eğer gıda yoğunluğu çok fazla ise bakteriler kenetlenerek grup halinde hareket edebilmektedirler (Başbuğ, 2008).

Parhi ve ark. tarafından yapılan çalışmada, bakterilerin beslenme davranışına dayanan Bakteriyel Besin Arama Optimizasyon Algoritması (BFOA) kullanılarak engellere çarpmaksızın başlangıç ve hedef nokta arasında mobil robot yol planlaması gerçekleştirilmiştir. Simülasyon uygulaması dört veya altı engelden oluşan iki farklı çevre türünde gerçekleştirilmiştir. Simülasyon sonuçları mobil robotun hedefe başarılı bir şekilde ulaştığını göstermektedir (Parhi ve ark., 2012). Montiel ve arkadaşları tarafından yapılan bir başka çalışmada ise statik ve dinamik engellerin bulunduğu gerçek dünya ortamında BFOA kullanılarak mobil robot yol planlaması için yeni bir yöntem önerilmiştir. Çeşitli simülasyon ortamlarında BFOA yöntemi, Yapay Potansiyel Alanı (APF), Genetik Potansiyel Alanı (GPF) ve Sahte Bakteriyel Potansiyel(PBPF) yöntemlerinin performansları karşılaştırılmıştır. Bu karşılaştırma sonuçlarına göre önerilen yöntem uygulanabilir, optimal ve güvenli bir yol sağlamıştır (Montiel ve ark., 2015).

#### d. Yapay arı koloni algoritması

Sürü zekâ temelli algoritmalarından biri olan Yapay Arı Koloni Algoritması (ABC) da son yıllarda optimizasyon problemlerinin çözümünde yaygın olarak kullanılmaktadır. Çok boyutlu ve çok modelli optimizasyon problemlerini çözmek için arı sürülerinin yiyecek arama davranışlarının taklit edilerek modellenmesine dayanan ABC algoritması 2005 yılında Karaboğa tarafından önerilmiştir. Arıların yiyecek aramak için gittikleri kaynaklar, algoritmada çözülmek istenen problemin olası çözümlerini, kaynaklardaki nektar miktarı ise çözümün kalitesini (amaç fonksiyon) ifade etmektedir. ABC algoritmasında, bir kolonide işçi arı, gözcü arı ve kâşif arı olmak üzere üç çeşit arı bulunmaktadır. Her bir kaynak için görevli bir arı bulunmaktadır. ABC algoritması en fazla nektara sahip kaynağı bularak olası çözümler arasından problemin maksimum ya da minimum noktasını bulmaya çalışmaktadır (Karaboğa, 2005).

Agarwal ve ark. tarafından yapılan bir çalışmada yapay arı koloni algoritmasına dayanan mobil robotların yol planlama probleminin çözümü için yeni bir metot önerilmiştir. Bu metotta kovana yakın farklı yiyecek kaynakları bulmak için bal arılarının toplu davranışlarından esinlenilmiştir. Önerilen yöntem iki adımdan oluşmaktadır. İlk adım başlangıç noktasından hedef noktasına kadar çarpışmaksızın bir başlangıç yolu oluşturulmasıdır. Engellere çarpmayı önlemek için engeller etrafında yeni (sanal) sınırlar tanımlanmıştır. İkinci adım ise başlangıç yolunun yapay arı koloni algoritması ile optimize edilmiştir. Optimize edilecek yol seçilirken engellerin sanal sınırları göz önünde bulundurulmuş ve bundan dolayı en kısa yol sanal sınırlar üzerinden geçirilmemiştir (Agarwal ve ark., 2013). Abbas ve arkadaşları ise çalışmalarında ABC algoritma tabanlı yönlendirilmiş yapay arı koloni(DABC) algoritması geliştirmişlerdir. DABC algoritmasında yol planlama problemleri en iyi arının mevcut yönü kullanılarak diğer arıların da bu yöne yönlendirilmesiyle çözülmektedir. Üç farklı çevre ortamında önerilen algoritmanın performansı ABC ve en yeni algoritmalarla karşılaştırılmıştır. İlk iki çevre ortamında DABC algoritması diğer algoritmalarla karşılaştırıldığında performans farkı çok

azdır. Engellerin çok olduğu çevre ortamında ise çok daha iyi performans göstermektedir (Abbas ve ark., 2014).

e. Yapay sinir ağı algoritması

Yapay Sinir Ağları (YSA), insan beyninin bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA ile basit biyolojik sinir sisteminin çalışma şekli simüle edilmektedir. Simüle edilen sinir hücreleri nöronlar içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünmeye ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

Sung ve ark. çalışmalarında kapalı ortamlarda mobil robot için hem engel önleme hem de hedefe ulaşmak için etkili bir yöntem gerçekleştirmişlerdir. Onlar duvar ve engellerin konumunu yapay sinir ağlarında geri yayılım yaklaşımıyla tahmin etmişlerdir. Global yol planlama mobil robotun hedefe giderken düz yol belirlemiştir. Düz yol üzerinde hareket eden mobil robot diğer bir deyişle yerel yol planlamada engel ile karşılaşır duvar takip algoritmasını kullanarak engellerden kaçınmaktadır. Ayrıca bu çalışmada yerel minimum noktasına düşmemek için basit bir karar mekanizması geliştirmişlerdir. Simülasyon sonuçları önerilen yöntemin hedefe daha kısa sürede ve hızlı bir şekilde vardığını göstermektedir (Sung ve ark., 1999). Shamsfakhr ve ark. tarafından gerçekleştirilen bir çalışmada ise istatistiksel boyut azaltma tekniği ile yapay sinir ağı yaklaşımını birleştirerek yeni bir yöntem geliştirilmiştir. Geliştirdikleri bu yöntemle doğru ve hızlı bir robot navigasyonu sağlamak aynı zamanda engellerden kaçınmayı da amaçlamışlardır. Önerilen yöntemde bir geri yayılım öğrenme algoritmali fonksiyon yaklaşımına dayanan iki farklı ileri beslemeli sinir ağı kullanılmaktadır. Robot çalışma ortamını görselleştirmek için 180° lazer mesafe sensör okumaları kullanılmaktadır. Yöntem

gerçek dünya verileri üzerinde test edilmiş ve deneysel sonuçlar önerilen yöntemin etkinliğini doğrulamak için kullanılmıştır. Robotun kinematik kısıtları göz önüne alınarak algoritma tarafından rasyonel bir davranış geliştirilmiştir. Navigasyon deneyleri, dönüş açısı için doğru değerler elde ettiğinden dolayı robotun kararlı hareket etmesini sağlamıştır. Bu sebeple hem yolun uzunluğunu hem de bilinmeyen bir ortamdaki navigasyon süresini en aza indirmişlerdir (Shamsfakhr ve ark., 2017).

#### f. Bulanık mantık

Bulanık mantık, insan zihnindeki öznel belirsizlikleri ifade etmek için düşünülmüştür. Uzman kişi dilsel niteleyiciler olarak tanımlanabilecek; uygun, çok uygun değil, yüksek, fazla, çok fazla gibi günlük yaşantımızda sıkça kullandığımız kelimeler doğrultusunda esnek bir denetim mekanizması geliştirir. Bulanık denetim bu tür mantıksal ilişkiler üzerine kurulmuştur (Zadeh, 1965; 1971).

Bulanık mantık algoritmasında, robot navigasyonunun karar verme işlemi IF-THEN kuralları dizisi ile ifade edilebilir. Bu işlemin kolayca gerçekleştirilmesi için, gezinme sorunu daha basit görevlere ve davranışlara dönüştürülür. H. Chang ve T. Jin mobil robot yol planlama problemini çözmek için bir bulanık çıkarım modeli geliştirmişlerdir. Hedefin / engellerin yerleri ve bilinmeyen dinamik ortamlarda robotun mevcut hızı sensörlerden algılanır. Araştırmacılar bu modelde optimal dönüş açısı  $\theta$ 'u bulmak için hedef yönlendirme, engelden sakınım ve dönüş hareketi olmak üzere üç büyük navigasyon hedefini maliyet fonksiyonuna dahil etmişlerdir. Mobil robot çevreye bağlı olarak ve maliyet fonksiyonlarının ağırlıklarını değiştirerek navigasyonu gerçekleştirmektedir (Chang ve ark., 2013). Pradhan ve ark. statik ve dinamik engellerin bulunduğu bir ortamda çoklu mobil robotların navigasyonunu farklı bulanık mantık denetleyiciler kullanarak gerçekleştirmişlerdir. Çalışmada mobil robotların farklı durumları dikkate alınarak bir bulanık kurallar dizisi belirlenmiştir. Bulanık kurallar dizisi planlanan yörüngeye bağlı olarak değişmektedir. Bulanık mantık kontrolerinin girdisi olarak sol, sağ, ön engel mesafesi ve hedef açı parametrelerini almışlardır. Çıktı olarak ise mobil robotun sağ ve sol tekerlek hızları alınmıştır. Sonuç olarak kurallar bütünü sayesinde bine yakın

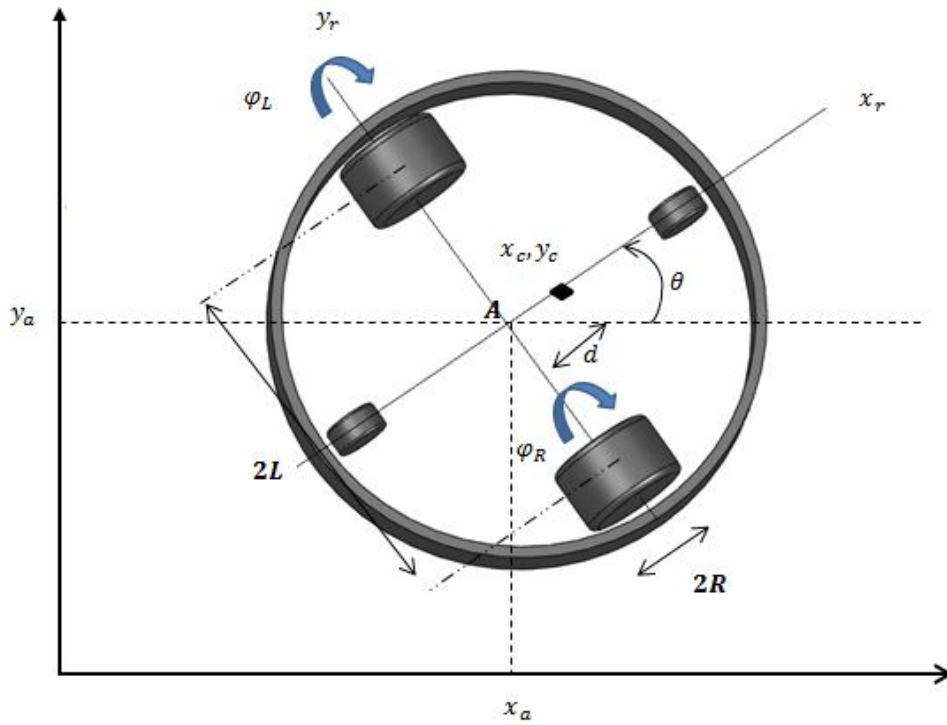


mobil robot, birbirlerine ve engellere çarpışmaksızın navigasyon işlemlerini sağlamışlardır (Pradhan ve ark., 2009).

## 1.2. Mobil Robot Sisteminin Modellenmesi

### a. Kobuki mobil robotun kinematik modeli

Diferansiyel sürüş tekniği mobil robotun ortak eksen üzerindeki bağımsız iki tekerleğinin farklı hızlara sahip olmasına dayalı bir sürüş biçimidir. Diferansiyel sürümlü robotlar modellenirken robotun sağ ve sol uçlarında bulunan tekerlekler esas alınmış, robotun dengesini sağlayan sarhoş tekerlerde ortaya çıkan sürtünme ihmal edilmiştir. Diğer taraftan bu modelleme, robotun tamamen düz bir zeminde ilerlediği varsayımına dayanmaktadır.



Şekil 1.5. Kobuki robotun kinematik model parametreleri

Şekil 1.5.'te uygulamada kullanılan Kobuki mobil robotun modeli gösterilmektedir. Kobuki mobil robot iki tahrik tekerleği (sol ve sağ tekerlekler), herhangi bir yük dağılımı için robotu destekleyen iki küçük boş tekerlekten oluşmaktadır (Renawi1 ve ark., 2017).

Çalışmada kullanılan diferansiyel sürüş mobil robotun kinematik modeli denklem eşitlikleri ile belirlenir. Aşağıdaki denklemlerde verilen kısaltmaların açıklamaları Tablo 1.1.'de verilmektedir.

Tablo 1.1. Denklem 1.3-17'de verilen kısaltmaların açıklamaları

$V_c$	: Mobil robotun lineer hızı [m/s]
$V_r$	: Sağ tekerlek lineer hızı [m/s]
$V_l$	: Sol tekerlek lineer hızı [m/s]
$L$	: Ön iki tekerleğin merkeze olan mesafeleri [m]
$x, y$	: Koordinat sistemi.
$\omega_c$	: Mobil robotun açısal hızı [rad/s].
$\varphi_R$	: Sağ tekerlek açısal hızı
$\varphi_L$	: Sol tekerlek açısal hızı
$\theta$	: Robotun dönme açısı [rad]
$I_{cc}$	: Anlık eğrilik merkezi
$R$	: Tekerleğin dinamik yarıçapı [m].
$\tau_R, \tau_L$	: Sağ ve sol motorun torqu

Kobuki robotun tekerleklerinin doğrusal hızı açısal hız açısından aşağıdaki gibi ifade edilmektedir.

Sağ ve sol tekerleğin doğrusal hızı;

$$V_R = \omega_R x r \quad (1.3)$$

$$V_L = \omega_L x r \quad (1.4)$$

$$V_C = \dot{\theta} r_{ICC} \quad (1.5)$$

$$V_L = \dot{\theta} (r_{ICC} - \frac{2L}{2}) \quad (1.6)$$

$$V_R = \dot{\theta} (r_{ICC} + \frac{2L}{2}) \quad (1.7)$$

$$V_C = \frac{V_L + V_R}{2} \quad (1.8)$$

$$\omega_C = \dot{\theta} = \frac{V_R - V_L}{2L} \quad (1.9)$$

Tekerlekler hızları sağa ve sola doğru hızlandırılarak yeniden düzenlenir:

$$V_R = V_C + \frac{d\omega_C}{2} \quad (1.10)$$

$$V_L = V_C - \frac{d\omega_C}{2} \quad (1.11)$$

Hızların matris görünümü:

$$\begin{bmatrix} V_R \\ V_L \end{bmatrix} = \begin{bmatrix} 1 & \frac{2L}{2} \\ 1 & \frac{-2L}{2} \end{bmatrix} \begin{bmatrix} V_C \\ \omega_C \end{bmatrix} \quad (1.12)$$

S durumunda tanımlanırsa:

$$S = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} \quad (1.13)$$

x,y koordinat sistemin hareket konumu,  $\theta$  ise z etrafındaki dönme açısını belirler.

$$\dot{x} = V_C \quad (1.14)$$

$$\dot{y} = 0 \quad (1.15)$$

$$\dot{\theta} = \omega_C \quad (1.16)$$

$$\dot{S} = R \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V_C \cos \theta \\ V_C \sin \theta \\ \omega_C \end{bmatrix} \quad (1.17)$$

b. Kobuki mobil robotun dinamik modeli

Dinamik model, mobil robotun hareketini etkileyen kuvvetleri göz önünde bulundurarak mekanik bir sistem hareketinin modellenmesidir. Mobil robotlarda dinamik model mobil robot hareketinin simülasyonu ve hareket kontrol algoritmalarının tasarımı için önemli bir yere sahiptir (Dhaouadi ve ark., 2013).

Kobuki mobil robotun dinamik modeli üç tekerleğe sahip diferansiyel sürüş mobil robotların dinamik modeline benzemektedir. Kobuki mobil robotun dinamik modelini elde edilmesinde Lagrange yaklaşımı kullanılmıştır. Lagrange yaklaşımı, mekanik sistemlerin hareket denklemlerini formüle etmek için çok güçlü bir yöntemdir. Lagrange yöntemi tarafından verilen sistemin kinetik ve potansiyel enerjileri göz önünde bulundurularak hareket denklemleri türetilir. Lagrange denkleminde potansiyel enerji sıfır olarak kabul edilmiştir ve genelleştirilmiş koordinat vektörünün seçilmesiyle birlikte kinetik enerji terimleri bulunmuştur (Dhaouadi ve ark., 2013).

Diferansiyel sürüş mobil robot dinamik modeli için (Denklem 1.19) verilen mobil robot yapısının kütlesi  $m_c$  tekerlekler ve motor kütlesi olmadan ve  $m$  ise tekerlekler ve motorun kütlesi dahil edilerek robot yapısının kütlesidir.

Şekil 1.5.'e göre genelleştirilmiş koordinatlar:

$$q = [x_a \ y_a \ \theta \ \varphi_R \ \varphi_L]^T \quad \text{olmak üzere} \quad (1.18)$$

Lagrange denklemi kullanılarak elde edilen denklem:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_1} \right) + \left( \frac{\partial L}{\partial q_1} \right) = F - \Lambda(q)^\lambda \quad (1.19)$$

Türev işlemlerinden sonra elde edilen model:

$$\left( m + \frac{2l_w}{R^2} \right) \dot{V}_c - m_c dW_c^2 = \frac{1}{R} (\tau_R + \tau_L) \quad (1.20)$$

$$\left(I + \frac{2L^2 I_w}{R^2}\right) \dot{W}_c - m_c dW_c V_c = \frac{L}{R} (\tau_R - \tau_L) \quad (1.21)$$

Yukarıda türetilen dinamik model, robotun doğrusal ve açısal hızlarını elde etmek için giriş olarak motor dönme momentine(tork) bağlıdır. Torklar belirtilen şartlara göre DC motor modelini kullanarak elde edilmiştir. Lineer ve açısal hız elde edildikten sonra sistem durumu S elde edilebilir.

### 1.3. Literatür Araştırması

Önceki başlıklarda, tez çalışması kapsamında yapılanları daha iyi takip edebilmek için gerekli olan yol planlama kavramı, mobil robotun kinematik ve dinamik modellenmesi, yol planlama algoritmaları ile ilgili genel bilgiler verilmiştir. Literatür araştırması bölümünde ise tez çalışmasının konusuyla daha yakından ilgili olan hibrit algoritma tabanlı mobil robotların yol planlamasında bahsedilmektedir.

Çok robotlu yol planlama problemi bir çeşit optimizasyon problemi olarak görülmektedir. Son zamanlarda çok kısıtlı optimizasyon problemlerini çözmek için heuristik tabanlı algoritmalar önerilmektedir. Literatürde meta-sezgisel algoritmaların birleştirilmesi sonucunda ortaya çıkan PSO-GSA, SA-ANN, SA-ACO, ACO-FUZZY, ACO-PSO, PSO-AG, PSO-Tabu Search, GA-Dijkstra, MO-FA, CS-ANFIS, BAT-PSO, GA-PSO, ACO-EA, ACO-GA, PSO-DV, ACO-APF gibi hibrit algoritmalar çoklu mobil robotların yol planlamasında kullanılmaktadır.

Literatürdeki çalışmalarda hibrit algoritmaların tercih edilmesinin nedeni olarak yol planlama problemi optimize edilirken global en iyi çözüme ulaşılmasıdır. Hibrit algoritmalarla algoritmaların çeşitliliği ve sağlamlığı arasındaki denge kurulmaktadır. Hibrit algoritma ile ilgili literatürde mobil robot yol planlama alanıyla ilgili çok sayıda makale bulunmaktadır.

2017 yılında Hidalgo-Paniagua ve ark. tarafından yapılan bir çalışmada doğada bulunan ateş böceklerinin yanıp sönmeye davranışını esas alarak yeni bir çok-amaçlı ateş böceği yaklaşımı(MO-FA) geliştirmişlerdir. Bu algoritmayı mobil robotların yol planlama problemini çözmek için kullanmışlardır. Önerilen MO-FA algoritması ile doğru ve etkin çözümleri elde etmek için üç farklı amaç fonksiyonu tanımlamışlardır.

Enerji tüketimini minimize eden amaç fonksiyonuyla yol güvenliği, uzunluğu ve düzgünlüğü istenmektedir. Sekiz farklı gerçek senaryoda önerilen algoritmalar ile yol hesaplanmıştır. Diğer taraftan MO-FA algoritmasının performansını kanıtlamak için Ahmed ve Deb (2013) ve Jun ve Qingbao (2010) tarafından önerilen farklı NSGA-II algoritmaları ile karşılaştırılmıştır. Özellikle değerlendirme yapmak için HV(hiper hacmi) ve SC(ayarlanmış alan) çok amaçlı kalite ölçütleri uygulanarak değerlendirme sağlamışlar ve istatistiksel bir analiz gerçekleştirmişlerdir. İstatistiksel analizler yol planlama problem çözümünde özgün MO-FA yaklaşımının diğer yaklaşımlara göre daha iyi sonuçlar verdiğini kanıtlamıştır (Jun ve Qingbao, 2010 ; Ahmed ve Deb, 2013; Hidalgo-Paniagua ve ark., 2017).

2017 yılında Dao ve ark. çalışmalarında ise arı tozlaşma optimizasyonuna(BPO) dayalı optimal mobil robot yol planlaması için çok amaçlı yeni bir yaklaşım önermişlerdir. Bu yaklaşım MBPO(çok amaçlı arı tozlaşma optimizasyon) algoritması optimizasyon probleminde meydana gelen yerel optimum noktaya takılmama, çeşitlilik sağlama ve paralel hesaplama gerçekleştirmek için yapay arı koloni(ABC) ve çiçek tozlaşma algoritmalarıyla(FPA) birleştirilerek oluşturulmuştur. ABC algoritmasının yapısında bulunan arıların ve FPA algoritmasındaki polenlerin paralel yapıları ve konumlarını değiş tokuş ederek güçlü bireyler en iyi konuma hareket edilmesini sağlamışlardır. En iyi konuma gelen bireyler de mobil robot hareket yerlerini belirlemektedir. Mobil robotun takip edeceği yol minimize edilmek amacıyla oluşturulan amaç fonksiyonunda en kısa mesafe ve pürüzsüz yol parametrelerini kullanmışlardır. Çalışma ortamı engellerin şekilleri ve konumları ayrıca Pareto çözümüne dayanan bir ayrıştırma çözümüyle eşleştirilen MBPO ajanları için kurulan robotun başlangıç ve bitiş konumlarını içermektedir. Simülasyon sonuçlarına bakıldığında önerilen MBPO algoritması ABC ve FPA algoritmalarıyla kıyaslandığında daha verimli performans elde ettikleri görülmektedir (Dao ve ark., 2017). ABC algoritmasıyla gerçekleştirilen diğer bir çalışma ise 2015 yılında Contreras-Cruz ve ark. tarafından yapılmış olup bu çalışmada evrimsel programlama(EA) algoritma ile hibrit bir algoritma geliştirmişlerdir. Önerilen yaklaşımda yerel arama işlemini yani robotun başlangıç ve hedef konumunu birleştiren bir başlangıç yolunu ABC algoritmasıyla gerçekleştirmişlerdir. Daha sonra

yerel arama işlemi tarafından bulunan olası yolun kısa, pürüzsüz ve çarpışma olmaksızın serbest yol olabilmesi için EA tarafından optimize işlemi gerçekleştirmişlerdir. Önerilen hibrit algoritma, planlama performanslarına göre Dijkstra algoritması tabanlı klasik olasılıklı yol haritası metodu (PRM) ile karşılaştırılmıştır. Her iki planlama sisteminin verimliliğini ölçmek için benchmark haritalarında test etmişler ve yol uzunluğu, yol pürüzsüzlüğü, arama maliyeti ve planlama başarı oranı gibi ölçüm parametreleri kullanılarak karşılaştırmışlardır. Önerilen yöntem, dağınık engellerin bulunduğu ortamlarda daha kısa sürede yollar ürettiğini bunun sebebi olarak da PRM'nin öğrenme evresinde görünürlük grafiği oluşturmadığından kaynaklandığını belirtmişlerdir. Bunlara ek olarak yöntemin uygulanabilirliği ve fizibilitesini göstermek için sistem deneysel bir robotik platformda uygulanmıştır (Contreras-Cruz ve ark., 2015).

2016 yılında Das ve arkadaşları yerçekimi araştırması algoritması (IGSA) ile parçacık sürüsü optimizasyon (IPSO) algoritmasının birleştirilmesi ile karmaşık çevre ortamlarında çoklu robotlar için en uygun yol yörüngesini belirlemek için yeni bir metodoloji önermişlerdir. Bu yaklaşımda IGSA'nın hareket mekanizma özelliğine IPSO'nun sosyal nitelik özelliğini eklemişlerdir. Ayrıca IPSO-IGSA, IGSA hızlanma ve parçacık konumlarını aynı anda IPSO hızıyla güncellendiğinden dolayı arama ve kullanma arasındaki etkin dengeyi korumuşlardır. Robot IPSO-IGSA algoritmasını kullanarak mevcut haritada hareket etme için bağımsız kararlar almak, koordineli çalışmak ve işbirliği yapmak gibi özelliklere sahiptir. Aynı araştırmacıların diğer çalışmalarından farklı olarak karmaşık ortamda çoklu mobil robotlar için geliştirilmiş PSO ve diferansiyel hız (DV) algoritması birlikte kullanılarak hibrit algoritma formunda yeni bir algoritma geliştirmişlerdir. Hibrit algoritma ile mobil robota en uygun yönü belirlemeyi amaçlamışlardır. Problemden mesafe, engelden sakınım, engellerin dinamik konumunun tahmini ve yolun pürüzsüzlüğü gibi farklı kriterlere dikkat çekmek için üç hedef fonksiyonu formüle edilmiştir. Farklı hedef fonksiyonlarını birleştirerek amaç fonksiyonunu oluşturmuşlardır. Araştırmacılar her iki çalışmada bulunan amaç fonksiyonlarının amacını, mobil robotların hedefe varma sürelerini ve gidecekleri yolu minimize edilmesi olarak belirlemişlerdir. Bunlara ek olarak performans kriteri olarak dönüş sırasındaki güç tüketimi, dönme

sayısı, yolculuk süresi, yol sapması ve mobil robotların kendi hedeflerine ulaşmak için gereken dönüş sayısı belirlenmiştir. Son olarak, çoklu mobil robot yol planlamasında benzer ortamlarda IPSO-DV hibrit algoritması, IPSO ve DE algoritmalar ile kıyaslanırken IPSO-IGSA hibrit algoritması da IPSO ve IGSA algoritmaları analitik ve deneysel olarak karşılaştırılmıştır. Khepera mobil robotun benzer ortam içeren simülasyon ve fiziksel deneylerde yapılan testler sonucunda, en uygun yörünge yol uzunluğu, dönüş sayısı ve varış süresi açısından IPSO-DV algoritmasının IPSO ve DE algoritmalarından daha iyi performans gösterirken, IPSO-IGSA algoritması da IPSO ve IGSA algoritmalarından daha iyi performans göstermektedir (Das ve ark., 2016, Behera ve ark., 2016). 2013 yılında Purcaru ve arkadaşları PSO ve GSA ile yaptıkları çalışmada ise GSA'nın yerel arama kabiliyeti ile PSO algoritmasının sosyal düşünme özelliği birleştirilerek yeni bir hibrit yol planlama algoritması tasarlamışlardır. Yol planlama problemini çözmek için hibrit algoritma tarafından optimizeyi sağlayan iki amaç fonksiyonu tanımlanmıştır. Algoritma mobil robotun çevredeki engellerden kaçınarak başlangıç noktasından hedef noktasına varana kadar takip etmesi gereken yol uzunluklarını en aza indirir. Aynı zamanda üretilen yörüngelerin tehlikeli bölgelerde çarpışma noktaları oluşturmamasını sağlamaktadır. Diğer taraftan da yörünge her noktasında statik çevrede var olan tehlikeli alanlara göre güvenli uzaklıkta kalmalarını sağlar. Önerilen hibrit PSO-GSA, PSO, GSA algoritmaları kullanılarak, farklı engellerin olduğu çoklu tehlike bölgelerine sahip farklı ortamlarda test edilmiştir. Bu test sonuçlarına göre PSO-GSA algoritmasının daha iyi sonuçlar verdiği gösterilmiştir (Purcaru ve ark., 2013).

2016 yılında Gigras ve ark. tarafından yapılan çalışmada ACO-PSO algoritmalarını birleştirilerek hibrit bir yöntem geliştirmişlerdir. Shi ve ark. yaptıkları çalışmada ACO-PSO algoritmasını kullanarak üç boyutlu çevre ve sualtı çevre modeli olmak üzere iki farklı çalışma ortamını kullanarak robotun yol planlama problemini çözmeyi amaçlamışlardır. Bu çalışmalarda ilk olarak çevre modeli Bitmap yöntemi ile oluşturulmuştur ve robotun hareketleri çerçeve alan metodu kullanılarak parçacık hareketleri belirlenmiştir. ACO olumlu geribildirim, paralellik ve diğer algoritmalarla kolayca hibrit olma gibi üstün özelliklere sahiptir. Parametre seçiminde deneyim ve



tekrar deneylere ihtiyaç duyulduğundan dolayı ACO ile PSO algoritmaları birleştirilmiştir. Böylece, ACO'daki parametreler PSO aracılığıyla optimize edilmekte ve parametreler kendiliğinden uyarlanabilmekte, böylece ACO'nun optimum performansı artmaktadır. Sonuç olarak, simülasyon deneyleri 3 boyutlu çevre modeli ve sualtı çevre modeli olmak üzere iki farklı ortamda araştırılmıştır. Simülasyon sonuçlarından elde ettikleri verilere göre ACO-PSO yönteminin verimli ve uygulanabilir olduğunu kanıtlamışlardır (Shi ve ark., 2008; Bu ve ark., 2008; Gigras 2015).

Huang ve ark. ve Ju ve ark. tarafından yapılan çalışmalar da mobil robot için başlangıç ve bitiş noktaları arasında en uygun yolu bulmak amacıyla hibrit bir meta-sezgisel algoritma geliştirmişlerdir. Hibrit meta-sezgisel algoritma oluşturmak için GA ve PSO algoritmalarını kullanmışlardır. İlk çalışmada PSO algoritmasının yapısında bulunan parçacık parametrelerine GA operatörlerinden çaprazlama ve mutasyon işlemi uygulanarak yeni çözümler bulmuşlardır. Geliştirilen hibrit algoritma ile geleneksel algoritmalarda karşılaşılan erken yakınsama ve zaman karmaşıklığı problemlerinin üstesinden gelmişlerdir. Hibrit GA-PSO algoritması tarafından üretilen olası ilk yol kübik B-spline tekniği ile geliştirilerek ortamda engellere çarpma olmaksızın optimale yakın bir şekilde yol pürüzsüzleştirilmiştir. Simülasyon sonuçları önerilen melez GA-PSO otonom robot navigasyonunda daha pürüzsüz global yol planlaması sağladığını göstermiştir. Araştırmacılar tarafından ikili ağaç kodlama yapısı ile yol planlama gerçekleştirmişlerdir. Çalışma alanında engeller olmaksızın, mobil robotun başlangıç ve bitiş noktaları ikili ağaç gösterim yönteminde bulunmaktadır. Düğümlerin sayısı çalışma ortamının karmaşıklığına bağlıdır. Önerilen ikili ağaç yapısı PSO'daki öğrenme problemini çözmektedir. PSO'da farklı uzunluklara sahip kromozomlar ile işlem yapabilmek için ikili ağacın yapısına sahte düğümler eklemişlerdir. Deneysel sonuçlar, önerilen hibrit yöntemin geleneksel grid tabanlı yöntemlere göre daha az düğüm noktası kullanıldığını ve engelsiz yol sayısının daha fazla olduğunu göstermiştir (Huang ve ark., 2011; Ju ve ark., 2014).

2015 yılında Bilbeisi ve ark. tarafından yapılan çalışmada çoklu mobil robotların yol planlaması PSO algoritmasıyla sağlanırken, mobil robotun engelden sakınma,

hareket kontrolü ve kullanımı Agoraphilic (AG) algoritmasıyla gerçekleştirilmiştir. Geliştirilen PSO-AG algoritmasını farklı sayıda robot(100, 200, 300, 400), değişen engel oranı (0%, 10%, 20%, 30%) içeren ve farklı zorluk seviyelerinde çevre ortamlarında test etmişlerdir. PSO-AG algoritmasını test etmek amacıyla çalışma süresi, yineleme sayısı, yakınsama hızı ve hedefe ulaşan robot yüzdesi gibi performans kriterlerini belirlemişlerdir. Simülasyon sonuçlarına göre PSO-AG algoritmanın farklı sayıda robot ve engel oranına göre tekrardan ölçeklenebilir yapıdadır. Bu nedenden dolayı robot sayısı ve engel oranının tekrarlanma sayısı üzerinde bir etkiye sahip olduğunu saptamışlardır. Ayrıca yakınsama hızı ve engel oranı ters orantılı iken robot sayısı ile yakınsama hızının doğru orantılı olarak değiştiğini tespit etmişlerdir (Bilbeisi ve ark., 2015).

2016 yılında Panda ve ark. tarafından gerçekleştirilen çalışmada karmaşık bir ortamda çoklu mobil robotların yol planlaması için Tabu Search(TABU) ile PSO algoritmalarının birleştirilmesiyle yeni bir yaklaşım elde edilmiştir. PSO-TABU algoritmasıyla minimum zamanda minimum adım sayısı ile hedef noktasına varılarak yol uzunluğunu optimize edilmiştir. PSO algoritmasıyla her iterasyon adımından sonra en iyi çözüm hesaplanmış daha sonra Tabu arama algoritmasıyla en yakın komşunun konumu belirlemek amacıyla arama gerçekleştirilmiştir. Simülasyon çalışmalarına göre PSO, TS ve PSO-Tabu algoritmaları yol uzunluğu açısından karşılaştırıldığında PSO-Tabu algoritmasının daha verimli sonuçlar verdiği görülmüştür (Panda ve ark., 2016).

2016 yılında Nandanwar ve ark. tarafından yapılan çalışmada bilinmeyen bir dış ortamda koordineli çoklu etmenlerin yol planlama, engellerden kaçınarak hedef izleme ve hedefe varma sorununu çözmek amacıyla BAPSO(BAT algoritma ve PSO) algoritması tasarlanmıştır. Yol planlamada bilinmeyen hedef konumunu bulmada performans kriteri olarak robotun hareketleri ve deney süresini belirlemişlerdir. Simülasyon ortamında gerçekleştirilen çalışmada BAPSO algoritmasının diğer algoritmalara göre daha iyi değerler verdiğini tespit etmişlerdir (Nandanwar ve ark., 2016).

2009 yılında Tan ve ark. tarafından yapılan çalışmada kısmen bilinen ortamlarda mobil robotların global yol planlaması için APF ve ACO algoritmaları birleştirilerek uygun bir yöntem geliştirmişlerdir. İlk olarak, serbest çalışma alanını oluşturmak için görünürlük grafik yöntemi kullanarak model oluşturmuşlardır. ACO algoritması, oluşturulan model üzerinde bilinen statik engellere çarpmaksızın Euclidean uzaklık fonksiyonunu kullanarak en kısa yolun bulunmasını sağlamıştır. Son olarak, APF algoritması bilinmeyen engellerle karşılaştığında, çarpışmayı önlemek için gerçek zamanlı olarak yerel bir yol planlaması oluşturmuştur. Simülasyon analizlerinden elde edilen verilere göre önerilen hibrit yaklaşımın yakınsama hızını arttırdığı buna ek olarak dinamik ve karmaşık ortamlarda da kullanımının uygun olduğunu belirlenmiştir (Tan ve ark., 2009).

2012 yılında Zhang ve arkadaşları tarafından yapılan çalışmada mobil robotların yol planlamasında verimliliği artırmak için Benzetim Tavlama(SA) , ACO ve çerçevesiz-dörtlü ağaç yaklaşımının da bulunduğu yeni bir SA-ACO olarak adlandırılan hibrit bir algoritma geliştirilmiştir. Onlar, çalışma ortamını düzeltmek ve haritaları görünümünü netleştirmek amacıyla çerçevesiz-dörtlü ağaç yaklaşımından yararlanmışlardır. SA algoritmasının performansı rastgele başlangıç noktalarına bağlı olduğundan dolayı bu noktalar oluşturulurken ACO algoritması kullanmışlardır. Teorik analiz ve simülasyon sonuçları çalışma ortamında mobil robotun hedef arama ve navigasyon hızını artırma işlemlerinde oluşturulan SA-ACO algoritmasının performans gösterdiği kanıtlanmıştır. Sonuç olarak bu algoritma sağlamlık ve kendi kendine uyarlanma avantajı sağlamıştır (Zhang ve ark., 2012).

SA algoritması ile ilgili diğer bir çalışmada ise GA algoritması ile birleştirilerek SA-GA algoritmasıyla mobil robot için global yol planlama tasarlanmıştır. SA algoritmasının yavaş yakınsama özelliğinden kaynaklanan problemi çözmek için GA algoritmasının yerel arama özelliği kullanılmıştır. Bu şekilde hibrit algoritma GA'dan üstün olarak farklı haritalara uyum yeteneği, yüksek başarı oranı ve daha kısa yol açısından daha çok verim elde etmiştir (Chen ve ark., 2011). 2013 yılında Wei tarafından SA algoritması ile Powell algoritmasını kullanarak geliştirilmiş Benzetimli Yapay Sinir Ağı(BYSA) algoritması oluşturmuşlardır. Mobil robotun

takip edeceği global yol planlanırken kritik öneme sahip olan enerji fonksiyonu olarak belirtilen yol uzunluğu, SA ile optimize edilerek en kısa yolu elde etmişlerdir. Haritada bulunan engellere çarpma olayı gerçekleştiği zaman SA algoritması ile optimize edilen uygunluk fonksiyonuna ceza fonksiyonundan elde edilen hata değerini eklemişlerdir. YSA ile ceza fonksiyonundan elde edilecek hata değerini belirlemişlerdir. Geliştirilmiş algoritma simülasyon ortamında test edildiğinde etkin bir yol planlama, hesaplama hızını arttırmakta ve yavaş yakınsama özelliklerinin üstesinden gelmektedir (Wei ve ark., 2013)

2017 yılında Pandey ve arkadaşlarının çalışmalarında bilinmeyen statik ve dinamik ortamda özerk mobil robotlar için navigasyon ve engellere çarpmayı önlemek için tek tip 1 bulanık mantık sistem (T1-SFSL) denetleyicisi ve Hibrit Bulanık-WDO(Rüzgara Dayalı Optimizasyon) hibrit bir algoritma gerçekleştirmişlerdir. WDO algoritmasını bulanık denetleyicinin giriş / çıkış üyelik fonksiyon parametrelerini optimize etmek ve ayarlamak için kullanmışlardır. WDO algoritması, N-boyutlu arama alanının üzerinde gezinen sonsuz küçük hava parsellerinin atmosferik hareketine dayalı olarak çalışmaktadır. Bu önerilen tekniğin performansını test etmek için, simülasyon ortamında elde edilen veriler ile Khepera-III mobil robot kullanarak gerçek zamanlı olarak gerçekleştirilen deneysel çalışmalarda verileri karşılaştırmışlardır. Simülasyon ve deneysel sonuçlar, Bulanık-WDO denetleyicinin T1-SFSL denetleyicisine kıyasla daha iyi performans sağladığını göstermektedir. Mobil robot navigasyonu için Bulanık-WDO algoritması T1-SFSL denetleyicisine göre daha etkin sonuçlar verdiğini belirtmişlerdir (Pandey ve ark., 2017). Bulanık mantık tabanlı yapılan diğer çalışmada ise optimizasyon yöntemi değiştirilmiştir. Belirtilen bu çalışma 2013 yılında Purian ve ark. tarafından dinamik çevrelerde mobil robot navigasyonunu gerçekleştirmek amacıyla ACO algoritmasının heuristik olma özelliği ve bulanık mantıkta bulunan kural parametreleri birleştirilerek oluşturulmuştur. Mobil robot için en kısa yol bulanık mantığın karar alma özelliğine göre başlangıç ve hedef noktaları arasındaki mesafe ve en yakın engellere olan açı farkından etkilenecek şekilde belirlenmiştir. ACO algoritması yol uzunluğu, zamanı ve pürüzsüzlüğü dahil en iyileme kriterlerine göre bulanık kural parametrelerini optimize etmiştir (Purian ve ark., 2013).

2012 yılında Chaari ve ark. zeki yol algoritması olarak adlandırdıkları ACO-GA hibrit algoritmasıyla global yol planlama çalışması gerçekleştirmişlerdir. Arama alanını azaltmak, arama süresini hızlandırmak ve çözüm kalitesini artırmak amacıyla ACO algoritmasını kullanmışlardır. Arama sürecini hızlandırmak için başlangıç yolları belirlenirken heuristik uzaklık bilgisinden yararlanılmıştır. ACO algoritmasında bulunan kontrol mekanizması ve GA'da bulunan mutasyon operatörü birlikte kullanılarak çözüm kalitesi arttırılmıştır. Son olarak yerel optimum noktasından sakınma işlemini GA'nın çaprazlama operatörüyle sağlamışlardır. Algoritmanın sağlamlığını ve kalitesini değerlendirmek için simülasyon ortamında test çalışmaları yapmışlardır. Simülasyon çalışmasında zeki yol algoritması, ACO, GA ve Bellman Ford algoritmaları en kısa yol parametresine göre karşılaştırılmıştır. Bu karşılaştırma sonucuna göre zeki yol algoritması arama yaklaşımı olarak daha iyi bir performans sağlamıştır. Çözüm kalitesinde ise ACO'ya göre %48.3, çalışma zamanında Bellman Ford en kısa yol metodundan %64.9 daha iyi olduğunu kanıtlamışlardır (Chaari ve ark., 2012). 2014 yılında aynı araştırmacılar tarafından simülasyon ortamında gerçekleştirilen zeki yol algoritması Wifibot Lab ortamında gerçek zamanlı olarak test edilmiştir. Deneysel çalışmaların sonucunda, ACO-GA hibrit algoritmasının optimal çözüm bulmada % 80'e kadar başarı sağlayabildiği kanıtlanmıştır. Gerçek zamanlı ortamda kısıtlamalara rağmen en uygun yolu bulmada hem ACO hem de GA algoritmalarının üstün özelliklerinden yararlanmışlardır (Chaari ve ark., 2014).

2016 yılında Bakdi ve ark. tarafından yapılan çalışmada mobil robotlar için çevrimdışı yol planlama gerçekleştirilmiştir. İlk adım olarak iki kinect kamera sistemi kullanarak görüntü işleme tekniği ile çevre modellenmesi sağlamışlardır. İkinci adım olarak GA, mobil robotun başlangıç ve hedef noktaları arasında engellere çarpmadan en uygun yolu bulmak amacıyla kullanılmıştır. Daha sonra üretilen yol Parçacıklı Kubik Hermit İnterpolasyon Polinomu kullanarak düzgünleştirilmiştir. Düzgünleştirilmiş yolun takip edilmesi için Lyapunov Global asimtotik yöntemi uyarlanabilir bir bulanık mantık denetleyicisinde kullanılmıştır. Bunlara ek olarak odometri ve kinect sensörlerden elde edilen veriler alınarak kalman filtrede birleştirilerek robotun mevcut konumu ve yönü tahmin edilmiştir. Önerilen yöntemin

doğrulaması için farklı yöne hareket eden robot "RobuTER" kullanmışlar ve kapalı ortamda engellerle çarpışmadan yol planlama başarıyla gerçekleştirilmiştir. Sonuç olarak yapılan çalışmada mobil robotun hızı artmakta, minimum yol uzunluğu ve maksimum güvenlik sağlanmakta, robotun yol yörüngesini takibi sırasında sensör ölçümleri ile hatalar izlenmekte ve en aza indirilmektedir (Bakdi ve ark., 2016).

2016 yılında Boğar ve ark. tarafından yapılan çalışmada robotun en kısa yol problemini çözmek amacıyla GA-Dijkstra hibrit algoritması önerilmiştir. GA'nın global arama özelliği ile Dijkstra algoritmasının yerel çözüm kabiliyeti birleştirilerek daha kısa yol için daha hızlı çözümler elde etmeyi amaçlamışlardır. Simülasyon ortamları 20x20 ve 50x50 engellerin bulunduğu bir ortamdan oluşmaktadır. Dijkstra, GA ve önerilen hibrit algoritmalar, iterasyon sayısı ve çalışma süresi kriterine göre karşılaştırılmıştır. Karşılaştırma sonuçlarına göre GA- Dijkstra hibrit algoritması 50x50 çalışma ortamında daha az iterasyon ve çalışma süresi ile diğer ortamda elde edilen çözümler ile aynı değerlere sahip olduğunu tespit etmişler. Bu sebepten dolayı GA- Dijkstra hibrit algoritması daha büyük çalışma alanları için uygun olduğu belirtilmiştir (Boğar ve ark., 2016). 2012 yılında Zhang, Boğar ve ark.'nın çalışmalarında kullandığı geleneksel yöntem olan Dijkstra algoritması yerine A\* algoritmasını kullanarak hızlı bir şekilde en kısa yolu aramak amacıyla global optimal yolu GA ile optimize etmiştir. Simülasyon sonuçlarından elde ettikleri verilere göre önerilen yöntemin arama hızı ve kalitesinin arttığını belirtmişlerdir (Zhang ve ark., 2012).

2012 yılında Liu ve ark. çalışma ortamı olarak seçilen dinamik bir çevrede GA ile Yapay Olası Alan (APF) algoritmalarını birleştirerek hibrit algoritmaya dönüştürmüşlerdir. Bu çalışmanın yeniliği, hareketli engellerin bulunduğu dinamik ortamda GA tabanlı global yol planlayıcı ile APF tabanlı yerel yol planlayıcının kombine edilmesidir. GA yakınsamayı verimli hale getirebilmek için evrimsel popülasyon ve genetik operatörleri kullanmıştır. Hibrit yaklaşımın işlem süresi açısından GA ile karşılaştırıldığında, daha iyi performans sağladığını göstermiştir. Hibrit yaklaşımın göstermiş olduğu performans sonucunda mobil robotun engellere çarpmasını önlemede hızlı yanıt alması çok önemlidir. Sonuç olarak hibrit yaklaşım

büyük ölçekli dinamik ortamlarda daha iyi sonuç verdiği açıkça belirtilmiştir (Liu ve ark., 2012).

2015 yılında Parhi ve ark. tarafından mobil robotun navigasyon problemini çözmek için guguk kuşu arama algoritması(CS) ve uyarlamalı nöron bulanık çıkarım sistemini(ANFIS) birleştirerek yeni bir yaklaşım geliştirilmiştir. Uyarlamalı nöron bulanık çıkarım sistemi (ANFIS), paralel hesaplama, sinir ağı, uyum kabiliyeti ve insana benzer bilgi gösterimini birleştiren bulanık sistem yaklaşımına dayalı bir sinir ağıdır. ANFIS'de giriş ve çıkış parametrelerinin eğitimi ve güncellenmesi sistem tasarlanma sorunlarından biridir. ANFIS için eğitim algoritmalarının çoğu eğitim açısına dayalıdır. Eğitim açısına bağlı olarak eğiten algoritmalar ANFIS'i geliştirir. Her basamaktaki eğitimlerin hesaplanması çok karmaşıktır ve kullanılan zincirleme kural modeli yerel minimuma da neden olabilir. Bu sebepten dolayı kullanılan model, ağın doğruluğunu etkilemektedir. Dezavantajları ortadan kaldırmak amacıyla meta-sezgisel yöntemlerden olan CS algoritmasını seçmişlerdir. CS algoritması ve LSE(en küçük kareleme) yöntemi, ANFIS'deki giriş ve çıkış parametrelerini optimize etmek ve mobil robotun dönüş açısını hesaplamak amacıyla kullanmışlardır. Petri-Net ağ modelini kullanarak mobil robotlar arasındaki çarpışma probleminden kaçınmak için bir dizi çarpışma önleme kuralları öngörülmüş ve etkili bir şekilde mobil robot kontrolörüne yerleştirilmiştir. Son olarak, CS algoritmasının yinelemesi yoluyla sistem çıktısı ile gerçek eğitim verileri arasındaki hata, asgari bir değere ulaştırmışlardır. ANFIS modelinin amaç fonksiyonu CS algoritma tabanlı kök ortalama karesi hatası (RMSE) olarak tanımlamışlardır. Hem simülasyon hem de deneysel ortamlarda yol planlama algoritmasının etkinliği, fizibilitesi ve sağlamlığını test etmişlerdir. Geliştirilen algoritma, diğer modellerle karşılaştırıldığında daha iyi sonuç vermiştir (Parhi ve ark., 2015). Diğer bir çalışma da ise bulanık mantık sistemi biyolojiden ilham alan iki katmanlı sinir ağıyla birleştirilerek, bilinmeyen ortamlarda verimli bir mobil robot navigasyon yöntemi geliştirmişlerdir. Navigasyonda robot modeli olarak Braitenberg'in araç modelini kullanmışlardır. Sinir sistemindeki nöron modelinde sensör girişlerini kullanan robot, çalışma ortamında hareket etmek amacıyla çevredeki ortamı tespit etme, hedefi izleme, engelleri tespit etme ve çarpışmama gibi olayları gerçekleştirir. Bulanık mantık, farklı katlardaki nöronlar





Tablo 1.2. (Devamı)

Algoritma	Kin. Model	Engel Şekli	Statik Engel	Dinamik Engel	Statik Hedef	Dinamik Hedef	Sim	Gerçek Sistem	Yıl
GA - A*	H	Kare	E	H	E	H	E	H	2012
ACO - GA	H	Dörtgen, Daire, Çokgen	E	H	E	H	E	H	2012
ACO - GA	H	Yok	H	H	E	H	E	E	2014
FUZZY - WDO	H	Kare Daire	E	E	E	H	E	E	2017
PSO - DV	H	Çokgen	E	E	E	H	E	H	2016
PSO - GA	H	Daire	E	H	E	H	E	H	2014
ACO - PSO	H	Belirsiz	E	H	E	H	E	H	2008
ACO - APF	H	Çokgen	E	H	E	H	E	H	2009
ACO - PSO	H	Şekilsiz	E	H	E	H	E	H	2008

#### 1.4. Tezin Amacı, Kapsamı ve Hedefler

Tez çalışmasının temel amacı, meta-sezgisel algoritma tabanlı mobil robot yol planlaması için özgün bir hibrit algoritma geliştirmektir. Geliştirilen özgün hibrit algoritmalar kullanılarak, engeller bulunan statik bir ortamda tekli ve çoklu mobil robotlar için etkili yol planlaması tasarlamak ve uygulamaktır. Bu çalışma iki ana başlık altında toplanmaktadır. İlk olarak meta-sezgisel tabanlı yol planlama algoritmalarının incelenerek daha etkili ve özgün bir hibrit algoritmanın geliştirilmesi için zemin hazırlanmasıdır. Kullanılan algoritmaların amacı meta-sezgisel algoritmaların verimliliğini ve yol planlamada yolun minimize edilerek belirlenen zaman, maliyet gibi en önemli performans kriterlerini geliştirmektir. Çalışmanın ikinci amacı olarak geliştirilen algoritmaları doğrulamak üzere fiziksel ortamda çeşitli uygulamalar gerçekleştirmek ve elde edilen sonuçların karşılaştırmasını yapmaktır.

Yukarıda bahsedilen amaçlara ulaşabilmek tespit edilen hedefler ve izlenen yol aşağıda verilmiştir:

- Üzerinde engeller bulunan statik bir ortamda tekli ve çoklu mobil robotların yol planlaması için özgün bir hibrit algoritma geliştirildi ve uygulandı. PSO(Parçacık Sürü Optimizasyonu), FA (Ateş Böceği Algoritması), CS (Guguk Kuşu Algoritması) gibi meta-sezgisel algoritmaların her biri ayrı ayrı kullanılarak elde edilen sonuçlar benzer çalışmalarla karşılaştırıldı. Daha sonra aynı ortam şartları ve robotlar için yol planlamaları yapmak üzere çalışmaya özgü CS-PSO(Guguk Kuşu Algoritması-Parçacık Sürü Optimizasyonu), CS-FA(Guguk Kuşu Algoritması-Ateş Böceği Algoritması), CS-PSO-FA (Guguk Kuşu Algoritması-Parçacık Sürü Optimizasyonu- Ateş Böceği Algoritması) hibrit algoritmalar geliştirildi. Doğadan ilham alan algoritmalar kullanıldığı için bu algoritmaların yapısında PSO’da parçacık, FA’da ateşböcekleri, CS’de konak yuva parametreleri başlangıç ile hedef nokta arasında oluşabilecek olası yolları tanımlamaktadır.
- Kobuki mobil robotlar ile geliştirilen algoritmaların çalıştığı server bilgisayar arasında iletişim Wifi haberleşme standardı aracılığıyla sağlanmaktadır. Algoritmalar tarafından üretilen global yol, gerçek uygulamada çalışan yerel yol planlama mobil robotun takip etmesini sağlamaktadır. Robot ile haberleşmede MATLAB tabanlı ROS programı kullanılmıştır. Görüntü almak amacıyla, laboratuvar tavanına bir kamera yerleştirilerek lokalizasyon gerçekleştirildi, çalışma ortamındaki robotların ve engellerin konumları belirlendi. Böylece, lokalizasyon ve yol bulma işlemleri bütünlük bir sistem olarak gerçek laboratuvar ortamında uygulanmıştır.
- Kameradan alınan görüntüyü işleyerek harita oluşturma, algoritmalar ile yol bulma, robot ile haberleşme ve robotun algoritmalar ile yol bilgisine göre ilerleyebilmesini gibi tüm faaliyetleri kolaylıkla gerçekleştirebilmek amacıyla MATLAB-GUI (Graphical User Interface) tabanlı bir ara yüz tasarlanmıştır.

- Geliştirilen yol planlama algoritmalarının etkinliği, uygulanabilirliği ve sağlamlığı hem simülasyon ortamında hem de gerçek dünya ortamında test edildi ve elde edilen sonuçlar karşılaştırıldı.
- Önerilen hibrit algoritmalar, hibrit algoritmaları oluşturan bileşen algoritmalar ve diğer meta-sezgisel yöntemlerden elde edilen sonuçlar simülasyon ortamında karşılaştırıldı.

## **BÖLÜM 2. MOBİL ROBOTLARIN YOL PLANLAMADA AMAÇ FONKSİYONUNUN BELİRLENMESİ**

### **2.1. Giriş**

Optimizasyon, mühendislik, enerji, ekonomi, tıp ve bilgisayar bilimi gibi birçok alanda gerçekleştirilmektedir. Optimizasyon, en iyileme anlamına gelmektedir. Belirli sınırları sağlayacak şekilde tanımlanan problemde bilinmeyen parametre değerlerinin hesaplanma işleminin gerçekleşmesi optimizasyon problemi olarak adlandırılmaktadır (Murty, 2003).

Kısıtlayıcı fonksiyonlar problem için uygun sınırlar belirlemekte ve aranan optimum amaç fonksiyon değeri çizilen bu sınırlar içerisinde yer almaktadır. Sınırlayıcılar bazı durumlarda eşitsizlik denklemleri ile tanımlanırken bazı durumlarda ise eşitlikler biçiminde ifade edilirler. Optimizasyon problemlerinde tasarımdan beklentiler ne kadar fazla ise problem için çözüm o derece zor olmaktadır (Karaboğa, 2011).

Optimizasyon problemleri için birçok algoritma önerilmiştir. Doğadan ilham alarak geliştirilen optimizasyon algoritmaları uygunluk fonksiyonunu en iyi seçeneklerle belirlemektedir. Optimizasyon algoritmaları heuristik veya meta-sezgisel bir yaklaşım ile tasarlanabilir. Heuristik tabanlı algoritma yaklaşımları problem tabanlı yaklaşımlarıdır. Meta-sezgisel tabanlı algoritmalar ise operatörlerini ve parametrelerini ayarlayarak çeşitli optimizasyon problemleri için uyarlanabilen genel yapıya sahiptir.

Akıllı otonom robot kontrolörleri verimli bir şekilde geliştirilebilmesi uygunluk fonksiyonunun formüle edilmesiyle gerçekleştirilmektedir. Amaç fonksiyonu evrimsel hesaplamaların merkezinde bulunmaktadır. Uygunluk fonksiyonu, optimizasyon probleminin çözümünde popülasyon içerisindeki çözümlerden en iyi çözümün belirlenmesinden sorumludur. Karmaşık görevleri yerine getirebilen otonom robot kontrolörlerinin geliştirilebilmesinde, uygunluk fonksiyonu

genellikle ulaşılabilir denetleyici kalitesinde sınırlayıcı faktör olarak belirlenmiştir. Bu sınır genellikle bir durma noktasına varana kadar sonraki nesillerin uygunluk fonksiyonlarının değerlendirilmesinde ortaya çıkmaktadır.

Genel olarak optimizasyon problemleri tanımlanırken belirli bir amaç ve kısıtlardan oluşmaktadır. Bu çalışmada aşağıdaki eşitlik kullanılarak (Denklem 2.1) Hatanın Mutlak İntegrali (IAE) amaç fonksiyonu olarak belirlenmiştir. Amaç fonksiyonunda kullanılan uygunluk fonksiyonları ise meta-sezgisel tabanlıdır. Amaç fonksiyonu ile problem minimize edilmeye çalışılmaktadır.

$$IAE = \int |e(i)| dt = \int |y(i) - x(i)| dt \quad (2.1)$$

Yol planlamada amaç fonksiyonu bir ölçüm grubu için IAE toplamı aşağıdaki eşitlik kullanılarak (Denklem 2.2) gibi tanımlanır:

$$F = \sum_{i=1}^N |f_i(x, y)| \quad (2.2)$$

## 2.2. Mobil Robotların Yol Planlamada Uygunluk Fonksiyonunun Oluşturulması

Yol planlama problemlerinde amaç, engellere çarpmadan başlangıç ile hedef arasında en uygun yolu bulmaktır. Uygunluk ölçüsü olarak en kısa ve optimum yol belirlenmiştir. Bu tez çalışmasında da, uygunluk fonksiyonu en kısa yol olarak belirlenmiştir. (Denklem 2.3) ve Tablo 2.1.'de bulunan Manhattan, Euclidean ve Diagonal heuristik fonksiyonlar çalışmada kullanılan amaç fonksiyonunu göstermektedir.

$$f = \begin{cases} \sum_{i=1}^{n-1} d(p_i, p_{i+1}) & \text{engel yoksa} \\ \sum_{i=1}^{n-1} d(p_i, p_{i+1}) + hata & \text{engel varsa} \end{cases} \quad (2.3)$$

$$hata = \sum_j^n \sum_i^n engel_{ji} \cdot yol_{ji} \quad (2.4)$$

Tablo 2.1. Heuristik Uzaklık Fonksiyonu

Heuristik Fonksiyon	Denklem
Manhattan	$d(p_i, p_{i+1}) =  x_s - x_g  +  y_s - y_g $
Euclidean	$d(p_i, p_{i+1}) = \sqrt{(x_s - x_g)^2 + (y_s - y_g)^2}$
Diagonal	$d(p_i, p_{i+1}) =  x_s - x_g  +  y_s - y_g  + (\sqrt{2} - 2) * \min\{ x_s - x_g ,  y_s - y_g \}$

Tablo 2.1.'de Manhattan uzaklık, Euclidean uzaklık ve Diagonal uzaklık fonksiyonlarının açıklamaları aşağıda verilmiştir.

– Manhattan uzaklık fonksiyonu

Manhattan uzaklık fonksiyonu kullanarak robotun gideceği yol planlanması yapılırken mevcut noktadan hedef noktaya ulaşmak için dikey veya yataydaki hareketin toplam kare sayısının hesaplanmasıyla bulunur (Denklem 2.5). Bu uzaklık fonksiyonunun dezavantajı hedefe ulaşmada kullanılan kare sayısının artması yolun gerçek maliyetini arttırmasıdır. Bu sebeple Manhattan uzaklık fonksiyonu yol planlamaları için uygun çözüm olmayabilir.

$$h_1(n) = |x_s - x_g| + |y_s - y_g| \quad (2.5)$$

– Euclidean heuristik fonksiyonu:

Euclidean uzaklık fonksiyonunu kullanarak robotun gideceği yol planlanması, olası gidilecek nokta ve hedef nokta arasında düz bir şekilde hareket ettirilmesi ile yapılır (Denklem 2.6). Genellikle hesaplama işlemi yapılırken gerçek maliyet büyük ölçüde görmezden gelinir. Bundan dolayı gereğinden fazla nokta ziyaret edilerek geçen zamanı arttırmaktadır.

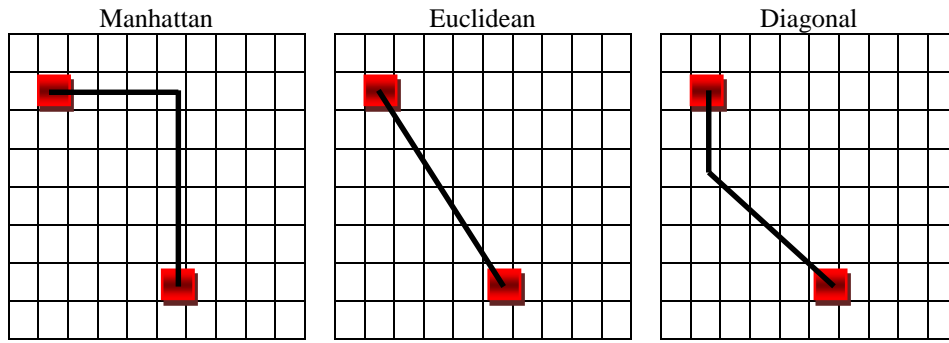
$$h_2(n) = \sqrt{(x_s - x_g)^2 + (y_s - y_g)^2} \quad (2.6)$$

– Diagonal heuristik fonksiyonu:

Diagonal uzaklık fonksiyonu, Manhattan ve Euclidean heuristik uzaklık fonksiyonların birleştirilmesiyle elde edilir. Robotun gideceği yol (Denklem 2.7) kullanılarak hesaplanır.

$$h_3(n) = h_1(n) + (\sqrt{2} - 2) * \min\{|x_s - x_g|, |y_s - y_g|\} \quad (2.7)$$

Manhattan, Euclidean ve Diagonal heuristik uzaklık fonksiyonlarının grafiksel gösterimi Şekil 2.1.'de gösterilmektedir.

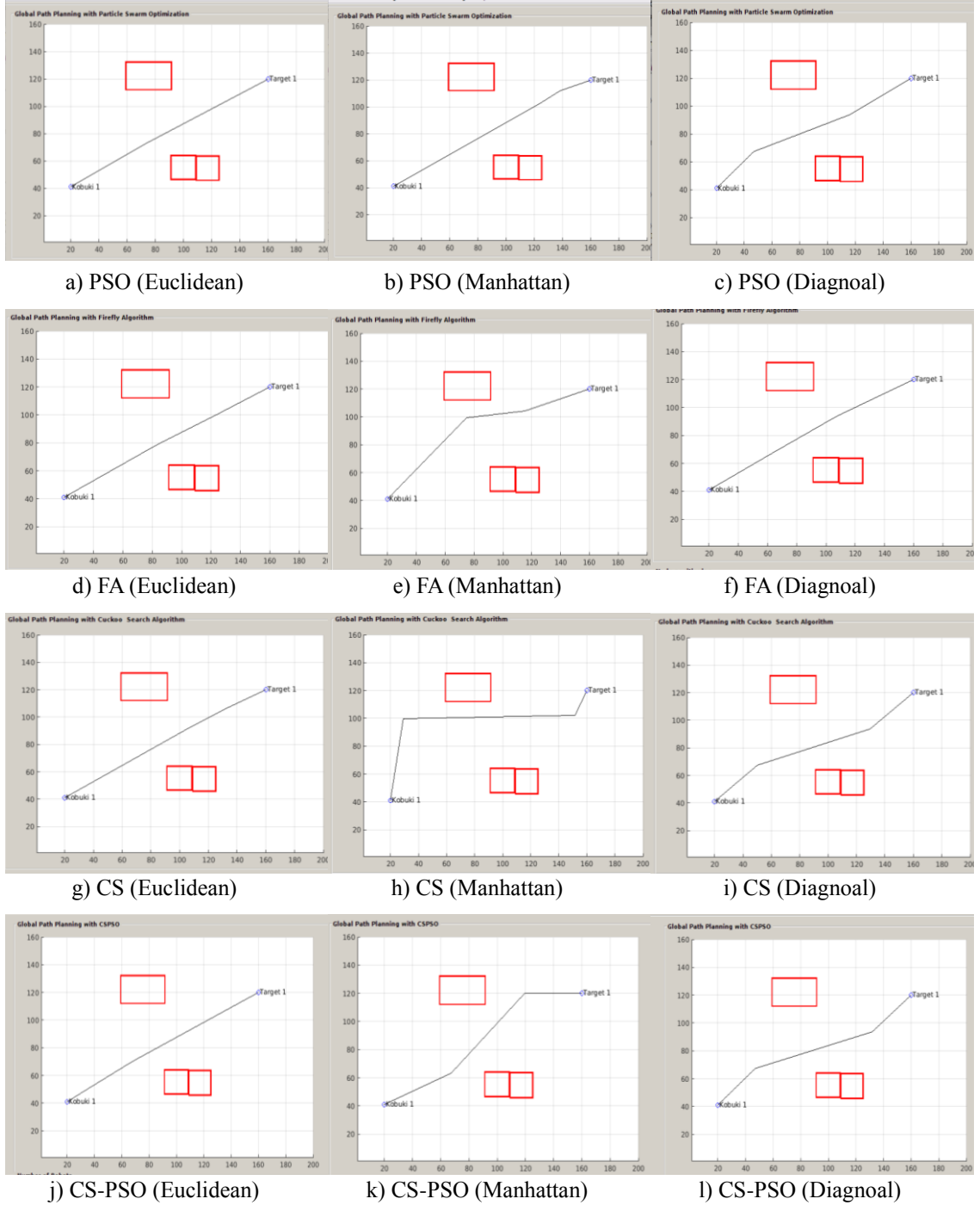


Şekil 2.1. Heuristik Fonksiyonlar

Burada  $f$  uygunluk fonksiyonu,  $p$  algoritmanın türüne göre değişen birey ve  $n$  ise popülasyon uzunluğudur. Amaç fonksiyonu en kısa yolu planlamak için seçilen bireyler arasındaki uzaklıkların toplamı olarak verilmektedir. Elde edilen değer ise yolun uzunluğu olarak tanımlanmaktadır. İki düğüm arasında bir engel varsa, uygunluk fonksiyona bir hata değeri eklenir. Hata değerini hesaplamak için (Denklem 2.4) verilen engelin bulunduğu alan ve rota scalar matris olarak çarpıldıktan sonra toplanmaktadır. Eğer yolda engel yoksa bu değer 0, engel varsa bu değer sıfırdan farklı olarak hesaplanan bir hata değeridir.

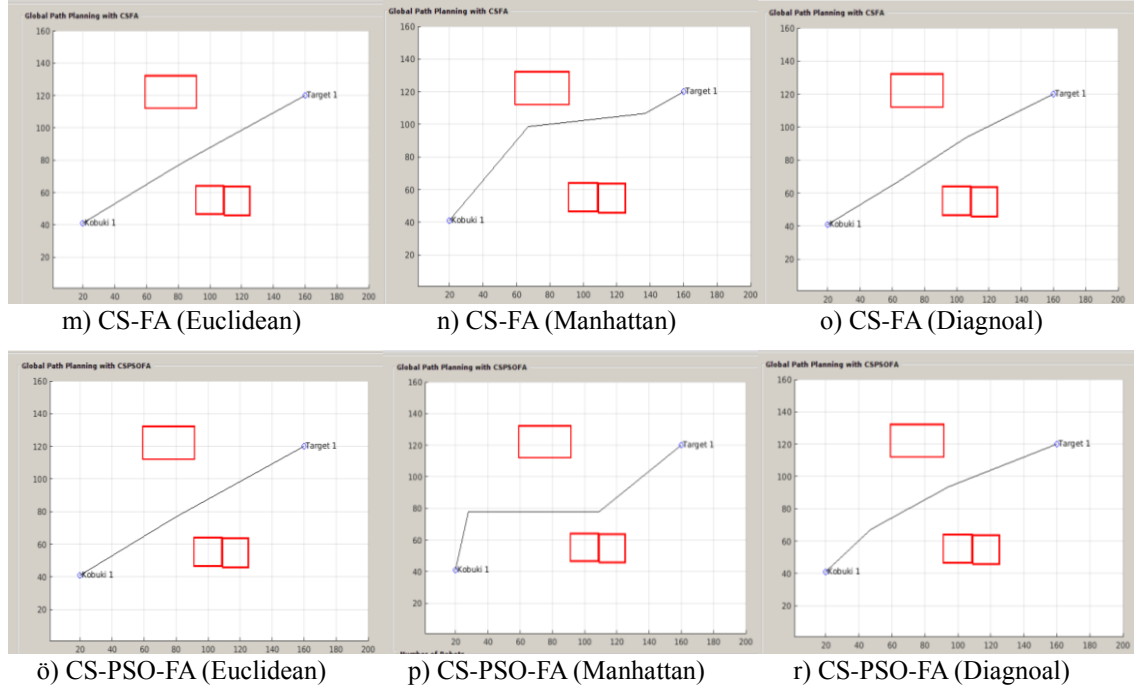
Bu tez çalışmasında, global yol planlamada kullanılacak olan heuristik uzaklık fonksiyonları aynı çalışma ortamında algoritmaların ortak parametre değerlerini aynı yaparak simülasyon ortamında analizler gerçekleştirilmiştir. Heuristik uzaklık fonksiyonu olarak Manhattan, Euclidean, Diagonal belirlenmiştir. Simülasyon ortamında geliştirilen algoritmalar farklı heuristik fonksiyonları ile 10 kez

çalıştırılarak ortalama yol uzunlukları hesaplanmış ve Tablo 2.2.'de sunulmuştur (Şekil 2.2.).



Şekil 2.2(a-r). Global yol planlamada heuristik fonksiyonların etkisi





Şekil 2.2(a-r). (Devamı)

Tablo 2.2. Sezgisel uzaklık fonksiyonların algoritmalar üzerinde etkisi

Algoritma	Heuristik Uzaklık Fonksiyonu		
	Euclidean	Manhattan	Diagonal
PSO	160.8147	184.1704	162.3357
FA	160.7900	166.8821	162.2319
CS	160.7949	181.0239	162.6487
CS-PSO	160.7930	173.7546	162.5824
CS-FA	160.8018	180.1783	166.5334
CS-PSO-FA	160.7918	176.8130	164.9624
Ortalama	160.7980	177.1370	163.5491

Tablo 2.2.'de geliştirilen algoritmalarla göre engellere çarpmaksızın en kısa yolu bulmak için kullanılan uzaklık fonksiyonları karşılaştırılmalı olarak gösterilmektedir. Burada en uygun değer Euclidean uzaklık fonksiyonuyla elde edildiği görülmektedir.

## **BÖLÜM 3. ÇOKLU MOBİL ROBOTLAR İÇİN PARÇACIK SÜRÜ OPTİMİZASYON ALGORİTMA TABANLI GLOBAL YOL PLANLAMA**

### **3.1. Giriş**

Son on yılda gezegen araştırması, gözetim ve mayın tespiti gibi uygulama alanlarında mobil robot yol planlama giderek artan bir araştırma alanı olmaya başlamıştır. Buna ek olarak yol planlamada insansız sistemlerin öneminden dolayı yoğun bir şekilde incelenmektedir. Yol planlamanın amacı yol uzunluğu ve yolun düzeltilmesi gibi önceden tanımlanan performans kriterlerini optimize ederek başlangıç konumundan hedef konuma doğru en uygun ve engelsiz yolu belirlemektir. Bununla birlikte, deterministik olmayan polinom zamanlı zor (NP-zor) problemlerin doğası nedeniyle, yol planlamanın etkin bir şekilde çözülmesi zor bir süreçtir.

Parçacık Sürüsü Optimizasyon (PSO) algoritması, sürekli doğrusal olmayan optimizasyon, kısıtlı ve kısıtlı olmayan, türevlenemez çok modlu fonksiyonlar için geliştirilmiş bir evrimsel hesaplama tekniğidir. PSO, basitlik ve yüksek yakınsama hızından dolayı, yol planlama problemini çözmek amacıyla yaygın ve başarılı bir şekilde uygulanmaktadır.

PSO kümelenme ve sınıflandırma, güvenlik ve askeri, modelleme, biyomedikal, tümleşik optimizasyon, iletişim ağları, kontrol, sensor ağları, sinyal işleme ve robotik gibi farklı alanlarda kullanılmaktadır (Poli, 2008). PSO uygulama alanlarına ek olarak biyolojik ve tıbbi alanlar, elektrik, elektronik ve elektromanyetik alanlar, bilgisayar istihbarat uygulamaları, kombinatoryal problem çözme, görüntü-sinyal işleme ve robotiklere kadar geniş bir uygulama alanı vardır (Chatterjee ve ark., 2005).

Literatürde PSO algoritması tabanlı robotik alanı ile ilgili çok sayıda makale bulunmaktadır. Bu makaleler, robotik manipulatörler ve kolların kontrolü, hareket planlama ve kontrol, robotun çalışması, toplu robotik arama, denetimsiz robotik öğrenme, yol planlama, engelden kaçınma, sürü robotu, insansız araç navigasyonu, futbol oynama, robot görme, nakliye robotları, koku kaynağı lokalizasyonu, çevre haritalama, robotların ses kontrolü gibi konuları içermektedir.

Bu çalışmada mobil robotların yol planlaması gerçekleştirileceği için literatürde mobil robot yol planlama problemini çözmek için PSO algoritmasının kullanıldığı çalışmalar incelenmiştir. 2012 yılında Deepak ve arkadaşları çalışmalarında mobil robot navigasyon problemlerini çözmek amacıyla PSO tabanlı bir sistem geliştirmişlerdir. Geliştirilen algoritma sayesinde bilinmeyen ortamlarda engellerden kaçınma ve en uygun yolları bulma ihtimali artırmıştır. Algoritma tarafından üretilen yörüngeler, her yinelemede global en iyi konuma(gbest) dayanmaktadır. Sürü arasındaki minimum uygunluğa sahip olan parçacık, en iyi global konum olarak düşünülmektedir. Bundan dolayı robot en iyi konuma doğru hareket eder ve hedef konuma varana kadar bu süreç iterasyonda devam etmektedir. Modellenen uygunluk fonksiyonunun kontrol parametrelerini ayarlamak için çok sayıda deney yapılmıştır. Yaptıkları bu çalışmada simülasyon sonuçları ile geliştirilen algoritmanın etkinliğini doğrulamışlardır (Deepak ve ark., 2012). Zhang ve arkadaşları tarafından yapılan çalışmada belirsiz konumlarda bulunan tehlikeli kaynakların olduğu yerlerde robot yol planlama problemini çözmek için çok amaçlı PSO geliştirilmiştir. Uygun yol belirlemede performans kriteri olarak yol uzunluğu ve risk derece parametrelerini seçmişlerdir. Risk derecelerini değerlendirmek amacıyla bulanık üyelik fonksiyonu kullanılmıştır. Bu algortmada yolların uygunluğunu geliştirmek için rastgele örnekleme ve tek biçimli mutasyon işlemi uygulanarak yeni bir güncelleme yöntemi önerilmiştir. Son olarak, önerilen algoritmanın verimliliğini ve uygulanabilirliğini doğrulamak için dört test problemi seçilerek simülasyon gerçekleştirilmiştir. Belirsiz tehlike kaynaklarının bulunduğu bir ortamda önerilen algoritmanın, yol planlama problemini çözmek için uygulanabilir bir alternatif olduğu belirtilmiştir (Zhang ve ark., 2013). Tang ve arkadaşları tarafından yapılan çalışmada “Etkin bölge” adı verilen daraltılmış bir alanda mobil robotların global yol planlaması amacıyla özelleştirilmiş bir PSO algoritması önerilmiştir. Parametrelerin etkisini (nüfus

boyutu, parçacık boyutu ve maksimum iterasyon sayısı) incelemek için simülasyonda hem basit hem de karmaşık çalışma ortamlarında yol planlama algoritması çalıştırmışlardır. Simülasyon analizlerine göre algoritma daha hızlı arama yaparak arama kalitesini artırmıştır. Dolayısıyla algoritmanın etkinliği artmıştır (Tang ve ark., 2010). Wang ve arkadaşları tarafından yapılan çalışmada futbol robotunun engellerin belirli bir hızda olduğu varsayılan bir dinamik ortamda global yol yaklaşımı çözmek için PSO'ya dayalı bir yol algoritması tasarlanmıştır. Önerilen algoritma basit, etkili kodlama şeması ve hatasız uygunluk fonksiyonu ile ön plana çıkmaktadır. Bu özellikler kullanılarak yerel optimal çözümden kaçınarak global optimal çözüm kabiliyetine dayanan az maliyetli düz bir yol oluşturmuşlardır. Futbol robotu için bu yaklaşımın uygunluğu simülasyon çalışmaları ile gösterilmiştir (Wang ve ark., 2006). Saska ve arkadaşları tarafından yapılan çalışmada futbol robotları için kübik eğriler dizisi kullanılarak yol planlamada özgün bir yaklaşım sunulmuştur. Bu eğrileri optimize etmek için PSO kullanılmıştır. Elde edilen düzgün yol robot tarafından kolaylıkla takip edilmiştir. Geliştirilen yöntem diğer optimizasyon yöntemleri (Görünürlük Grafiği ve Potansiyel Alanı) ile karşılaştırıldığında, PSO tekniği yerel optimum noktasına takılmamıştır (Saska ve ark., 2006). Nasrollahy ve arkadaşlarının çalışmasında dinamik ortamda engellere çarpmaksızın başlangıç ve hedef noktalar arasındaki yol planlaması PSO kullanılarak gerçekleştirilmiştir. PSO'nun kullanım amacı yerel optimum noktalarından kaçınmak toplam süreyi en aza indirmektir. Önerilen algoritmanın doğruluğunu kanıtlamak için simülasyon ve deneysel çalışmalar yapılmıştır (Nasrollahy ve ark., 2009). Solea ve ark. tarafından yapılan çalışmada mobil robotun ortamda bulunan sabit ve hareketli engellere çarpmaksızın uygun yolu bulmak için PSO algoritması kullanılmıştır. Mobil robot lazer tarayıcı modülü ile sabit ve hareketli çevreyle ilgili doğru bilgiye ulaşmıştır. Önerilen PSO algoritması kullanılarak hem global hem de yerel yol planlaması gerçekleştirilmiştir. Hesaplama verimliliği, simülasyon ortamında ve gerçek deney ortamında yapılan çalışmalarla kanıtlanmıştır. Simülasyon ortamında üretilen global yol gerçek deney ortamında yerel yol planlamayla örtüştüğünden dolayı doğruluğu kanıtlanmıştır (Solea ve ark., 2016).

Tablo 3.1.'de mobil robot yol planlamasında PSO algoritması ile ilgili verilen literatür çalışmalarının özet hali verilmiştir.

Tablo 3.1. Mobil robot yol planlamada PSO algoritmasının uygulanması

Yazarlar	Kin. Model	Engel Şekli	Statik Engel	Dinamik Engel	Statik Hedef	Din. Hedef	Sim	Sis.
B. Deepak, D. R. Parhi.	H	Kare	E	H	E	H	E	H
Y. Zhang, D.Gong, J.Zhang	H	Çokgen	E	H	E	H	E	H
Y. Tang, Q. Li, L.Wang, C.Zhang, Y.Yin	H	Çokgen	E	H	E	H	E	H
L. Wang, Y. L. H.Deng	H	Kare	H	E	H	E	E	H
M. Saska, M. Macas, L.Preucil, L. Lhotska	H	Kare	E	E	H	H	E	H
A. Z. Nasrollahy, H. H. S.Javadi	H	Daire	E	E	H	E	E	H
R. Solea, D. Cernega	H	Kare, Daire	E	H	E	H	E	E

Bu çalışmada statik bir çevrede çoklu mobil robotların navigasyon problemini çözmek için PSO algoritma tabanlı bir yaklaşım geliştirilmiştir. Bu yaklaşımın amacı PSO'nun verimliliğini yol planlamada kullanarak belirlenen zaman, maliyet gibi en önemli performans kriterlerini geliştirmektir.

### 3.2. PSO Algoritması için Matematiksel Modelleme

Parçacık Sürü Optimizasyonu kuş, balık ve havyan sürülerinin çevrelerine adapte olma, zengin yiyecek kaynağı bulabilme ve avcılardan kaçabilme yeteneklerinden

esinlenilerek Kennedy ve Eberhart tarafından geliştirilmiş bir algoritmadır (Kennedy ve Eberhart, 1995).

PSO kodundan da anlaşıldığı gibi, algoritmada ilk olarak kullanılacak parametreler belirlenir. İkinci olarak popülasyon değerleri rastgele belirlendikten sonra bu değerlere karşılık gelen amaç fonksiyon değeri hesaplanır. Başlangıç çözümündeki her bireye ait çözüm aynı zamanda o parçacığın yerel en iyi değeridir. Amaç fonksiyon değeri, en iyi olan parçacık global en iyi olarak atanır ve bu parçacık değerleri global komşular olarak saklanır. Aynı şekilde başlangıçtaki her bir parçacığın amaç fonksiyon değerleri yerel en iyiler olarak saklanıp parçacık değerleri de yerel komşular olarak saklanır. Genellikle başlangıç çözümünde başlangıç hızları rastgele belirlenmektedir. Başlangıç çözümünden sonra PSO algoritması tam anlamıyla başlar. Başlangıç çözümünde elde edilen hız değerlerinden yararlanarak yeni pozisyonlar belirlenir. Yeni bulunan pozisyonlar için amaç fonksiyonu değerleri hesaplanır ve hızlar güncellenir. Daha sonra her parçacık için o iterasyona kadar bulunan en iyi amaç fonksiyonu değerini veren çözüm yerel en iyi olarak atanır. Tüm parçacıklar için yerel en iyiler bulunarak yerel en iyi komşular kümesi oluşturulur. Her iterasyonda bulunan global en iyiler karşılaştırılarak en iyi amaç fonksiyonuna sahip parçacığın çözümü global en iyi olarak atanır. Bu işlem bir durdurma kriterine kadar devam eder. Genel PSO algoritmasının sahte (pseudo) kodu Şekil 3.1.'de gösterilmektedir.

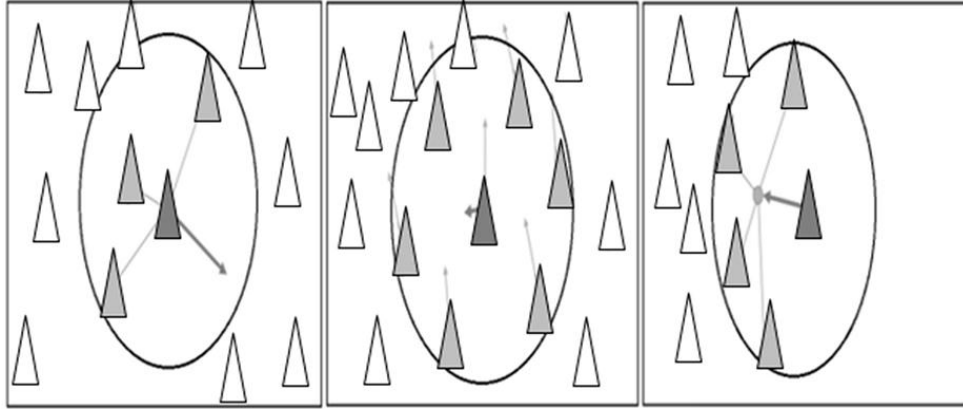
```

Begin
    Başlangıç parametrelerini ata
    Başlangıç popülasyonunu oluştur
    Her bir parçacık için
        Uygunluk değerini hesapla
    pBest'i bul
    gBest'i bul
Do
    Her bir parçacık için
        (1) denklemine göre parçacık hızını hesapla
        (2) denklemine göre parçacık pozisyonunu hesapla
        Uygunluk değerini hesapla
    pBest'i güncelle
    gBest'i güncelle
    v değerini güncelle
While Durdurma Kriteri
End

```

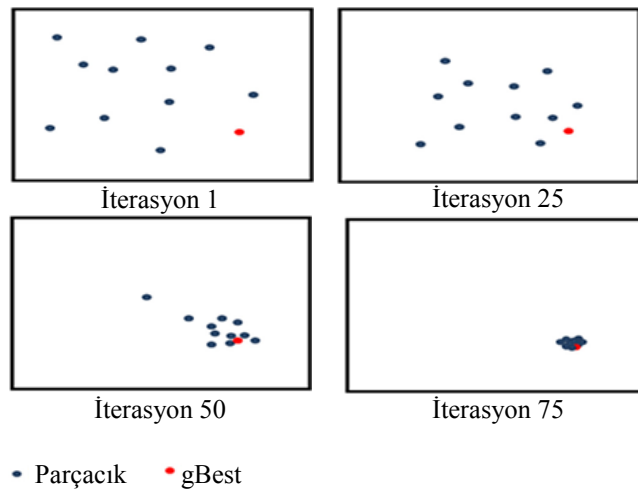
Şekil 3.1 PSO algoritması sözde kodu

Şekil 3.2.'de gösterildiği gibi PSO'yu sırasıyla ayırma, hizalama ve birleştirme gibi üç basit davranışla açıklamıştır. Ayırma, kalabalık yerel sürü arkadaşlarından kaçınma davranışdır. Hizalama, yerel sürülerin ortalama yönüne doğru hareket etme davranışdır. Birleştirme yerel sürünün ortalama konumuna doğru hareket etme davranışdır (Del Valle ve ark., 2008).



Şekil 3.2. PSO temel davranışları (Del Valle ve ark., 2008)

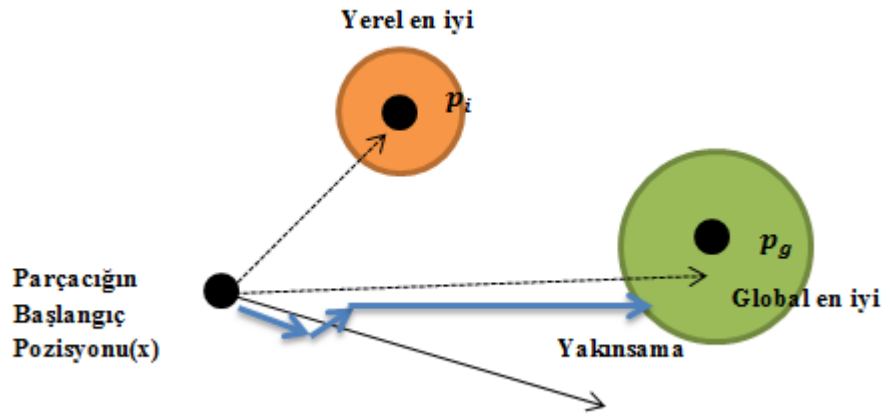
Şekil 3.3.'te, iterasyon sayısına göre PSO algoritmasındaki parçacıkların hareketini göstermektedir. İlk iterasyonda, en iyi çözümü bulmak için tüm parçacıklar yayılır. Her parçacık değerlendirilir. En iyi çözümler komşu topolojide bulunması beklenir. Daha sonra sürünün her bir üyesi için global en iyi parçacık güncellenir. Tüm parçacıklar en iyi çözümü kullanarak parçacığa doğru hareket ederek yakınsamayı gerçekleştirmektedir.



Şekil 3.3. İterasyon sayısına göre PSO algoritmasının hareketi

### 3.2.1. PSO algoritmasının temel parametreleri

Popülasyon temelli olan bu algoritma, sürü içerisinde bilgi paylaşımı ile durumlarını güncellemektedir. Bu sayede neredeyse bütün arama uzayı araştırılarak en iyi çözüme yakın bir çözüm bulunabilir. Sürü içerisindeki her bir bireyin kendi hafızası bulunmaktadır ve ayrıca en iyi konum değerini de bilmektedirler.



Şekil 3.4. Global en iyi yakınsamanın ana yapısı

Sürü içerisindeki her birey, parçacık olarak adlandırılmakta ve her parçacığın sürü içerisinde konum( $p$ ) ve hız( $v$ ) bilgileri bulunmaktadır. Bu konum ve hız bilgileri, parçacığın iterasyon içindeki en iyi pozisyonuna( $p_{best}$ ), sürü içerisindeki en iyi pozisyonuna( $g_{best}$ ) ve o andaki hız bilgisine göre her iterasyon da güncellenmektedir. İterasyonda pozisyon ve hız bilgisi aşağıdaki denklemlerdeki (Denklem 3.1-3.2) gibi güncellenir (Gao ve ark., 2008).

$$v_{t+1} = vw_t + c_1r_1(p_{best} - p_t) + c_2r_2(g_{best} - p_t) \quad (3.1)$$

$$p_{t+1} = p_t + v_{t+1} \quad (3.2)$$

(Denklem 3.1-2)'de bulunan terimleri detaylı açıklamaları aşağıda verilmiştir.



- Hız; parçacığın bir sonraki konumunu ifade etmektedir. PSO algoritması ile gerçekleştirilen çözümde etkili bir değişkendir. Hız  $v_{t+1}$  ile gösterilmektedir.
- Atalet Ağırlığı (Inertia Weight); bir önceki hız vektörünün mevcut hız vektörü üzerindeki etkisini kontrol eden parametreye atalet ağırlığı denir ve  $w$  şeklinde gösterilir. Atalet ağırlığının parçacıkta global ve yerel arama özellikleri arasında değiştirme etkisi bulunmaktadır. Eğer atalet ağırlığı yüksek bir değer alınırsa parçacık global bir şekilde arama gerçekleştirir, bu değer küçük alınması durumunda ise yerel bir arama gerçekleşir (Kennedy ve ark., 2001). Shi ve Eberhart, atalet ağırlığının etkisinin incelenmesinde  $w \in [0.8,1.2]$  olarak seçmenin daha hızlı yakınsama sağlanacağını belirtmişlerdir (Kennedy ve ark., 2001). Atalet ağırlık değerinin 1.2'den daha büyük olduğu durumlarda hatalı yakınsama gerçekleşebilmektedir. Bu sonuçlara ek olarak atalet ağırlığın çok küçülmesi, parçacıkların hareket etmemesine sebep olmaktadır.
- $r_1$  ve  $r_2$  programın ürettiği değeri 0-1 arasında değişen rastgele sayılardır.
- Parçacığın tecrübesi ( $c_1$ ) ve Sürünün tecrübesi ( $c_2$ ):  $c_1$  sosyal parametre olarak adlandırılır.  $c_2$  ise kavramsal parametre olarak adlandırılır. Sürüde önceki iterasyonlardaki en iyi sonuca sahip parçacığın sonuçlarının bir sonraki iterasyona etkisidir.  $c_1$  ve  $c_2$  değişkenleri aynı zamanda öğrenme faktörü, korelasyon katsayısı veya hızlanma katsayıları olarak bilinmektedir. Bu katsayılar rastgele vektörler  $r_1$  ve  $r_2$  ile birlikte bir parçacığın hızı üzerinde bilişsel ve sosyal bileşenlerin stokastik etkisini kontrol eder. Çoğu uygulamada  $c_1 = c_2$  eşit alınırken bu sabitler arasındaki oran probleme bağlı olarak değişir.  $c_1$  ve  $c_2$  düşük değer alırsa yavaş değişimlerle oldukça düzgün bir parçacık yörüngesi oluştururken, yüksek değerler hızlı değişimlerle keskin hareketler ortaya çıkarır. Genellikle  $c_1$  ve  $c_2$  için sabit değerler kullanılmaktadır. Shi ve Eberhart tarafından yapılan çalışmada katsayılar için

genel olarak  $c_1 = c_2 = 1,494$  alınabileceği belirtilmiştir. Farklı çalışmalarda ise genellikle 1,8-2 aralığında değerler alınmaktadır.

- Yerel En İyi Değer (pbest): Yerel en iyi, bir parçacığın o ana kadar elde edilmiş en iyi amaç fonksiyon çözüm değeridir. Sürüde bulunan parçacık sayısı kadar her iterasyonda yerel en iyi bulunur. Amaç fonksiyonu yerine uygunluk fonksiyonları da kullanılabilir.
- Global En İyi Değer (gbest): Elde edilen en iyi amaç fonksiyonu ya da uygunluk fonksiyonu değeri ve bu değere sahip parçacığın çözümüdür.

PSO algoritma aşağıdaki basamaklardan oluşur

Adım 1 – Başlangıç sürüsü rastgele üretilen başlangıç pozisyonları ve hızları ile oluşturulur.

Adım 2 – Sürüdeki tüm parçacıkların uygunluk değeri hesaplanır.

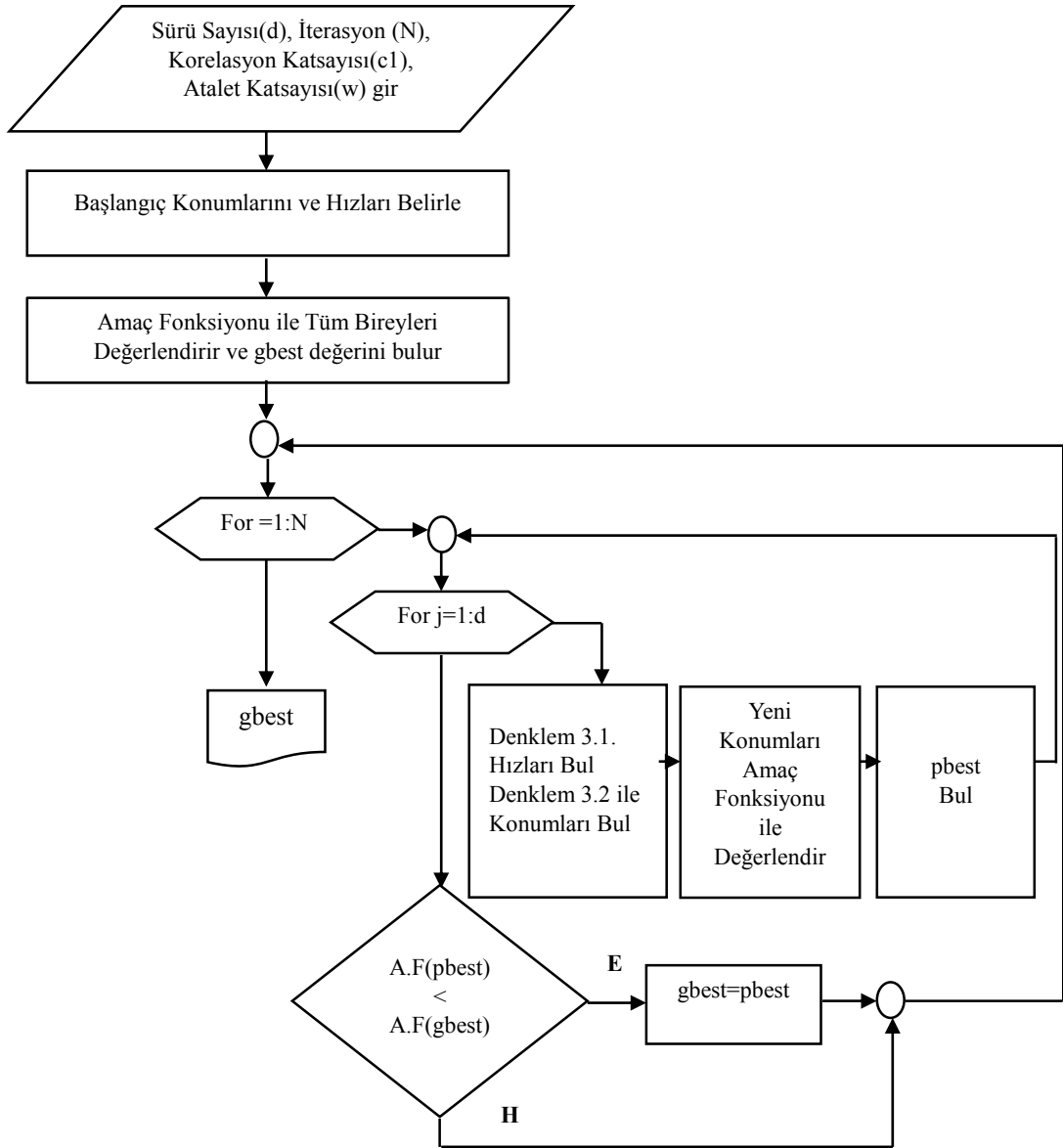
Adım 3 – Her bir parçacık için mevcut nesilde yerel en iyi pozisyon (pbest) bulunur. Sürü içerisinde en iyilerin sayısı parçacık sayısı kadardır.

Adım 4 – Mevcut nesile kadar bulunan yerel en iyiler içerisinde global en iyi pozisyon (gbest) belirlenir.

Adım 5 – Parçacık hızları ve pozisyonları (Denklem 3.1-3.2)'e göre güncellenir.

Adım 6 – Durdurma kriteri sağlanıncaya kadar 2-3-4-5 adımlar tekrar edilir.

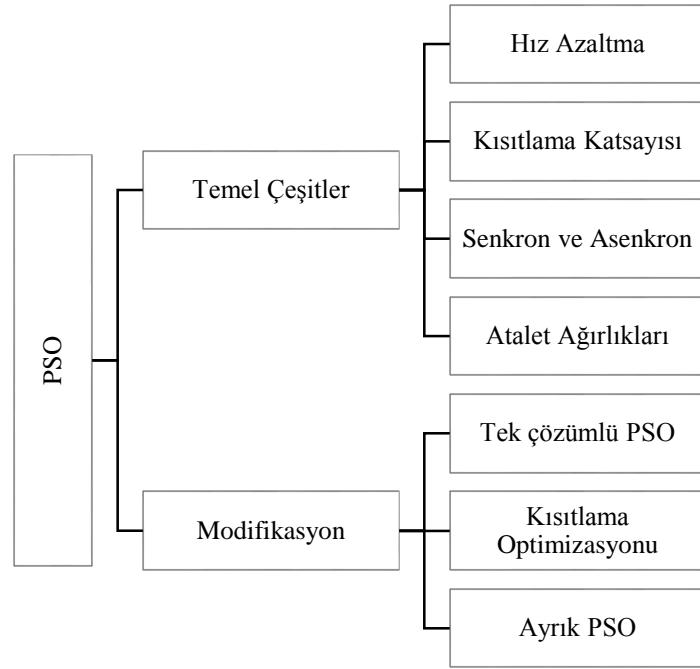
PSO algoritmasının akış diyagramı Şekil 3.5.'te verilmiştir.



Şekil 3.5. PSO algoritmasının akış diyagramı (Tozan ve ark., 2008)

### 3.2.2. PSO algoritmasının çeşitleri

Son yıllarda PSO algoritması hızlı bir şekilde gelişerek çeşitlilik göstermeye başlamıştır. Şekil 3.6.'da PSO algoritmasının çeşitleri gösterilmektedir. PSO'da bulunan yakınsama hızı ve çözüm kalitesini artırmak için hız azaltma, atalet ağırlıkları, kısıtlama katsayısı, senkron ve asenkron güncellemeler gibi parametreler temel çeşitliliği geliştirmek için kullanılmıştır (Rini ve ark., 2011)



Şekil 3.6. PSO algoritmasının çeşitleri

### 3.2.3. PSO algoritmasının avantaj ve dezavantajları

PSO algoritmasının genel yapısı incelendikten sonra avantaj ve dezavantajları aşağıdaki gibi özetlenebilir.

Avantajları:

- PSO algoritmasının uygulanması kolaydır. Ayarlama için birkaç parametreye sahip, optimizasyon kabiliyeti gelişmiş ve tamamlanması kolaydır.
- PSO algoritmasının örtüşüm ve mutasyon gibi hesaplama işlemleri yoktur. Arama, parçacığın hızı tarafından gerçekleştirilir. Birkaç kuşağın gelişimi sırasında, sadece en iyi parçacık diğer parçacıklara bilgi iletebilir. Bu özelliklere ek olarak arama hızı çok yüksektir.

Dezavantajları:

- Optimum yakınsamaya ulaşmak için güçlü bir şekilde atalet ağırlıklarından etkilenmektedir.

- Algoritma yakınsamaya başlayınca parametrelerin sabit değerlerinden dolayı parçacıkların sürekli değişimine sebep olabilmektedir.
- Çok modlu fonksiyon optimizasyonları gerçekleştirilirken erken ve yavaş yakınsama meydana gelmektedir.

### 3.3. Mobil Robotlar İçin PSO Algoritma Tabanlı Yol Planlama

Bu tez çalışmasının amaçlarından biride mobil robotların yol planlamasında karşılaşılan problemlerin çözümüne katkı sağlamak için PSO tabanlı bir yaklaşımın geliştirilmesidir. Geliştirilen algoritma ile statik farklı ortamlarda engellerden kaçınma ve hedefe ulaşmak için optimal yollar elde edilmiştir. Optimal yolları elde etmek için uygunluk fonksiyonu tanımlanmıştır. Mobil robot hedef noktasına varana kadar parçacıkların en iyi konumlarını(gbest) kullanarak hareket etmektedir.

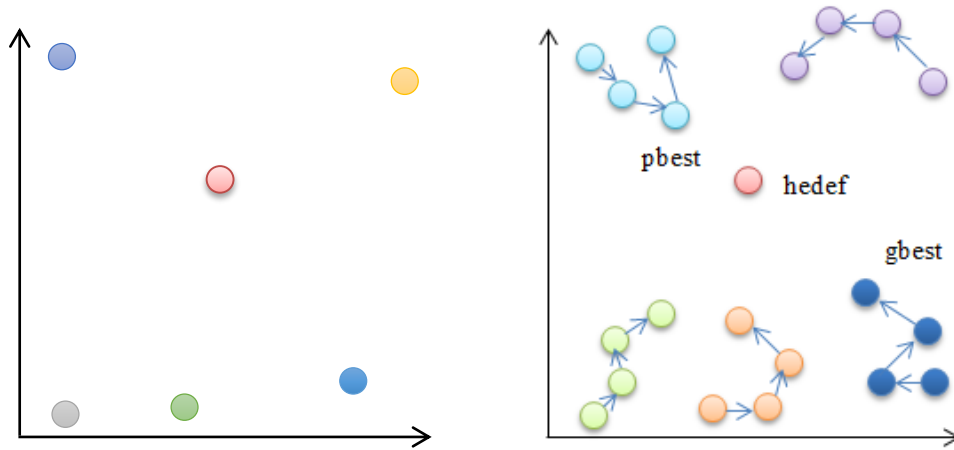
#### 3.3.1. PSO tabanlı uygunluk fonksiyonunun oluşturulması

PSO algoritmasında tanımlanan uygunluk fonksiyonu, mobil robotun yol planlamasında en kısa yolu bulmak amacıyla uygulanmaktadır. Yol planlama problemlerinde amaç, engellere çarpmadan başlangıç ve hedef arasındaki en uygun yolu bulmaktır. Uygunluk ölçüsü olarak en kısa yol ve minimum zaman parametreleri seçilmiştir. Yol planlama da her parçacığın uygunluk değerinin hesaplanması için amaç fonksiyonunun belirlenmesi gerekmektedir. Uygunluk fonksiyonu olarak Bölüm 2’de verilen (Denklem 2.3) ve Tablo 2.1.’de bulunan Manhattan, Euclidean ve Diagonal heuristik fonksiyonları belirlenmiştir. Bu algoritmadaki uygunluk fonksiyonunda  $f$  uygunluk fonksiyonunu,  $p$  parçacık ve  $n$  ise popülasyon uzunluğudur. Amaç fonksiyonu en kısa yolu planlamak için seçilen parçacıklar arasındaki uzaklıkların toplamı olarak bulunmaktadır. Amaç fonksiyonundan elde edilen değer en kısa yolu vermektedir.

#### 3.3.2. Geliştirilen PSO tabanlı mobil robotlarda global yol planlama

PSO algoritması ile yol uzunluğu optimize edilerek en uygun yol elde edilmiştir. Engelli bir statik ortamda mobil robot yol planlama probleminde, parçacıkların en iyi

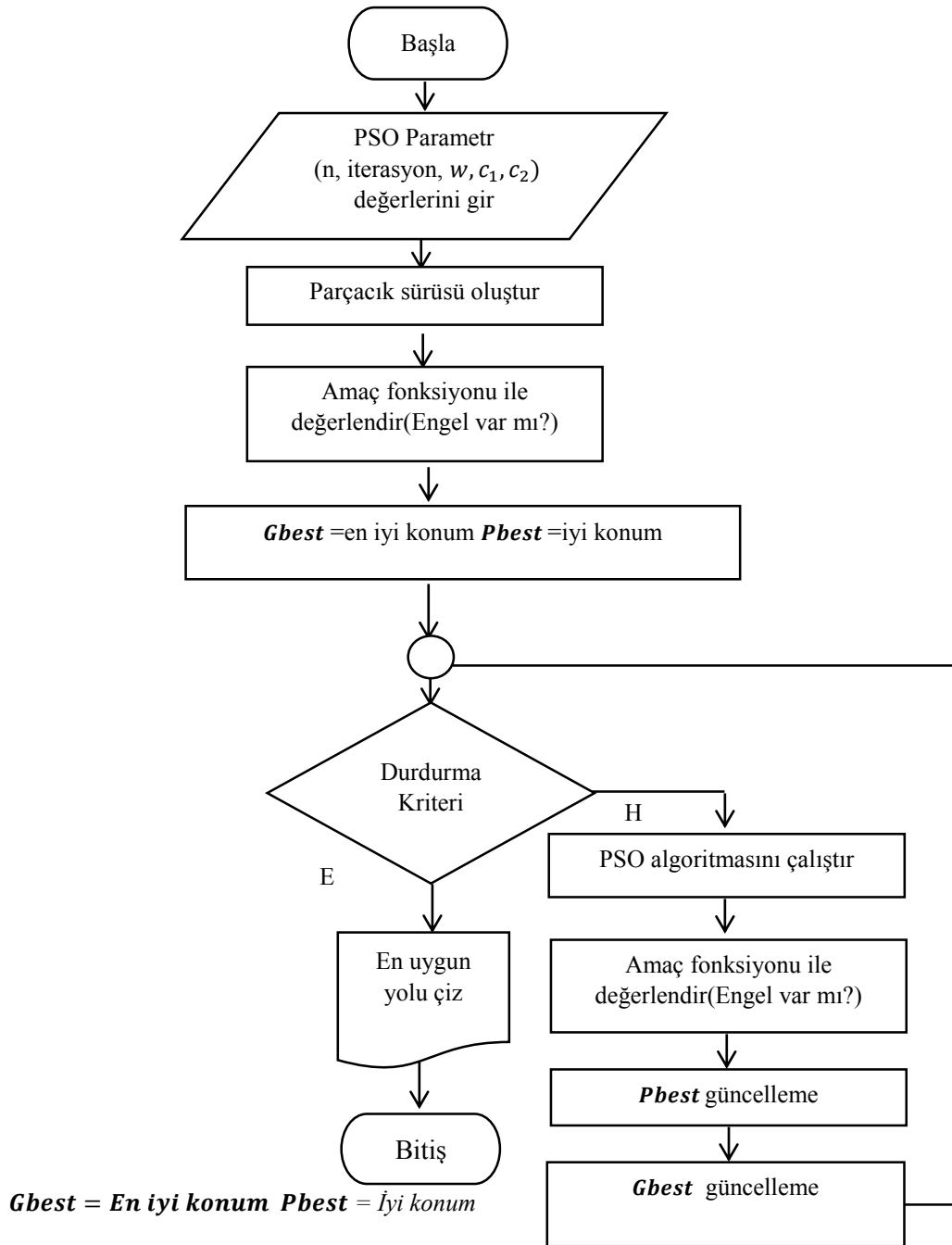
konum bulma özelliği kullanılarak en uygun yol bulunmuştur. En iyi konum yani global konum( $g_{best}$ ) seçilerek sürü içerisindeki minimum uygunluğa sahip olan parçacık belirlenmiştir. Yol planlamanın gerçekleştirilebilmesi amacıyla mobil robotun ve engellerin konumları belirlendikten sonra PSO algoritması uygulanmaya başlanmıştır. Uygulamada her yinelemede parçacıkların en minimum olan global konum( $g_{best}$ ) değeri seçilmiştir. Mobil robot hedefe ulaşmaya kadar seçilen global en iyi konum değerlerine göre hareket etmektedir(Şekil 3.7.).



Şekil 3.7. Parçacıkların hedefe doğru ilerlemesi

Algoritma probleme uyarlanırken başlangıç olarak parçacık hızları 0 olarak belirlenmiştir. Ardından parçacıklara başlangıç değeri olarak, belirlenen kısıtlar arasındaki konumlarda rastgele reel sayılar verilmiştir. Döngü içinde hızlar hesaplanıp, ardından bu hızlar ile parçacıklar hareket ettirilmekte ve bulunan parçacıkların yeni konumlarının uygunluk değerleri belirlenmektedir. İyi olanların yerleri değiştirilerek  $p_{best}$  ve  $g_{best}$  değerleri belirlenmektedir. Her bir parçacık için hız değerleri hesaplanarak döngü tekrar başa gelmekte ve işlemler tekrar edilmektedir. Döngü istenen durdurma kriteri sağlanana kadar devam etmektedir.

Çoklu mobil robotlar için önerilen PSO tabanlı yol planlamanın özetlenmiş işlem basamakları ve akış diyagramı(Şekil 3.8.) verilmiştir.

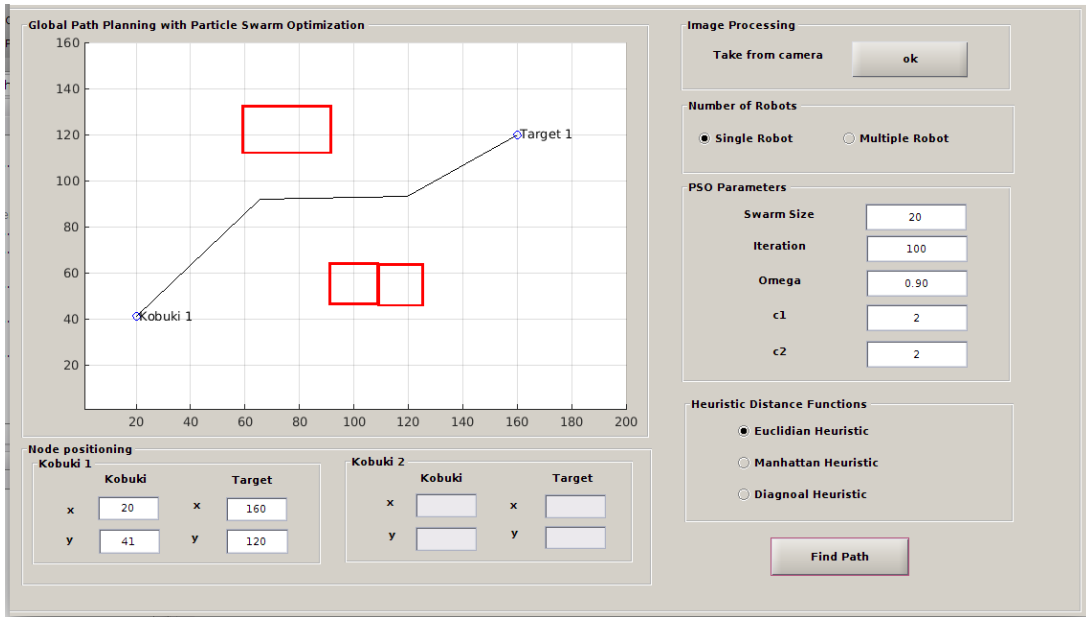


Şekil 3.8. Geliştirilen PSO tabanlı Global Yol Planlamanın Akış Diyagramı

### 3.4. Simülasyon ve Deneysel Sonuçlar

Önerilen PSO tabanlı yol planlayıcının etkinliği simülasyon ve deneysel ortamda test edilerek belirlenmiştir. Tekli ve çoklu mobil robotların global yol planlaması için Şekil 3.9. ve Şekil 3.11. simülasyon sonuçlarını gösterirken Şekil 3.10. ve Şekil 3.12.

deneysel sonuçları göstermektedir. Tablo 3.2.-3.'de PSO tabanlı yol planlayıcının simülasyon ve deneysel çalışmalardan elde edilen yol uzunluk değerleri ve hata oranları verilmektedir. PSO algoritmasında popülasyon büyüklüğü 20, momentum katsayısı 0.9,  $c_1$  ve  $c_2$  2 olarak belirlenen ideal parametre değerleri ile algoritmanın çalışmasında ilk olarak parçacıkların hız değerine 0 değeri atanmaktadır. Ardından parçacıklara başlangıç değeri olarak, belirlenen kısıtlar (gerçek çevre ortamının boyutları) arasındaki değerlerden rastgele reel sayılar verilmektedir. Algoritmada bulunan parçacıkların hızları hesaplandıktan sonra, bu hızlara bağlı olarak parçacıklar hareket ettirilmekte ve yeni konumların uygunluk değerleri hesaplanmaktadır. İyi olan parçacıkların içerisinde en iyi konuma sahip olan ( $g_{best}$ ) parçacıklar mobil robotun gideceği konumları belirlemektedir.



Şekil 3.9. Statik engelli bir ortamda mobil robot için PSO tabanlı global yol planlama





Sahne 1



Sahne 2



Sahne 3



Sahne 4



Sahne 5

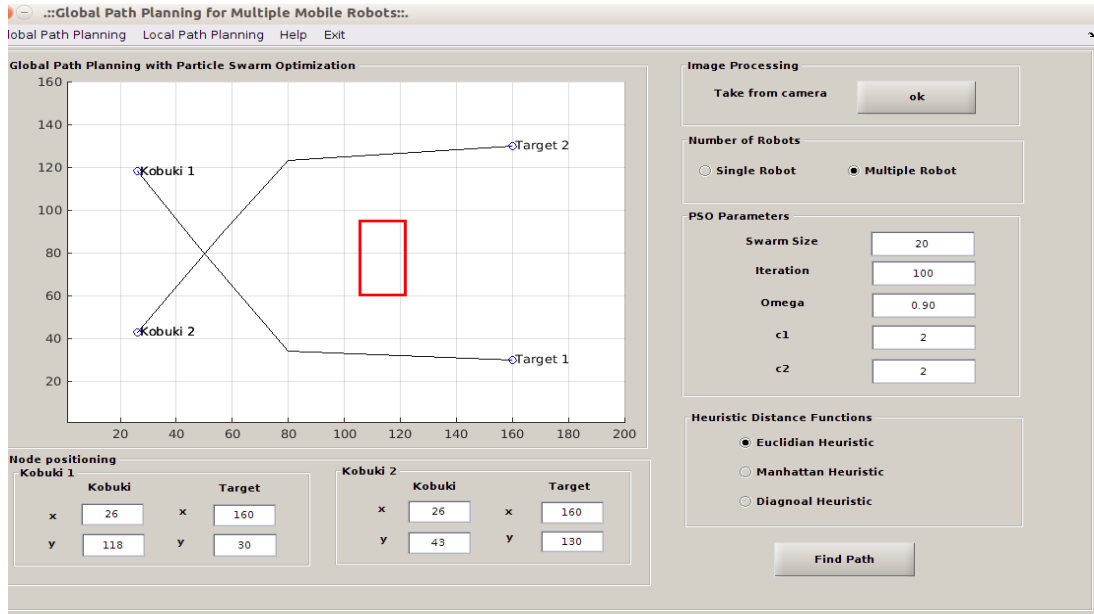


Sahne 6

Şekil 3.10. PSO tabanlı yol planlayıcının gerçek ortamda uygulanması

Tablo 3.2. Şekil 3.9. ve 3.10.'da gösterilen mobil robot için PSO tabanlı yol planlayıcı tabanlı simülasyon ve deneysel çalışmadaki yol uzunluk değerlerinin karşılaştırılması

Çalışma Sayısı	Simülasyondan elde edilen yol uzunluğu(cm)	Deneysel çalışmadan elde edilen yol uzunluğu(cm)	% Hata
1	171,0599	144,0212	15,08
2	196,4468	152,6463	22,29
3	170,8966	145,9441	14,60
4	171,0623	144,9763	15,24
5	171,0559	144,0722	15,77
6	171,0559	143,1596	16,31
7	171,0598	143,0313	16,38
8	171,0598	145,8586	15,11
9	171,0604	144,1234	15,74
10	169,6510	145,3899	14,30
<b>Ortalama</b>	<b>173,4408</b>	<b>145,3222</b>	<b>16,08</b>



Şekil 3.11. Statik engelli bir ortamda çoklu mobil robot için PSO tabanlı global yol planlama

Tablo 3.3. Şekil 3.11. gösterilen çoklu mobil robotlar için PSO tabanlı yol planlayıcının simülasyon ve deneysel çalışmadaki yol uzunluklarının karşılaştırılması

Çalışma Sayısı	Robot	Simülasyondan	Deneysel çalışmadan	%	%
		elde edilen yol uzunluğu(cm)	elde edilen yol uzunluğu(cm)	Kobuki 1 Hata	Kobuki 2 Hata
1	Kobuki 1	179,9649	152,6780		
	Kobuki 2	176,8809	200,9646		
2	Kobuki 1	179,9174	153,4625		
	Kobuki 2	176,7726	200,0883		
3	Kobuki 1	179,9174	152,2681		
	Kobuki 2	180,8647	203,0776		
4	Kobuki 1	180,5257	153,4157		
	Kobuki 2	180,8647	203,5790		
5	Kobuki 1	179,9882	153,0881		
	Kobuki 2	176,7958	199,8194		
6	Kobuki 1	182,2488	154,5844	14,5	13,23
	Kobuki 2	176,8335	199,2436		
7	Kobuki 1	179,9276	152,2057		
	Kobuki 2	179,2090	204,3352		
8	Kobuki 1	179,9189	152,6413		
	Kobuki 2	176,8116	200,1684		
9	Kobuki 1	179,9573	153,7811		
	Kobuki 2	176,8808	205,1410		
10	Kobuki 1	182,2488	155,9020		
	Kobuki 2	183,1796	206,5688		
Ort	<b>Kobuki 1</b>	<b>180,4615</b>	<b>153,4026</b>		
	<b>Kobuki 2</b>	<b>178,5093</b>	<b>202,1359</b>		

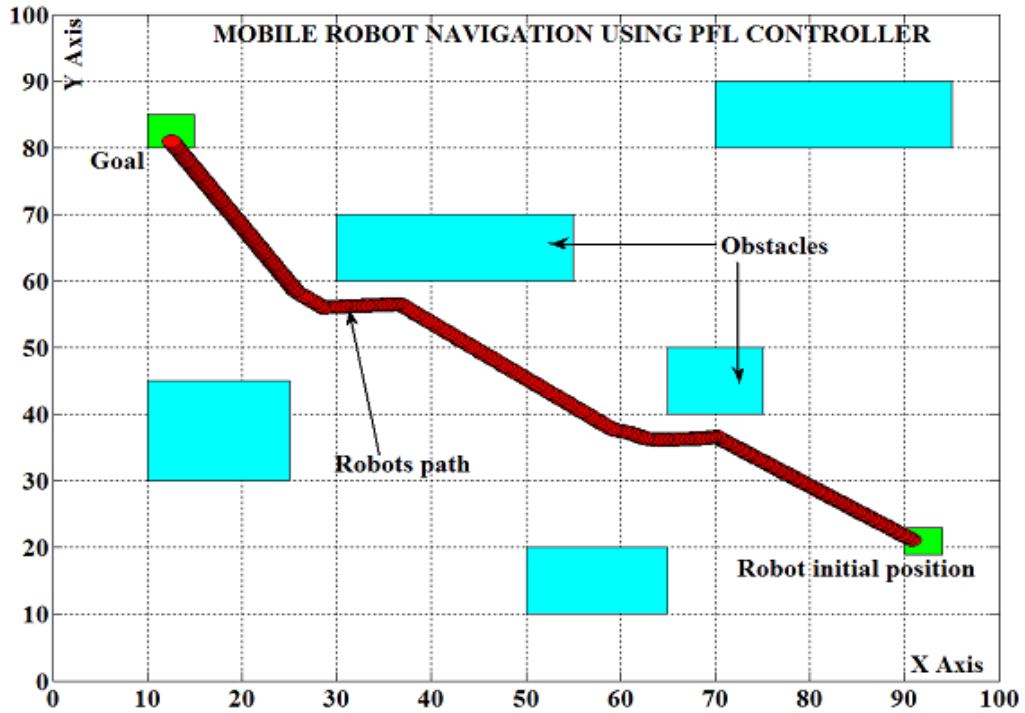
Tablo 3.2.'ye göre tekli mobil robot için PSO tabanlı yol planlayıcı kullanılarak oluşturulan simülasyon ve deneysel çalışmadaki yol uzunluk değerlerinin karşılaştırma sonucuna göre ortalama hata yaklaşık olarak %16 olarak bulunmuştur.

Tablo 3.3.'deki sonuçlara göre çoklu mobil robotlar için PSO tabanlı yol planlayıcı kullanılarak oluşturulan simülasyon ve deneysel çalışmadaki yol uzunluk değerlerinin karşılaştırma sonucuna göre Kobuki robot 1 için yaklaşık olarak %14,05 daha kısa olduğu belirlenmiştir. Bu sonuçtan dolayı daha kısa sürede daha az enerjiyle yol planlaması gerçekleştirilmiştir. Diğer Kobuki robot 2 için ise deneysel

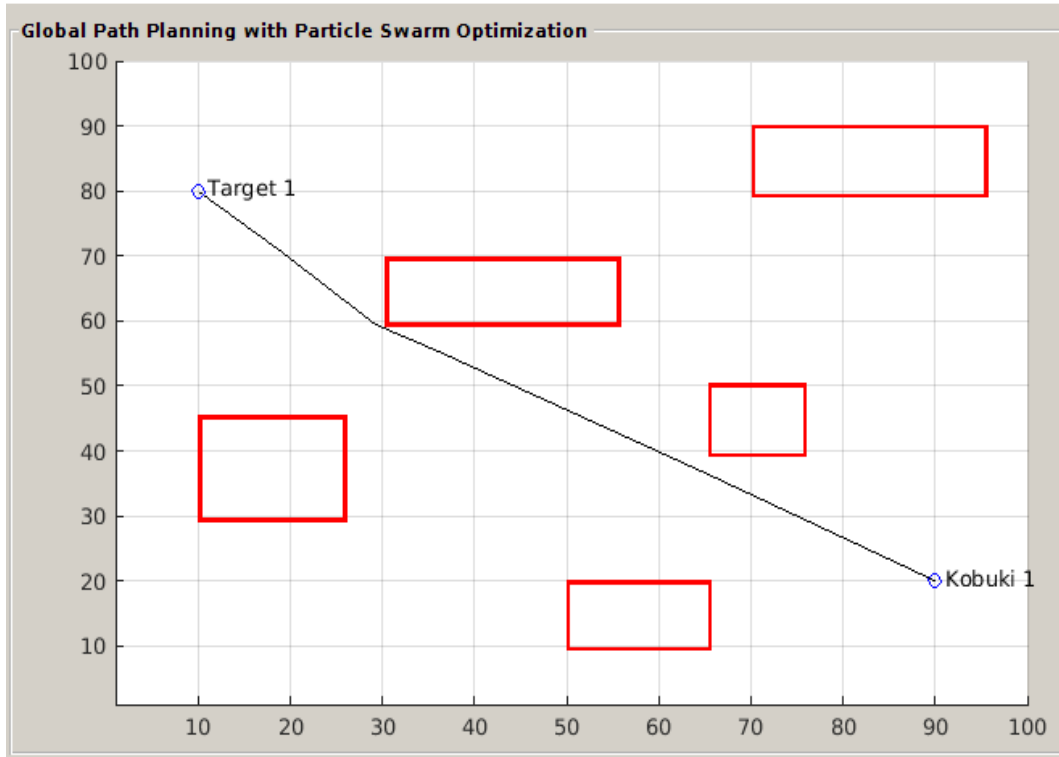
çalışmada yol uzunluğu yaklaşık olarak %13.23 daha uzun olduğu belirlenmiştir. Kobuki 2 mobil robotunun deneysel ortamda yol uzunluk değerlerinin fazla çıkmasının sebebi ise Kobuki 1 mobil robot ile çakışmaması için beklenen süreden kaynaklanmaktadır.

### 3.5. Geliştirilen PSO Algoritmasının Diğer Algoritmalarla Karşılaştırılması ve Performans Analizi

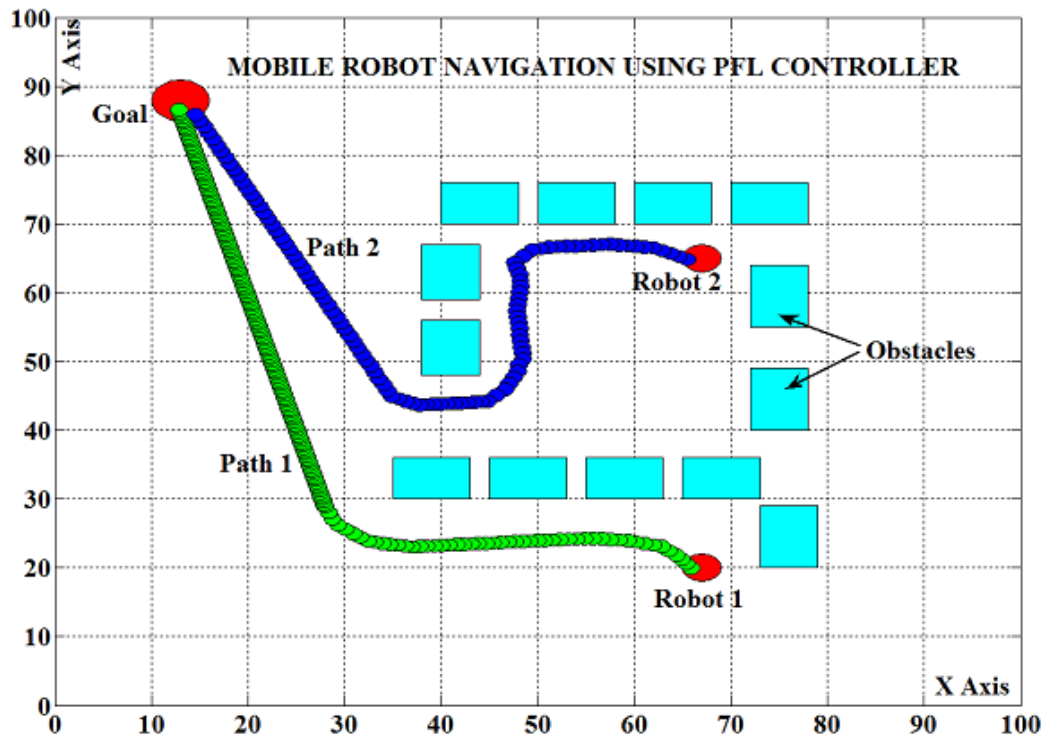
Bu bölümde optimum yol uzunluk parametresinin değerlerini karşılaştırmak için tarafımızdan geliştirilen PSO tabanlı yol planlayıcısı ile diğer yol planlama algoritmaların performans analizi gerçekleştirilmiştir. Her iki algoritma için de benzer ortamlarda tekli ve çoklu mobil robotlar yol planlaması için simülasyon çalışmaları yapılmıştır.



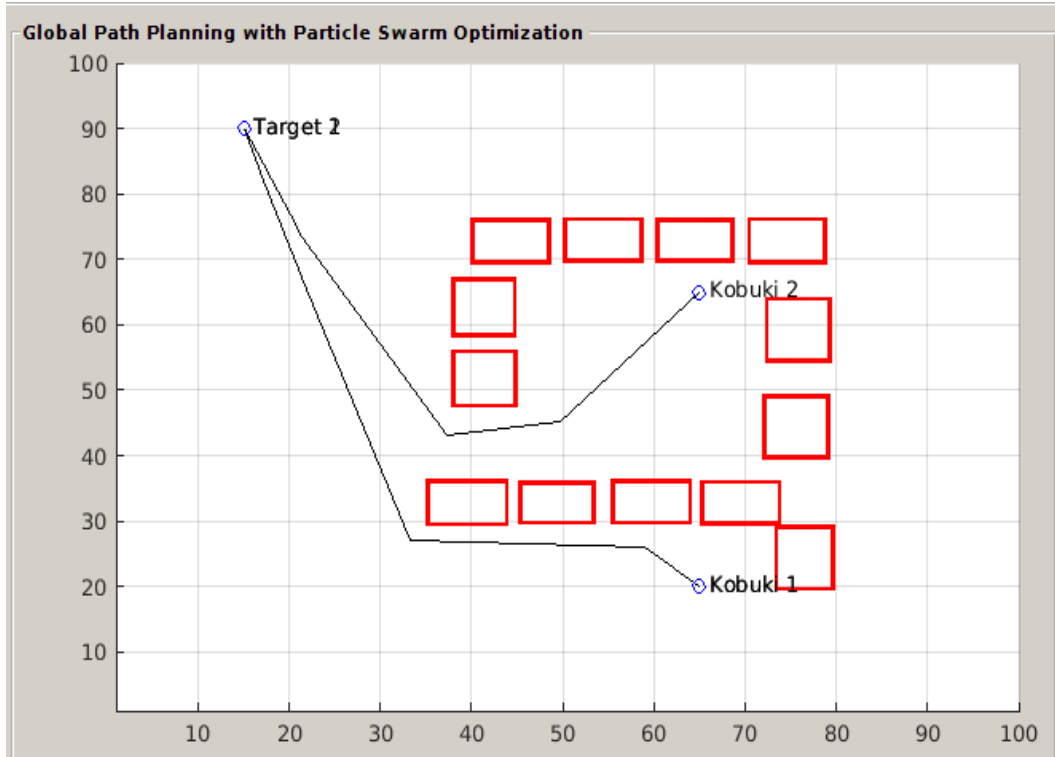
Şekil 3.12. Tek robot için PFL tabanlı mobil robot navigasyonu (Patle, 2016)



Şekil 3.13. Tarafımızdan geliştirilen PSO tabanlı global yol planlama



Şekil 3.14. Çoklu robot için PFL tabanlı mobil robot navigasyonu (Patle, 2016)



Şekil 3.15. Çoklu mobil robot için tarafımızdan geliştirilen PSO tabanlı global yol planlama

Tablo 3.4. Simülasyon sonuçlarının karşılaştırılması

	Tarafımızdan geliştirilen PSO kullanılarak elde edilen simülasyon yol uzunluğu("cm")	PFL kullanılarak elde edilen simülasyon yol uzunluğu("cm") (Patle, 2016)	% Performans Farkı
Senaryo 1	100,5761(Şekil 3.13.)	150,3(Şekil 3.12.)	33,08

Tablo 3.5. Simülasyon sonuçlarının karşılaştırılması

	Tarafımızdan geliştirilen PSO kullanılarak elde edilen simülasyon yol uzunluğu("cm")	PFL kullanılarak elde edilen simülasyon yol uzunluğu (Patle, 2016)	% Performans Farkı
Robot 1	99,8597 (Şekil 3.15.)	162,00 (Şekil 3.14.)	38,30
Robot 2	89,6200 (Şekil 3.15.)	122,80 (Şekil 3.14.)	27,01

Şekil 3.12.-15. ve Şekil 3.13.-14.'de sırasıyla önerilen tarafımızdan geliştirilen PSO tabanlı ve Patle tarafından geliştirilen Olası Bulanık mantık(PFL) tabanlı mobil robot yol planlayıcılardan simülasyon ortamındaki analizler gösterilmektedir (Patle, 2016).

Tekli ve çoklu mobil robotların global yol planlamada tarafımızdan geliştirilen PSO ve PFL tabanlı yol planlayıcılardan elde edilen yol uzunluk değerleri Tablo 3.4. ve Tablo 3.5.'te verilmiştir. Verilere göre önerilen PSO algoritmasının tekli mobil robot yol planlamasında %33.08, çoklu robot yol planlamasında ise Kobuki robot 1 için %38,03 diğer Kobuki robot 2 için ise %27,01 yol uzunluğunun daha kısa olduğu saptanmıştır.

## **BÖLÜM 4. ÇOKLU MOBİL ROBOTLAR İÇİN ATEŞ BÖCEĞİ ALGORİTMASI TABANLI GLOBAL YOL PLANLAMA**

### **4.1. Giriş**

Son zamanlarda heuristik tekniklerden biri olan Ateş Böceği(FA) algoritması karmaşık optimizasyon problemlerin çözümünde kullanılmaya başlanmıştır. 2008 yılında Yang tarafından ateşböceklerinin ışık yoğunluklarındaki değişikliklerden faydalanarak geliştirilen FA, geleneksel algoritmalarından daha uyarlanabilir olma özelliği ile ayrılmaktadır.

FA sürekli, tümeşik, kısıtlı, çok amaçlı, dinamik ve gürültülü optimizasyon, makine öğrenmede sınıflandırma problemleri, veri madenciliği ve sinir ağları, görüntü işleme, endüstriyel optimizasyon, kablosuz sensör ağları, anten tasarımı, iş optimizasyonu, robotik, web, kimya ve inşaat mühendislik alanları gibi farklı dallarda kullanılmaktadır (Fister ve ark., 2013). FA'nın yakınsama oranı ve sağlamlık özelliklerinden dolayı tercih edilen bir algoritma haline gelmiştir. Gauss dağılım fonksiyonu ile FA'nın yakınsama hızı artırılarak algoritmanın etkinliği artmıştır.

FA, kendi kendine adapte olma ve organize etme yeteneğine sahip olan evrimsel bir tekniktir. Sağlamlık ve yüksek yakınsama oranı gibi özellikleri kullanarak karmaşık sistemlerin değişen ve istenmeyen durumlarını uygun hale getirir. Başlangıç pozisyonundan daha iyi bir pozisyona gelme yeteneğini kullanarak yineleme sayısını en aza indirir. Son yıllarda FA, güçlü optimizasyon yeteneğinden dolayı mobil robotların yol planlamasında da yol bulmada kullanılmaktadır.

Literatürde mobil robotların yol planlamada en kısa yolu bulmak için FA tabanlı bir çok çalışma bulunmaktadır. 2017 yılında Patle ve arkadaşları tarafından yapılan çalışmada Normal Olasılık Dağılımı (NPD) ile değiştirilmiş FA kullanarak verimli



seyrüsefer kontrolörü geliştirilmiştir. En uygun yolu bulmak için FA denetleyicisi kullanılırken navigasyon süresini en aza indirmek için ise NDP kullanılmıştır. NPD, ateşböceklerini rastgele olarak oluşturmakta ve iterasyon önleyerek hesaplamaları en aza indirmiştir. Önerilen denetleyicinin bilinmeyen çevrede robotun yol planlamasında etkili olduğunu kanıtlamıştır. Diğer yapay zeka yaklaşımlarıyla karşılaştırıldığında optimal yol ve çözüm kalitesi açısından daha iyi performans göstermiştir (Patle ve ark., 2017). 2014 yılında Li ve arkadaşları tarafından yapılan çalışmada bir mobil robot için bilinen çevrede en kısa olası yolun bulunması amacıyla FA ve Bezier eğrisine dayanan yeni bir planlama algoritması sunulmuştur. FA'nın tercih edilme sebebi, GA ve PSO algoritmalarında olduğu gibi tercih edilmeyen yerel optimum noktalarda bulunmayarak global optimum noktalarda bulunmasıdır. Başlangıç ve bitiş noktaları arasındaki engellere çarpmaksızın en kısa olası yolu tanımlayan Bezier eğrisinin optimal kontrol noktalarını belirlemek ve optimize etmek için FA kullanılmıştır. Farklı karmaşıklıklara sahip olan haritalarda yol planlama problemini çözmek için kriter olarak yol uzunluğu seçilerek FA, GA ve atalet ağırlıklı parçacık sürü optimizasyonu (PSO-w) kıyaslamıştır. Deneysel sonuçlar FA'nın GA ve PSO-w algoritmalarına göre daha yüksek başarı oranına sahip olduğunu göstermiştir (Li ve ark., 2014). 2013 yılında Brand ve ark. tarafından yapılan çalışmada mobil robotun çarpışmadan en kısa yolu planlaması için FA iki boyutlu statik ve dinamik çalışma ortamı tasarlanmıştır. Başlangıç ve bitiş noktaları arasındaki yol uzunluğunu en aza indirgeyen optimal bir çözüm hedeflenmiştir. Mobil robot hareketli olan engellerden gerçek zamanlı olarak kaçınarak yol planlanma gerçekleştirilmiştir. Bu çalışmaya ek olarak, FA ve ACO algoritmaları kullanılarak farklı çalışma alanlarında gerçekleştirilen yol planlama hem yol uzunluğu hem de hesaplama maliyeti açısından karşılaştırılmıştır (Brand ve ark., 2013). 2012 yılında Liu ve arkadaşları tarafından ateş böceği algoritmasının iterasyon sayısına bağlı olarak değişen emilim ve rassal parametreleri baz alınarak geliştirilen adaptif ateş böceği (AFF) algoritması ile yeni bir planlama algoritması tasarlanmıştır. AFF algoritması, temel FA'nın çözüm kalitesini ve yakınsama özelliğini iyileştirmiştir. Bu algoritmanın uygulanabilir olduğu simülasyon deneyleriyle doğrulanmıştır (Liu ve ark., 2012). Parhi ve arkadaşları çalışmalarında mobil robotların seyrüseferinde, ateşböceklerinin fiziksel ve doğal izlenimini

kullanarak FA'nın genel görünüm ve doğal zeka mekanizmasının geliştirilmesi hedeflenmiştir. Önerilen çalışmada karar değişkenleri olarak parlaklık ve ışık seçilmiştir. Mobil robot seyrüseferinde, en kısa yol üretmek ve çok değişkenli korelasyon fonksiyonu ile eş merkezli küre tabanlı geometrik bir teknikle engellerden kaçınma amaçlanmaktadır. Simülasyon ve deneysel sonuçlar optimal yolun sağlandığını ve hata yüzde oranının azaldığını göstermiştir. FA'nın performansı PSO ve bakteri potansiyel alanı (BPF) yaklaşımlarıyla karşılaştırılmıştır. Karşılaştırılması sonucuna göre önerilen yöntemin en kısa yol bulma ve çözümlene kalitesinde daha iyi performans gösterdiği kanıtlanmıştır (Parhi ve ark., 2017). Tablo 4.1.'de FA mobil robot yol planlamaya ilgili verilen literatür çalışmalarının özet hali verilmiştir.

Tablo 4.1. Mobil robot yol planlamada FA uygulaması

Yazarlar	Kin. Model	Engel Şekli	Statik Engel	Dinamik Engel	Statik Hedef	Dinamik Hedef	Sim .	Gerçek Sistem
B. K. Patle, D. Parhi, A. Jagadees O. P. Sahu	H	Dörtgen	E	H	E	H	E	E
B. Li,L. Liu, Q. Zhang, D. Lv, Y. Zhang, J. Zhang, X. Shi	H	Daire	E	H	E	H	E	H
M. BRAND, X. YU	H	Kare	E	E	E	H	E	H
C.Liu Z.Gao, W.Zhao	H	Çokgen	E	H	E	H	E	H
Md. Tajmiruzza man Md. Al-Mamun	H	Belirsiz	E	H	E	H	E	H
B.K. Patle , D. R. Parhi	H	Çokgen	E	H	E	H	E	H

## 4.2. FA için Matematiksel Modelleme

Ateş böceği yöntemi K. N. Krishnanand ve D. Ghose tarafından 2005 yılında geliştirilmiş olup rastgele sayısal optimizasyon arama yöntemlerinden biridir (Krishnanand ve ark., 2005).

Popülasyon tabanlı meta heuristik(sezgisel) algoritma olan ateş böceği yöntemi çoklu model fonksiyonlarını optimize etmek amacıyla kullanılmaktadır. Bu yöntem ateş böceklerinin sosyal davranışları baz alınarak geliştirilmiştir. Çok modelli fonksiyonların optimizasyonunda kullanılan FA ve diğer optimizasyon algoritmaları arasındaki en önemli fark maksimum noktaların verimli bir şekilde bulunması ve sürüdeki bireylerin dinamik karar almasıdır.

Ateşböceği sürüsünde her bir birey komşularını seçmek için aldıkları sinyal gücüne göre hareketini belirlemektedir. Buna ek olarak hareket belirlemede karar alanını kullanmaktadır. Ateş böceklerinin eşlerini veya avlarını çekmek için kullandığı ışığın yanıp sönmeye neden olan oksijenin "lusiferaz" adlı bir kimyasalla reaksiyona girerek "lusiferin" adlı bir bileşenin oluşmasıdır. Bir ateş böceği ne kadar parlak olursa diğer ateşböcekleri içinde o kadar çekici olmaktadır.

Ateş böceği algoritması (FA), Dr. Xin-She Yang tarafından geliştirilen ve tropikal iklim bölgelerindeki ateş böceklerinin sosyal davranışlarını baz alan bir meta-sezgisel optimizasyon algoritmasıdır (Yang, 2009). Bu algoritma diğer sürü zekâ tabanlı algoritmalarla birçok benzerliğe sahip olmasına rağmen anlaşılabilirliği ve uygulanabilirliği daha basittir. Bir ateş böceğinin ışıklarını yakıp söndürme davranışının birinci amacı olarak diğer ateş böceklerini çekmek için bir sinyal sistemi oluşturmasıdır. Araştırmacılar, yanıp sönen ışıkların, ateş böceklerine arkadaşlarını bulmada, olası avlarını çekmede ve avcılardan kendilerini korumada yardımcı olduklarını düşünmektedirler. Bu özelliklere ek olarak ateş böceklerinin tek cinsiyette olması ve birbirlerini çekmeleri algoritmanın temelini oluşturur. Dolayısıyla erkek ve dişi ayrımı olmadığı gibi, bütün ateş böcekleri diğer ateş böcekleri tarafından cezbedilmektedir. Ateş böceklerinin çekicilikleri ve parlaklıkları arasında doğru orantı bulunmaktadır. Yani bir ateş böceği ne kadar parlak olursa diğer ateş böcekleri içinde o kadar çekici hale gelmektedir. Ateş böceklerinin aralarındaki mesafe, parlaklığı azalttığı için çekiciliği de azaltmaktadır. Ortamda bir ateş böceğinden daha parlak ateş böceği varsa, bu ateş böceği parlak olana doğru hareket edecektir (Yang, 2010; Apostolopoulos ve ark., 2011). Eğer ortamda daha parlak ateş böceği yoksa ateş böcekleri rastgele yönlerde hareket ederler. Ateş böceği algoritmasında, verilen bir optimizasyon probleminin amaç fonksiyonunda optimum

çözümler elde etme, ateş böceği sürüsüne parlak ve daha çekici yerlere gitmede yardım eden yanıp sönen ışık ya da ışık şiddeti ile ilişkili olmaktadır (Gandomi ve ark., 2011).

Ateş Böceklerinin davranışları kurallaştırıldığında aşağıdaki sonuçlar ortaya çıkar.

- Ateş böcekleri cinsiyetsiz olma özelliklerini kullanarak diğer ateş böceklerini etkileyerek kendilerine doğru çekebilirler.
- Ateş böceklerinde çekicilik yaydıkları ışık yani parlaklıkları ile doğru orantılıdır. Yaydıkları ışığa göre çok parlak olanlar az parlak olanları kendilerine doğru hareket ettirirler.
- Ateş böceklerinin parlaklıklarına bağlı olarak optimize edilecek problemin çözümünde kullanılacak uygunluk fonksiyonun değeriyle belirlenmektedir (Kaya ve ark., 2008).

Üç temel davranış kurallarına dayanarak FA'nın temel işlem adımları Şekil 4.1.'de verilen sözde kodda özetlenmiştir.

```

Amaç fonksiyonu belirlenir  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ .
n adet ateş böceği başlangıç popülasyonu değerleri üretilir
 $x_i$ 'deki Işık yoğunluğunu ( $I_i$ )  $f(x_i)$  ile hesaplanır
Işık emilimi ( $\gamma$ ) belirlenir.
while (durdurma kriteri),
  for  $i = 1 : n$ 
    for  $j = 1 : n$ 
      if ( $I_i < I_j$ )
        Ateşböcekleri  $i$ 'den  $j$ 'ye hareket eder.
      end if
       $r$  ve  $\exp[-\gamma r^2]$  yoluyla çekicilik değişir.
      Yeni çözümleri hesaplar ve ışık yoğunluğu güncellenir.
    end for  $j$ 
  end for  $i$ 
  Ateşböcekleri sıralanır ve global en iyi ( $g^*$ ) bulunur.
end while
Sonuçlar görüntülenir.

```

Şekil 4.1. FA algoritması sözde kodu (Yang, 2014)

#### 4.2.1. FA'nın temel parametreleri

FA'da iki temel konu vardır: Bunlar ışık yoğunluğunun değişimi ve çekiciliğin formüle edilmesidir. Ateş böceklerinin çekicilikleri parlaklıkları ile belirlenmektedir. Elde edilen çekicilikle amaç fonksiyonu ilişkilendirilmiştir.

Ateş böceklerinin ışık yoğunluğu( $I$ ) değeri (Denklem 4.1) verilmiştir. Bu değer başlangıç ışık yoğunluğuna( $I_0$ ), gammaya( $\gamma$ ) ve uzaklığa( $r$ ) bağlıdır.

$$I = I_0 e^{-\gamma r} \quad (4.1)$$

Eşitlik-4'de Ateş böceklerinin çekiciliğini( $\beta$ ) veren denklemdeki (Denklem 4.2) başlangıç değeri( $\beta_0$ ),aslında iki ateşböceğinin arasındaki uzaklığın sıfır olduğu andaki değeri göstermektedir. Çekicilik, uzaklığın karesi, gamma ve başlangıç değerine bağlıdır.

$$\beta = \beta_0 e^{-\gamma r^2} \quad (4.2)$$

(Denklem 4.2) Denklemi kullanarak çekiciliği hesaplama bazen zor olabildiğinden, işlemleri kolaylaştırmak için (Denklem 4.3) kullanılabilir[9].

(Denklem 4.3) Denklemde Ateş böceklerinin çekiciliğini( $\beta$ ) veren denklemdeki (Denklem 4.2) başlangıç değeri( $\beta_0$ ),aslında iki ateşböceğinin arasındaki uzaklığın sıfır olduğu andaki değeri göstermektedir. Çekicilik, uzaklığın karesi, gamma ve başlangıç değerine bağlıdır.

$$\beta = \frac{\beta_0}{1+\gamma r^2} \quad (4.3)$$

Ateş böceklerinin birbirine olan uzaklıkları önem arz etmektedir, çünkü uzaklıkla hem ışık yoğunluğu hem de çekicilik değişmektedir. Dolayısıyla bu değişimler, ateş böceği için hareketi belirleyecektir. Uzaklık, denklemdeki (Denklem 4.4) gibi hesaplanabilir.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (4.4)$$

Ateş böcekleri daha parlak ve daha çekici olan ateş böceklerine doğru hareket ederler. Bu hareket, (Denklem 4.5) aşağıdaki gibi ifade edilebilir. Eşitlikte bulunan ikinci ifade çekicilik formülünden gelmektedir. Yani, iki ateş böceği arasındaki uzaklık ve çekiciliğin çarpılması ile ifade edilmektedir. Üçüncü ifadede bulunan  $\alpha$  rastlantı parametresi,  $\varepsilon$  ise gaussian dağılımından gelen bir vektörü ifade etmektedir (Yang, 2014).  $\alpha$  rastlantı parametresi genellikle  $[0,1]$  arasında rastgele olarak üretilmektedir.

$$x_i^{t+1} = x_i + \beta_0 e^{-\gamma r^{2ij}} (x_j - x_i) + \alpha \varepsilon_i^t \quad (4.5)$$

$\beta_0=0$  olduğu zaman hareket sadece rastgele yürüyüşe bağlıdır. Diğer yandan,  $\gamma$  parametresi yakınsama hızı üzerinde çok önemli bir etkiye sahiptir. Bu parametrenin değeri teorik olarak  $\gamma \in [0, \infty)$  aralığından herhangi bir değeri alabilir. Optimize edilmesi probleme bağlı olarak değişmektedir. Genellikle 0.1 ve 10 arasında değişebilen değerler alabilmektedir.

Ateş böceği algoritmasının adımları aşağıda özetlenmiş ve akış şeması Şekil 4.2.'de gösterilmiştir.

FA algoritmasının işlem basamakları;

Adım 1 - Her birey gücünü, hareket durumunu ve görsel aralığı rastgele üretilir.

Parçacık hareket durumunu, uzay hızı, konumu ve hız sınırını içerir.

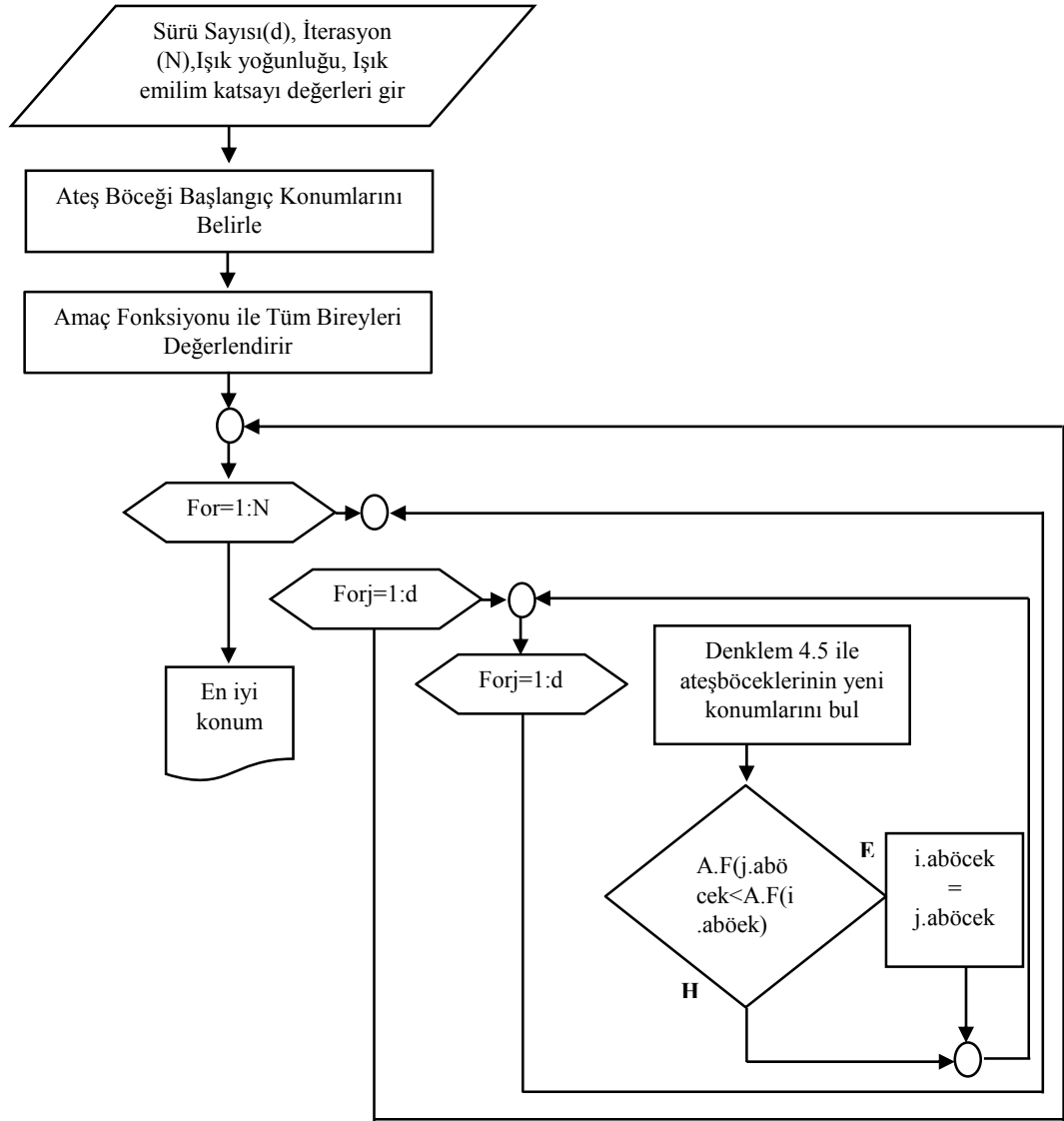
Adım 2 - Çekiciliği hesaplamak için komşuları arar.

Adım 3 - Hareket durumunu belirler ve kendisiyle en çekici olanı karşılaştırır.

Adım 4 - Eğer diğer ateş böceklerinin çekiciliği kendinden daha iyi ise hareket durumunu değiştirir.

Adım 5 – Eğer diğer ateş böceklerinin kendisinden daha çekici değilse hareket durumunu ve rastgele hareketini korur.

Adım 6 –Sonlandırma şartı sağlanana kadar 2 ve 5 adımlar tekrarlanır.

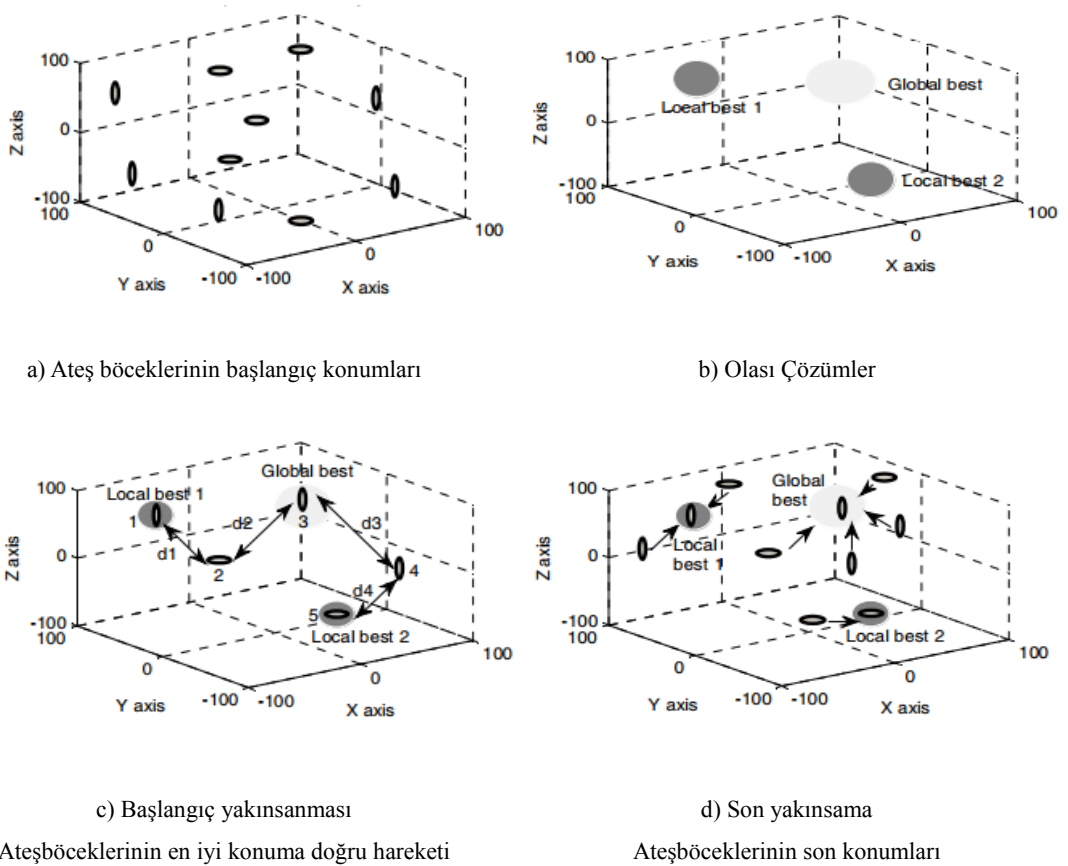


Şekil 4.2. FA akış diyagramı(Yang, 2014)

#### 4.2.2. FA'nın çalışma prensibi

Geleneksel FA'nın çalışma prensibi, üç boyutlu bir optimizasyon problemi üzerinde gösterilmiştir. Ateş böceği sürü sayısı 10 olarak belirlenmiştir. Algoritma başlatıldığında, tüm ateşböcekleri Şekil 4.3(a). 'da gösterildiği gibi arama alanında rastgele olarak dağıtılır. Bu optimizasyon probleminde, arama alanı Şekil 4.3(b). 'de gösterildiği gibi, en iyi iki yerel değer (Lbest) ve en iyi global değer(Gbest) olduğu varsayılmaktadır. İlk arama sırasında, bazı ateş böcekleri, Lbest değerlerine doğru hareket eder bazıları ise Şekil 4.3(c). 'de gösterildiği gibi Gbest değerine hareket

ederek ulaşır. Şekil 4.3(c)'.den ateşböceği 1'in LB1'de, ateş böceği 3'ün global Gbest'de ve ateş böceği 2'ninde Lbest 1 ve Gbest arasında bulunur. Ateş böceği 3 tarafından üretilen ışık yoğunluğu ateş böceği 1'in ürettiği ışık yoğunluğundan daha fazladır. Bu durumda ateş böceği 2 Kartezyen uzaklık (Denklem 4.5) denklemine bağlı olarak LB1 veya GB'ye doğru hareket eder. Bu problemde ateşböceği 1 ile ateşböceği 2 arasındaki mesafe  $d2$  ile karşılaştırıldığında  $d1$  daha az olduğu görülmektedir, bundan dolayı ateş böceği 2 GB noktasına değil de LB1 noktasına doğru hareket eder. Benzer şekilde, ateş böceği 4 ile ateş böceği 5 arasındaki Kartezyen uzaklık ( $d4$ ),  $d3$ 'ten daha kısadır, dolayısıyla ateş böceği 4 muhtemelen LB2'ye doğru çekilir. Arama tekrarlanması artıkça GB'de ateş böcekleri korunur. Yerel en iyi değerdeki ateş böcekleri arasındaki çekicilik sinyali arama yinelemesinde artışıyla üssel olarak azalır ve GB yönünde ilerleyecektir. Son olarak önemli sayıda ateşböceği Şekil 4.3(d)'.de gösterildiği gibi optimizasyon araştırmasının sonunda global en iyi değerde toplanmıştır (Raja ve ark., 2013).

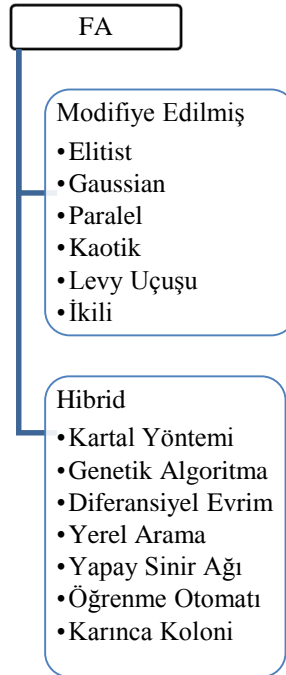


Şekil 4.3. Ateş Böceklerinin hareketlerinin aşamaları(Raja ve ark., 2013)



### 4.2.3. FA'nın türevleri

FA meta-sezgisel algoritmalar arasında verimli algoritmalarından biridir. Son beş yıldır araştırmacılar FA'nın performansını ve yakınsama hızını artırmak amacıyla çalışmaktadırlar. Klasik FA değiştirilmiş ve hibrit olarak ayrılarak analiz edilmektedir(Şekil 4.4.). Klasik FA ağırlıklı olarak sürekli optimizasyon problemlerinde kullanılmaktadır. Çeşitli problem sınıflarını çözerek optimal sonuçlar sağlamak amacıyla FA üzerinde çeşitli değişiklikler ve hibrit algoritmalar geliştirilmiştir. Bu değişiklikler temel olarak Elitist, ikili FA, Gaussian, Levy uçuşları, kaos tabanlı FA ve paralel FA olarak geliştirilmiştir. Diğer yandan, klasik FA diğer algoritmalar ile birleştirilerek hibrit algoritmalar ortaya çıkmıştır. FA ile hibrit algoritma elde edilirken kullanılan algoritmalar Kartal stratejisi, genetik algoritma, diferansiyel evrim, yerel arama, sinir ağı, öğrenme otomatı ve karınca koloni algoritmasıdır.



Şekil 4.4. FA türevleri

Şekil 4.4.'te verilen türevlere ek olarak literatür incelendiğinde FA algoritmasının türevleri olarak Ayrık Ateş Böceği (DFA), Kaotik Ateş Böceği (KFA), Lagranj Ateş Böceği(LFA), Memetik Ateş Böceği(MFA), Çok Amaçlı Ayrık Ateş Böceği(MDFA), Çok Amaçlı Ateş Böceği(MOFA), Çok Amaçlı Ateş Böceği(MOFA), Çok Amaçlı

Geliştirilmiş Ateş Böceği(MOEFA), Hibrit Ateş Böceği(HFA), Yırtıcı Paralel Ateş Böceği(pFAP) gibi algoritmalar ile çalışmalar gerçekleştirilmiştir (Fister ve ark., 2013).

#### 4.2.4. FA'nın avantaj ve dezavantajları

Ateş böceği algoritmasının avantaj ve dezavantajları aşağıda özetlenmiştir.

Avantajları;

- Yüksek yakınsama oranına sahiptir.
- Her ateş böceği bağımsız olarak hareket eder ve mevcut durumuyla kendisine daha iyi konum arar. Böylece yerel optimumdan kaçır ve global bir optimum sayısının daha az yineleme sayısı ile sahip olur.
- Sağlam algoritmadır.
- Paralel uygulamalar için uygun bir algoritmadır.

Dezavantajları;

- Yerel optimum noktalara takılabilir.
- FA parametreleri sabittir ve zamanla değişmezler.
- FA eski konumlarını hatırlayamadıklarından dolayı bu da iyi konumlarından bağımsız olarak hareket etmelerine neden olmaktadır (Yang ve ark., 2013).

### 4.3. Mobil Robotlar için Tarafımızdan Geliştirilen FA Tabanlı Yol Planlama

Bu çalışmada engelli ortamlarda çoklu mobil robotların yol planlama problemini çözmek için FA tabanlı bir yaklaşım geliştirilmiştir. Bu yaklaşımın geliştirilmesinin amacı olarak FA'nın verimliliğini kullanarak yol planlamada istenen performansın

sağlanmasıdır. Ateş böceklerinin başlangıç konumundan daha iyi bir konuma gelme yeteneği kullanılır ve yineleme sayısı en aza indirilerek en kısa/uygun yolu bulmayı kolaylaştırır. Son yıllarda FA, güçlü optimizasyon yeteneğinden dolayı mobil robotların yol planlamasında en kısa ve optimum yol bulmada kullanılmaktadır.

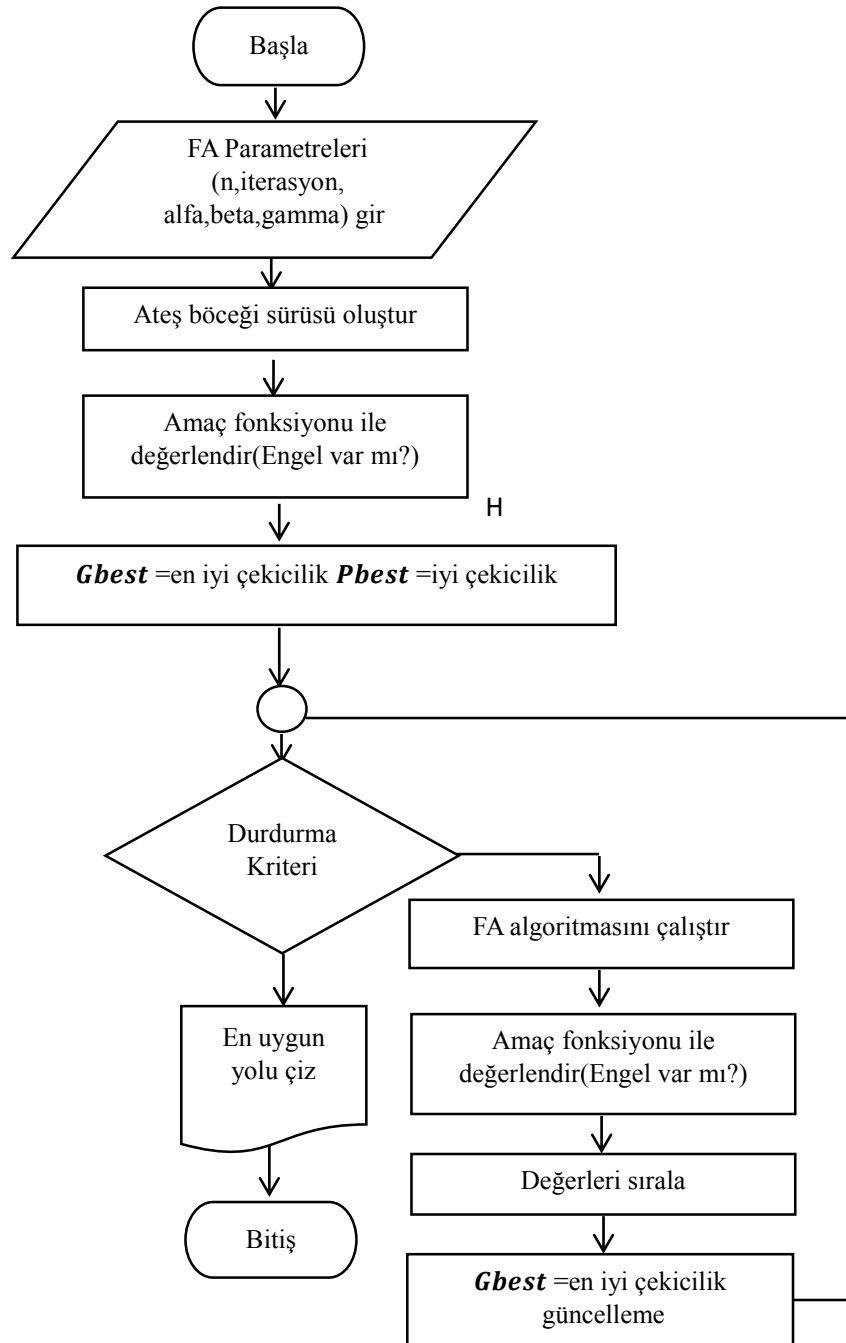
#### 4.3.1. FA tabanlı uygunluk fonksiyonun oluşturulması

FA algoritmasının temelinde bulunan ateş böceklerin parlaklık değerlerine göre hareket etmesi mobil robotun başlangıç ve hedef arasında gideceği yolu bulmada kullanılacaktır. Uygunluk fonksiyonu olarak yol uzunluğu ve yol güvenliği(engellerden kaçınma) optimallik seçeneği olarak belirlenmiştir. Uygunluk fonksiyonu olarak Bölüm 2’de verilen (Denklem 2.3) ve Tablo 2.1.’de bulunan Manhattan, Euclidean ve Diagonal heuristik fonksiyonları belirlenmiştir. Uygunluk fonksiyonunda  $f$  algoritmanın uygunluk fonksiyonu,  $p$  ateşböcekleri ve  $n$  ise popülasyon uzunluğudur. Amaç fonksiyonu en kısa yolu planlamak için seçilen ateşböcekleri arasındaki uzaklıkların toplamı olarak verilmektedir. Elde edilen değerde yolun uzunluğunu vermektedir.

#### 4.3.2. Tarafımızdan geliştirilen FA tabanlı mobil robotlarda global yol planlama

Mobil robotlarda yol planlama için global yol bulma minimizasyon problemine dönüştürülür. Yol planlamada optimizasyon engel tespiti, engelden kaçınma ve optimal yol üretilme gibi problemleri çözmek amacıyla algoritmanın en iyi parametrelerin seçilmesidir. Engellerin bulunduğu statik bir ortamda mobil robot yol planlama probleminde, ateş böceğinin yanıp sönme davranışı en uygun yolun planlanması amacıyla kullanılır. Bu çalışmada FA yaklaşımıyla mobil robot yol optimizasyonunu yol uzunluğu ve engellerden sakınımın verimli bir şekilde sağlanmasıdır. Çalışma ortamının tepesinde bulunan kamera ile çevre bilgilerini elde etmek amacıyla ortamın resmi çekilir. Görüntü işleme tekniği kullanılarak elde edilen görüntüden robot ve engellerin konumu renklerine göre belirlenir. FA kontrolörünü kullanarak mobil robot ve hedefin konumuna göre mobil robotlar için en uygun ve güvenli yol planlaması yapılır. Uygulama sırasında, her yinelemede küresel olarak daha parlak ateş böceğinin konumu seçilmekte ve mobil robot bu konumlara göre

hareket etmektedir. Döngü içinde ilk olarak ateş böceklerinin birbirine olan çekiciliğine bakılmış daha parlak ve çekici olanlara doğru diğerleri hareket etmiştir. Hareket sonrası tekrardan çekicilik ve ışık yoğunluğu hesaplanmıştır. Tüm ateş böcekleri hareket ettirildikten sonra, sonuçlar belirlenen amaç fonksiyonuna göre sıraya konularak, o iterasyon kullanılarak en iyi sonuç belirlenmiştir. Çoklu mobil robotlar için önerilen FA tabanlı yol planlamasının özetlenmiş işlem basamakları ve Şekil 4.5.'te akış diyagramı verilmiştir.



Şekil 4.5. Global Yol Planlama için Geliştirilen FA tabanlı Akış Diyagramı

Yol planlamasının özetlenmiş işlem basamakları;

Adım 1- Robot, hedef ve engellerin konumları belirlenmektedir.

Adım 2- Ateş böceği sürüsünü rastgele olarak oluşturmaktadır.

Adım 3-Robotun hareket edeceği noktaları amaç fonksiyonu ile hesaplanmaktadır.

Bu noktalar üzerinde engel olup olmadığı belirlenecektir.

Adım 4-FA algoritması aktif edilmektedir.

Adım 5-Amaç fonksiyonuna göre sürü içerisinde en çekici olan ateş böceği seçilmektedir.

Adım 6-Mobil robot en çekici ateş böceğinin konumuna doğru hareket etmektedir.

Adım 7- 3-6 arasındaki adımlar engelden sakınarak devam etmektedir.

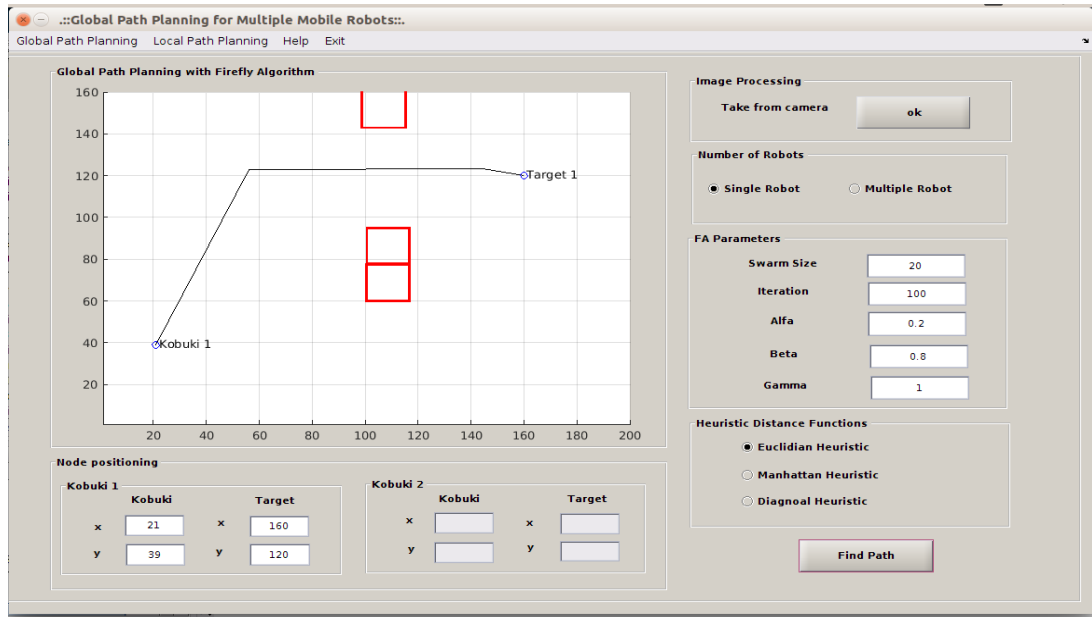
#### 4.4. Simülasyon ve Deneysel Sonuçlar

Çoklu mobil robotlar için geliştirilen farklı çevre ortamlarında FA temelli yol planlamanın doğruluğu ve uygulanabilirliği simülasyon ve deneysel ortamda incelenmiştir. FA etkili olarak kullanılması amacıyla parametrelerin doğru seçilmesi gerekir. FA'nın kontrol parametrelerinin test edilmesinin amacı yakınsama oranını artırarak algoritmanın performansı geliştirmektir. FA'nın verimliliğini en etkin hale getirmek için Tablo 4.2.'de kontrol parametrelerinin kullanıldıkları aralıklar verilmiştir. Bu değerlere göre  $\alpha$ ,  $\beta_0$ ,  $\gamma$ , n parametrelerin etkileri incelenmiştir. Bu parametrelerin test edilme aşamasında bir parametrenin değişimi kontrol edilirken diğer parametreler sabitlenmiştir. Kontrol parametreleri yani  $\alpha$ ,  $\beta_0$ ,  $\gamma$  kararlaştırıldıktan sonra çözümü daha da iyileştirmek için sürü sayısı değiştirilmiştir. Seçilen parametrelere göre algoritmanın duyarlılığı test edilmiş ve önerilen yol planlama problemini çözmek için birlikte kullanılacak en iyi parametrelerin seçimi sağlanmıştır.

Tablo 4.2. FA Kontrol Parametreleri

FA Parametreleri	Başlangıç Değeri	Bitiş Değeri	Artış miktarı
Sürü Sayısı	20	50	10
Alfa( $\alpha$ )	0,1	1	0,1
Başlangıç değeri( $\beta_0$ )	0,1	1	0,1
Işık yoğunluğu( $\gamma$ )	0,9	1	0,1

Test çalışmaları her bir kontrol parametresi için Şekil 4.6. 'de bulunan çevre ortamı için algoritma on kez çalıştırılarak elde edilen yol uzunluğu değerlerinin ortalaması alınarak parametre seçimi gerçekleştirilmiştir. Mobil robot yol planlamada hesaplama karmaşasını en aza indirmek amacıyla gerçekleştirilen optimizasyon test çalışmalarından sonra kontrol parametreleri  $\gamma = 1$ ,  $\alpha = 0.2$  ve  $\beta = 0.8$  belirlenmiştir. FA tarafından elde edilen kontrol parametreleri yol planlama probleminin robotun başlangıç konumundan hedef konumuna kadar olan kısımda kabul edilmiştir. Kontrol parametreleri olan popülasyon büyüklüğü(n) 20, rastgele sayı parameterisi alfa 0.2, çekiciliğin başlangıç değeri 0.8, ışık yoğunluğu ise 1 olarak belirlendikten sonra konum sınırları belirlenerek, sınır değerleri arasında rastgele değerler atanmıştır. Döngü içinde ilk olarak ateş böceklerinin birbirine olan çekiciliğine bakılmış daha parlak ve çekici olanlara doğru diğerleri hareket ettirilmiştir. Hareket sonrası tekrardan çekicilik ve ışık yoğunluğu hesaplanmıştır. Tüm ateş böcekleri hareket ettirildikten sonra, sonuçlar belirlenen amaç fonksiyonuna göre sıraya konularak, o iterasyon için en iyi sonuç belirlenmiştir.



Şekil 4.6. Statik bir ortamda tekli mobil robot için FA tabanlı global yol planlama



Sahne 1



Sahne 2



Sahne 3



Sahne 4



Sahne 5



Sahne 6

Şekil 4.7. FA tabanlı yol planlayıcının gerçek ortamda uygulanması



Sahne 7

Sahne 8

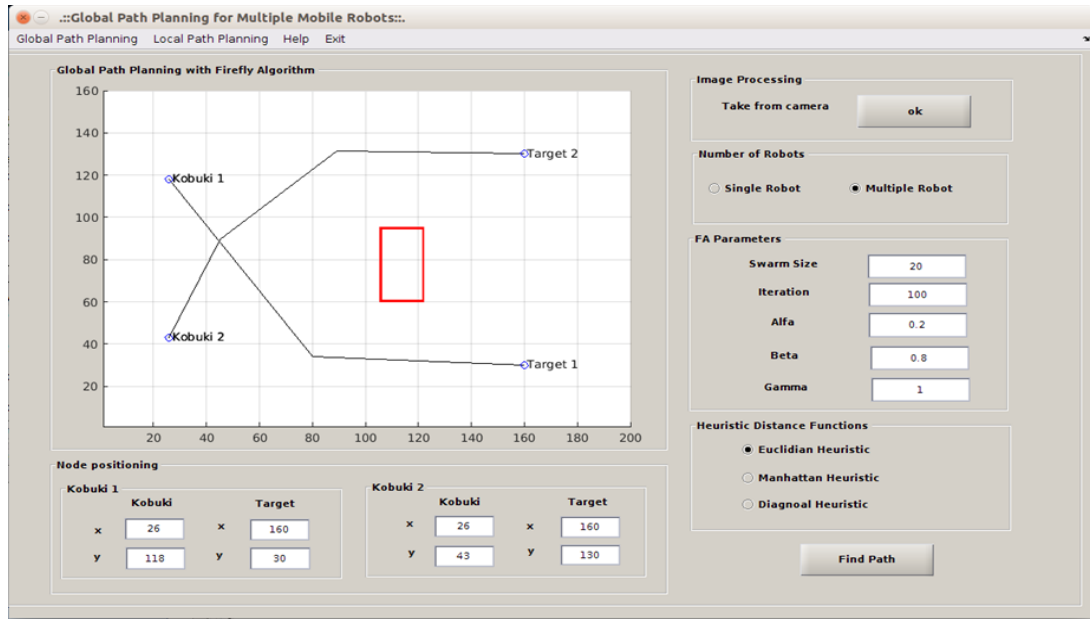
Şekil 4.7. (Devamı)

Tablo 4.3. Şekil 4.6. ve 4.7.'da gösterilen tekli mobil robot için FA tabanlı yol planlayıcıdan elde edilen simülasyon ve deneysel çalışmalardaki yol uzunluklarının karşılaştırılması

Çalışma Sayısı	Simülasyondan elde edilen yol uzunluğu(cm)	Deneysel çalışmadan elde edilen yol uzunluğu(cm)	% Hata
1	195,1882	160,5631	17,73934
2	183,1772	168,0440	8,261509
3	183,4603	155,2540	15,37461
4	190,0984	159,3169	16,1924
5	187,2797	155,3428	17,05305
6	192,4985	153,0581	20,48868
7	184,9489	155,5402	15,90099
8	189,6182	155,9872	17,73617
9	193,1433	152,6494	20,96573
10	191,0592	158,4544	17,06529
<b>Ortalama</b>	<b>189,0470</b>	<b>157,4210</b>	<b>16,6777</b>

Simülasyonda ortamında (Şekil 4.6.) gerçekleştirilen FA tabanlı yol planlayıcının uygulanabilirliğini ve doğruluğunu göstermek için deneysel çalışma(Şekil 4.7.) yapılmıştır. Tablo 4.3.'de simülasyon ve deneysel çalışmalardan elde edilen sonuçlar verilmektedir. Simülasyon ve deneysel çalışma arasında yol uzunluğu performans kriterine göre ortalama hataların yaklaşık olarak % 16 olduğu tespit edilmiştir.





Şekil 4.8. Engelli bir ortamda çoklu mobil robotlar için FA tabanlı global yol planlama

Tablo 4.4. FA tabanlı yol planlayıcı kullanılarak gerçekleştirilen simülasyon (Şekil 4.8.) ve deneysel sonuçlardan elde edilen yol uzunluklarının karşılaştırılması

Çalışma Sayısı	Robot Adı	Simülasyondan elde edilen yol uzunluğu(cm)	Deneysel çalışmadan elde edilen yol uzunluğu(cm)	%	
				Kobuki 1 Hata	Kobuki 2 Hata
1	Kobuki 1	179,9177	154,3000	15,44	13,24
	Kobuki 2	176,9551	203,6634		
2	Kobuki 1	180,9250	156,7516		
	Kobuki 2	176,9252	202,8305		
3	Kobuki 1	180,5872	155,0226		
	Kobuki 2	185,2608	207,0430		
4	Kobuki 1	179,9191	152,7167		
	Kobuki 2	179,9564	202,4833		
5	Kobuki 1	198,2531	164,2201		
	Kobuki 2	176,8004	199,9003		
6	Kobuki 1	180,0551	152,5935		
	Kobuki 2	177,0340	201,0898		
7	Kobuki 1	185,3176	157,6232		
	Kobuki 2	176,8814	201,5494		
8	Kobuki 1	179,9482	152,1097		
	Kobuki 2	183,8080	204,2482		

Tablo 4.4.(Devamı)

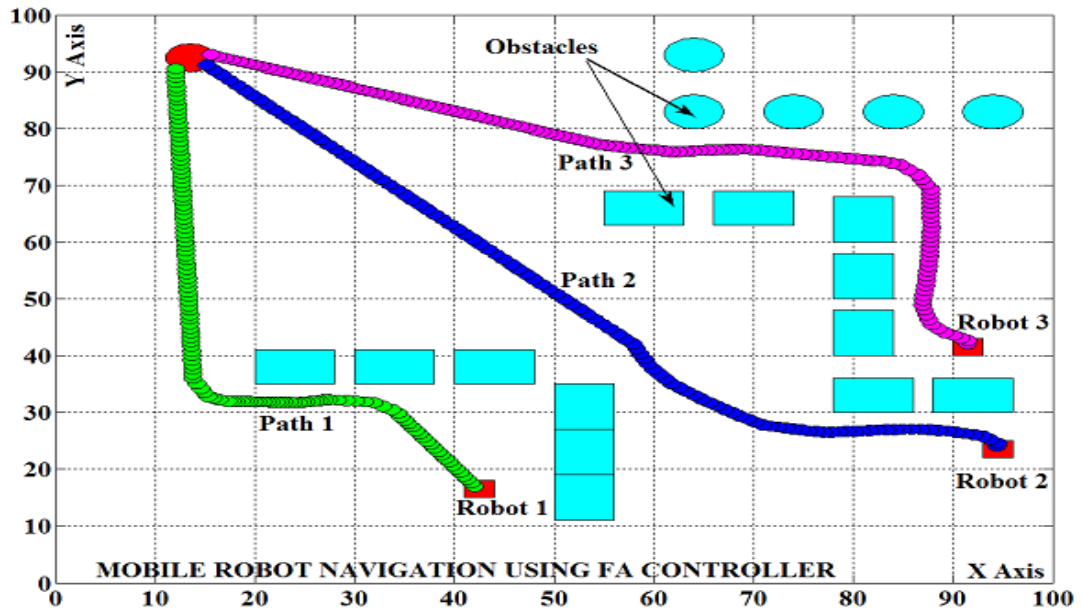
Çalışma Sayısı	Robot Adı	Simülasyondan	Deneysel çalışmadan	%	%
		elde edilen yol uzunluğu(cm)	elde edilen yol uzunluğu(cm)	Kobuki 1 Hata	Kobuki 2 Hata
<b>9</b>	Kobuki 1	182,8142	152,2471		
	Kobuki 2	177,0508	200,0020		
<b>10</b>	Kobuki 1	184,9329	152,0467		
	Kobuki 2	176,8387	201,4557		
<b>Ort</b>	<b>Kobuki 1</b>	<b>183,2670</b>	<b>154,9631</b>		
	<b>Kobuki 2</b>	<b>178,7511</b>	<b>202,4266</b>		

Tablo 4.4.'deki sonuçlara göre Önerilen FA tabanlı çoklu robot yol planlamasında simülasyon ve deneysel çalışma arasındaki fark Kobuki robot 1 için %15,44 daha kısa iken kobuki robot 2 için ise %13.23 yol uzunluğu deneysel çalışmada daha uzun bir şekilde gerçekleşmiştir. Daha kısa olan yol daha kısa sürede ve daha az enerjiyle yol planlaması sağlamıştır.

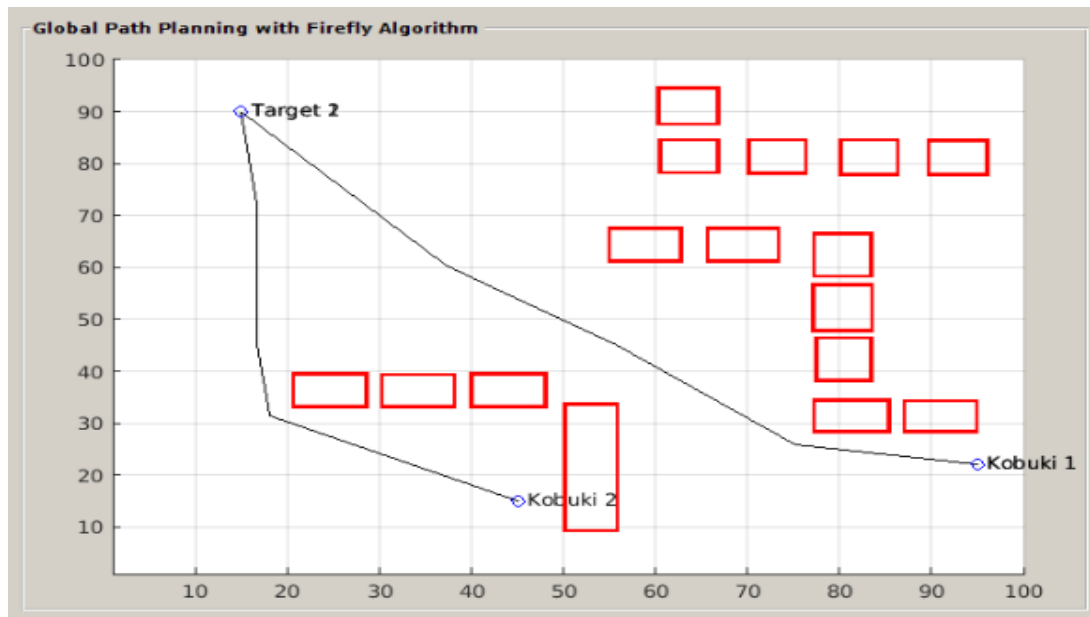
#### 4.5. Diğer Yapılan Çalışmalarla Karşılaştırma

Geliştirilen FA tabanlı global yol planlamanın performansını test etmek amacıyla literatürde var olan diğer zeki tabanlı yol planlamalarla karşılaştırılmıştır. Aynı başlangıç ve hedef noktaları kullanılarak benzer simülasyon ortamında yapılan karşılaştırmalarda performans kriteri olarak yol uzunluk parametresi seçilmiştir. Tekli ve çoklu mobil robotlar için geliştirilen FA tabanlı yol planlamayı doğrulamak için 2016 yılında Patle ve ark. tarafından yapılan çalışmadaki FA tabanlı global yol planlama kullanılmıştır (Patle ve ark., 2016).





Şekil 4.11. Diğer FA tabanlı çoklu mobil robotlar için global yol planlaması (Patle ve ark., 2016)



Şekil 4.12. Bu çalışmada tarafımızdan geliştirilen FA tabanlı çoklu mobil robotlar için global yol planlaması

Tablo 4.5. Tekli mobil robot için simülasyon sonuçlarının yol uzunluğuna göre karşılaştırılması

Robot	Önerilen FA kullanılarak elde edilen simülasyon yol uzunluğu("cm")	FA kullanılarak elde edilen simülasyon yol uzunluğu("cm")	% Performans Farkı
<b>Kobuki</b>	109,2416 (Şekil 4.9.)	180,20 (Şekil 4.10.)	39,38

Tablo 4.6. Çoklu mobil robotlar için simülasyon sonuçlarının yol uzunluğuna göre karşılaştırılması

<b>Senaryo 1</b>	<b>Önerilen FA kullanılarak elde edilen simülasyon yol uzunluğu("cm")</b>	<b>FA kullanılarak elde edilen simülasyon yol uzunluğu("cm")</b>	<b>% Performans Farkı</b>
Kobuki 1	90,3556 (Şekil 4.12.)	154 (Şekil 4.11.)	41,33
Kobuki 2	108,5143 (Şekil 4.12.)	219,8 (Şekil 4.11.)	50,6

Tablo 4.5.-6.'da çalışmada geliştirilen FA ve araştırmacıların geliştirdiği FA yol uzunluğuna göre karşılaştırılmasını göstermektedir. Sonuçlara göre tekli robot yol planlamasında %39,38, çoklu robot yol planlamasında ise robot 1 için %41,33 robot 2 için ise de %50,6 daha kısa yol planlaması gerçekleştirilmiştir. Bu sonuçlara göre önerilen FA tabanlı global yol planlayıcısının mobil robot yol planlama probleminin çözümünde etkili bir şekilde geliştirildiği görülmektedir.

## **BÖLÜM 5. ÇOKLU MOBİL ROBOTLAR İÇİN GUGUK KUŞU ALGORİTMASI TABANLI GLOBAL YOL PLANLAMA**

### **5.1. Giriş**

Cuckoo Search Algoritması(CS) algoritması, meta-sezgisel evrimsel hesaplama tekniklerinden olan yeni bir optimizasyon kavramıdır. CS, bilimsel uygulamalar ve mühendislik optimizasyon problemlerinde uygulanabilen zeki tabanlı bir algoritmadır. CS algoritmaları mühendislik uygulamaları, görüntü işleme, tahmin, sınıflandırmalar, güç sistemleri, özellik seçimi, ısı alış verişi, robotik gibi alanlarda uygulamaları hızlı bir şekilde artmaktadır (Fister ve ark., 2014). Ayrıca CS, NP-zor tümeleşik optimizasyon problemlerinin çözümüne de uyarlanabilir.

CS guguk kuşunun çoğalma stratejilerinden faydalanılarak Xin-She ve Suash Deb tarafından geliştirilmiş bir algoritmadır. Guguk kuşlarının kuluçka parazitliğinden(brood parasitism) esinlenilerek ve yumurtalarını diğer konak kuşların yuvalarına yerleştirmesinden ilham alarak geliştirmişlerdir (Yang, 2009). CS algoritmasının başarı sebebi yerel ve global arama arasında kurduğu önemli denge ve algoritmanın çalışması için daha az parametrenin gerekli olmasıdır. CS algoritmasının global araması sonsuz ortalama ve varyansla birlikte standart olmayan rassal dağılımlı Levy uçuşlarını kullanmaktadır. Bu nedenle kaynakların ayrık olduğu noktalarda CS arama alanının araştırılması doğaya daha yakındır. Bundan dolayı CS algoritmasında global arama sağlanabilmektedir (AH ve ark., 2013).

Bu çalışmada mobil robotların yol planlaması gerçekleştirileceği için literatürde CS algoritmasının kullanıldığı mobil robot yol planlama problemini çözen çalışmalar incelemiştir. 2013 yılında Parhi ve ark. tarafından yapılan çalışmada mobil robotlarda yol planlama problemini çözmek için CS algoritması kullanılmıştır. Bu algoritma

bazı guguk kuşlarının parazit davranışı ve diğer konukçu kuşların yuvasına yumurtalarını bırakmasından esinlenerek gerçekleştirilmiştir. Çevrede bulunan robot, hedef ve engeller arasında yeni bir objektif bir fonksiyon geliştirilmiştir. Sürüdeki her yuvanın hedef fonksiyon değerine bağlı olarak robot engellerden kaçınarak hedefe doğru ilerlemiştir. Statik engellerin bulunduğu çeşitli ortamlarda önerilen algoritmanın etkinliğini doğrulanmıştır. Algoritmanın etkinliği simülasyon sonuçları ile karşılaştırılmalı olarak gösterilmiştir. Simülasyon sonuçlarına ek olarak CS algoritmasının parametrelerinin değişiminin yol uzunluğuna etkisi gösterilmiştir. Simülasyon sonuçlarına göre CS algoritması diğer meta-sezgisel GA ve PSO algoritmalarıyla karşılaştırıldığında daha verimli bir yol planlama algoritmasını sağlandıklarını belirtmişlerdir (Parhi ve ark., 2013). 2014 yılında Parhi ve arkadaşlarının diğer bir çalışmada ise seyrüsefer probleminde robot, hedef ve engeller arasında yeni bir hedef tabanlı optimize edilebilen formül geliştirmişlerdir. Robot hedefe doğru yol alırken eğer önünde engel yoksa euclidiean heuristik fonksiyonu kullanarak en kısa yoldan hedefe ulaşır. Eğer yolda engel varsa CS algoritmasıyla engelden kaçınarak yol optimize edilir. Geliştirilen algoritmanın etkinliği ve uygunluğunu göstermek için literatürde bulunan çalışmalarla simülasyon ortamı karşılaştırmalı olarak sunulmuştur. Önerilen sistemi doğrulamak amacıyla simülasyon ve deneysel çalışma sonuçları kıyaslanmıştır (Parhi ve ark., 2014). Tablo 5.1.'de literatür çalışmalarında CS algoritma tabanlı mobil robot yol planlamalarının özet hali verilmiştir.

Tablo 5.1. Mobil robot yol planlamada CS algoritmasının uygulanması

Yazarlar ve Yıl	Kin Model	Engel Şekli	Statik Engel	Dinami k Engel	Statik Hedef	Dinamik Hedef	Sim	Gerçek Sistem
P. K. Mohanty D. R. Parhi (2013)	H	Yuvarlak	E	H	E	H	E	H
P. K. Mohanty D. R. Parhi (2014)	H	Dörtgen	E	H	E	H	E	E

Robotik alanda mobil robot navigasyonunda algoritmanın uygulanması ve geliştirilmesi önemli bir yer tutmaktadır. Bu çalışmada engellerle dolu statik bir çevrede çoklu mobil robotların navigasyon problemlerini çözmek için CS algoritma tabanlı bir yaklaşım geliştirilmiştir.

## 5.2. CS Algoritması için Matematiksel Modelleme

CS algoritmasında guguk kuşları başka yuvalara uçarak, oradaki yuvaya yumurtlamaktadır. Yumurtladıkları yuvada/konakta, yumurtadan çıkma olasılığı yüksek olan konağın yumurtalarını da atarlar. Asıl yuvanın sahibi geldiğinde yabancı yumurtayı fark ederse yabancı yumurtayı ya aşağı atar ya da yuvayı terk eder. Eğer tanımazsa, o yumurtaların üzerine oturarak kuluçka asalaklığına maruz kalır. Eğer yumurtayı tanırsa, yumurtayı yuvadan aşağı atabilir veya yuvayı terk ederek kendisine yeni bir yuva inşa edebilir. CS algoritmasında karşılaşılan bu durumlar Yang ve Deb tarafından kurallaştırılmıştır (Yang ve Deb, 2009).

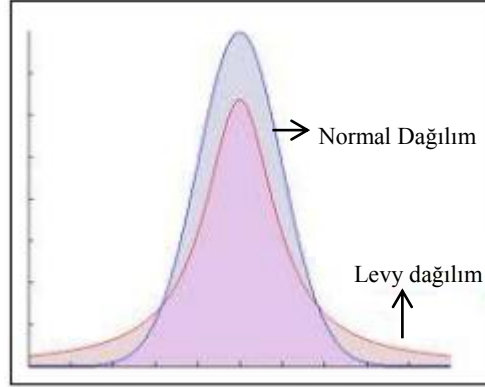
- Bir guguk kuşu rastgele seçtiği bir yuvaya bir tane yumurta bırakabilir.
- Yuva içerisindeki kaliteli yumurta bir sonraki jenerasyona aktarılmaktadır.
- Yuvanın sahibi guguk kuşu, yuvaya geri döndüğünde bırakılmış olan yabancı yumurtayı belirli bir  $p_a$  (0,1) olasılığı ile tanıyabilmektedir. Eğer ev sahibi guguk kuşu bırakılmış olan yabancı yumurtayı tanırsa ya kendine yeni yuva kurmak üzere yuvayı terk edecek ya da yabancı yumurtayı aşağı atacaktır.

CS algoritması, global ve yerel rassal yürüyüşün dengelenerek kullanıldığı bir algoritmadır. Algoritmanın dengelenme işlemi global ve yerel rassal yürüyüş parametreleri ile sağlanmaktadır.

CS algoritmasında yeni nesiller üretebilmek için Şekil 5.1.'de görüldüğü gibi Levy dağılımı kullanılmaktadır. Problemin çeşidine ve büyüklüğüne bağlı olarak Levy uçuşları ölçeklendirilmektedir. (Denklem 5.2) görüldüğü gibi bir sonraki konum şuanki konuma ve geçiş olasılıklarına bağlı olduğu görülmektedir. En iyi çözümün



yerel noktalarda kalmaması için yeni çözüm kümelerinin o anki en iyi çözümden rassal bir şekilde uzaklaşması gerekmektedir.



Şekil 5.1. Levy uçuş Dağılımı

CS algoritmasında global rassal yürüyüş Levy uçuşuyla gerçekleştirilmektedir. (Denklem 5.1) verilen Levy uçuşunun etkisi üretilen rastgele değer olan  $\alpha$  gibi bir değişkenle belirlenmektedir. Bir sonraki işlem olarak eski konum bilgisi ile toplanarak guduk kuşlarının yeni konumlarının bulunması sağlanmaktadır. She'nin önermiş olduğu Levy Flight kontrol parametresi  $\lambda$ ' dır. Global rassal yürüyüşü sağlayan  $\lambda$  parametresi genellikle 1 ile 3 arasında değer almaktadır (Yang ve Deb, 2009).

$$x_i^{t+1} = x_i^t + \alpha \otimes Levy(\lambda) \quad (5.1)$$

$$L = \frac{\lambda \Gamma(\lambda) \sin \frac{\pi \lambda}{2}}{\pi} \frac{1}{s^{1+\lambda}}, \quad s \gg s_0 > 0$$

Yerel aramayı gerçekleştiren yerel rassal yürüyüş (Denklem 5.2) ile gerçekleştirilmiştir. Bu denklemde bulunan rastgele değer olan  $\alpha$  (Denklem 5.1) ile aynı değerleri göstermektedir. Adım uzunluğu  $s$  ile belirlenmiştir ve değerinin denklem üzerinde kritik değere sahiptir.  $s$  değeri büyük bir değer seçilirse yakınsama gerçekleşmemektedir.  $s$  değeri küçük bir değer seçilirse çok fazla iterasyon gerektirmektedir (Karagül, 2014).

$H(u)$  adım fonksiyonu olup,  $\epsilon$  parametresi ise gauss dağılımdan gelen rassal reel bir sayıdır.  $x_j^t$  ve  $x_k^t$  ise çözüm uzayındaki vektörlerinin birbiri ile rastgele değişmesini sağlayan permütasyon değerleridir.

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \quad (5.2)$$

CS algoritmasının sahte (pseudo) kodu Şekil 5.2.'de gösterilmektedir.

```

Amaç fonksiyonu belirlenir  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ .
n adet konak yuvası olan başlangıç popülasyonu üret
while (durdurma kriteri)
  Levy uçuşu ile rassal guguk kuşu al
   $f_i$  maliyetini bul
  n yuva içinden rassal yuva seç
  if ( $f_i < ff$ ),
    j'yi yeni çözümle değiştir.
  end
En kötü yuvaların  $p_a$  oranı kadarını terk et ve yenileri inşa et
İyi Çözümleri tut
Çözümleri sırala ve o anki en iyi çözümü bul
end while
Sonuçlar görüntülenir.

```

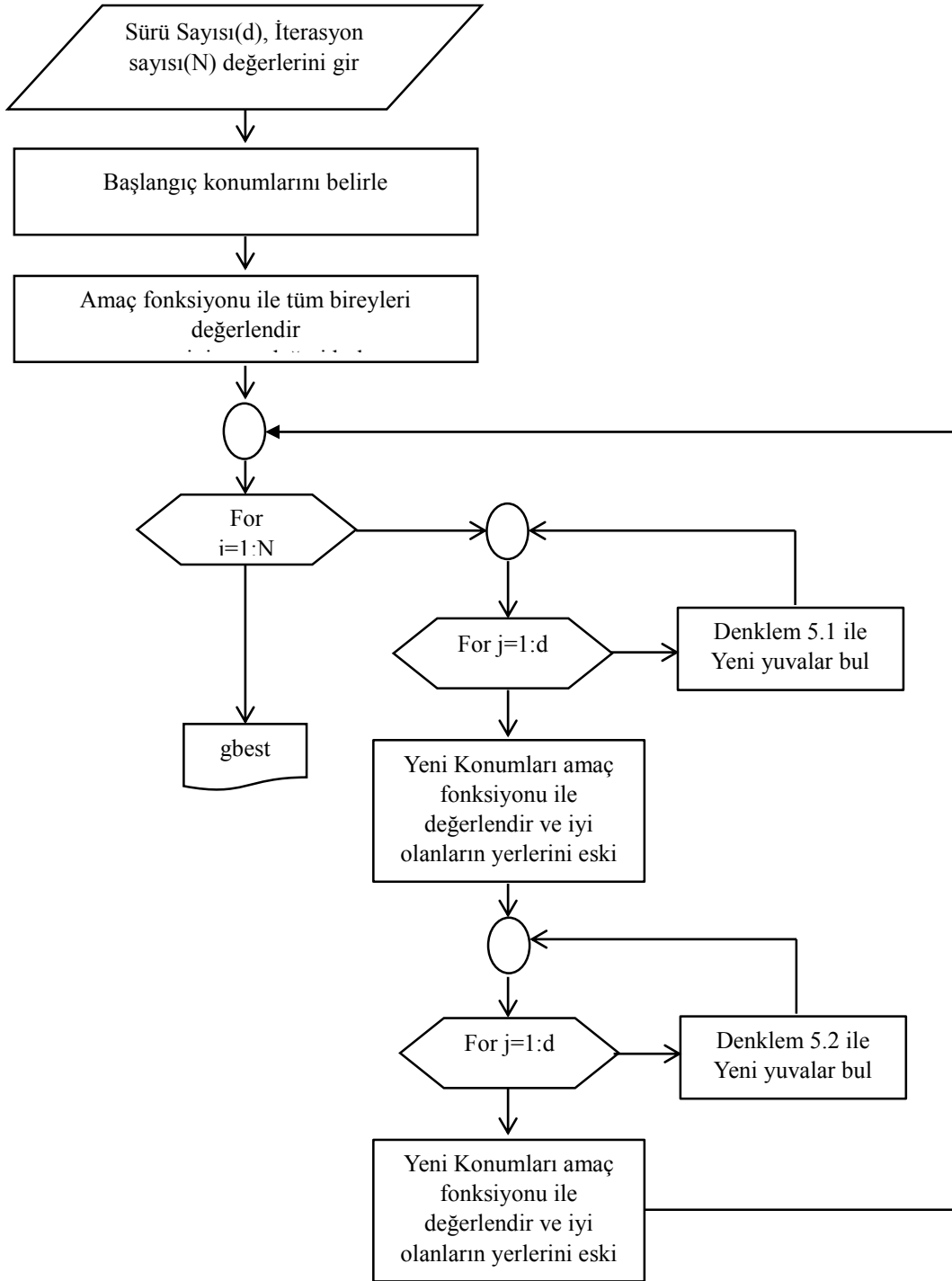
Şekil 5.2. CS algoritma sözde kodu

CS algoritmasının avantajlarından biri de diğer meta-sezgisel algoritmalara göre daha az parametre kontrolü olmasıdır. Tablo 5.2.'de CS algoritmasının parametre değer aralıkları ve genellikle kullanılan değerler özetlenmiştir (Shehaba ve ark., 2017).

Tablo 5.2. CS algoritmasının parametreleri(Shehaba ve ark., 2017)

Parametre	Sembol	Aralık	Genellikle Kullanılan
<b>Yuva Sayısı</b>	N	[15,50]	N=15
<b>Oran</b>	$p_a$	[0,1]	$p_a = 0,25$
<b>Adım Boyutu</b>	$\alpha$	$\alpha > 1$	$\alpha = 1$

CS algoritmasının akış diyagramı şekil 5.3'de verilmiştir.



Şekil 5.3. CS algoritmasının akış diyagramı (Yang, 2009)

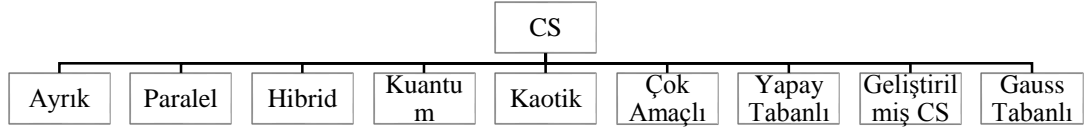
CS algoritmasının prensipleri Yang tarafından özetlendiğinde algoritma aşağıdaki üç bileşene bağlı olarak gelişir.

- Bütünleşmiş aramalar etkin çözüm sağlaması için Levy uçuşları kullanılmaktadır.

- Yerel aramada rastgele yürüyüşler kullanılmaktadır.
- En iyi çözümleri seçmek ve korumak, en iyi konum değerlerine erişmek için uygulanan yöntem genetik algoritmadaki işleyişle aynıdır.

### 5.2.1. CS algoritmasının çeşitleri

CS algoritması geliştirildikten itibaren çeşitli alanlarda kullanılmaya başlanmıştır. CS algoritmasının gelişmesiyle birlikte çeşitleri ortaya çıkmıştır. Şekil 5.4.'de CS algoritmasının çeşitleri tablo halinde gösterilmiştir.



Şekil 5.4. CS algoritmasının çeşitleri (Fister ve ark., 2014)

### 5.2.2. CS algoritmasının avantaj ve dezavantajları

CS algoritmasının genel yapısı incelendikten sonra avantaj ve dezavantajları aşağıda özetlenmiştir.

Avantajları;

- Global yakınsama özelliği gereksinimleri karşılar.
- Yerel ve global arama özelliklerini desteklemektedir.
- Global arama özelliğinde Lévy uçuşlarını küresel bir arama stratejisi olarak kullanmaktadır (Wang ve ark., 2012).
- CS algoritması diğer optimizasyon algoritmalarıyla karşılaştırıldığında daha basit, daha az parametreye sahiptir. Aynı zamanda diğer optimizasyon algoritmaları ile hibritleştirme işlemi daha kolaydır.

Dezavantajları;

- CS algoritmasında bulunan kontrol parametre değerlerinin sabit tutulması algoritmanın verimliliğini azaltmaktadır.
- CS algoritması matematiksel analiz yönünden zayıftır. Bu sebepten dolayı algoritmanın nasıl çalıştığının anlaşılması zordur (Shehaba ve ark., 2017).

### 5.3. Mobil Robotlar İçin Geliştirilen CS Algoritma Tabanlı Yol Planlama

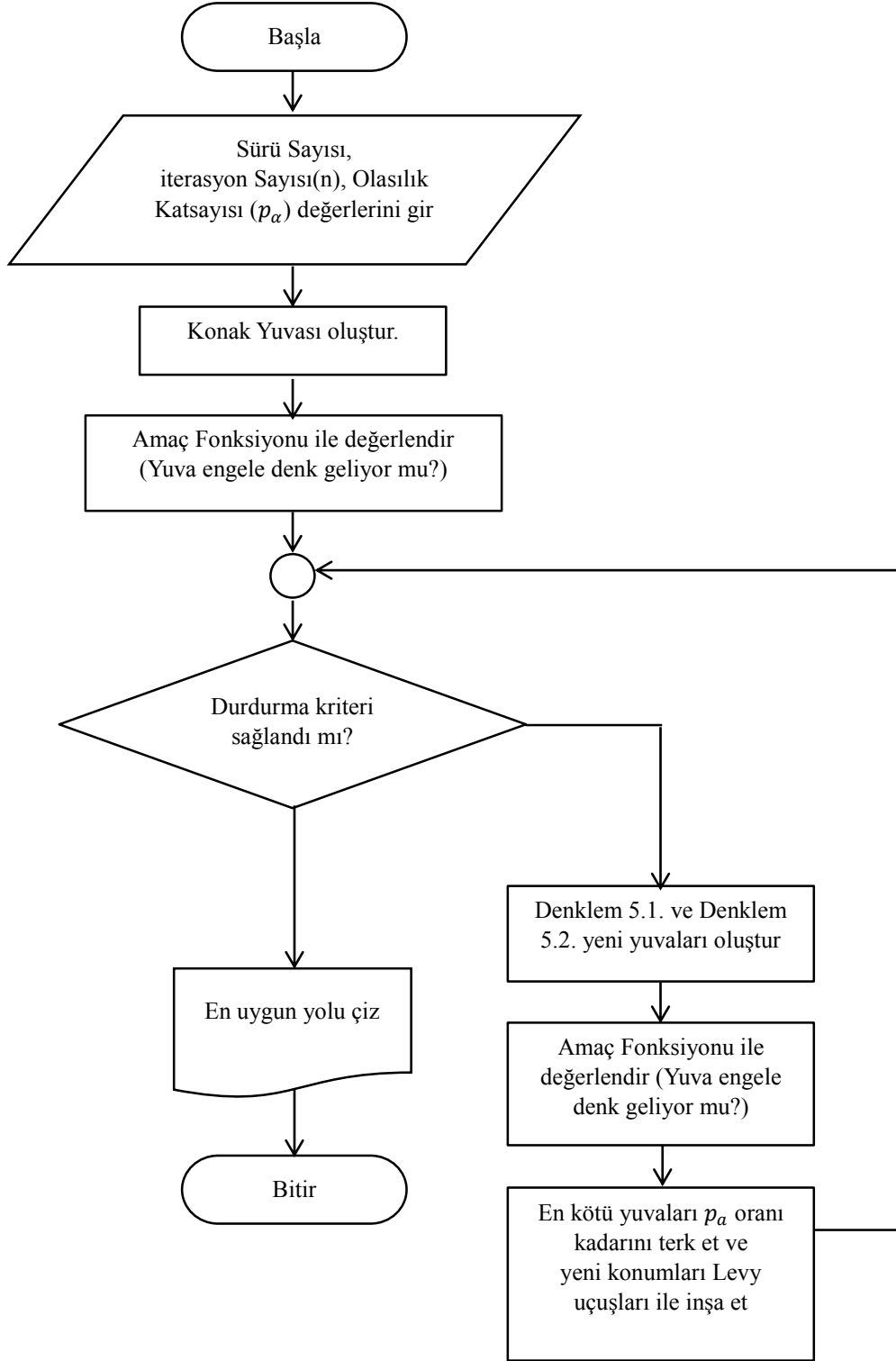
En kısa / en uygun yol planlaması otonom araçların verimli çalışması için gereklidir. Bu bölümde, çeşitli statik engeller bulunan, bilinmeyen veya kısmen bilinen bir ortamda mobil robot yol planlaması için yeni bir doğadan ilham alan meta-sezgisel CS algoritması uygulanmıştır. Robot navigasyon alanındaki temel sorunlardan biri yol planlama problemidir. Bu bölümde ele alınan sorunun amacı, çeşitli statik engeller tarafından doldurulan bir ortamda robot yolunun planlamasıdır.

#### 5.3.1. Geliştirilen CS algoritma tabanlı uygunluk fonksiyonunun oluşturulması

Uygunluk fonksiyonu, engellerden kaçınma ve mobil robotun en kısa yolu arama davranışını gerçekleştirmek için kullanılır. Guguk kuşu sürüsünde bulunan her yuvanın uygunluk fonksiyonuna bağlı olarak robot engellerden kaçınarak hedefe doğru ilerlemektedir. Her yinelemenin sonucunda robotun izlediği konumlar elde en iyi yuvalardır. CS algoritması uygulanırken uygunluk fonksiyonu olarak Bölüm 2’de açıklanan (Denklem 2.3) ve Tablo 2.1.’de bulunan Manhattan, Euclidean ve Diagonal heuristik fonksiyonları kullanılmaktadır. Uygunluk fonksiyonunda  $f$  algoritmadaki uygunluk fonksiyonu,  $p$  yuvalar ve  $n$  ise popülasyon uzunluğudur. Amaç fonksiyonu en kısa yolu planlamak için en iyi yuvaların arasındaki uzaklıkların toplamı olarak bulunmaktadır.

### 5.3.2. Geliştirilen CS algoritma tabanlı mobil robotlarda global yol planlama

Çoklu mobil robotlar için önerilen CS tabanlı yol planlamasının özetlenmiş işlem basamakları ve Şekil 5.5.'te akış diyagramı verilmiştir.



Şekil 5.5. Global Yol Planlama için Geliştirilen CS tabanlı Akış Diyagramı

Mobil robot yol planlama için CS algoritma uygulama basamakları;

Adım 1: Robotun başlangıç ve hedef noktaları belirlenir.

Adım 2: Başlangıç sürüsü konak yuvaları ile oluşturulur.

Adım 3-Robotun hareket edeceği noktalar amaç fonksiyonu ile hesaplanır. Bu noktalar üzerinde engel olup olmadığı belirlenir.

Adım 4-CS algoritması aktif edilir.

Adım 5- Amaç fonksiyonuna göre mevcut en iyi yuva seçilir.

Adım 6- Kötü yuvalar terk edilir ve Levy uçuşuyla yeni yuvayı inşa edilir.

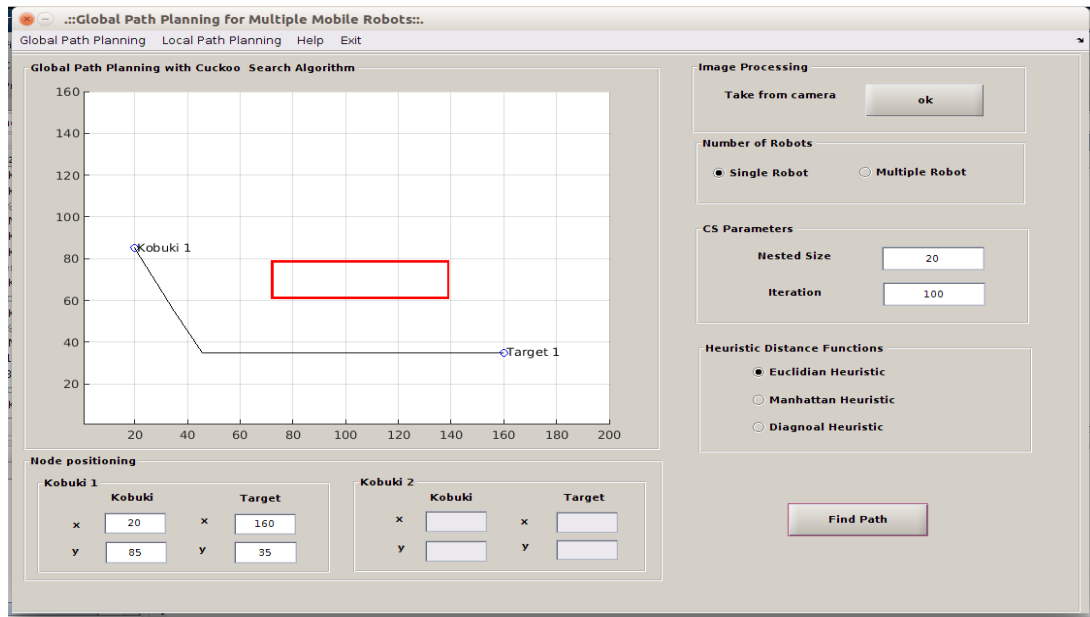
Adım 7-Mobil robot en iyi yuvaya doğru hareket eder.

Adım 8- 3-6 arasındaki adımlar engelden sakınarak devam eder.

Adım 9-Hedefe ulaşıldı.

#### **5.4. Simülasyon ve Deneysel Sonuçlar**

Önerilen CS tabanlı yol planlayıcı ile aynı çevre ortamları için simülasyon ve deneysel çalışmalar yapılarak yol uzunluğuna göre karşılaştırma yapılmaktadır. CS algoritmasının yol planlama problemine aktarılması aşağıda açıklanmaktadır. CS algoritmasının parametreleri diğer algoritmalarla aynı olması için popülasyon büyüklüğü(n) 20,  $p_a$  parametresi 0,25 olarak belirlendikten sonra amaç fonksiyonu belirlenmektedir. Konak yuvaların başlangıç değerleri robotun gezinebileceği alanın sınır değerleri arasında rastgele olarak belirlenmektedir. Her bir yuva diğer bir deyişle yuvada bulunan yumurtalar(robot gezindiği konumlar) için uygunluk değerleri hesaplanmaktadır. CS algoritmasının global arama özelliği Levy uçuşu (Denklem 5.1) ile birleştirilerek gerçekleştirilir. Buradan gelen yeni çözümlerin uygunluk değerleri hesaplanarak, bulunan çözümlerde kötü olanlarının yerleri değiştirilmektedir. Global arama uygulandıktan sonra lokal arama adımı Denklem 5.2 kullanılmaktadır. Lokal arama sonucundan elde edilen çözümlerin uygunluk değerleri bulunarak, en iyi uygun değere sahip olan yuva ile robotun konum yerleri değiştirilmektedir.



Şekil 5.6. Tekli mobil robot için CS tabanlı global yol planlama



Sahne 1



Sahne 2



Sahne 3



Sahne 4

Şekil 5.7. Gerçek ortamda yerel yol planlayıcının uygulanması





Sahne 5



Sahne 6



Sahne 7

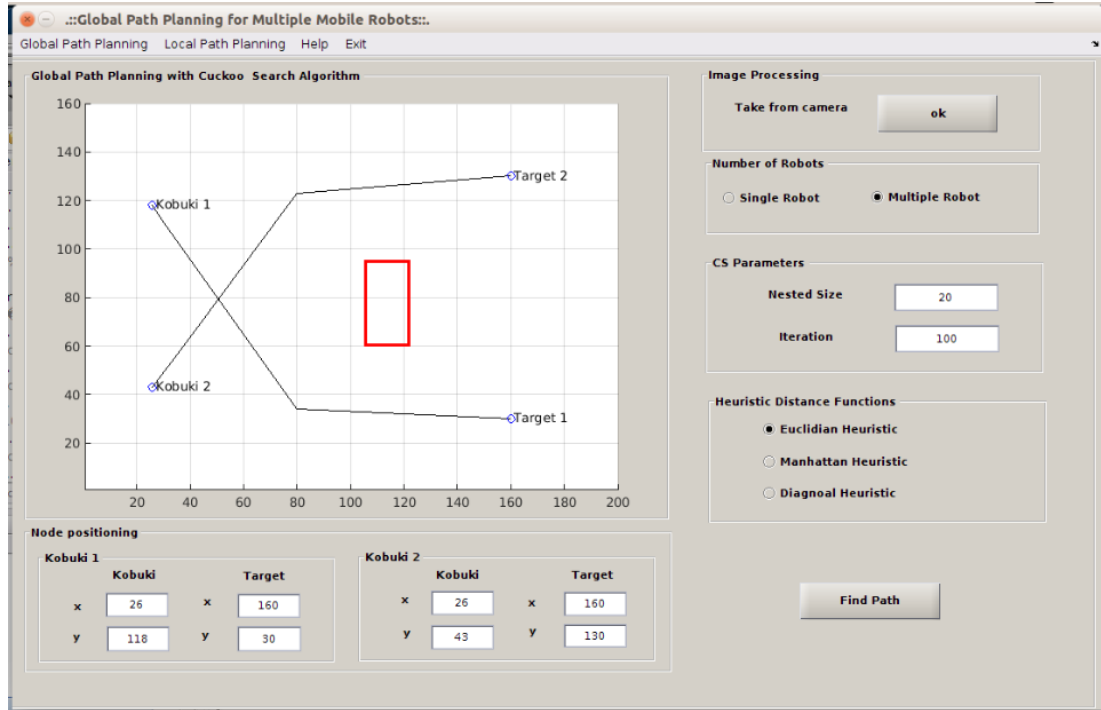


Sahne 8

Şekil 5.7. (Devamı)

Tablo 5.3. Şekil 5.6. ve 5.7.'de gösterilen mobil robotun CS tabanlı global yol planlayıcı için simülasyon ve deneysel çalışmadan elde edilen yol uzunluklarının karşılaştırılması

Çalışma Sayısı	Simülasyondan elde edilen yol uzunluğu(cm)	Deneysel çalışmadan elde edilen yol uzunluğu(cm)	% Hata
1	170,5281	138,2421	18,9329
2	170,4433	138,2025	18,9158
3	170,4256	144,1741	15,4035
4	170,2833	138,9537	18,3985
5	170,1147	137,7324	19,0355
6	170,3017	136,0943	20,0863
7	170,4545	135,8119	20,3236
8	170,2941	136,9132	19,6019
9	170,0794	138,7475	18,4219
10	170,4286	138,6977	18,6183
<b>Ortalama</b>	<b>170,3353</b>	<b>138,3569</b>	<b>18,7738</b>



Şekil 5.8. Engelli statik bir ortamda çoklu mobil robot için CS tabanlı global yol planlama

Tablo 5.4. Şekil 5.8.'de gösterilen çoklu mobil robotlar için CS tabanlı yol planlayıcının bulunduğu simülasyon analiz ve deneysel çalışmalardan elde edilen yol uzunluklarının karşılaştırılması

Çalışma Sayısı	Robot Adı	Simülasyondan	Deneysel çalışmadan	%	%
		elde edilen yol uzunluğu(cm)	elde edilen yol uzunluğu(cm)	Kobuki 1 Hata	Kobuki 2 Hata
1	Kobuki 1	179,9773	152,1332	15,29	13,49
	Kobuki 2	176,9257	200,6209		
2	Kobuki 1	180,3350	152,1691		
	Kobuki 2	176,6218	201,4124		
3	Kobuki 1	180,0815	152,9175		
	Kobuki 2	177,0976	201,7393		
4	Kobuki 1	180,0666	152,4116		
	Kobuki 2	176,8434	201,2775		
5	Kobuki 1	180,1298	152,7473		
	Kobuki 2	176,9913	201,1100		
6	Kobuki 1	180,1146	152,0157		
	Kobuki 2	177,1778	199,8918		
7	Kobuki 1	180,1697	152,1438		
	Kobuki 2	176,9991	200,5955		
8	Kobuki 1	180,0952	151,8982		
	Kobuki 2	176,9126	200,8982		

(Tablo 5.4. Devamı)

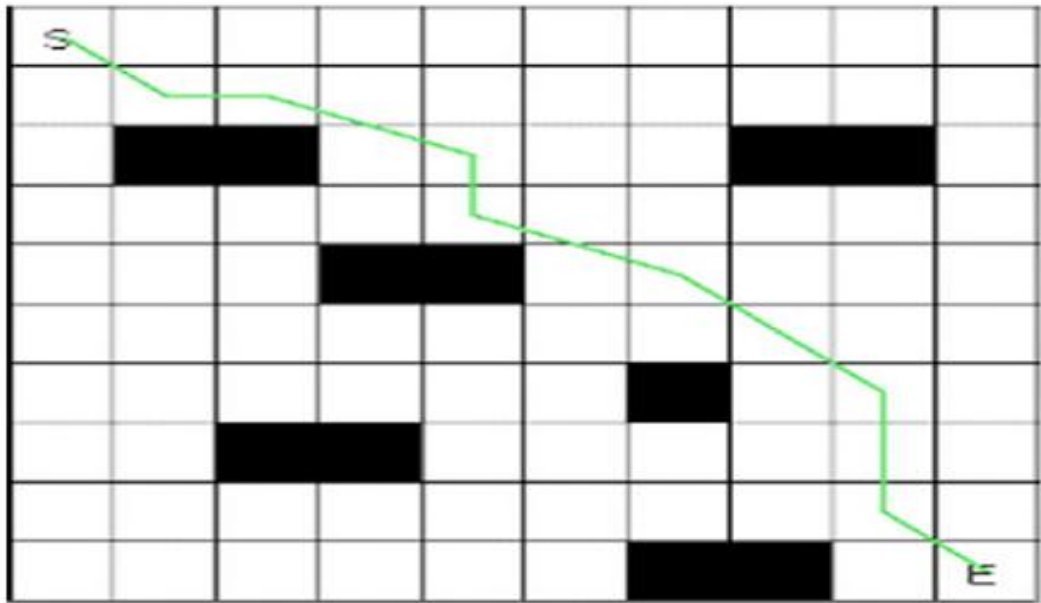
Çalışma Sayısı	Robot Adı	Simülasyondan elde edilen yol uzunluğu(cm)	Deneysel çalışmadan elde edilen yol uzunluğu(cm)	% Kobuki 1 Hata	% Kobuki 2 Hata
9	Kobuki 1	179,9791	154,0219		
	Kobuki 2	177,1478	199,8250		
10	Kobuki 1	180,1019	153,0547		
	Kobuki 2	177,1033	201,2912		
Ort	<b>Kobuki 1</b>	180,1051	152,5513		
	<b>Kobuki 2</b>	176,9820	200,8662		

Tekli mobil robotun yol planlaması Şekil 5.6.-7. ve çoklu mobil robotun yol planlaması ise Şekil 5.8.'de gösterilmekte ve elde edilen sonuçlar Tablo 5.3. –5.4.'te sunulmaktadır.

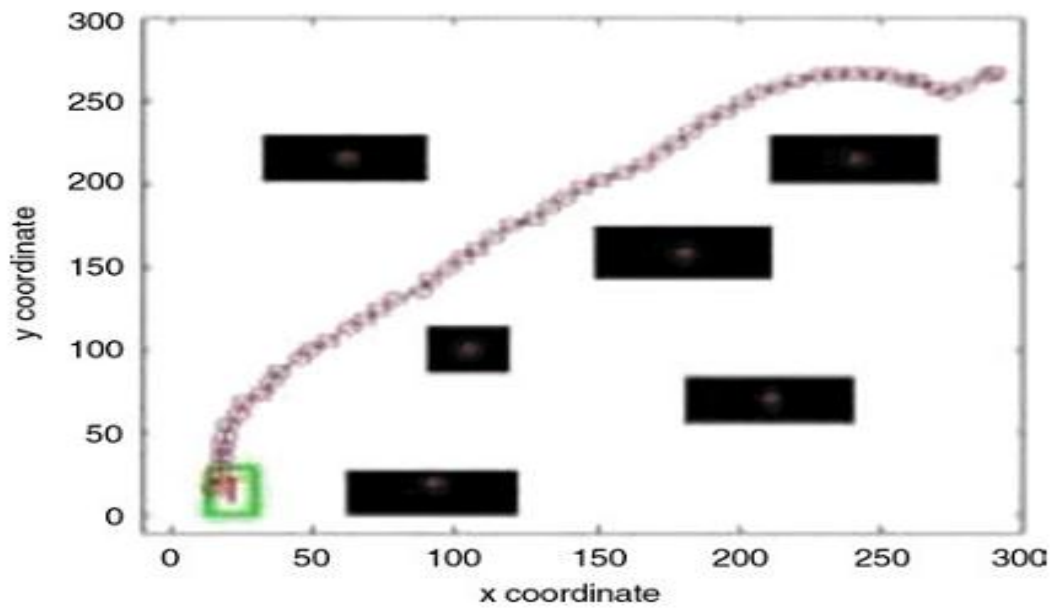
Tablo 5.3.'e göre tekli mobil robot için CS tabanlı global yol planlayıcı ve lokal yol planlayıcının bulunduğu simülasyon ve deneysel çalışmadaki yol uzunluklarının karşılaştırma sonucuna göre ortalama değer %18 olarak hesaplanmıştır.

### 5.5. Diğer Yapılan Çalışmalarla Karşılaştırma

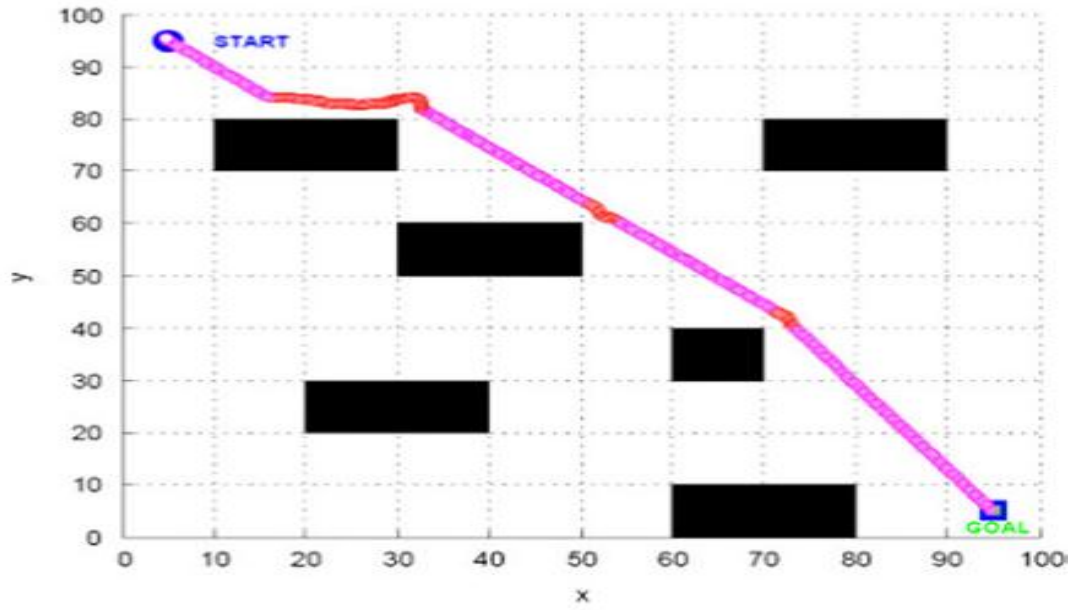
Bu bölümde CS tabanlı global yol planlayıcı ve meta-sezgisel tabanlı yol planlayıcılar farklı ortamlarda yapılan simülasyon çalışmalarından elde edilen sonuçlar karşılaştırılmıştır. Wang ve ark. (2010) tarafından yapılan GA, Mohamed ve ark. (2011, 2012) ) tarafından yapılan PSO ve Mohantyve (2017) ) tarafından yapılan CS tabanlı çalışmalardan elde edilen sonuçlar Şekil 5.9-5.11. arasında gösterilmektedir. Geliştirilen CS algoritma tabanlı planlamadan elde edilen sonuçlar Şekil 5.12.'de verilmiştir.



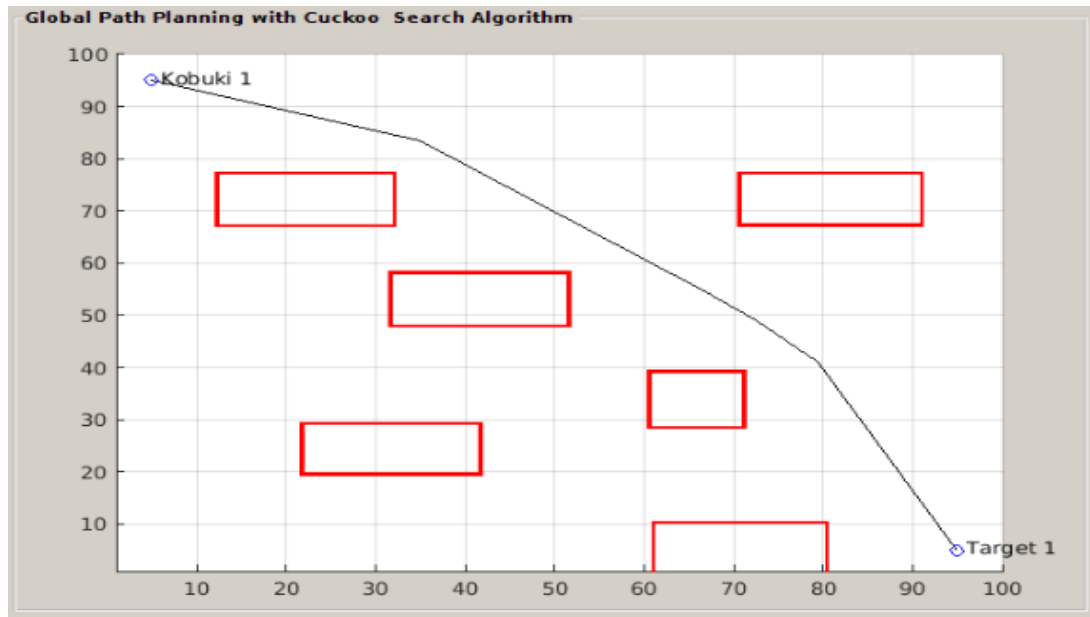
Şekil 5.9. GA algoritma tabanlı yol planlama (Wang ve ark. 2010)



Şekil 5.10. PSO algoritma tabanlı yol planlama (Mohammed ve ark. 2011)



Şekil 5.11. CS algoritma tabanlı yol planlama (Mohanty ve ark. 2017)



Şekil 5.12. Tarafımızdan Geliştirilen CS algoritma tabanlı yol planlama

Tablo 5.5. Tekli mobil robot için farklı algoritmalarla gerçekleştirilen simülasyon sonuçlarının yol uzunluğuna göre karşılaştırılması(Şekil 5.6.-9.)

Şekil	Yol uzunluğu(cm)	Algoritma	% Performans Farkı
Şekil 5.9.	171,4	GA	22,49
Şekil 5.10.	147,6	PSO	9,9
Şekil 5.11.	145,25	CS	8,53
Şekil 5.12.	<b>132,85</b>	Geliştirilen CS	

Simülasyon çalışmalarının sonuçlarına göre önerilen CS tabanlı global yol planlayıcının diğer yöntemlerle yol uzunluğu açısından karşılaştırıldığında daha kısa ve pürüzsüz yol sağladığı açıkça görülmektedir. Simülasyon sonuçlarından elde edilen yol uzunluğu sonuçları Tablo 5.5.'te verilmektedir. Sonuçlara göre tekli robot yol planlamasında sırasıyla %22,49, %9,9 ve %8,53 daha kısa yol planlaması gerçekleştirilmiştir.

## **BÖLÜM 6. ÇOKLU MOBİL ROBOTLAR İÇİN GELİŞTİRİLEN HİBRİT ALGORİTMALAR TABANLI GLOBAL YOL PLANLAMA**

### **6.1. Giriş**

Günümüzde, bilim ve mühendisliğin çeşitli alanlarında, doğadan ilham alan meta heuristik algoritmalar kullanılmaktadır. Genellikle GA, SA, ACO, FA ve PSO, CS algoritmaları gibi doğadan ilham alan en popüler meta-sezgisel algoritmalar mobil robotların yol planlamasında kullanılmaktadır.

Önceki bölümlerde çoklu mobil robotlarda yol planlamak amacıyla üç farklı meta-sezgisel algoritma kullanılmıştır. PSO, FA ve CS meta-sezgisel algoritmalar mobil robotların yol planlamasında ayrı ayrı kullanılmıştır. Bu algoritmalar farklı çalışma ortamlarında yol planlamadaki etkinlikleri doğrulanmak amacıyla çeşitli simülasyon ve deneysel çalışmalar gerçekleştirilmiştir.

Tek bir meta-sezgisel algoritma her zaman yol planlamada optimal sonuçlar vermemektedir. Bu sebepten dolayı hibrit meta-sezgisel yaklaşımlar geliştirilmektedir. Hibrit meta-sezgisel yaklaşımların en önemli avantajlarından biri farklı algoritmaların çoklu özelliklerini tek bir algortmada birleştirilmesidir.

Bu bölümde çoklu mobil robotların yol planlaması amacıyla PSO, FA ve CS algoritmaları birleştirilerek birden fazla hibrit algoritmanın oluşturulması sağlanmıştır. Guguk Kuşu Algoritma-Parçacık Sürü Optimizasyonu(CS-PSO), Guguk Kuşu Algoritması-Ateş Böceği algoritma(CS-FA), Guguk Kuşu Algoritma-Parçacık Sürü Optimizasyonu-Ateş Böceği algoritma(CS-PSO-FA) oluşturulan hibrit algoritmalar mobil robotların yol planlamasında kullanılmıştır. Hibrit algoritmalar oluşturulurken bir algoritmanın çıktı değeri diğer algoritmanın girdi verisi olarak

kullanılmaktadır. Sonuç olarak oluşturulan hibrit algoritmalar ile çoklu mobil robotların yol planlama performansının artırılması amaçlanmaktadır.

CS algoritması Bölüm 4’te detaylı olarak açıklanmıştır. Bu bölümde CS algoritması hibrit algoritmaların temelinde bulunduğu için kısa bir açıklama yapılmıştır.

CS algoritması gerçek global optimum yakınsama için global optimize edici karakteristik özelliklere sahiptir. Bu özelliğinden dolayı oluşturulan hibrit algoritmaların temelinde CS algoritması kullanılmaktadır. Bunlara ek olarak CS algoritması, kontrol parametreleri ve yüksek verimlilik gibi birçok avantaja sahiptir, fakat yakınsama hızı ve düşük doğruluk gibi bazı dezavantajlara da sahiptir. CS algoritmasında kullanılan Levy uçuşunun sağladığı yüksek dereceli rassallığı, arama sürecini bir alandan diğerine hızlı bir şekilde geçilmesini sağlar. Buda, algoritmanın global arama kabiliyetini güçlendirmektedir. Levy uçuşunun yüksek rassallığı göz önüne alındığında algoritma kör bir arama sürecini başlatır, yakınsama hızı yavaşlar ve arama etkinliği optimal çözümü belirgin bir şekilde azaltmaktadır.

## **6.2. CS ve PSO Tabanlı Geliştirilen Hibrit Algoritmanın Matematiksel Modellenmesi**

PSO işbirliğine dayalı özelliklere sahip popülasyonların toplumsal davranışlarını taklit eder ve matematiksel optimizasyon problemlerini çözmekte yaygın olarak kullanılır. PSO algoritmasındaki teorik çalışmalar bu algoritmanın popülasyon çeşitliliğini ağırlık katsayısından dolayı hızlı bir şekilde kaybettiğini göstermektedir. Tüm çözümlerin amacı mevcut en iyi çözümü elde etmektir. Diğer bir deyişle, tüm çözümler, bir arama alanının küçük bir bölgesinde mevcut en iyi çözüm etrafında gerçekleşmektedir. Bu gibi durumlarda PSO algoritması global optimum çözüm elde edilmeden önce lokal bir noktaya yakınsayabilmektedir. Sonuç olarak PSO algoritması yerel optimuma hızla yakınsadığı yerde, CS algoritması ise global optimum bir yol bulabilir (Wang ve ark. 2012a).



Geliştirilen özgün CS-PSO hibrit algoritma yapısını oluşturmak amacıyla CS ve PSO algoritmaları birleştirilmiştir. Algoritmaların üstün özellikleri seçilerek hibrit algoritma oluşturulmuştur. CS algoritmasının performansını ve uygulanabilirliğini geliştirmek amacıyla PSO algoritmasının güçlü özellikleri kullanılarak CS algoritmasının verimliliği artırılmıştır. CS-PSO hibrit algoritmasının temelinde, CS-PSO hibrit algoritmasının temelinde, global arama için CS algoritmasının kullandığı Levy uçuş yöntemiyle, yerel arama özelliği içinde PSO algoritmasının hız ve konum bulma özelliği birleştirilmiştir. Guguk kuşları, yumurtalarının bulunma olasılığını azaltmak amacıyla daima daha iyi bir yuva aramaktadır. Daha iyi yuvayı bulmak için komşu yuvaların konumlarından haberdar olmak ve birbirlerini daha iyi bir yere göç etmelerine yardımcı olmaktadır. Bu hibrit algoritmada her bir guguk kuşu kendi hayatında en iyi kişisel deneyimi  $p_{best}$  olarak kaydeder. Tüm kuşlar arasında en iyi  $p_{best}$  ise  $g_{best}$  denilmiştir. Guguk kuşlarının iletişimi  $p_{best}$ ,  $g_{best}$  yoluyla belirlenir ve bu parametreleri ve her birinin hızını kullanarak konumları güncellenmektedir.

$$v_{t+1} = v w_t + c_1 r_1 (p_{best} - p_t) + c_2 r_2 (g_{best} - p_t) \quad (6.1)$$

Matlab Kodu:  $nest(j,:) = s * w + c_1 * rand(1,r) * (p_{best} - s) + c_2 * rand(1,r) * (g_{best} - s);$

(Denklem 6.1) gösterilen  $w$  atalet,  $v$  ise CS algoritmasından gelen mevcut konum,  $c_1$  ve  $c_2$  PSO algoritmasının öğrenme faktörleridir. Hibrit algoritmanın sözde kodu Şekil 6.1.'de verilmiştir.

```

Amaç fonksiyonu belirlenir  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ .
n adet konak yuvası olan başlangıç popülasyonu üret
while (durdurma kriteri)
Levy uçuşu ile rassal guguk kuşu al
pbest, gbest belirlenir
PSO ile guguk kuşlarının konumları güncellenir
 $f_i$  maliyetini bul
n yuva içinden rassal yuva seç
if ( $f_i < f_j$ ),
j'yi yeni çözümle değiştir.
end
pbest, gbest güncelle
En kötü yuvaların  $p_a$  oranı kadarını terk et ve yenileri inşa et
İyi Çözümleri tut
Çözümleri sırala ve o anki en iyi çözümü bul
end while
Sonuçlar görüntülenir.

```

Şekil 6.1. Geliştirilen CS-PSO algoritmasının sözde kodu

### 6.3. CS ve FA tabanlı Geliştirilen Hibrit Algoritmanın Matematiksel Modellenmesi

FA, tropikal bölgelerde yaşayan ateş böceklerinin parlaklık ve hareket yönlerini baz alınarak geliştirilen bir meta-sezgisel algoritmadır. FA'nın yerel arama kabiliyeti güçlüdür. FA parametreleri iterasyon sırasında değişmez. Çekicilik katsayısı ve rasgelelik katsayısı olarak adlandırılan iki temel parametresi bulunmaktadır. Bu parametrelerin değerleri yakınsama hızını ve algoritmanın davranışını belirlemede çok önemlidir.

Geliştirilen özgün CS-FA hibrit algoritmasının yapısını oluşturmak içinde CS-PSO algoritmasında olduğu gibi guguk kuşlarının daha iyi yuva konumu bulabilmeleri için CS algoritması ile FA birleştirilerek hibrit bir çalışma gerçekleştirilmiştir.

CS-PSO hibrit algoritma yapısında olduğu gibi CS algoritmasında guguk kuşlarının global arama için kullandığı Levy flight yöntemi ile yerel arama için FA'nın kullandığı (Denklem 6.2) ile yerel arama özelliği birleştirilmiştir. Bu şekilde global ve yerel arama arasında denge oluşturulmuştur. CS algoritmasından elde edilen en iyi yuva değerine FA'nın parlak ve daha çekicilik parametreleri eklenerek daha iyi bir konumda bulunan yuva hedeflenmektedir. Böylelikle CS algoritması yerel bir optimum noktada takılı kalması engellenerek daha iyi çözüm yuvaları elde edilmiştir. Algoritmanın daha anlaşılır olması için geliştirilen CS-FA hibrit algoritmasının sözde kodu Şekil 6.2.'de verilmiştir.

$$x_i^{t+1} = x_i + \beta_0 e^{-\gamma r^2 ij} (x_j - x_i) + \alpha \varepsilon_i^t \quad (6.2)$$

Matlab Kodu: `nest(i,:)=nest(i ,:).*(1-beta)+nest(j,:).*beta+tmpf;`

```

Amaç fonksiyonu belirlenir  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ .
n adet konak yuvası olan başlangıç popülasyonu üret
while (durdurma kriteri)
  Levy uçuşu ile rassal guguk kuşu al
  FA ile CS algoritmasından alınan konumların etkililiği çekicilik ve parlaklık parametresi ile güncellenir.
  fi maliyetini bul
  n yuva içinden rassal yuva seç
  if ( $f_i < f_j$ ),
    j'yi yeni çözümlerle değiştir.
  end
  En kötü yuvaların  $p_a$  oranı kadarını terk et ve yenileri inşa et
  İyi Çözümleri tut
  Çözümleri sırala ve o anki en iyi çözümü bul
end while
Sonuçlar görüntülenir.

```

Şekil 6.2. Geliştirilen CS-FA hibrid algoritmanın sözde kodu

#### 6.4. CS, PSO ve FA tabanlı Geliştirilen Hibrit Algoritmanın Matematiksel Modellenmesi

Geliştirilen özgün CS-PSO-FA hibrid algoritmanın yapısı CS, PSO ve FA meta-sezgisel algoritmalarının güçlü özellikleri birleştirilerek oluşturulmuştur. CS algoritmasının yüksek verimliliği, PSO algoritmasının hızlı yakınsama hızı ve FA'nın yerel arama özellikleri birleştirilerek sağlam bir hibrit meta-sezgisel algoritma geliştirilmiştir.

CS algoritması global ve yerel rassal yürüyüş yaklaşımları olası tüm çözümleri taradığı için iki yaklaşımı dengelemektedir. Algoritma dengelenme işlemini global ve yerel rassal yürüyüş parametreleriyle sağlamaktadır. CS algoritması global aramayı gerçekleştirmek için Levy uçuşunu kullanmaktadır. Diğer hibrit meta-sezgisel yaklaşımların yapısına benzer olarak CS algoritmasında global rassal yürüyüş gerçekleştirildikten sonra yerel rassal yürüyüşde PSO algoritmasının pozisyon denklemleri kullanılmıştır. Guguk kuşlarına daha iyi konak yuva sağlamak için PSO algoritmasının yapısında bulunan  $p_{best}$ ,  $g_{best}$  kullanarak konumlarını güncellenmektedir.

$$v_{t+1} = vw_t + c_1 r_1 (p_{best} - p_t) + c_2 r_2 (g_{best} - p_t) \quad (6.3)$$

$$\text{Matlab Kodu: } \text{nest}(j,:) = s * w + c1 * \text{rand}(1,r) * (p_{best} - s) + c2 * \text{rand}(1,r) * (g_{best} - s); \quad (6.4)$$

PSO algoritmasından elde edilen en iyi konumun etkinliğini artırmak amacıyla FA algoritmasında bulunan ateş böceklerinin daha parlak ve daha çekici olana doğru hareket etme özelliği kullanılarak en iyi konum değerleri ile güncellenmektedir. Geliştirilen özgün CS-PSO-FA hibrit algoritmasının sözde kodu Şekil 6.3.'te verilmiştir.

$$x_i^{t+1} = x_i + \beta_0 e^{-\gamma r^2 i j} (x_j - x_i) + \alpha \varepsilon_i^t \quad (6.5)$$

$$\text{Matlab Kodu: } \text{nest}(i,:) = \text{nest}(i, :) * (1 - \beta) + \text{nest}(j, :) * \beta + \text{tmpf};$$

Amaç fonksiyonu belirlenir  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ .  
n adet konak yuvası olan başlangıç popülasyonu üret  
**while** (*durdurma kriteri*)  
Levy uçuşu ile rassal guguk kuşu al  
**pbest, gbest belirlenir**  
**PSO ile guguk kuşlarının konumları güncellenir.**  
**FA ile PSO algoritmasından alınan konumların daha iyi konuma gelmesi sağlamak için**  
**ateş böceklerinin çekicilik ve parlaklık parametresi ile güncellenir.**  
*f<sub>i</sub> maliyetini bul*  
n yuva içinden rassal yuva seç  
**if** ( $f_i < f_j$ ),  
j'yi yeni çözümlerle değiştir.  
**end**  
En kötü yuvaların  $p_a$  oranı kadarını terk et ve yenileri inşa et  
İyi Çözümleri tut  
Çözümleri sırala ve o anki en iyi çözümü bul  
**end while**  
Sonuçlar görüntülenir.

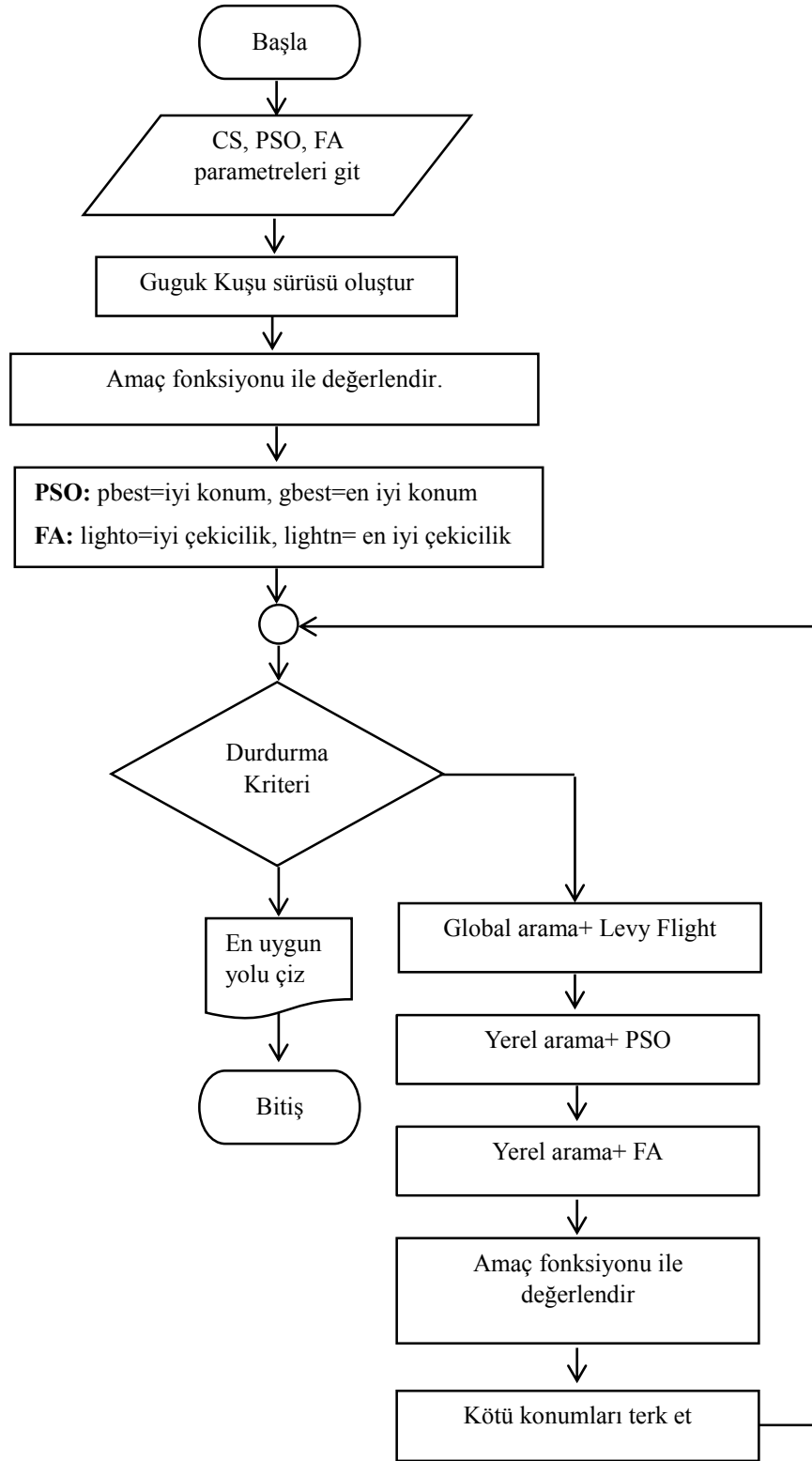
Şekil 6.3. Tarafımızdan geliştirilen CS-PSO-FA hibrit algoritmanın sözde kodu

### 6.5. CS, PSO ve FA tabanlı Bu Tez Çalışmasına Özgü Geliştirilen Yeni Bir Hibrit Algoritma ile Global Yol Planlama

Bu çalışmada kare şeklinde engellerin bulunduğu statik bir çevrede çoklu mobil robotların navigasyon problemlerini çözmek için PSO, FA ve CS algoritma tabanlı bir yaklaşım geliştirilmiştir. Bu yaklaşımın amacı heuristik algoritmaların verimliliğini kullanarak yol planlamada yolun minimize edilerek belirlenen zaman, maliyet gibi en önemli performans kriterlerini geliştirmektir. Yol planlamada hedefe giden yolu oluşturan PSO'da en iyi parçacık, FA'da en iyi ateş böceği, CS'de en iyi konak yuva parametreleri yoldaki bir adımı temsil etmektedir.

Tasarlanan hibrit algoritmalarının temelinde CS algoritması yer almaktadır. PSO ve FA sırasıyla CS algoritmasıyla birleştirilerek ikili hibrit algoritmalar oluşturulmuştur. Son olarak PSO ve FA algoritmaları CS algoritmasıyla birleştirilerek üçlü hibrit algoritma geliştirilmiştir. Bu algoritmalar engelli bir ortamda en kısa/optimal yol planlamasında kullanılmıştır. CS algoritmasında oluşturan her bir yuva için uygunluk değerleri hesaplanmıştır. Ardından global arama yapabilmek için Levy uçuşu gerçekleştirilmiştir. Buradan gelen yeni çözümlerin uygunluk değerleri ölçülerek, iyi çözümler kötü çözümler ile yer değiştirmiştir. Sonraki adımda ise yerel arama gerçekleştirilmiştir. Yerel arama sonucundan elde edilen çözümlerin uygunluğu hesaplanarak, en iyi uygunluk değerine sahip olanların yerleri değiştirilmiştir. Yerel aramadan elde edilen iyi konum değeri PSO algoritmasının pozisyon denklemleri ve FA algoritmasının hareket denklemleri sırasıyla uygulanarak konum güncellemesi sağlanmıştır.

Şekil 6.4.'te çoklu mobil robotların yol planlaması için geliştirilen özgün CS-PSO-FA hibrit algoritmanın akış diyagramı verilmiştir.



Şekil 6.4. Çoklu mobil robotların yol planlaması için geliştirilen CS-PSO-FA hibrid algoritmanın akış diyagramı

Mobil robot yol planlama için geliştirilen yeni bir CS-PSO-FA algoritmasının uygulama basamakları;

Adım 1 - Robotun başlangıç ve hedef noktaları belirlenir.

Adım 2 - Başlangıç sürüsü, konak yuvaları ile oluşturulur.

Adım 3 - Robotun hareket edeceği noktalar amaç fonksiyonu ile hesaplanır. Bu noktalar üzerinde engel olup olmadığı belirlenir.

Adım 4 - CS algoritması aktif edilir.

Adım 5 - Amaç fonksiyonuna göre mevcut en iyi yuva seçilir.

Adım 6 - Kötü yuvalar terk edilir ve Levy uçuşuyla yeni yuva inşa edilir.

Adım 7 - PSO konum denklemleri kullanılarak iyi konumdaki yuva bulunur.

Adım 8 - FA hareket denklemleri kullanarak PSO'dan elde edilen konum değerini iyileştirmek amacıyla daha parlak olan yuvaya doğru inşa etmesini sağlar.

Adım 9 - Mobil robot en iyi yuvaya doğru hareket eder.

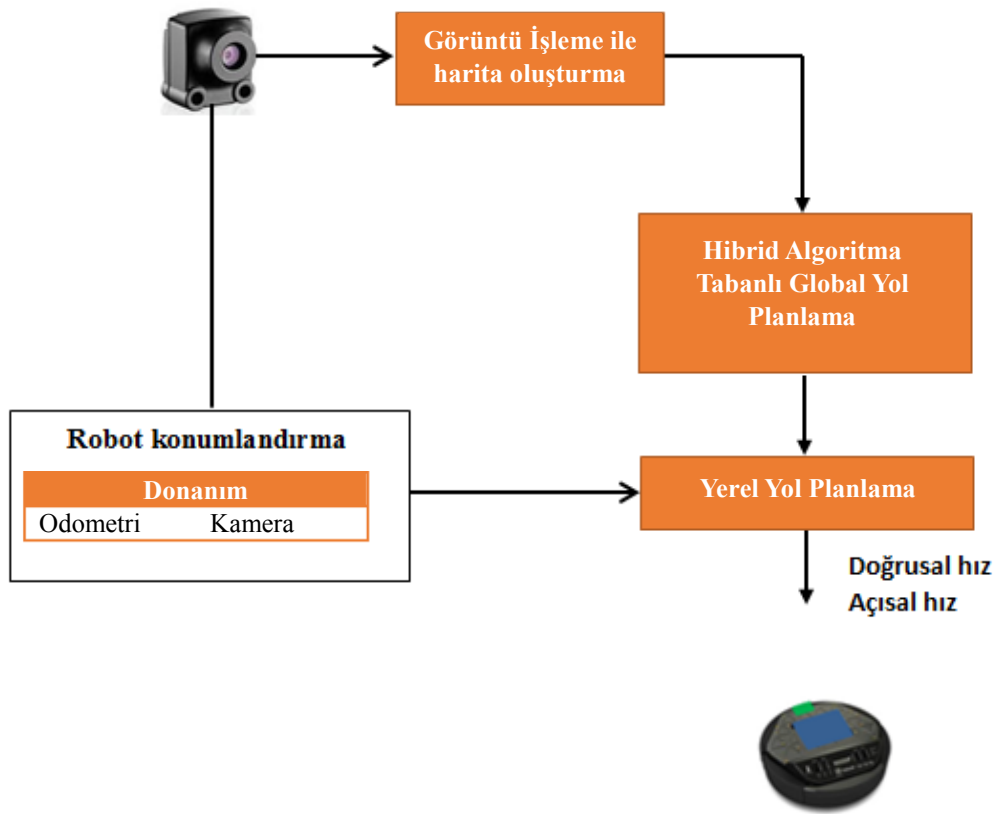
Adım 10 - 3-8 arasındaki adımlar engelden sakınarak devam eder.

Adım 11 - Hedefe ulaşıldı.

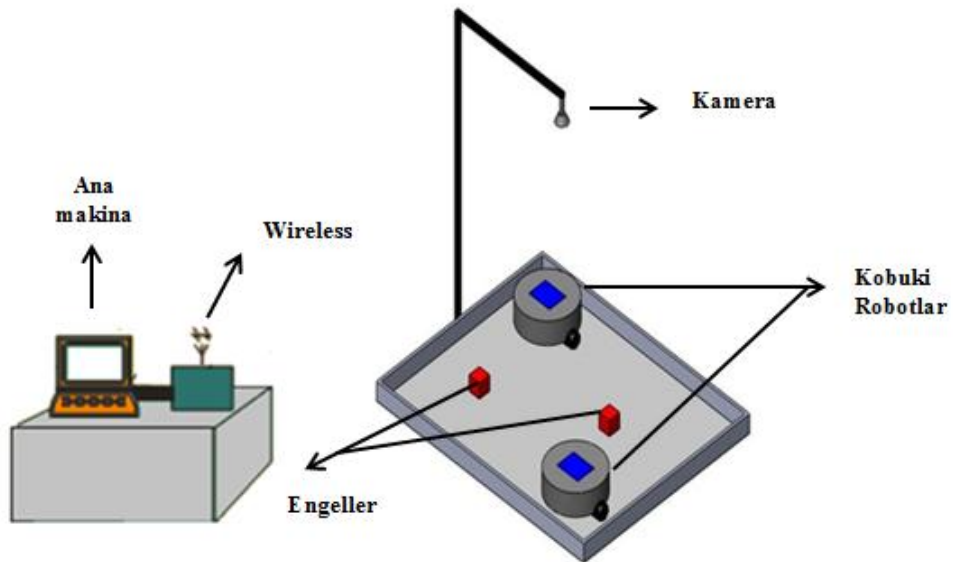
### **6.5.1. Yol planlama için gerçekleştirilen planlama sistemi**

Tez çalışmasında kullanılan yol planlama sisteminin blok diyagramı Şekil 6.5.'de gösterilmektedir. Sistemin genel yapısı incelendiğinde; kameradan alınan görüntü işlenerek çalışma çevresi haritaya dönüştürülmekte ve bu statik çevrede çoklu mobil robotlar engellere çarpmaksızın başlangıç ve bitiş noktası arasındaki en kısa yolu bulmaktadırlar. Yerel yol planlamada çoklu mobil robotlar global yol planlamadan elde edilen yolu takip etmektedirler.

Gerçekleştirilen çalışma, Sakarya Üniversitesi Mekatronik mühendisliği bölümünde bulunan laboratuvar ortamında tasarlanan platformda uygulanmıştır. Yapılan çalışmada platformun tamamını görecektir şekilde yerleştirilen Usb kamera, 2 adet Kobuki robot, robotları çalıştırmak için Raspberry pi 3, engeller, veri iletişimini sağlamak amacıyla kablosuz iletişim(Wi-Fi) standardı kullanılmıştır. Bunlara ek olarak robotlar ve program arasındaki iletişimi sağlayacak server bir bilgisayar kullanılmaktadır.



Şekil 6.5. Statik çevrede çoklu mobil robotlarının yol planlaması için tasarlanacak sistem

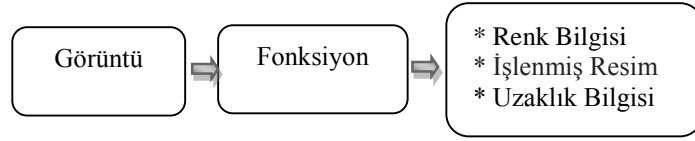


Şekil 6.6. Laboratuvar Ortamının Modeli



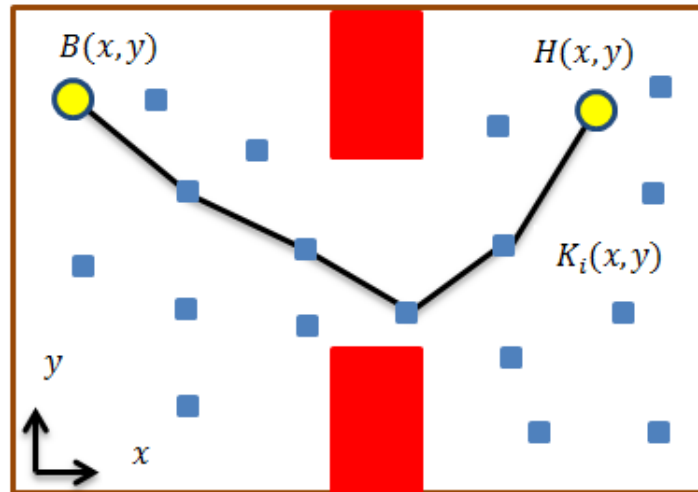
İlk adımda, laboratuvar ortamında Şekil 6.6.'da ki gibi kurulan platformun ortasına yerleştirilmiş olan kamera ile çalışma ortamının görüntüsü elde edilmektedir.

Elde edilen görüntü Şekil 6.7.'de verilen görüntü işleme fonksiyonuna girdi değeri olarak verilmektedir. Fonksiyona giren görüntüden mobil robotların ve engellerin konumları bulunmaktadır. Bu konumlar harita bilgisinde kullanılması amacıyla gerçek dünya koordinatlarına çevrilerek statik çevrenin alt yapısı oluşturulmuştur.



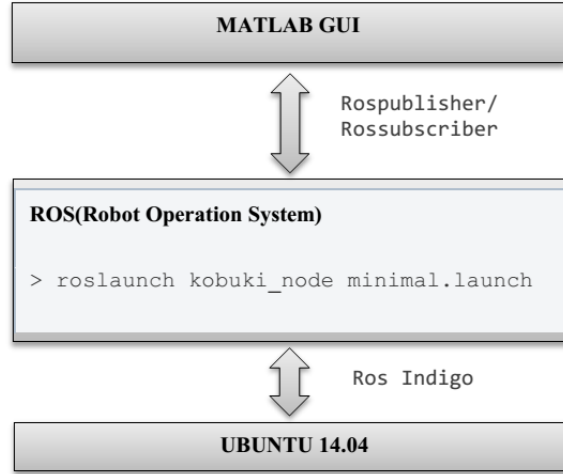
Şekil 6.7. Görüntü işleme fonksiyonu blok şeması

Mobil robotun gezindiği engelli statik çevre tanımlanmasında konumlandırma tamamlandıktan sonra ortamda rastgele  $K_i(x, y)$ ,  $(i = 1, \dots, N)$  olarak tanımlanan örnekler oluşturulmuştur. Şekil 6.8.'de ki gibi tanımlanan çevrede çoklu mobil robotlar, hücrelerin  $(x, y)$  koordinat düzlemini kullanarak hareket etmektedir. B başlangıç noktaları (mobil robotun konumları), H bitiş noktası ve  $K_i(x, y)$  rastgele tanımlanan konumları temsil etmektedir. Robot başlangıç ve hedef noktaları arasında uygulanabilir yolu bulana kadar 4 farklı yerel hareketi kullanmaktadır.



Şekil 6.8. Statik çevrenin tanımlanması

İkinci adımda, oluşturulan çevreye göre engellere çarpmaksızın en kısa yolun hesaplanması amacıyla PSO, FA, CS ve geliştirilen özgün hibrit algoritmalar ile global yol planlama gerçekleştirilmiştir.



Şekil 6.9. Önerilen Yazılımın Taslağı

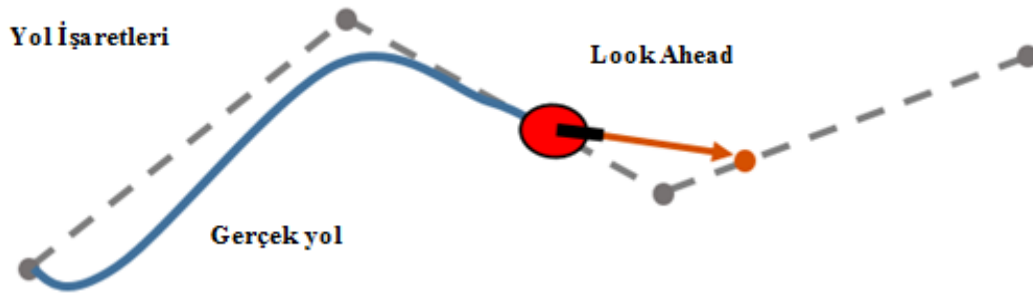
Üçüncü adımda ise global yol planlamadan elde edilen yol ve mobil robotdan elde edilen odometri verileri birleştirilerek gerçek zamanlı olarak yeni bir yol üretmek için Pure Pursuit algoritması kullanılarak yerel yol planlaması gerçekleştirilmiştir. Pure Pursuit ile elde edilen doğrusal ve açısal hız bilgileri Wireless üzerinden Kobuki mobil robotlara gönderilmekte ve robotlar yol koordinatları boyunca hareket edip hedefe ulaşması sağlanmaktadır. Yerel yol planlamasında mobil robotun bileşenlerini kullanabilmek için ROS (Robot Operation System) yazılımı kullanılmıştır(Şekil 6.9.).

### 6.5.2. Yerel yol planlama için kullanılan planlama sistemi

Pure Pursuit, mobil robotlarda sıklıkla kullanılan yöntemlerden biridir. Bu yöntem geometrik tabanlı çalışan yol izleme algoritmasıdır. Pure Pursuit, robotun başlangıç noktasından(mevcut konumun) hedef noktasına ulaşmaya kadar farklı yol işaretlerine dönüştürmek amacıyla farklı dairesel yayların uydurulmasına dayanmaktadır. Bu yol işaretleri, robotun mevcut konumu ile referans yolu arasındaki en yakın noktadan geçen sabit bir look ahead(ileri görüş) noktası olarak

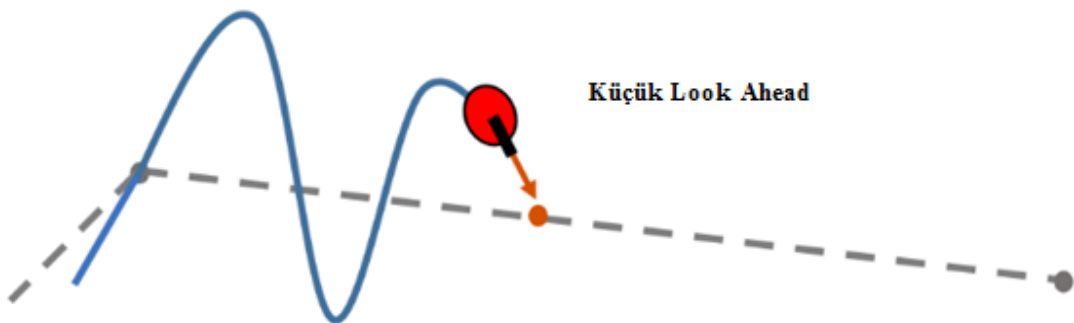
belirlenmektedir. İleri görüş uzaklık(look ahead distance) mesafesi, Pure Pursuit denetleyicisinin en önemli parametrelerinden biridir. İleri görüş uzaklık (Look ahead) parametresini ayarlamak denetleyicinin performansında önemli bir rol oynamaktadır. Denetleyici çıkış olarak robotun sürüş tekerlek hızlarına gönderdiği dönüş açısını kullanmaktadır.

Şekil 6.10.'da yol üzerindeki yol işaretlerini, mobil robot ve ileri görüş noktasını göstermektedir. Buna ek olarak gerçek yol ile yol işaretlerinin eşleşmediği görülmektedir.



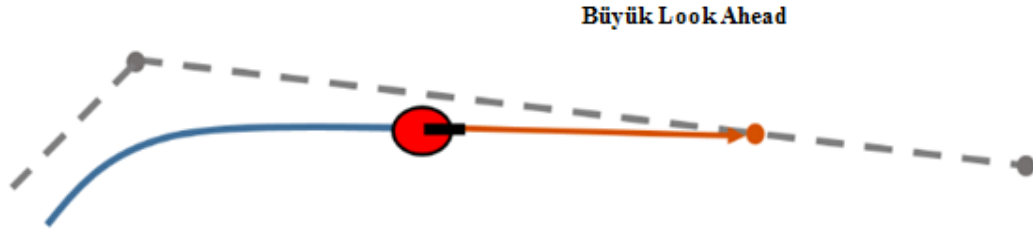
Şekil 6.10. Pure Pursuit algoritma yapısı

İleri görüş(Look ahead) parametrelerinin değiştirilmesi robotun yolu nasıl izlendiğini de değiştirmektedir. Bu parametre belirlenirken dikkat edilmesi gereken yolun takip edilmesi ve tekrardan izlenmesini sağlamaktır. Yol işaretleri arasındaki yolu takip etmek için küçük Look Ahead mesafesi seçilirse mobil robotun yola doğru hızlı bir şekilde ilerlemesi sağlanırken Şekil 6.11.'deki gibi robot yoldan çıkabilir ve salınımlar gerçekleştirebilir.



Şekil 6.11. Küçük Look Ahead noktasının seçilmesi

Yol boyunca salınımları azaltmak için daha büyük Look Ahead mesafesi seçilebilir fakat bu seçimde robotun köşe noktalarda daha büyük eğri çizmesine sebep olmaktadır.



Şekil 6.12. Büyük Look Ahead noktasının seçilmesi

Look Ahead mesafe belirleme özelliği, takip edilen yol ve robot sistemine göre uygun bir şekilde ayarlanmalıdır. Bu çalışmada MATLAB Robotik Sistem’de bulunan Pure Pursuit yol izleme denetleyicisi kullanılmıştır. Pure pursuit algoritmasının başlangıç tanımlamaları Şekil 6.13.’deki gibi gerçekleştirilmiştir.

```
robotGoal1=0.18;
controller1 = robotics.PurePursuit;
controller1.Waypoints = path1;
controller1.DesiredLinearVelocity = 0.2;
controller1.MaxAngularVelocity = 1;
controller1.LookaheadDistance = 0.6;
distanceToGoal1 = norm(robotCurrentLocation1 - robotGoal1);
robotGoal1 = path1(end,:);
```

Şekil 6.13. Pure Pursuit başlangıç tanımlamaları

Yol işaret noktalarını takip eden robot aşağıdaki beş işlem basamağı kullanarak başlangıç noktasından hedef noktaya doğru hareket ettirilir:

1. Mevcut ortamda robotun global yol planlamadan elde edilen konumu ve odometri konumunu aşağıdaki gibi tanımlanır:

```
odomdata1 = receive(sub1,1);
quat1 = odomdata1.Pose.Pose.Orientation;
angles1 = quat2eul([quat1.W quat1.X quat1.Y quat1.Z]);
odomPose1 = [odomdata1.Pose.Pose.Position.X odomdata1.Pose.Pose.Position.Y angles1(1)];
robotCurrentPoseOffset1 = robotCurrentPose1 - odomPose1;
robotCurrentLocation1 = path1(1,:);
robotCurrentPose1 = [robotCurrentLocation1 0];
```

Şekil 6.14. Pure Pursuit konum tanımlamaları

2. Robotun mevcut konumu hedef noktasına ulaşana kadar odometri verisiyle birleştirilerek güncelleyen komutlar Şekil 6.15.'te verilmiştir.

```
odomdata1 = receive(sub1,1);
quat1 = odomdata1.Pose.Pose.Orientation;
angles1 = quat2eul([quat1.W quat1.X quat1.Y quat1.Z]);

robotCurrentPose1 = [odomdata1.Pose.Pose.Position.X odomdata1.Pose.Pose.Position.Y
angles1(1)] + robotCurrentPoseOffset1;
```

Şekil 6.15. Pure Pursuit konum tanımlamaları

3. Denetleyici kullanarak doğrusal ve açısal hızlar hesaplanır.

```
[v1, omega1] = step(controller1, robotCurrentPose1);
```

Şekil 6.16. Açısal ve doğrusal hızların hesaplanması

$v$  = Doğrusal hız,  $\omega$  = Açısal hız

4. Bu hızlar robota gönderilir.

```
velmsg1.Linear.X = v1;
velmsg1.Angular.Z = omega1;
send(pub1, velmsg1);
```

Şekil 6.17. Açısal ve doğrusal hızların robota gönderilmesi

5. Robot hedefe ulaşıncaya kadar 2 ve 4 arasındaki adımlar tekrarlanır.

### 6.5.3. Robot ve algoritmalar için MATLAB ve ROS tabanlı kullanıcı arayüzü tasarımı

MATLAB, özellikle mühendislik alanındaki sistemlerin analizi ve grafiksel çizimler için kullanılan, görüntü işleme, yapay sinir ağları, sayısal işaret işleme, optimizasyon, veri elde etme, veritabanı, süzgeç tasarımı, bulanık mantık, sistem kimliklendirme, dalgacıklar ve robotik gibi araçlar ile gelişmiş bir simülasyon tasarımı ve hesaplama ortamı sunmaktadır (Çankaya ve ark., 2013). Grafiksel arayüz oluşturmak için MATLAB yazılımının MATLAB GUI(Graphical User Interface) bölümü kullanılmıştır. MATLAB GUI ile kullanıcı, uygulamaya yönelik basit ve kolay arayüz tasarlayabilmektedir. Bu çalışmada, belirlenen fonksiyonların

simülasyonları ve hesaplamalarını gerçekleştirmek için MATLAB yazılımı kullanılmış olup grafiksel arayüz GUI aracılığı ile tasarlanmıştır.

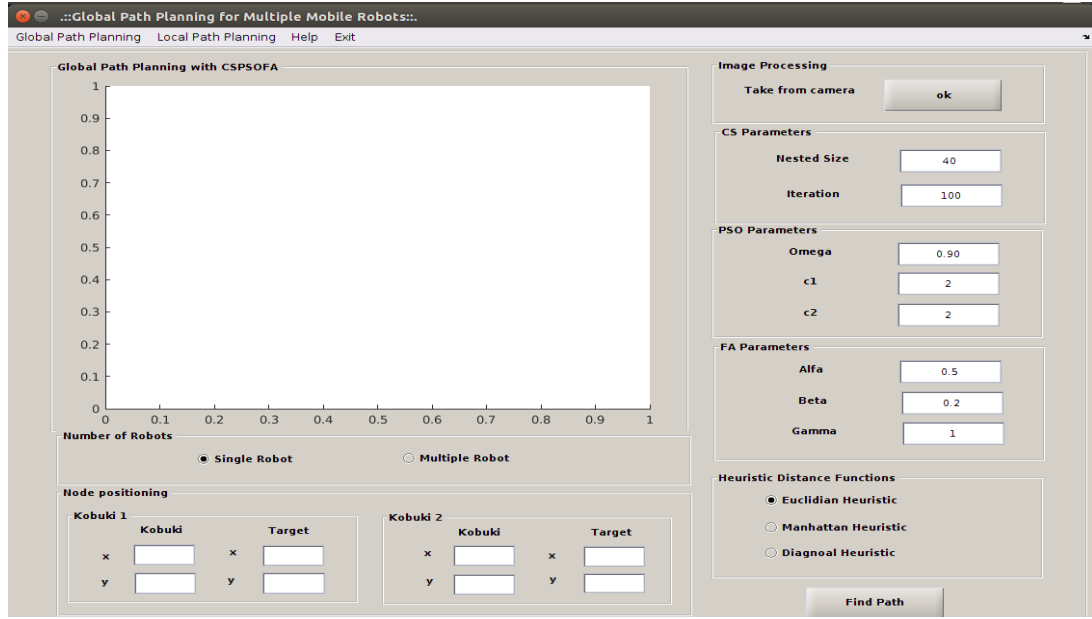
ROS (Robot Operation System) başlangıçta laboratuvar ortamında daha sık kullanılmaktaydı fakat günümüzde endüstriyel robot sistemlerinde geniş bir uygulama alanına sahiptir. Yayın ve katılım temel görevlerinin gelişmesine rağmen, uygulamaya yönelik mesajlar, servis çağırmak ve paylaşım parametre özellikleri sabit kalmaktadır. ROS, geleneksel işletim sistemlerinden farklı süreç yönetimi ve planlamaya sahiptir.

ROS'un temel özellikleri aşağıda özetlenmiştir:

- Uçtan uca
- Araç kutusu tabanlı
- Çok dilli
- Zayıf
- Serbest ve açık kaynak kodlu

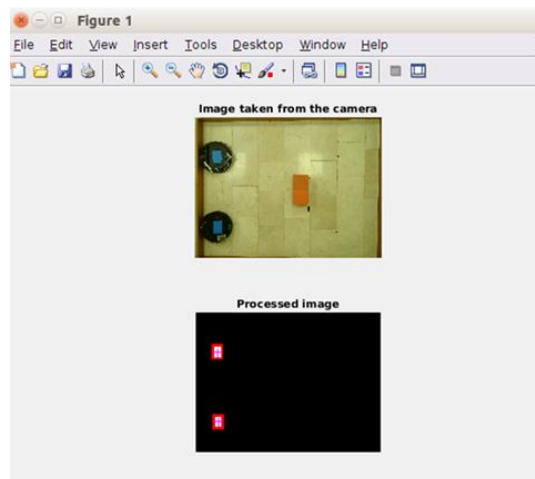
ROS'un programlama çerçevesinde genellikle C++ ve Python programlama dili vardır fakat son zamanlarda MATLAB da kullanılmaya başlanmıştır. MATLAB kontrol sistemleri ve robotik simülasyonlar için güçlü bir programlama dilidir (Quigley ve ark., 2009). MATLAB 2015a sürümünün sağlamış olduğu Robotik Sistemler araçlarıyla MATLAB ve ROS'un entegre olarak kullanılmasına olanak vermiştir.

Tez çalışmasında önerilen metodun etkisini incelemek ve daha kolay bir şekilde anlaşılması için kullanıcı dostu MATLAB-GUI ve ROS tabanlı grafiksel bir ara yüz tasarlanmıştır. Ara yüzün tasarlanma amacı çoklu mobil robotların yol planlamasını etkin ve kolay bir şekilde gerçekleştirmektir. Ara yüz iki ana menüden oluşmaktadır. Bunlar “Global Path Planning” ve “Local Path Planning” bölümleridir.



Şekil 6.18. Tasarlanan Arayüz

Global Path Planning (Global Yol Planlama) kısmında; Şekil 6.18.'de görüldüğü gibi “Image processing(Görüntü işleme)” kısmında kameradan alınan görüntü yardımıyla veya kameradan bağımsız olarak kullanıcı tarafından oluşturulan çalışma ortamının tanımlandığı ekran yer almaktadır. Tekli ve çoklu mobil robot için kamerayla alınan ortamın gerçek görüntüsü görüntü işleme aşamalarından geçirilerek haritaya dönüştürülmektedir (Şekil 6.19.).



Şekil 6.19. Kameradan alınan çalışma alanı görüntüsü ve haritaya dönüştürülmesi

“Number of robots (Robot sayısı)” kısmından mobil robotun tekli veya çoklu olacağı belirlenmektedir.

“Node positioning (Nokta konumları)” kısmında görüntü işleme tekniğinden Kobuki mobil robotların konum değerleri elde edilmektedir. Mobil robotların hedef noktaları kullanıcı tarafından belirlenmektedir.

“Heuristic distance functions (Heuristik uzaklık fonksiyonları) ” kısmında uzaklık fonksiyonu olarak (Euclidian, Manhattan, Diagonal) seçilebilmekte ve algoritmanın türene göre (PSO, FA, CS, CS-PSO, CS-FA, CS-PSO-FA) parametreler kısmından algoritma ile ilgili parametre değerleri değiştirilebilir.

“Global Path Planning” kısmında 200x160 cm olan statik çevrede Kobuki mobil robotlar ile başlangıç ve hedef noktası yuvarlak şekille temsil edilmiştir. Engellerin sınırları kırmızı çizgilerle belirlenmiştir.

“Find Path (Yol Bul)” butonu tıklandığında, kullanıcı tarafından belirlenen parametreler kullanılarak, geliştirilen hibrit algoritma ile en uygun yol bulunur ve bulunan yol çalışma ortamı üzerinde çizdirilir.

“Local Path Planning (Lokal Yol Planlama)” kısmında global yol planlamada belirlenen yol “Pure Pursuit” algoritması kullanılarak mobil robotun takip etmesi sağlanmaktadır.

## **6.6. Simülasyon ve Deneysel Sonuçlar**

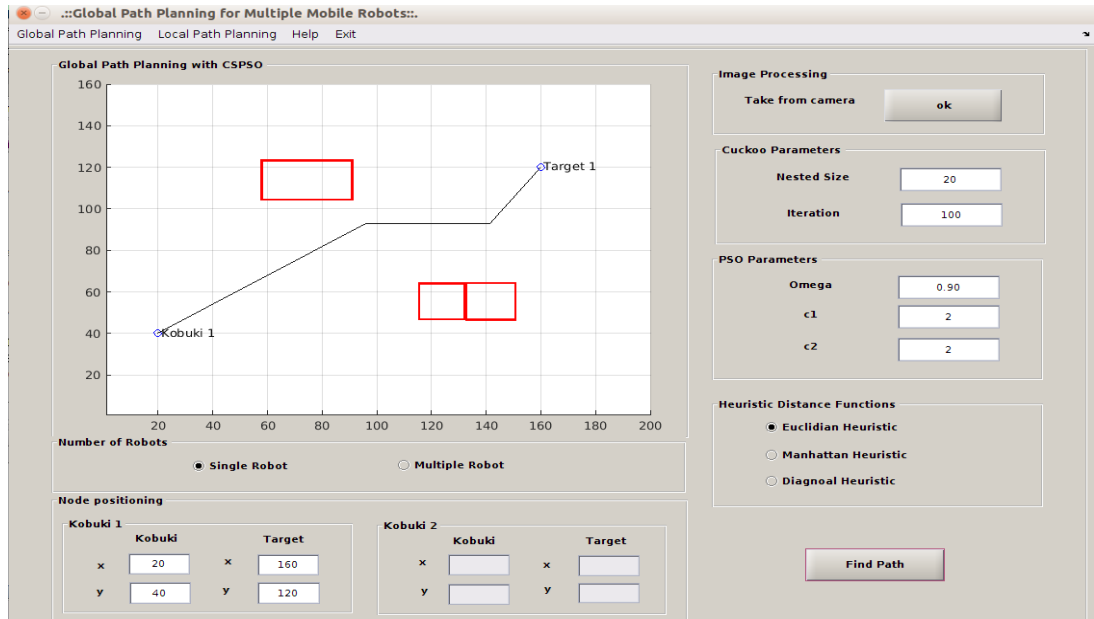
Geliştirilen özgün hibrit algoritmalar tabanlı yol planlayıcılar ile aynı çevre ortamları için simülasyon ve deneysel çalışmalar yapılarak yol uzunluğuna göre karşılaştırma yapılmıştır. Gerçek zamanlı deneysel analiz önerilen yol planlama yöntemlerinin geçerliliği, etkililiği ve uygulanabilirliğini göstermektedir. Simülasyon çalışmaları Intel(R) Core(TM) i5-5200U CPU @ 2.70 Ghz, 64 Bit, 4GB RAM'e sahip bir bilgisayarda Ubuntu 14.04.5 LTS işletim sistemi üzerinde çalıştırılmaktadır. Kullanılan robot işletim sistemi ROS indigo ve Kinect dağılımlarıdır. Deneysel



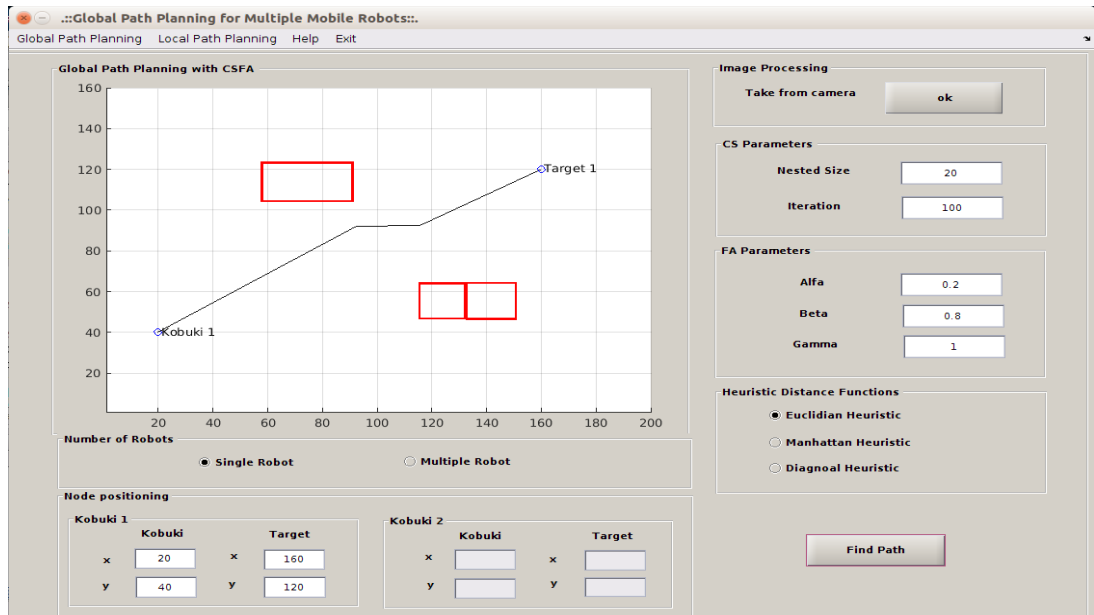
analizler ise laboratuvar ortamında (200x160cm) Kobuki mobil robotlar kullanarak gerçekleştirilmiştir. Kobuki mobil robotun teknik özellikleri Ek'te verilmektedir.

Geliştirilen meta-sezgisel tabanlı hibrit algoritmalar, statik bir ortamda tekli ve çoklu mobil robotların başlangıç ve hedef noktası arasında engellere çarpmadan global yol planlayıcıyla hesaplanan amaç fonksiyon değerlerine göre en kısa yol hesaplanmaktadır. Lokal yol planlamada ise mobil robot elde edilen global yolu takip ederek hedefe ulaşmaktadır. Hibrit algoritmaların performans analizleri yapılırken daha önceki Bölüm 3-5'te verilen PSO, FA, CS algoritmalarında kullanılan parametreler seçilmiştir. Simülasyon ve deneysel çalışmalarda performans kriteri olarak ortalama yol uzunluğu(cm) seçilmiştir. Ortalama yol uzunluğunun hesaplanması için hibrit tabanlı global yol planlayıcılar 10 kez çalıştırılmıştır. Her simülasyon analizden sonraki işlem ise global planlayıcıdan elde edilen yolun deneysel çalışmayla da desteklenmesidir.

Geliştirilen CS-PSO-FA hibrit algoritmanın yol planlama problemine uygulanma adımları verilmiştir. Konak yuvalarının başlangıç değerleri robotun gezinebileceği sınır değerleri arasında rastgele olarak belirlenir. CS algoritmasının global arama özelliği Levy uçuşu (Denklem 5.1) ile birleştirilir. Sonraki adım ise lokal arama için denklemi (Denklem 5.2) kullanılarak gerçekleştirilir. Lokal aramadan elde edilen çözümler PSO algoritmasının parçacıkların hızlarına bağlı konum değiştirme özelliğinden yararlanarak en iyi konuma sahip yuva belirlenmektedir. En iyi konuma sahip olan yuva FA'nın çekicilik ve ışık yoğunluğuna doğru hareket etme özelliğinden yararlanarak en iyi uygunluğa sahip olan yuva belirlenir. Bunlarda mobil robotun hareket edeceği konumları belirlemektedir.

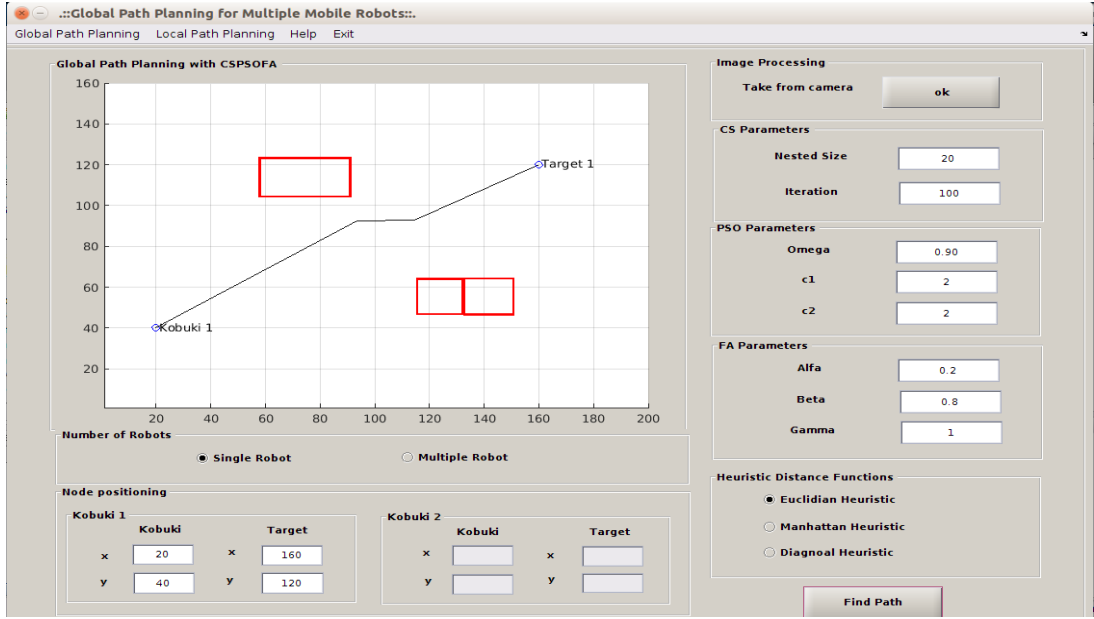


(a)



(b)

Şekil 6.20(a-c). Tekli mobil robot için geliştirilen hibrit algoritma tabanlı global yol planlama



(c)

Şekil 6.20(a-c). (Devamı)



Sahne 1



Sahne 2



Sahne 3



Sahne 4

Şekil 6.21. Tekli mobil robot için özgün CS-PSO-FA hibrit algoritmasının deneysel analizi



Sahne 5

Sahne 6

Şekil 6.21. (Devamı)

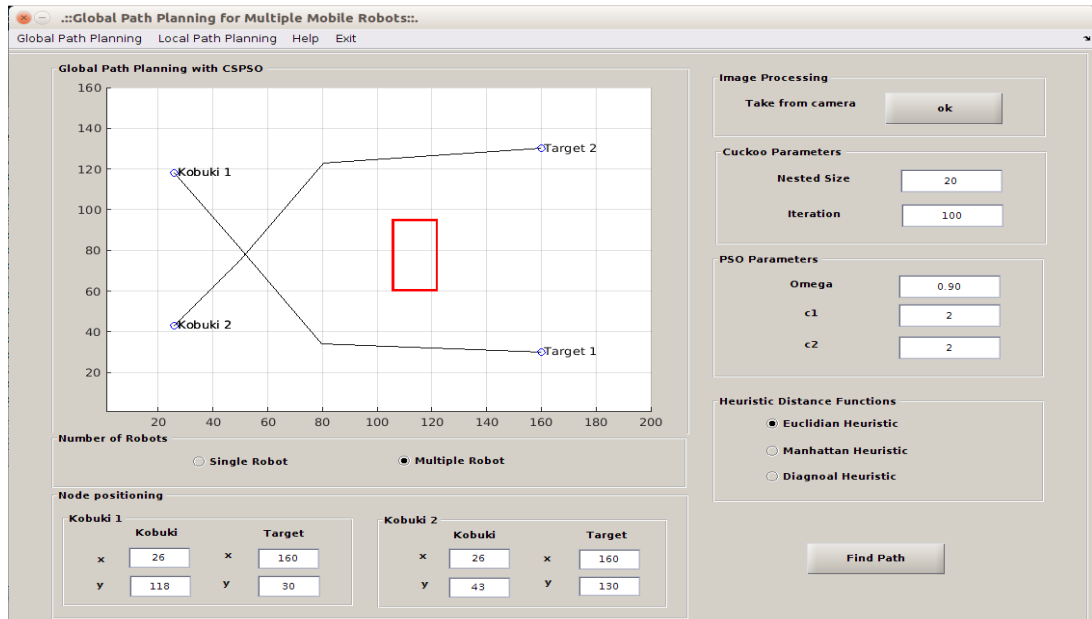
Tablo 1.1. Tekli mobil robot için özgün hibrit algoritmaların simülasyon ve deneysel analizinden elde edilen yol uzunluklarının karşılaştırılması

No	CS-PSO		CS-FA		CS-PSO-FA	
	Simülasyon (cm)	Deneysel (cm)	Simülasyon (cm)	Deneysel (cm)	Simülasyon (cm)	Deneysel (cm)
1	170,8190	143,9840	164,8035	154,6926	164,3069	144,2667
2	164,6911	144,3350	169,5267	143,9458	164,6483	144,6756
3	170,9685	144,3335	165,1175	143,9553	163,4247	144,8713
4	163,3983	149,2102	162,4956	149,7577	164,9759	143,5602
5	164,8487	144,0532	162,6763	144,8473	164,2183	144,2958
6	175,1582	142,5651	175,5749	145,9557	166,7813	144,2958
7	175,0556	145,2010	169,0251	145,3558	165,1424	143,4644
8	164,2391	193,3844	163,8008	144,5495	166,2212	142,3645
9	166,0237	144,4585	163,7681	144,4683	165,4503	159,4871
10	176,1780	144,3563	164,4550	145,0791	164,4290	144,2199
Ort	<b>169,1380</b>	<b>149,5881</b>	<b>166,1244</b>	<b>146,2607</b>	<b>165,0598</b>	<b>145,5501</b>

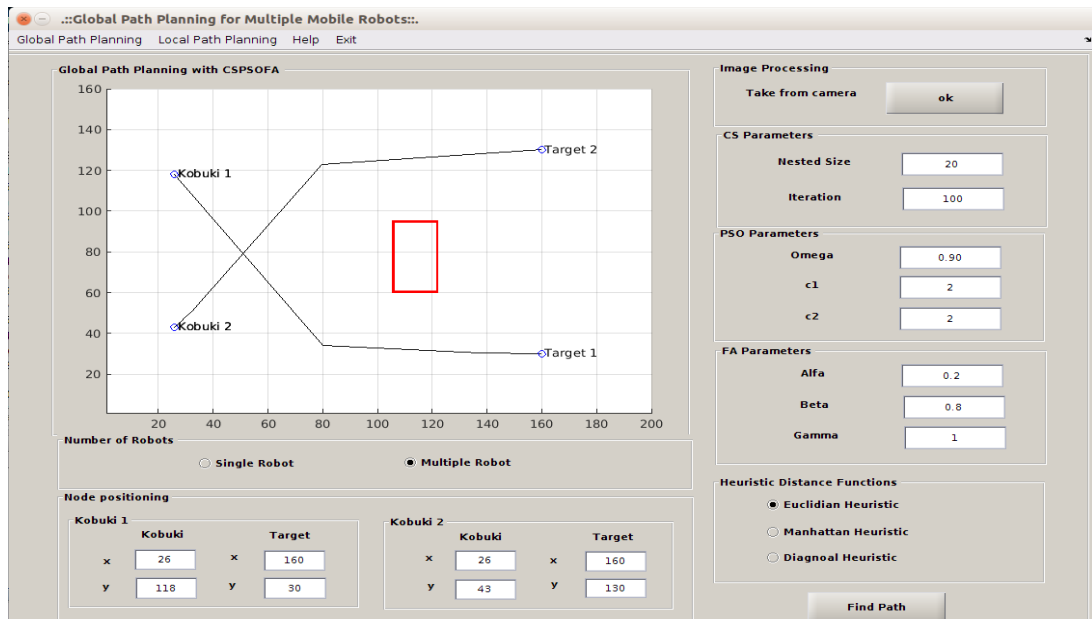
Tablo 2.1.(Devamı)

%Hata		
CS-PSO	CS-FA	CS- PSO- FA
11,55	11,95	11,81

Şekil 6.20. ve 6.21.'de gösterilen aynı ortam için tekli mobil robotun yol planlamasında kullanılan CS-PSO, CS-FA, CS-PSO-FA hibrit yol planlayıcılardan elde edilen simülasyon ve deney sonuçları Tablo 6.1'de verilmiştir. Yol uzunluğu performans parametresine göre deneysel ve simülasyon değerleri arasındaki ortalama hata yüzdesi sırasıyla yaklaşık olarak % 11,55, % 11,95 ve % 11,81'dir.

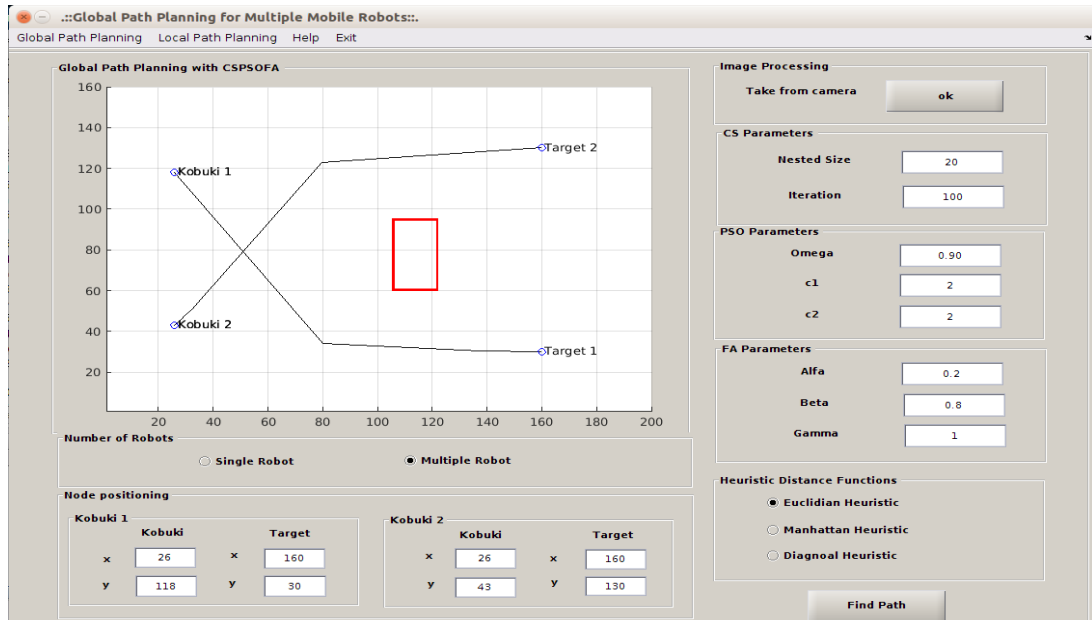


(a) CS-PSO



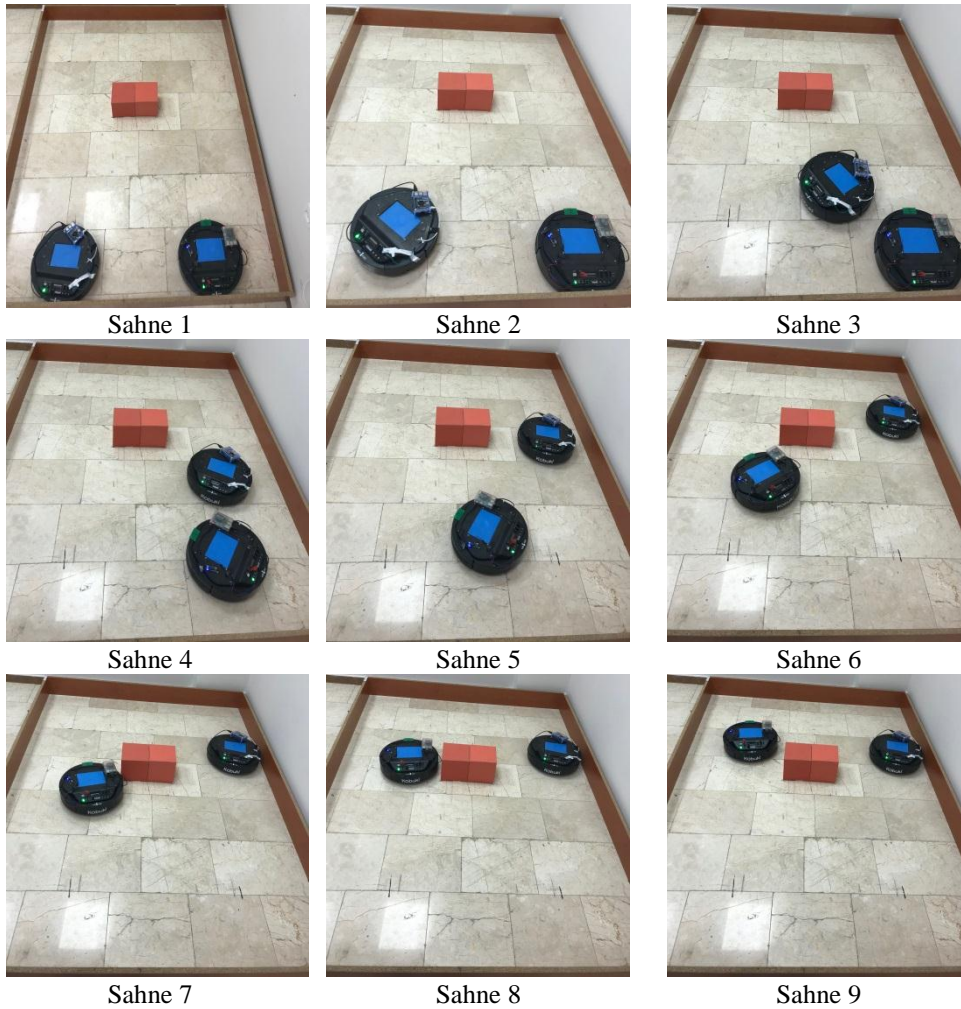
(b) CS-FA

Şekil 6.22(a-c). Çoklu mobil robotlar için geliştirilen hibrit algoritmalar tabanlı global yol planlama



(c) CS-PSO-FA

Şekil 6.22(a-c). (Devamı)



Şekil 6.23. Çoklu mobil robotlar için geliştirilen özgün CS-PSO-FA hibrit algoritmasının deneysel analizi

Tablo 6.2. Çoklu mobil robotlar için özgün hibrit algoritmaların simülasyon ve deneysel analizinden elde edilen yol uzunluklarının karşılaştırılması

No	Robot	CS-PSO		CS-FA		CS-PSO-FA	
		Simülasyon (cm)	Deneysel (cm)	Simülasyon (cm)	Deneysel (cm)	Simülasyon (cm)	Deneysel (cm)
1	Kobuki 1	180,0614	154,1328	179,9225	152,5079	179,9777	153,0185
	Kobuki 2	176,7536	201,2220	176,8110	199,7260	176,0908	200,2754
2	Kobuki 1	180,2307	154,2309	180,0669	152,7770	180,2100	154,3114
	Kobuki 2	176,9329	200,5548	176,8700	199,8705	176,9275	201,3440
3	Kobuki 1	180,0452	153,7398	179,9323	153,5577	180,0486	151,6640
	Kobuki 2	176,9625	200,8757	176,9045	200,4022	176,8432	200,3865
4	Kobuki 1	180,1072	153,5720	180,0941	152,8189	179,9483	153,4419
	Kobuki 2	176,7585	200,0114	176,8920	199,3974	176,9690	199,5225
5	Kobuki 1	180,0207	153,2959	180,0353	151,5573	179,0390	153,8643
	Kobuki 2	177,0868	200,4815	176,9112	199,8626	176,9415	201,4225
6	Kobuki 1	180,1665	151,6592	180,0825	154,5707	180,1068	151,7643
	Kobuki 2	176,9561	200,0236	176,8695	199,5223	176,8530	200,2163
7	Kobuki 1	180,0602	153,3169	180,0813	151,7434	179,9464	154,3929
	Kobuki 2	177,0537	201,0599	177,0198	200,6377	176,9801	201,4982
8	Kobuki 1	180,3187	154,0855	180,0813	151,9099	180,0503	152,0860
	Kobuki 2	176,8911	199,4981	176,8663	201,4090	176,8596	199,6687
9	Kobuki 1	180,0352	153,0869	180,0899	153,3819	180,2011	152,1070
	Kobuki 2	176,9324	200,7105	176,6508	199,4787	176,8888	200,1079
10	Kobuki 1	180,0240	153,3859	180,1352	152,9946	179,7800	153,9144
	Kobuki 2	177,0693	199,8641	177,2361	201,2632	176,9028	201,8960
Ort	<b>Kobuki 1</b>	<b>180,1069</b>	<b>153,4505</b>	<b>180,0521</b>	<b>152,7819</b>	<b>179,9777</b>	<b>152,8564</b>
	<b>Kobuki 2</b>	<b>176,9396</b>	<b>200,4301</b>	<b>176,9031</b>	<b>200,1569</b>	<b>176,0908</b>	<b>200,6338</b>

Tablo 6.3. Çoklu mobil robotlarda hibrit algoritmaların simülasyon ve deneysel çalışmalarının arasındaki ortalama hata oranları

% Ortalama Hata					
CS-PSO		CS-FA		CS-PSO-FA	
Kobuki 1	Kobuki 2	Kobuki 1	Kobuki 2	Kobuki 1	Kobuki 2
14,8	13,27	15,14	13,14	15,04	13,46

Tablo 6.3.'de, çoklu mobil robotun yol planlama sistemi için CS-PSO, CS-FA ve CS-PSO-FA özgün hibrit yol planlayıcılarının simülasyon ve deneysel sonuçlardan elde edilen yol uzunluk değerleri verilmiştir. Tablo 6.3.'e göre ortalama hata yüzdesi

Kobuki 1 için sırasıyla yaklaşık olarak % 14,8 ve % 15,14 ve % 15,04 dır ve simülasyon çalışmasındaki yol uzunluğu daha kısa iken Kobuki 2 için sırasıyla yaklaşık olarak %13,27, %13,14 ve %13,46 olarak hesaplanmıştır. Deneysel çalışmalardan elde edilen yol uzunluğu daha uzundur. Bunun sebebi olarak Kobuki 1 ve Kobuki 2 mobil robotun birbirine çarpması için gerekli olan bekleme süresidir.

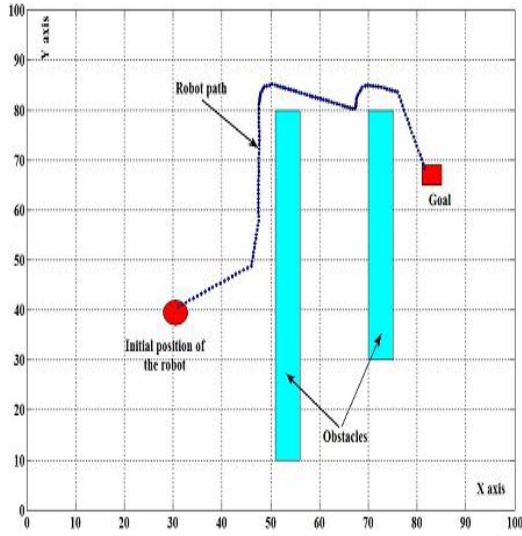
Deneysel analizlerden kobuki mobil robotların engellere çarpmadan başarıyla hedefe ulaştıkları açıkça görülmektedir. Geliştirilen yol planlayıcıları arasında en iyi performansa sahip olan hibrit CS-PSO-FA algoritmasıdır. Bundan dolayı çoklu mobil robotların yol planlamasında hibrit CS-PSO-FA algoritmasının yol planlama problemini çözmek için en uygun olduğu kanıtlanmıştır.

### **6.7. Geliştirilen Özgün Hibrit Algoritmaların Diğer Hibrit Algoritmalarla Karşılaştırılması ve Performans Analizi**

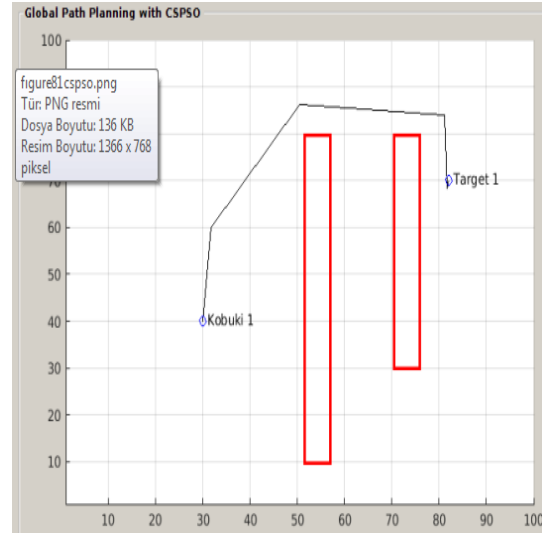
Geliştirilen hibrit algoritma tabanlı yol planlayıcılar ile literatürde gerçekleştirilmiş meta-sezgisel tabanlı hibrit algoritmali yol planlayıcılar en kısa/optimal yol uzunluğu belirlemek amacıyla performansları simülasyon ortamında karşılaştırılmıştır. Patle ve ark. FA tabanlı üç farklı hibrit algoritma yaklaşımı geliştirmişlerdir (Patle ve ark., 2016). Bu tez çalışmasında geliştirilen hibrit algoritmalar CS tabanlı olarak geliştirildiği halde içerisinde FA yaklaşımı da bulunmaktadır. Bu sebepten dolayı hibrit algoritmaların karşılaştırılması uygun bulunmuştur. Ayrıca algoritmaların her biri meta-sezgisel tabanlı algoritmadır.

Bu yaklaşımlar, farklı çevre ortamlarında tekli ve çoklu mobil robotların yol planlaması şeklinde karşılaştırılmıştır. Şekil 6.24. - 6.35. statik ortamda global yol planlaması için tekli mobil robotun simülasyon sonuçlarını gösterirken Şekil 6.36.- 6.41. ise çoklu mobil robotların simülasyon sonuçlarını göstermektedir.

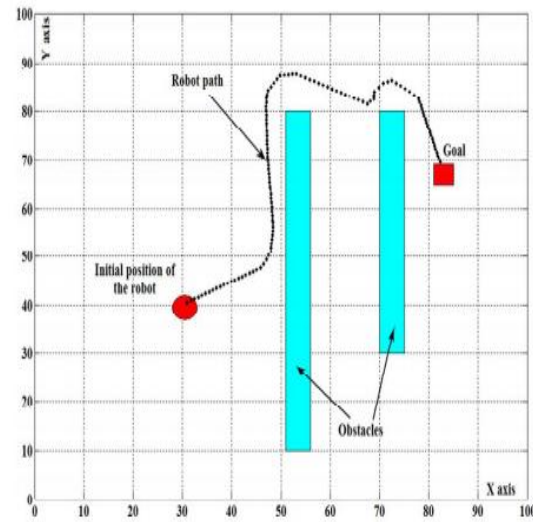




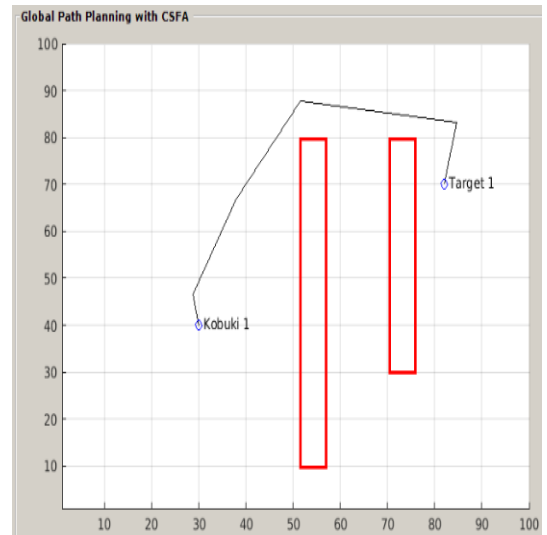
Şekil 6.24. FA-PFL tabanlı navigasyon



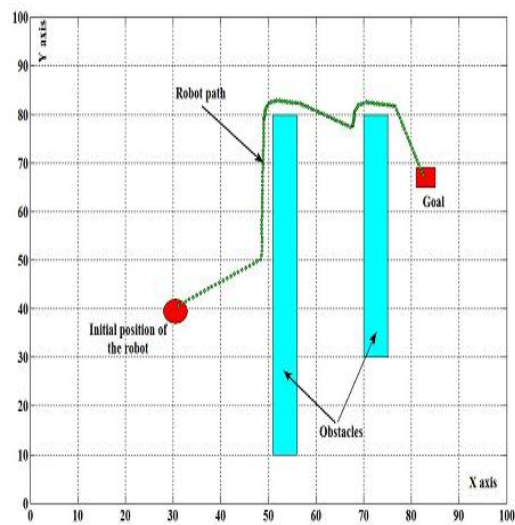
Şekil 6.25. CS-PSO tabanlı navigasyon



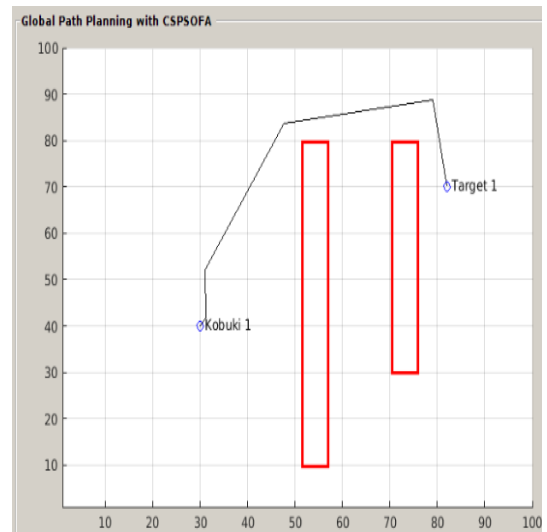
Şekil 6.26. FA-MGA tabanlı navigasyon



Şekil 6.27. CS-FA tabanlı navigasyon



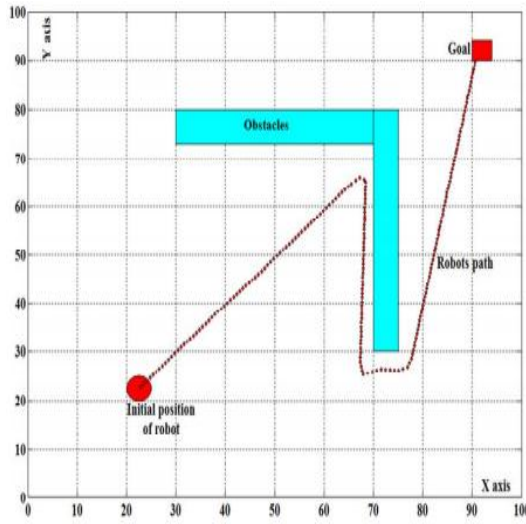
Şekil 6.28. FA-PFL-MGA tabanlı navigasyon



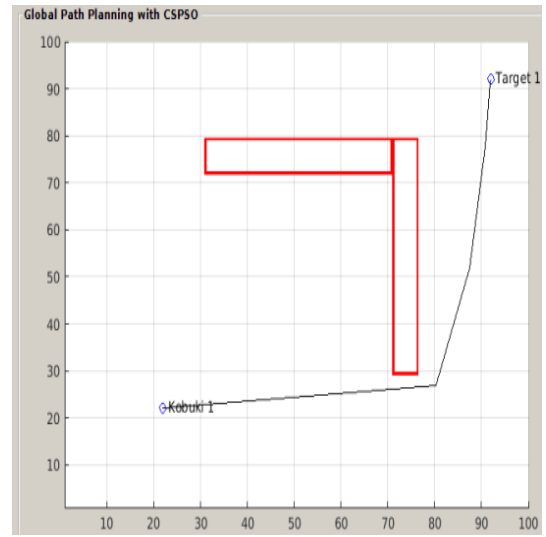
Şekil 6.29. CS-PSO-FA tabanlı navigasyon

Tablo 6.4. Statik bir ortamda yol uzunluğuna göre kıyaslanması(Şekil 6.24.-6.29.)

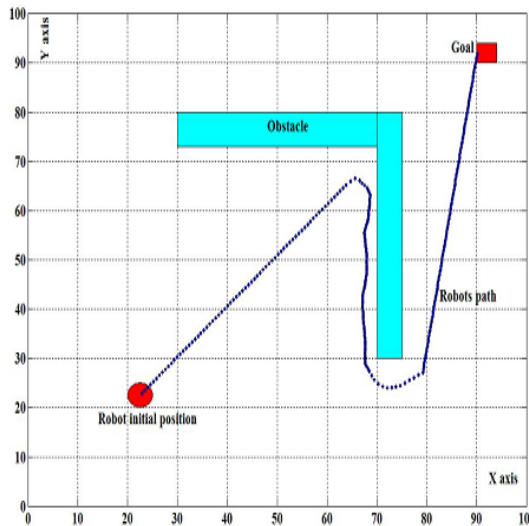
Algoritma	Simülasyondan elde edilen yol uzunluğu("cm")	% Performans Farkı
FA-PFL	146,5700(Şekil 6.24.)	32,40
FA-MGA	138,9500(Şekil 6.26.)	28,69
FA-PFL-MGA	136,6800(Şekil 6.28.)	27,50
CS-PSO	100,2400(Şekil 6.25.)	1,15
CS-FA	100,6918(Şekil 6.27.)	1,40
CS-PSO-FA	99,0875(Şekil 6.29.)	



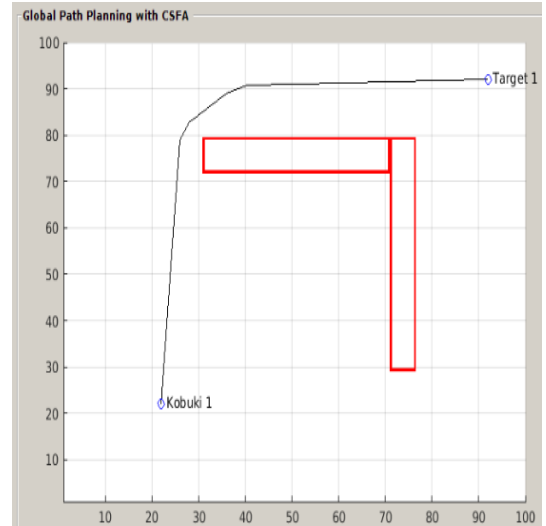
Şekil 6.30. FA-PFL tabanlı navigasyon



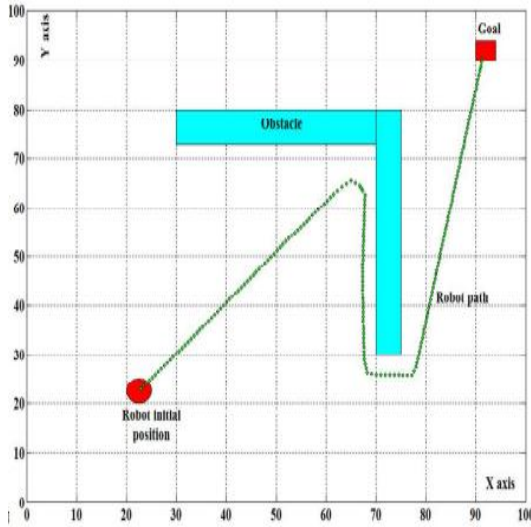
Şekil 6.31. CS-PSO tabanlı navigasyon



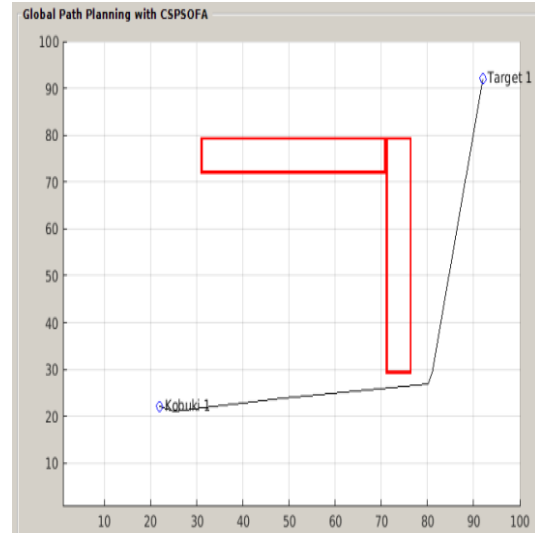
Şekil 6.32. FA-MGA tabanlı navigasyon



Şekil 6.33. CS-FA tabanlı navigasyon



Şekil 6.34. FA-PFL-MGA tabanlı navigasyon



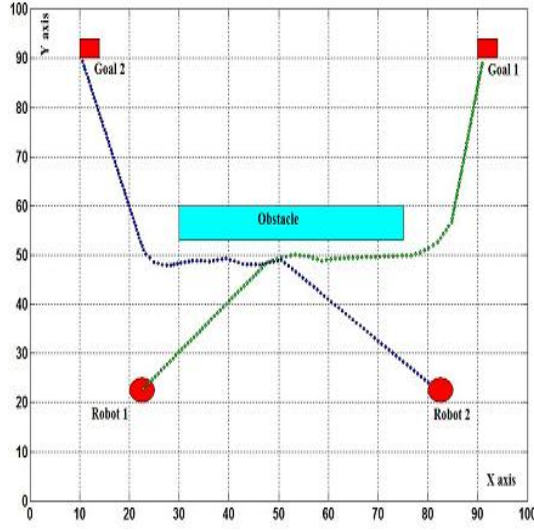
Şekil 6.35. CS-PSO-FA tabanlı navigasyon

Tablo 6.5. Statik bir ortamda tekli mobil robot için yol uzunluğuna göre karşılaştırma(Şekil 6.30.-6.35.)

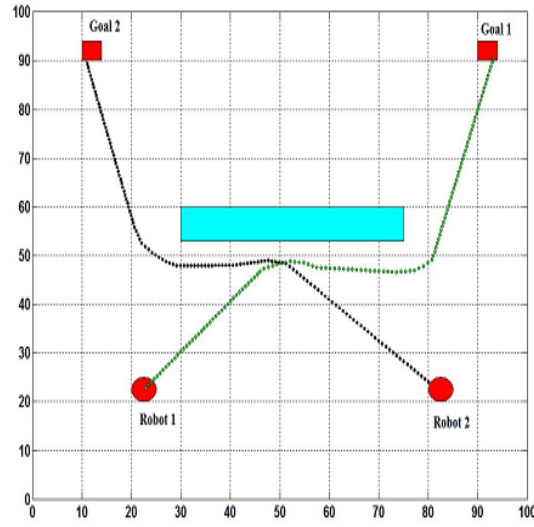
Algoritma	Simülasyondan elde edilen yol uzunluğu("cm")	% Performans Farkı
FA-PFL	262,5 (Şekil 6.29.)	52,47
FA-MGA	252,3 (Şekil 6.31.)	50,55
FA-PFL-MGA	239,7 (Şekil 6.33.)	47,95
CS-PSO	124,8676 (Şekil 6.30.)	0,08
CS-FA	127,9321 (Şekil 6.32.)	2,48
CS-PSO-FA	124,7551 (Şekil 6.34.)	

Şekil 6.24.-6.35.'de iki farklı çevre ortamının simülasyon analizlerinden elde edilen yol uzunluk değerlerine göre bu tez çalışmasında geliştirilen hibrit algoritmaların performansı Patle ve ark. tarafından geliştirilen hibrit algoritmalara göre daha iyi olduğu görülmektedir( Tablo 6.4-5.). Tekli mobil robotların yol planlamasında kullanılan algoritmalar arasında performans kriteri olarak en kısa/optimal yol uzunluğu seçildiğinde en iyi olan algoritma hibrit CS-PSO-FA algoritmadır. Hibrit algoritmalarda minimum performans kazancı yaklaşık olarak %27,50 iken maksimum performans kazancı yaklaşık olarak %52,47 olarak hesaplanmıştır. Bu karşılaştırmalara ek olarak geliştirilen hibrit algoritmalarda birbirleriyle karşılaştırılmıştır. Hibrit CS-PSO-FA algoritmasına göre diğer geliştirilen hibrit

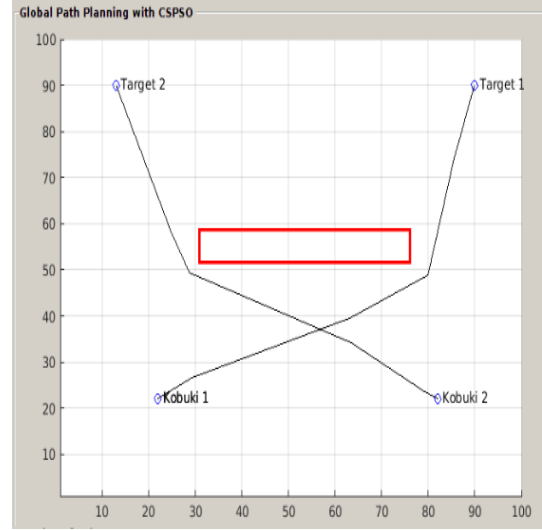
algoritmaların performansları karşılaştırıldığında yol uzunluklarının minimum % 0,08 iken maksimum % 2,48 daha kısa yol hesapladığı görülmektedir.



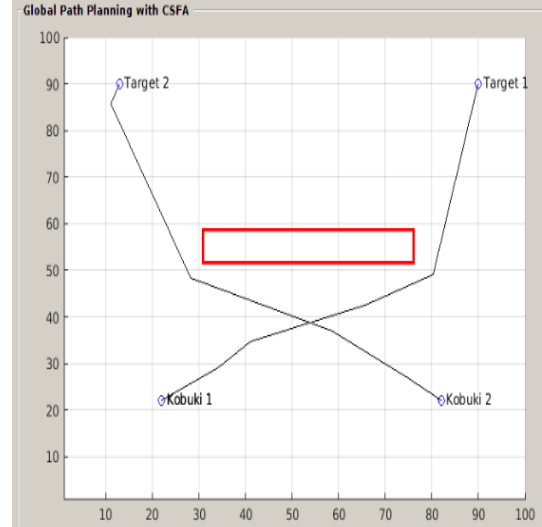
Şekil 6.36. FA-PFL tabanlı navigasyon (Patle ve ark.).



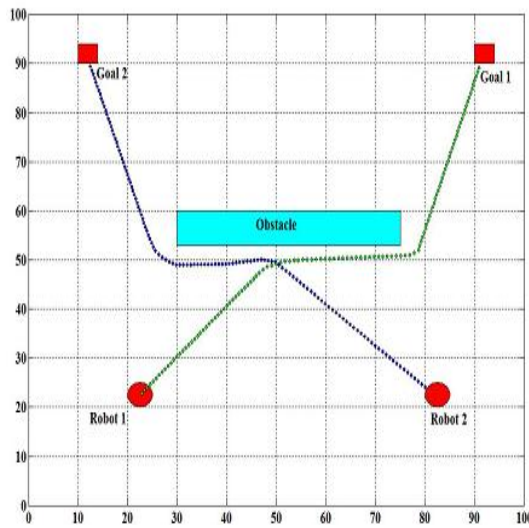
Şekil 6.38. FA-MGA tabanlı navigasyon



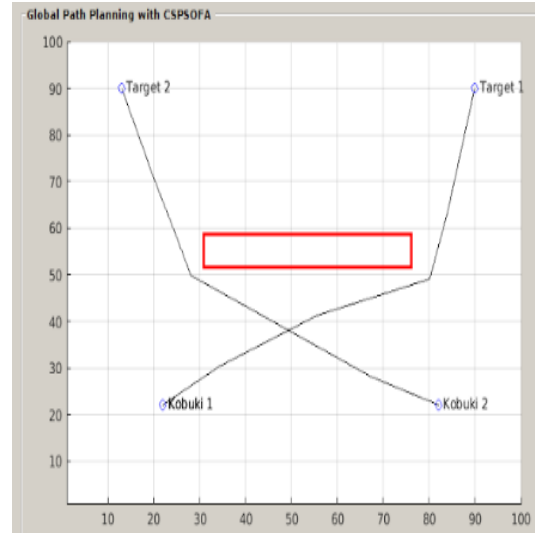
Şekil 6.37. CS-PSO tabanlı navigasyon



Şekil 6.39. CS-FA tabanlı navigasyon



Şekil 6.40. FA-PFL-MGA tabanlı navigasyon



Şekil 6.41. CS-PSO-FA tabanlı navigasyon

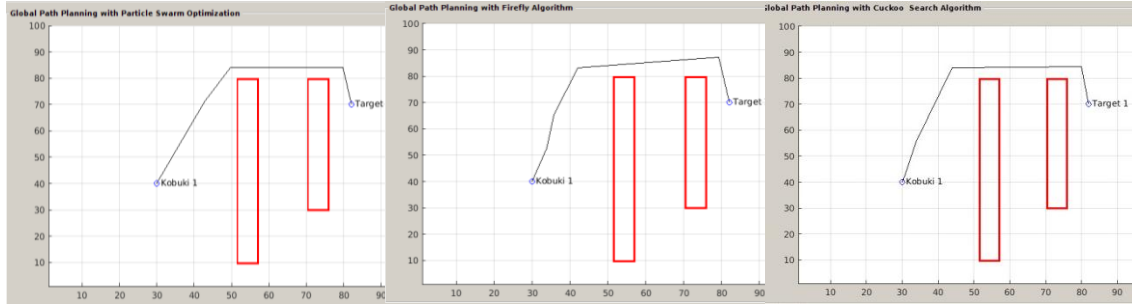
Tablo 6.6. Statik bir ortamda çoklu mobil robot için yol uzunluk değerlerini karşılaştırma

Algoritma	Robot	Simülasyondan elde edilen yol uzunluğu("cm")	% Performans Farkı
FA-PFL	Robot 1	159,23 (Şekil 6.36.)	33,06
	Robot 2	179,61 (Şekil 6.36.)	42,26
FA-MGA	Robot 1	156,82 (Şekil 6.38.)	32,03
	Robot 2	177,13 (Şekil 6.38.)	41,45
FA-PFL-MGA	Robot 1	154,47 (Şekil 6.40.)	31,00
	Robot 2	174,68 (Şekil 6.40.)	40,63
CS-PSO	Robot 1	<b>106,4490</b> (Şekil 6.37.)	0,12
	Robot 2	<b>103,7294</b> (Şekil 6.37.)	0,019
CS-FA	Robot 1	<b>106,9283</b> (Şekil 6.39.)	0,3
	Robot 2	<b>105,9792</b> (Şekil 6.39.)	2,14
CS-PSO-FA	Robot 1	<b>106,5756</b> (Şekil 6.41.)	
	Robot 2	<b>103,7093</b> (Şekil 6.41.)	

Şekil 6.36.-6.41.'de çoklu mobil robotların yol planlamasında aynı çevre ortamı kullanılarak gerçekleştirilen simülasyon çalışmalarında algoritmaların performans analizinden elde edilen sonuca göre CS-PSO-FA hibrit algoritması en kısa/optimal yol uzunluğunu vermiştir.

Patle ve arkadaşlarının geliştirdiği hibrit algoritmaların simülasyon sonuçları karşılaştırıldığında performans farkı minimum % 31 iken maksimum kazanç % 41,45

olarak hesaplanmıştır. Geliştirilen hibrit CS-PSO-FA algoritması diğer hibrit algoritmalara göre performans kazançları bakımından minimum % 0,12 iken maximum % 2,14 daha kısa yol hesapladığı görülmektedir.



(a) PSO

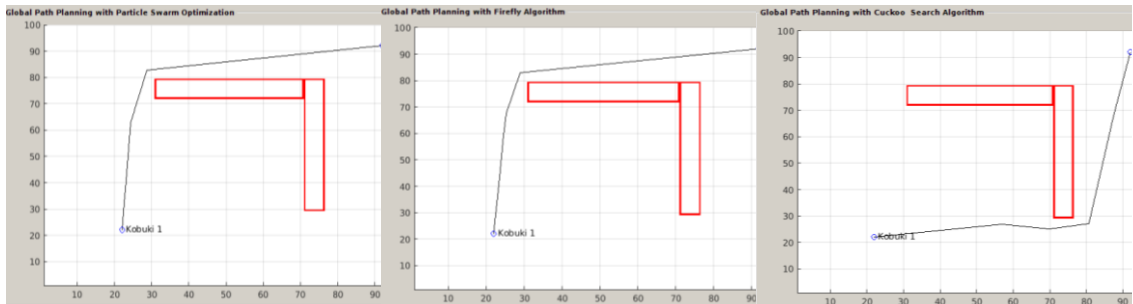
(b) FA

(c) CS

Şekil 6. 42. Statik bir ortamda tekli mobil robot için yapılan yol planlama

Tablo 6.7. Statik bir ortamda yol uzunluk değerlerine göre kıyaslama(Şekil 6.42.)

Algoritma	Simülasyondan elde edilen yol uzunluğu(cm)
PSO	94,7507 (Şekil 7.1a.)
FA	98, 2546 (Şekil 7.1b.)
CS	98, 2546 (Şekil 7.1c.)
CS-PSO	<b>94,7774 (Şekil 6.24.)</b>
CS-FA	<b>96, 4864 (Şekil 6.26.)</b>
CS-PSO-FA	<b>93,4710(Şekil 6.28.)</b>



(a) PSO

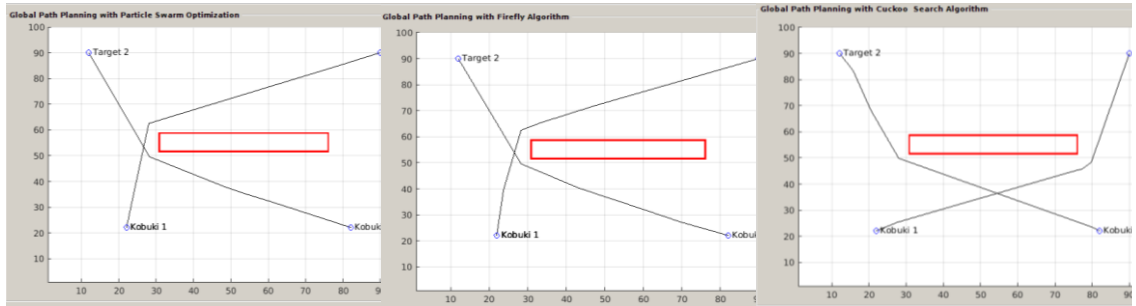
(b) FA

(c) CS

Şekil 6. 43. Statik bir ortamda tekli mobil robot için yol planlama

Tablo 6.8. Statik bir ortamda tekli mobil robot için yol uzunluğuna göre karşılaştırma(Şekil 6.43.)

Algoritma	Simülasyondan elde edilen yol uzunluğu("cm")
PSO	125,3585 (Şekil 7.2a.)
FA	134,8114 (Şekil 7.2b.)
CS	125, 2856(Şekil 7.2c.)
CS-PSO	<b>124,8676 (Şekil 6.30.)</b>
CS-FA	<b>127,9321 (Şekil 6.32.)</b>
CS-PSO-FA	<b>124,7551 (Şekil 6.34.)</b>



(a) PSO

(b) FA

(c) CS

Şekil 6.44. Statik bir ortamda çoklu mobil robot için yol planlama

Tablo 6.9. Statik bir ortamda çoklu mobil robot için yol uzunluğuna göre karşılaştırma(Şekil 6.44(a-c))

Algoritma	Robot	Simülasyondan elde edilen yol uzunluğu("cm")
PSO	Robot 1	108,79 (Şekil 7.3a.)
	Robot 2	104,00 (Şekil 7.3a.)
FA	Robot 1	108,70 (Şekil 7.3b.)
	Robot 2	104,03 (Şekil 7.3b.)
CS	Robot 1	106,73 (Şekil 7.3c.)
	Robot 2	104,22 (Şekil 7.3c.)
CS-PSO	Robot 1	<b>106,4490 (Şekil 6.36.)</b>
	Robot 2	<b>103,7294 (Şekil 6.36.)</b>
CS-FA	Robot 1	<b>106,9283 (Şekil 6.38.)</b>
	Robot 2	<b>105,9792 (Şekil 6.38.)</b>
CS-PSO-FA	Robot 1	<b>106,5756 (Şekil 6.40.)</b>
	Robot 2	<b>103,7093 (Şekil 6.40.)</b>

## BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasının amacı, meta-sezgisel algoritma tabanlı mobil robot yol planlaması için özgün bir hibrit algoritma geliştirmektir. Geliştirilen hibrit algoritmalar ile yol minimize edilerek yol planlamanın verimliliğinin ve performansının artırılması hedeflenmektedir. Bu çalışmada üzerinde engeller bulunan statik bir ortamda tekli ve çoklu mobil robotların yol planlaması için özgün hibrit bir algoritma geliştirildi ve uygulandı. PSO(Parçacık Sürü Optimizasyonu), FA (Ateş Böceği Algoritması), CS (Guguk Kuşu Algoritması) gibi meta-sezgisel algoritmalarından her biri ayrı ayrı kullanılarak elde edilen sonuçlar benzer çalışmalarla karşılaştırıldı. Daha sonra aynı ortam şartları ve robotlar için yol planlamaları yapmak üzere çalışmaya özgü CS-PSO(Guguk Kuşu Algoritması-Parçacık Sürü Optimizasyonu), CS-FA(Guguk Kuşu Algoritması-Ateş Böceği Algoritması), CS-PSO-FA (Guguk Kuşu Algoritması-Parçacık Sürü Optimizasyonu-Ateş Böceği Algoritması) hibrit algoritmalar geliştirildi. Geliştirilen tüm algoritmalar ile mobil robotların engellere çarpmaksızın yol planlama sağladıkları görülmüştür.

Önerilen çalışmadan elde edilen en önemli sonuçlar şunlardır:

- Önerilen PSO, FA ve CS tabanlı yol planlayıcı ile çoklu mobil robotlar statik bir ortamda engellere çarpmaksızın hedefe etkin bir şekilde en kısa/optimum yoldan ulaşmışlardır.
- PSO yol planlayıcıda parçacıkların en iyi konum bulma, FA yol planlayıcıda ateş böceklerinin daha parlak ve çekici olana doğru hareket etme, CS yol planlayıcısının ise en iyi konak yuva bulma özellikleri ile algoritmaların performansları artırılmıştır.
- CS algoritma tabanlı geliştirilen CS-PSO, CS-FA ve CS-PSO-FA hibrit yol planlayıcılar çoklu mobil robotların yol planlama amacıyla geliştirilmiştir.



Hibrit yol planlayıcıların tabanında bulunan CS algoritmasının seçilme nedeni diğer algoritmalara göre daha optimum ve kararlı sonuçlar vermesidir. Ek olarak CS algoritmasının global ve yerel arama özellikleri arasındaki denge vasıtasıyla yerel arama özelliği sağlanmıştır. Hibrit algoritmalar oluşturulurken CS algoritmasının yerel arama özelliğine diğer algoritmaların(PSO, FA) spesifik özellikleri eklenmiştir.

- Geliştirilen yol planlama algoritmaların etkinliği, fizibilitesi ve uygulanabilirliği hem MATLAB yazılımında simülasyon ortamında hem de Kobuki mobil robot kullanılarak gerçek zamanlı olarak deneysel ortamda analiz edilmiştir. Ek olarak geliştirilen algoritmaların etkinliği ve uygunluğu doğrulamak amacıyla literatürde diğer araştırmacıların geliştirdiği yöntemler simülasyon ortamında karşılaştırmalı olarak sunulmuştur.
- CS-PSO-FA hibrit algoritmasıyla elde edilen yolun diğer algoritmalara göre daha kısa olduğu ve böylece daha yüksek bir performansa sahip olduğu kanıtlanmıştır.
- Tek bir mobil robot için yol planlama, aynı simülasyon ortamında çalıştırılan algoritmaların performans analizleri Şekil 6.42., Şekil 6.43. ve Şekil 6.23.-6.34.'den elde edilen sonuçlara göre yapılmıştır. Simülasyon çalışmalarından elde edilen yol uzunlukları Tablo 6.7. ve 6.8.'de verilmiştir. Sonuçlara göre hibrit algoritmanın bileşenlerinden biri olan CS algoritması, PSO ve FA yol planlayıcılarına göre yol uzunluk ve hassaslık açısından daha iyi sonuçlar vermiştir. Hibrit algoritmalarda ise CS-PSO-FA yaklaşımı hem hibrit hem de bireysel algoritmalara göre daha etkin ve iyi sonuçlar vermiştir.
- Şekil 6.35. - 6.40. ve Şekil 6.44.'de görüldüğü gibi simülasyon ortamında gerçekleştirilen çoklu mobil robotların yol planlamadaki performansları karşılaştırılmıştır. Yol uzunluk performans kriterine göre yol planlayıcılardan bireysel algoritmalarda en iyi sonuç veren algoritma CS iken, hibrit yol planlayıcılarda ise CS-PSO-FA algoritmasıdır.

- Başlangıç ve hedef noktası arasındaki yol planlama için gerçekleştirilen simülasyon ve deneysel sonuçlar karşılaştırıldığında yol uzunluğu açısından geliştirilen algoritmalarda(PSO, FA, CS, CS-PSO, CS-FA, CS-PSO-FA) ortalama hata yüzdeleri sırasıyla %16, %16, %18, %11, %11, %11 olarak hesaplanmıştır.
- Önerilen yaklaşımlar tekli veya çoklu mobil robotlarla çeşitli çalışma ortamlarında aynı veya farklı hedef konumlarıyla simülasyon ve deneysel ortamda test edilmiştir.

Sonuç olarak, tez çalışması kapsamında; PSO, FA, CS ve geliştirilen özgün hibrit algoritmalar (CS-PSO, CS-FA, CS-PSO-FA) mobil robotların yol bulma problemi için uyarlandı, sisteminin tamamı gerçek bir laboratuvar ortamında test edildi ve uygulama için bir arayüz tasarlandı. Yol planlamada meta-sezgisel algoritmalar arasında en iyi performans gösteren CS algoritması ve hibrit algoritmalarda ise hibrit CS-PSO-FA algoritması olduğu kanıtlanmıştır.

Elde edilen çalışma sonuçlarının endüstriyel robotlara uyarlanmasıyla önemli oranda zaman ve enerji, tasarrufu sağlayarak verimlilik artışına katkıda bulunabileceği ümit edilmektedir.

Tez çalışması kapsamında makale ve bildiri çalışmaları yapılmıştır. Yayın ile ilgili ilave bilgiler tez çalışmasının sonundaki Ekler bölümünde yer almaktadır.

### **7.1. Önerilen Çalışmalar**

1. Ortamın haritası mobil robot üzerine yerleştirilecek LIDAR veya 360° görüntü alabilen kinect kamera gibi bazı eklemeler yapılarak sistem geliştirilebilir. Böylece, robot kısıtlı bir laboratuvar ortamında değil de istenen herhangi bir ortamda da çalıştırılabilir.
2. Mobil robot yol planlama tasarımında farklı meta-sezgisel algoritmalar kullanılarak hibrit algoritmalar geliştirilebilir.

3. Mobil robot çalışma ortamı engeller ve hedef noktası hareketli olarak belirlenebilir.

## KAYNAKLAR

- Abbas, N.H. Mahdi, F.A. 2014. Path Planning of an Autonomous Mobile Robot using Directed Artificial Bee Colony Algorithm, *International Journal of Computer Applications* Volume 96(11): 0975 – 8887.
- Ahmed, F. Deb, K. 2013. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Computing*, 17(7): 1283–1299.
- Agarwal, M. Goel, P. 2013. Path Planning of Mobile Robots using Bee Colony Algorithm. *International Journal of Computer Science & Information Technology*, 3(2): 86–89.
- Apostolopoulos, T. Vlachos, A. 2011. Application of the firefly algorithm for solving the economic emissions load dispatch problem, *Hindawi Publishing Corporation International Conference Journal of Combinatorics*.
- Bakdi, A. Hentout, A. Boutami, H. Maoudj, A. Hachour, O. Bouzouiaa, B. 2017. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robotics and Autonomous Systems* 89: 95–109.
- Başbuğ, S. 2008. Bakteriyel Besin Arama Algoritması ile Lineer Anten Dizilerinin Diyagram Sıfırlaması, *Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi*.
- Bhadoria, A. Singh, R.K. 2014. Optimized Angular a Star Algorithm for Global Path Search Based on Neighbor Node Evaluation. *I.J. Intelligent Systems and Applications*, 46-52.
- Bilbeisi, G. Al-Madi, N. Awad, F. 2015. PSO-AG: a multi-robot path planning and obstacle avoidance algorithm. *Applied Electrical Engineering and Computing Technologies*.
- Boğar, E. Beyhan, S. 2013. A hybrid genetic algorithm for mobile robot shortest path problem, *IJISAE*, 1(4): 264–267.
- Brand, M. Yu, X.H. 2013. Autonomous robot path optimization using firefly algorithm . *Machine Learning and Cybernetics*, Tianjin.
- Chaari, I. Koubaa, A. Bennaceur, H. Trigui, S. Al-Shalfan, K. 2012. smartPATH: A hybrid ACO-GA algorithm for robot path planning. *WCCI*, Brisbane, Australia.
- Châari, I. Koubaa, A. Trigui, S. Bennaceur, H. Ammar, A. Al-Shalfan, K. 2014. SmartPATH: an efficient hybrid ACO-GA algorithm for solving the global path planning problem of mobile robots, *International Journal of Advanced Robotic Systems*.

- Chang, H. Jin, T. 2013. Command fusion based fuzzy controller design for moving obstacle avoidance of mobile robot. *Future Information Communication Technology and Applications*, 905–913.
- Chang, W.C. Ling, H.C. 2014. *Visual Navigation and Control of Mobile Robots Based on Environment Mapping*. Advanced Mechatronic Systems.
- Chatterjee, A. Pulasinghe, K. Watanabe, K. Izumi, K. 2005. A particle-swarm-optimized fuzzy-neural network for voice controlled robot systems. *IEEE Transactions on Industrial Electronics*, 52(6):1478–1489.
- Chen, W. Qin, H. 2011. Path planning of mobile robot based on hybrid cascaded genetic algorithm, *Intelligent Control and Automation*, 501-504.
- Chia, S.H. Su, K.L. Guo, H. Chung, C.Y. 2010. Ant Colony System Based Mobile Robot Path Planning. *Fourth International Conference on Genetic and Evolutionary Computing*.
- Contreras-Cruz, M.A. Ayala-Ramirez, V. Hernandez-Belmonte, U.H. 2015. Mobile robot path planning using artificial bee colony and evolutionary programming, *Applied Soft Computing* 30: 319–328.
- Çankaya İ., Akgün D., Kaçar S., 2013. *Mühendislik uygulamaları için MATLAB*, Seçkin Yayıncılık.
- Dao, T.K. Pan, J.S. Pan, T.S. Nguyen, T. 2017. Optimal path planning for motion robots based on bees pollen optimization algorithm, *Journal of Information and Telecommunication*, 1(4): 4, 351–366.
- Das, P.K. Behera, H.S. Das, S. Tripathy, H.K. Panigrahi, B.K. Pradhan, S.K. 2016. A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment, *Neuro computing* 207: 735–753.
- Das, P.K. Behera, H.S. Panigrahi, B.K. 2016. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation*, 28:14–28.
- Deepak, B. Parhi, D.R. 2012. Pso based path planner of an autonomous mobile robot. *Cent. Eur. J. Comp. Sci.* 2(2):152-168.
- Deepu, R. Honnaraju, B. Murali, Sc. 2015. Path Generation for Robot Navigation using a Single Camera. *Procedia Computer Science*, 1425 – 1432.
- Dhaouadi, R. Hatab, A.A. 2013. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: a unified framework, *Advances in Robotics & Automation*, 2 (2).
- Fister, I. Yang, X.S. Brest, J. 2013. A comprehensive review of firefly algorithms, *Swarm and Evolutionary Computation*, 34–46.
- Fister, I. Yang, X.S. Fister, D. 2013. Cuckoo search: a brief literature review. *Cuckoo Search and Firefly Algorithm*, 49-62.
- Gandomi, A.H. Yang, X.S. Alavi, A.H. 2011. Mixed variable structural optimization using firefly algorithm, *Computers and Structures*, 89(23–24): 2325–2336.

- Gandomi, AH. Yang, XS. Alavi, AH. 2013. Cuckoo search algorithm: a meta heuristic approach to solve structural optimization problems. *Eng Comput*, 29(1):17–35.
- Gao, Z. Zeng, X. Wang, J. Liu, J. 2008. FPGA implementation of adaptive IIR filters with particle swarm optimization algorithm. *Communication Systems*.
- Gigras, Y. Choudhary, K. Gupta, K. Vandana. 2015. A hybrid ACO-PSO technique for path planning. *INDIACom*, 1616-1621.
- Hamani, M. Hassam, A. 2013. Mobile robot navigation in unknown environment using improved Apf method. *Arab Conference on Information Technology*, 444-456.
- Hayat, S. Kausar, Z. 2015. Mobile Robot Path Planning for Circular Shaped Obstacles Using Simulated Annealing. *Control, Automation and Robotics*.
- Hidalgo-Paniagua, A. Vega-Rodríguez, M.A. Pavón, J.F.N. 2017. Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach. *Soft Comput*, 21: 949–964.
- Huang, H.C. Tsai, C.C. 2011. Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm. *SICE*, 1338-1343.
- Jianguo, W. Biao, D. Guijuan, M. Jianwu, B. Xuedong Y. 2011. Path Planning of Mobile Robot Based on Improving Genetic Algorithm. *ICCE*, 535–542.
- Jing, X. Xiao-Liang, J. Yin, Y. Ding, L. 2016. Visual Navigation for Mobile Robot with Kinect Camera in Dynamic Environment. *Proceedings of the 35th Chinese Control Conference*, Chengdu, China.
- Joshi, H.N. Shinde, J.P. 2014. An Image Based Path Planning and Motion Planning for Autonomous Robot. *International Journal of Computer Science and Information Technologies*, 5 (4): 4844-4847.
- Ju, M.Y. Wang, S.E. Guo, J.H. 2014. Path planning using a hybrid evolutionary algorithm based on tree structure encoding, Hindawi Publishing Corporation *Scientific World Journal*.
- Jun, H. Qingbao, Z. 2010. Multi-objective mobile robot path planning based on improved genetic algorithm. *ICICTA*, 2: 752–756.
- Karaboga, D. 2005. An idea based on honey bee swarm for numerical optimization. *Technical Report*.
- Karaboğa, D. 2011. *Yapay Zekâ Optimizasyon Algoritmaları*, Nobel Yayın Dağıtım.
- Karagül, K. 2014. Guguk Kuşu Algoritması: Bir Plastik Atık Toplama Uygulaması. *Operations Research and Statistic*, 15: 775-784.
- Kaya, T. İnce, M. C. 2008. Genetik algoritmaların aktif filtrelerde kullanımı. *ELECO*, 512-515.
- Kennedy, J. Eberhart, R.C. 1995. Particle Swam Optimization. *IEEE International Conference on Neural Networks*, 1942-1948.

- Krishnanand, K.N. Ghose, D. 2005. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, *IEEE Swarm Intelligence*, 84-91.
- Li, B. Liu, L. Zhang, Q. Lv, D. Zhang, Y. Zhang, J. Shi, X. 2014. Path planning based on firefly algorithm and Bezier curve. *Information and Automation*, Hailar, China.
- Liu, C. Gao, Z. Zhao, W. 2012. A new path planning method based on firefly algorithm, *Computational Sciences and Optimization*.
- Liu, Y. Bharadwaj, K. K. 2011. A hybrid artificial potential field: genetic algorithm approach to mobile robot path planning in dynamic environments, *Computer Science and Convergence*, 325-333.
- Mohanty, P.K. Dayal, R.P. 2013. Cuckoo search algorithm for the mobile robot navigation, *SEMCCO*, 527–536.
- Mohanty, P.K. Parhi, D.R. 2015. A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach, *Memetic Computing*, 7:255–273.
- Mohanty P.K. Parhi, D.R. 2016. Optimal path planning for a mobile robot using cuckoo search algorithm, *Journal of Experimental & Theoretical Artificial Intelligence*, 28(1–2): 35–52
- Montiel, O. Orozco-Rosas, U. Sepúlveda R. 2015. Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles. *Expert Systems with Applications*. 42(12): 5177-5191.
- Mnubi, S.A. 2016. Motion Planning and Trajectory for Wheeled Mobile Robot. *International Journal of Science and Research*, 5(1).
- Nandanwar, M.K. Janghel, R.R. Zadagaonkar, A.S. 2016. Path Planning in Outdoor and Unknown Environment using BAPSO Algorithm, *IJRASET*, 4(3): 177-184.
- Nasrollahy, A. Z. Javadi, H. H. S. 2009. Using particle swarm optimization for robot path planning in dynamic environments with moving obstacles and target. *Sim European Symposium on Computer Modeling and Simulation*.
- Özkan, S.S. Karayel, D. Atalı, G. 2016. Morfolojik Görüntü İşleme Tekniği ile Yapay Sinir Ağlarında Görüntü Tahribat Analizi, *APJES* 4(1): 01-07.
- Panda, M.R. Priyadarshini, R. Pradhan, S.K. 2016. Autonomous Mobile Robot Path Planning Using Hybridization of Particle Swarm Optimization and Tabu Search. *Computational Intelligence and Computing Research*.
- Pandey, A. Parhi, D.R. 2017. Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind driven optimization algorithm. *Defence Technology* 13: 47-58.
- Parhi, D.R. Jha, A.K. 2012. Path Planning of Mobile Robot using Bacteria Foraging Optimization. *IJAICR*, 4(1): 1-5.
- Patle, B.K. Parhi, D.R. 2017. On firefly algorithm: optimization and application in mobile robot navigation”, *World Journal of Engineering* 14(1): 65–76.

- Patle, B.K. Parhi, D. Jagadeesh, A. Sahu, O. P. 2017. Real time navigation approach for mobile robot. *Journal of Computers*,12(2).
- Pradhan, S.K. Parhi, D.R. Panda, A. K. 2009 Fuzzy logic techniques for navigation of several mobile robots. *Applied Soft Computing*, 290–304.
- Poli, R. 2008. Analysis of the publications on the applications of particle swarm optimisation. Hindawi Publishing Corporation *Journal of Artificial Evolution and Applications*.
- Purcaru, C. Precup, R.E. Iercan, D. Fedorovici, L.O. David, R.C. 2013. Hybrid PSO-GSA robot path planning algorithm in static environments with danger zones. *ICSTCC*, 434-439
- Purian, F.K. Sadeghian, E. 2013. Mobile robots path planning using ant colony optimization and Fuzzy Logic algorithms in unknown dynamic environments. *Automation, Robotics and Embedded Systems*.
- Raja, N.S.M. Manic, K.S. Rajinikanth, V. 2013. Firefly Algorithm with Various Randomization Parameters: An Analysis. *SEMCCO*, 110–121.
- Renawi, A. Mohammad, A. J. Abdel-Hafez M. 2017. ROS validation for non-holonomic differential robot modeling and control case study: Kobuki robot trajectory tracking controller. *Modeling, Simulation, and Applied Optimization*.
- Rini, D.P. Shamsuddin, S.M. Yuhaniz, S.S. 2011. Particle swarm optimization: technique, system and challenges, *International Journal of Computer Applications* 14(1): 0975 – 8887.
- Samadi, M. Othman, M. F. 2013. Global Path Planning for Autonomous Mobile Robot using Genetic Algorithm. *International Conference on Signal-Image Technology & Internet-Based Systems*.
- Saska, M. Macas, M. Preucil, L. Lhotska, L. 2006. Robot path planning using particle swarm optimization of ferguson splines. *Emerging Technologies and Factory Automation*.
- Sevilgen, A.F.E. İnce, O. 2007. Sensör yerleştirme probleminin parçacık sürü optimizasyonu ile çözümü. *Elektrik-Elektronik-Bilgisayar Mühendisliği*.
- Shamsfakhr, F. Sadeghibigham B. 2017. A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments. *Turkish Journal of Electrical Engineering & Computer Sciences*, 25: 1629 – 1642.
- Shehaba, M. Khadera, A.T. Al-Betarb, M.A. 2017. A survey on applications and variants of the cuckoo search algorithm, *Applied Soft Computing*, 1-42.
- Shi, C. Bu, Y. Liu, J. 2008. Mobile robot path planning in three-dimensional environment based on ACO-PSO hybrid algorithm, *IEEE/ASME*, 252-256, Xi'an, China.
- Shi, C. Bu, Y. Li, Z. 2008. Path planning for deep sea mining robot based on ACO-PSO hybrid algorithm, *Intelligent Computation Technology and Automation*, 125-129.



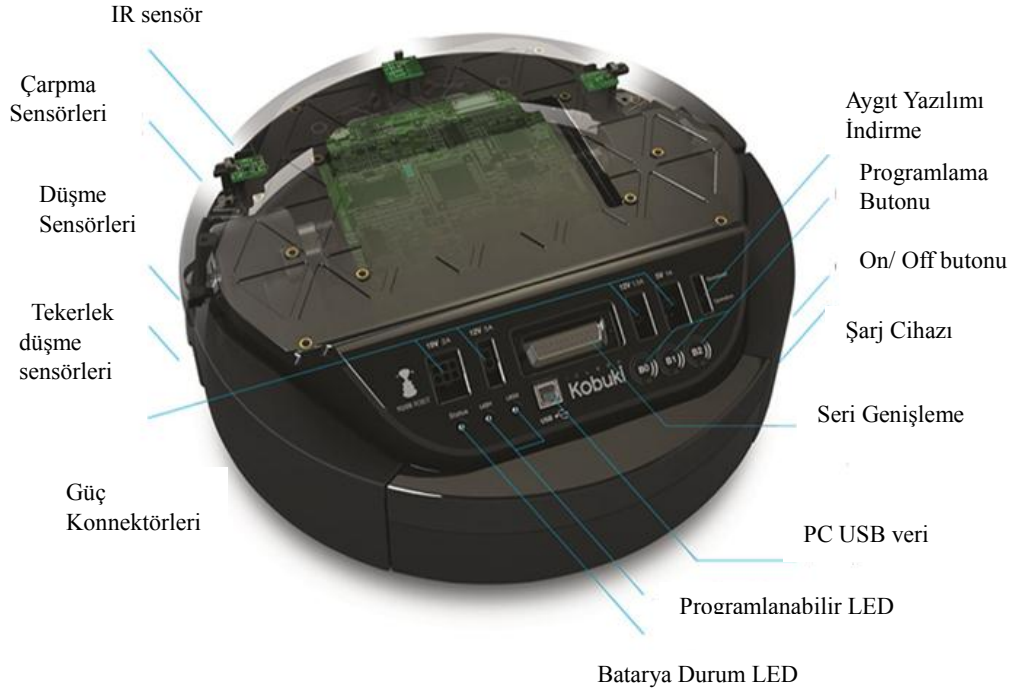
- Siegwart, R. Nourbakhsh, I.R. 2004. Introduction to Autonomous Mobile Robot. Massachusetts Institute of Technology Press, Cambridge, USA.
- Solea, R. Cernega, D. 2016. Online path planner for mobile robots using particle swarm optimization, ICSTCC, Sinaia, Romania.
- Stenz A. 1994. Optimal and efficient path planning for partially-known environments. Proceedings of the IEEE International Conference on Robotics and Automation, 4: 3310-3317.
- Sung, P.K. Hong, K.B. Hong, S.K. Hong, S.C. 1999. Path Planning of Mobile robot using Neural Network. ISIE, Bled, Slovenia.
- Quigley, M. Gerkey, B. Conley, K. Faust, J. Foote, T. Leibs, J. Berger, E. Wheeler, R. Ng, A. 2009. ROS: an open-source Robot Operating System, ICRA.
- Tan, A. Chen, D. 2009. Hybrid approach of path planning for mobile robots based on the combination of ACO and APF algorithms. Intelligent Systems and Applications.
- Tang, Y. Li, Q. Wang, L. Zhang, C. Yin, Y. 2010. An improved pso for path planning of mobile robots and its parameters discussion, Intelligent Control and Information Processing, Dalian, China
- Thomaz, C. Pacheco, M. Vellasco, M. 2006. Mobile Robot Path Planning Using Genetic Algorithms. Lecture Notes in Computer Science, Springer Book Series. 1606: 671-679.
- Tu, J. Yang, S. X. 2003. Genetic Algorithm Based Path Planning for a Mobile Robot, Proceedings of the International Conference on Robotics and Automation, Taipei, Taiwan.
- Vadakkepat, P. Lee, TH. Xin, L. 2001. Application of evolutionary artificial potential field in robot soccer system. NAFIPS, 2781-2785.
- Valle, D.Y. Venayagamoorthy, G.K. Mohagheghi, S. Hernandez, JC. Harley, RG. 2008 Particle swarm optimization: basic concepts, variants and applications in power system. IEEE Trans Evolutionary Computer, 171-195.
- Wang, F. He, X.S. Wang, Y. Yang, S.M. 2012. Markov model and convergence analysis based on cuckoo search algorithm. Jisuanji Gongcheng/Computer Engineering, 38(11):180-185.
- Wang, L. Liu, Y. Deng, H. Xu, Y. 2006. Obstacle-avoidance path planning for soccer robots using particle swarm optimization, Robotics and Biomimetics, Kunming, China.
- Wang, L. Yang, S. X. Biglarbegan, M. 2012. A Fuzzy Logic Based Bio-inspired System for Mobile Robot Navigation, Multisensor Fusion and Integration for Intelligent Systems (MFI), 219-224.
- Wei, X. 2013. Robot path planning based on simulated annealing and artificial neural networks, Research Journal of Applied Sciences, Engineering and Technology 6(1): 149-155.

- Yang, X.S. 2009. Firefly algorithms for multimodal optimization. *Foundations and Applications, Lecture Notes in Computing Sciences*, 5792:178-178, Sapporo, Japan.
- Yang, X.S. Deb, S. 2009. Cuckoo Search via L'evy Flights. *Nature & Biologically Inspired Computing*, 210 - 214.
- Yang, X.S. 2010. Firefly algorithms for multimodal optimization. *Foundations and Applications*,169-178.
- Yang, X.S. He, X. 2013. Firefly algorithm: Recent advances and applications, *Artificial Intelligence and Intelligent Systems*, 1: 36-50.
- Yang, X.S. 2014. *Nature-Inspired Optimization Algorithms*, 1st Edition Elsevier.
- Yin, L. Yin, Y. Lin, C. 2009. A new potential field method for mobile robot path planning in the dynamic environments. *Asian Journal of Control*, 1(2): 214-225.
- Yun, S.C. Ganapathy, V. Chien, T.W. 2010. Enhanced D' Lite algorithm for mobile robot navigation. *Industrial Electronics and Application (ISIEA)*, 545-550.
- Zadeh, A.L. 1965. Fuzzy sets, *Information & Control*, 8: 185-195.
- Zadeh, A.L. 1971. On fuzzy algorithms, *Electron. Res. Lab., Univ. California, Berkeley*.
- Zar, M.T. Sein, M. 2015. Using A\* algorithm for public transportation system in yangon region. *International Journal of Advanced Computational Engineering and Networking*, 3(3): 2320-2106.
- Zhang, Y. Gong, D. Zhang, J. 2013. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neuro computing* 103: 172–185.
- Zhang, Q. Ma, J. Liu, Q. 2012. Path planning based quad tree representation for mobile robot using hybrid-simulated annealing and ant colony optimization algorithm. *Intelligent Control and Automation*, 2537-2542.
- Zhang, L. Min, H. Wei, H. Huang, H. 2012. Global path planning for mobile robot based on A\* algorithm and genetic algorithm, *Robotics and Biomimetics* December, 1795-1799.
- Zhangqi, W. Xiaoguang, Z. Qingyao, H. 2011. Mobile Robot Path Planning based on Parameter Optimization Ant Colony Algorithm. *Procedia Engineering*, 2738 – 2741.

## EKLER

### EK 1: Kobuki Mobil Robot

Kobuki mobil robot, robotik eğitim ve araştırma için tasarlanmış düşük maliyetli teknolojik bir mobil cihazdır. Kobuki mobil robot üzerine eklenecek sensörler, aktüatörler ve harici bilgisayarlar için güç kaynağı sağlamaktadır. Hassas navigasyon sağlamak için üretimden kalibre edilmiş jiroskop entegre edilerek oldukça yüksek doğrulukta odometri sonuçları elde edilmektedir.



Şekil 1. Kobuki mobil robotun genel özellikleri

Şekil 1-2.'de, tez çalışmasında kullanılan Kobuki mobil robotun genel görünümü, üstten, alttan, kontrol paneli görünümü ve genel özellikleri gösterilmektedir. Kobuki mobil robotta bulunan standart yazılım ve donanımsal özellikler aşağıda verilmektedir;

### Yazılım

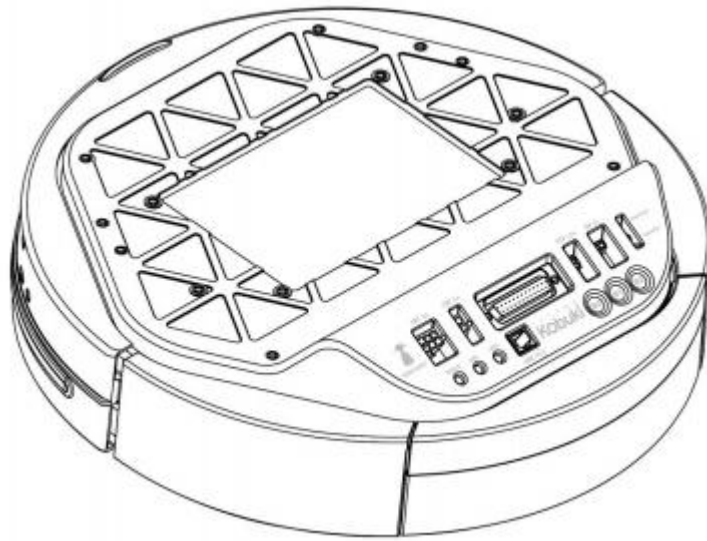
- Linux’de C++
- ROS sürücü
- Gazebo simülasyon

### Donanım

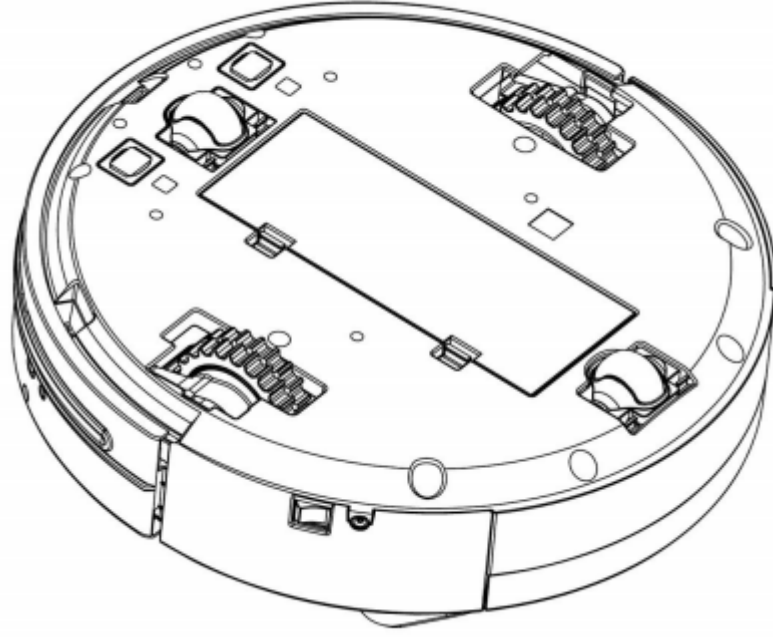
- PC Bağlantısı: USB veya paralel port üzerinden RX/TX pinleri
- Motor Aşırı Yük Algılama: yüksek akım tespitinde ( $> 3A$ ) güç devre dışı
- Fırçalı DC Motor
- Odometri: 52 tik/ kapalı devir, 2578.33 tik/ tekerlek devir, 11.7 tik/ mm
- Gyro: Fabrikada kalibre edilmiştir, 1 eksen (110 derece / s)
- Sensör Veri Hızı: 50 Hz



Şekil 2. Kobuki mobil robotun kontrol paneli



Şekil 3. Kobuki Mobil Robotun Üstten Görünüşü



Şekil 4. Kobuki Mobil Robotun Alttan Görünüşü

## EK 2: Tez Kapsamındaki Yayın Çalışmaları

Tablo 1. Makale çalışmaları

Yazarlar	Makale Başlığı	Dergi	Cilt/Sayfa	Tarih
Z.B. GARİP, D. KARAYEL, S.S.ÖZKAN, G.ATALI	Path Planning for Multiple Mobile Robots Using A* Algorithm	Acta Physica Polonica A	3/685-688	2017
Z.B. GARİP, D. KARAYEL, S.S.ÖZKAN, G.ATALI	Localization Of Odometri And Camerabased Mobile Robot Using Extended Kalman Filter	Acta Physica Polonica A	-	Baskıda
Z.B. GARİP, D. KARAYEL, S.S.ÖZKAN, G.ATALI	Path Planning for Multiple Mobile - Robots in Static Environment using HybridAlgorithm	-	-	Hakemlerde

Tablo 2. Bildiri çalışmaları

Yazarlar	Bildiri Başlığı	Etkinlik	Yer	Tarih
Z.B. GARİP, D. KARAYEL, S.S.ÖZKAN, G.ATALI	A Comparative Study on Heuristic Distance Functions in Path Planning of Mobile Robots	INTERNATIONAL CONFERENCE ON MATHEMATICS AND ENGINEERING	İstanbul	2016

### EK 3: Tez Kapsamındaki Proje Çalışmaları

Tablo 3. Proje çalışmaları

Destekleyen	Görevi	Proje no	Proje adı	Tarihler	Bütçe
SAÜ BAP	Tez	2015-50-02-	Mühendislik Sistemleri için Robot	Aralık 2015	9976
Tez	Öğrencisi	001	Ajanların Kontrol ve Koordinasyonu	Devam ediyor	TL

## **ÖZGEÇMİŞ**

Zeynep B. GARİP, 1988 yılında Sakarya'da doğdu. İlköğretimini tamamladıktan sonra Sakarya Anadolu Kız Meslek ve Teknik Lisesi Bilgisayar Bölümü'nde orta öğretime başladı ve 2005 yılında mezun oldu. Aynı yıl Sakarya Üniversitesi Bilgisayar Sistemleri Öğretmenliği Bölümü'nde lisans eğitime başladı ve Haziran 2009'da mezun oldu. Eylül 2009'da Sakarya Üniversitesi Fen Bilimleri Enstitüsü başladığı yüksek lisans eğitimini, Temmuz 2011 tamamladı. Eylül 2013'de Sakarya Üniversitesi Fen Bilimleri Enstitüsü mekatronik mühendisliğinde doktora eğitime başladı. 2014 yılından itibaren Sakarya Üniversitesi Adapazarı Meslek Yüksek Okulunda öğretim görevlisi olarak görev yapmaktadır.