

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÜÇ BOYUTLU UZAYDA ÇOK YÖNETİCİLİ MOBİL
ROBOTLAR İLE SÜRÜ HAREKETİ PLANLAMASI

Züla TOSUNOĞLU

YÜKSEK LİSANS TEZİ
Elektronik Haberleşme Mühendisliği Anabilim Dalı
Elektronik Programı

Danışman
Dr. Öğr. Üyesi Serkan KURT

Ocak, 2020

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÜÇ BOYUTLU UZAYDA ÇOK YÖNETİCİLİ MOBİL ROBOTLAR İLE
SÜRÜ HAREKETİ PLANLAMASI

Zülal TOSUNOĞLU tarafından hazırlanan tez çalışması 20.01.2020 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik Haberleşme Mühendisliği Anabilim Dalı Elektronik Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Serkan KURT
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Serkan KURT, Danışman
Yıldız Teknik Üniversitesi

Doç. Dr. Ahmet SERBES, Üye
Yıldız Teknik Üniversitesi

Doç. Dr. Aysel ERSOY, Üye
İstanbul Üniversitesi-Cerrahpaşa

Danışmanım Dr. Öğr. Üyesi Serkan KURT sorumluluğunda tarafımca hazırlanan Üç Boyutlu Uzayda Çok Yöneticili Mobil Robotlar ile Sürü Hareketi Planlaması başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Zülal TOSUNOĞLU

İmza

Anneme



TEŞEKKÜR

Tez çalışmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren olan değerli danışman hocam sayın Dr. Öğr. Üyesi Serkan KURT'a, sonsuz teşekkür ve saygılarımı sunarım.

Yüksek lisans eğitimim boyunca benim gelişimime önem veren beni destekleyen, çatısı altında çalışmaktan dolayı gurur duyduğum değerli Arçelik A.Ş'ye teşekkürlerimi sunarım.

Çalışmalarım boyunca yardımlarını hiç esirgemeyen, beni motive eden, değerli iş arkadaşım Rahmet Aybüke GÖZÜTOK BİLİR'e ve değerli yöneticim Bekir YALIN'a teşekkürü bir borç bilirim.

Beni ben yapan, her koşulda destekleyen, hayatımda rol model aldığım kıymetli annem Gülcan TOSUNOĞLU'na bana kattığı ve katacağı herşey için sonsuz teşekkürlerimi sunarım.

Zülal TOSUNOĞLU

İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	ix
ŞEKİL LİSTESİ	x
TABLO LİSTESİ	xi
ÖZET	xii
ABSTRACT	xiv
1 Giriş	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	3
1.3 Hipotez	4
1.4 Tezin organizasyonu	4
2 Sürü Zekası ve Optimizasyonu	5
2.1 Parçacık Sürü Optimizasyonu	5
2.2 Karınca Kolonisi Optimizasyonu	7
2.3 Yapay Arı Kolonisi Optimizasyonu	10
2.4 Ateş Böceği Algoritması Optimizasyonu	13
2.5 Bakteriyel Besin Arama Optimizasyonu	16
2.6 Guguk Kuşu Arama Optimizasyonu	16
2.7 Sıçrayan Kurbağa Optimizasyonu	18
2.8 Mirket Klan Algoritması Optimizasyonu	19
2.9 Örümcek Maymun Optimizasyonu	20
2.9.1 Örümcek Maymun Optimizasyonu Algoritması	21
2.10 Çok Yöneticili Mobil Robotların Sürü Hareketi	25
2.10.1 Çok Yöneticili Mobil Robotların Sürü İçerisindeki Hiyerarşik Dağılım	26
2.10.2 Mobil Robotların Ortam Sınırlarının Belirlenmesi	26
2.11 Optimizasyon Yöntemlerinin Karşılaştırılması	26

2.12 Konum Belirleme Teknikleri	30
2.12.1 Alınan Sinyalin Gücü Göstergesi (RSSI)	30
2.12.2 Parmak İzi Analizi	31
2.12.3 GPS (Global Positioning System)	31
3 ÜÇ BOYUTLU UZAYDA KONUMUNUN MATEMATİKSEL OLARAK MODEL- LENMESİ VE HATA KALIBRASYONU	32
3.1 Üç Boyutlu Uzayda Sisteminde Konumun Matematiksel Olarak Modellenmesi	32
3.2 Hatanın Kalibrasyonu	33
3.3 Sözde Ters Matris	36
4 Sonuçlar ve Öneriler	40
A MATLAB Kodları	41
Kaynakça	43
Tezden Üretilmiş Yayınlar	46

SİMGE LİSTESİ

ω	Eylemsizlik faktörü
$p_i(t)$	i. parçacığın t anındaki en iyi konumu
c_1	Parçacığın hızlandırma katsayısı
c_2	Sürünün hızlandırma katsayısı
r_1 ve r_2	0 ve 1 arasında bağımsız rasgele değişkenler
BS	Çalışan arı sayısına veya izleyen arı sayısına eşit besin kaynağı sayısı
fit_i	i. pozisyondaki çözümün uygunluk (fitness) değeri
p_i	i. pozisyondaki çözüme ait uygunluk değerinin keşfedilen tüm besin kaynakların bulunduğu pozisyonundaki toplam uygunluk değerine oranı
k ve j	$k \in \{1, 2, \dots, BS\}$ ve $j \in \{1, 2, \dots, D\}$ rasgele olarak seçilen indisler
ϕ_{ij}	$[-1, 1]$ arasında rasgele bir sayı
β_0	$r = 0$ mesafesinde iken ateş böceğinin sahip olduğu çekicilik
α	$r = 0$ mesafesinde iken ateş böceğinin sahip olduğu çekicilik
ε_i^t	t anında Gauss veya homojen dağılımdan çizilebilen rasgele sayıların bir vektörü
D	Optimizasyon problemindeki değişkenlerin sayısı
z	Popülasyondaki örümcek maymun etiketi
OM_z	Popülasyondaki z . örümcek maymun
$U(0, 1)$	0 ve 1 arasında rasgele bir sayı
OM_{maxj}	OM_z 'ye ait j yönündeki en üst sınır
OM_{minj}	OM_z 'ye ait j yönündeki en alt sınır
LL_{kj}	k. lokal grup liderinin j . eksenindeki konumu
OM_{zj}	z . örümcek maymununun j . eksenindeki güncel konumu

$OM_{r,j}$	k.grup içerisinde $r \neq j$ şartı ile rasgele seçilen r. maymunun j. eksenindeki konumu
GL_j	Global grup liderinin j. eksenindeki konumu
$maxfitness_z$	Grubun en büyük uygunluk değeri
$fitness_z$	z. maymunun uygunluk değeri
G	Robotun gerçek konumu
B	Robotun bilinen konumu
D	Durum etiketi
Y	Yönetici (Kaşif) robot
T	Takipçi robot
i	Robot etiketi
ϵ	Hata
${}^D_G X_i(t)$	i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki X eksenine göre gerçek konumu
${}^D_G Y_i(t)$	i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Y eksenine göre gerçek konumu
${}^D_G Z_i(t)$	i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Z eksenine göre gerçek konumu,
${}^D_B X_i(t)$	i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki X eksenine göre bilinen konumu
${}^D_B Y_i(t)$	i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Y eksenine göre bilinen konumu
${}^D_B Z_i(t)$	i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Z eksenine göre bilinen konumu,
$\epsilon_{X_i}(t)$	i. robotun t anındaki X eksenine göre hatası
$\epsilon_{Y_i}(t)$	i. robotun t anındaki Y eksenine göre hatası
$\epsilon_{Z_i}(t)$	i. robotun t anındaki Z eksenine göre hatası
ϵ_T	Toplam hata

KISALTMA LİSTESİ

SZ	Sürü Zekası
RSSI	Received Signal Strength Indication
PSO	Parçacık Sürü Optimizasyonu
KKO	Karınca Kolonisi Optimizasyonu
YAKO	Yapay Arı Kolonisi Optimizasyonu
ABAO	Ateş Böceği Algoritması Optimizasyonu
BBAO	Bakteriyel Besin Arama Optimizasyonu
GKAO	Guguk Kuşu Arama Optimizasyonu
SKO	Sıçrayan Kurbağa Optimizasyonu
MKAO	Mirket Klan Algoritması Optimizasyonu
ÖMO	Örümcek Maymun Optimizasyonu
GPS	Global Positioning System
3B	Üç Boyutlu
OM veya ÖM	Örümcek Maymun
LL	Lokal Lider
GL	Global Lider

ŞEKİL LİSTESİ

Şekil 2.1	PSO Algoritması akış diyagramı [20]	8
Şekil 2.2	KKO Algoritması akış diyagramı [20]	10
Şekil 2.3	YAKO Algoritması akış diyagramı [20]	14
Şekil 2.4	ABAO akış diyagramı [20]	15
Şekil 2.5	BBAO akış diyagramı [20]	17
Şekil 2.6	Mirket Hayvanı	19
Şekil 2.7	Örümcek Maymunu	20
Şekil 2.8	Örümcek Maymun Sürüsü Avlanma Davranışı	22
Şekil 2.9	Mobil Robot Ortam Sınırları	27
Şekil 2.10	RSSI ve mesafe arasındaki ilişki [25]	30
Şekil 3.1	MATLAB yardımıyla üç adet kürenin karşılaştırılması	33
Şekil A.1	Çakıştırılan üç küreye ait MATLAB kodu	41
Şekil A.2	J matrisi tersinin MATLAB yardımıyla bulunması	41
Şekil A.3	K matrisinin MATLAB programında sözde ters matris komutu kullanılarak bulunması	42

TABLO LİSTESİ

Tablo 2.1	Üyeler ve Sürü Zekası Algoritmaları	22
Tablo 2.2	İncelenen Algoritmaların hiyerarşi bakımından karşılaştırması . . .	29
Tablo 2.3	ÖMO algoritması ve Çok Yöneticili Mobil Robotların Sürü Hareketi İlişkilendirilmesi	29
Tablo 3.1	Verilen üç robotun konumu ve hedefe olan mesafeleri	37

Üç Boyutlu Uzayda Çok Yöneticili Mobil Robotlar ile Sürü Hareketi Planlaması

Zülal TOSUNOĞLU

Elektronik Haberleşme Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Serkan KURT

Bu çalışmada üç boyutlu uzayda çok yöneticili bir yöntem ile mobil robotların ortak hareketi ve sürekli kalibrasyon ile optimum konum belirlenmesi sağlanmıştır. Mobil robotların bilinen bir hedef görevi tamamlaması süresince izlemesi gereken uygun yolların belirlenmesi üzerine çeşitli Sürü Zekası (SZ) algoritmaları incelenmiştir. İncelenen bu algoritmalarından yola çıkılarak Örümcek Maymun Optimizasyon (ÖMO) algoritmasının hedeflenen optimum yol belirleme metodolojisine uygun olduğu görülmüştür. Çoklu mobil robotların kullanım alanlarının genişlemesi ile belirlenen optimum yol boyunca senkronize paralel hareketi sırasında birbirine göre konum belirlemesi ve mobil robotların birbiri ile sürekli iletişimde olması hedef görevin başarılı şekilde tamamlanması açısından kritik hale gelmiştir. Bahsedilen kritik konulara çözüm geliştirmek amacıyla belirli bir hedefe sahip ve belirli bir rota içerisinde senkronize paralel hareket eden çoklu mobil robotların üç boyutlu uzayda birbirine göre konumunun matematiksel olarak modellenmesi ve bu konumdaki hata miktarının kalibre edilmesi üzerine çalışılmıştır. Matematiksel olarak modellenen üç boyutlu konumdaki hata miktarının kalibrasyonu mobil robotların GPS verileri ve sözde ters matrisine göre hesaplanan konum bilgisi ile ifade edilmiştir. Mobil robotların birbiri ile sürekli halde iletişimde olması ve birbirlerini belirlenen ortam sınırları içerisinde kaybetmemeleri için alınan sinyal gücü göstergesi (RSSI-received signal strength indication) bilgilerinden yararlanılmıştır.

Anahtar Kelimeler: Mobil robotlar, sürü algoritması, optimizasyon, üç boyutlu uzay, sözde ters matris

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



Swarm Movement Planning with Multi Master Robots in 3D Space

Zülal TOSUNOĞLU

Department of Computer Engineering
Master of Science Thesis

Advisor: Dr. Serkan KURT

In this study, it is provided to determine the optimum position with the joint movement of mobile robots and continuous calibration with a multi-master method in 3D space. Various Swarm Intelligence (SI) algorithms have been examined to determine the appropriate paths that mobile robots must follow during the completion of a known target task. Based on these algorithms, it was found that the Spider Monkey Optimization (SMO) algorithm is in accordance with the targeted optimum path methodology. The position of the mobile robots in relation to each other during synchronized parallel movement along the optimum path determined and the uninterrupted communication of the mobile robots with each other have become critical for the successful completion of the target task by the expansion of the usage areas of the multiple mobile robots. In order to solve these critical issues, mathematical modeling and calibrating of the error position in three dimensional space of multiple mobile robots which have a specific target and synchronized parallel movement in a given route have been studied. Calibration of the error amount in three-dimensional mathematically modeled position is expressed by the position data calculated according to pseudo inverse matrix and GPS data of mobile robots. RSSI (received signal strength indication) information was used to keep mobile robots in uninterrupted communication with each other and not to lose each other within the specified environment.

Keywords: Mobil robots, swarm algorithm, optimization, three dimensional space, pseudo inverse matrix

**YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**



1.1 Literatür Özeti

Günümüzde sürü zekası veya sürü hareketi tabanlı optimizasyon teknikleri mobil robot uygulamaları alanındaki pek çok çalışmada geniş çapta yer almaktadır [1–7]. Sürü robotiği olarak da bilinen bu alan çeşitli hayvanların sürü halinde avlanma, beslenme, bir araya gelme ve saldırma gibi vb. iç güdüsel davranışlarından esinlenilerek geliştirilmiştir. Sürü hareketini baz alan bu robotlar bireyden ziyade sürü olmanın avantajını kullanarak grupları dahilinde birbirleriyle sürekli olarak iletişimde kalırlar. Bu sayede daimi olarak güvenli veriye ulaşmayı hedeflerler. Aynı zamanda da devre dışı kalan robotların görevlerini üstlenerek oluşabilecek herhangi bir zaafiyette riski telafi etme amacı taşırlar [8, 9].

Mobil robotlar deprem bölgelerinde arama kurtama çalışmalarından maden, petrol, doğalgaz vb. bir keşif için veya da besin arama gibi pek çok geniş alan arama uygulamalarında kullanılmaktadır. Bu alanda üretilen çalışmalar incelendiğinde genel kapsamın çoğunlukla iki boyutlu koordinat sisteminde ve yönetici vasfına sahip tek bir mobil robotun bulunduğu sürü hareketi planlaması olduğu görülmüştür [1–7]. Bu çalışmada ise üç boyutlu ortamlar için analizler ve çok yöneticili mobil robot sistem tasarımı yapılmıştır.

Bhattacharjee ve arkadaşları tarafından önerilen çalışmada Yapay Arı Kolonisi Optimizasyon tekniği kullanılarak çoklu mobil robotlarla yol planlaması yapılmıştır. Engellerin bulunduğu iki boyutlu olarak değerlendirilen bir ortamda önceden bilinen bir hareket noktasından istenilen hedef noktaya giden mobil robotlarının en kısa yolu bulmasına yönelik geliştirilmiştir [1].

Banerjee ve arkadaşları ise dinamik ve statik engellerin bulunduğu iki boyutlu ortamda çarpışma olasılığının minimize edildiği ve hedefe giden optimum yolun belirlenmesinde Parçacık Sürü Optimizasyonu tekniğinin taban alındığı mobil robot hareket planlaması üzerine çalışma yapmışlardır [2].

Hidalgo-Paniagua ve arkadaşları tarafında ateş böceklerinin davranışlarından esinlenilerek mobil robotlarla çok amaçlı yol hareketi planlaması alanında çalışmalar yapmışlardır. Bu çalışmada önerilen ana amaç üç farklı görevin mobil robotlar tarafından etkili ve doğru şekilde gerçekleştirilmesidir. Görevler gerçekleştirilirken enerji tüketimi ile ilişkilendirilerek planlanan yolun güvenli olması, uzunluğu ve sürtünmeden kaynaklanan enerji kaybının istenmemesi nedeniyle pürüzsüzlüğü oldukça önem taşımaktadır [3].

Brand ve Yu, Ateş Böceği Algoritması Optimizasyonu (ABAO) tekniğini temel alarak dinamik ve statik engellerin bulunduğu iki boyutlu koordinat sisteminde hareket planlaması üzerinde çalışma yapmışlardır. Bunun yanında PSO tekniği ile kıyaslama yaparak ABAO tekniğinin daha geniş çalışma alanlarında hızlı yakınsama özelliğine sahip olması açısından avantajlı olduğunu belirtmiştir [4].

Guan-Zheng, Huan ve Aaron tarafından önerilen çalışmada Karınca Kolonisi Optimizasyonu tekniği kullanılarak iki boyutu koordinat sisteminde gerçek zamanlı olarak mobil robotların optimum hareket planlaması konu edilmiştir. Optimum sonuca yakınsama hızı, çözüm varyasyonları ve dinamik yakınsama davranışı açısından değerlendirildiğinde genetik algoritma tabanlı robot hareketi planlama tekniklerinden daha iyi performans gösterdiği ifade edilmiştir [5].

Sierakowski ve Coelho, bakterilerin besin arama davranışından esinlenerek iki boyutlu koorinat sisteminde engellerin bulunduğu statik bir ortamda hedefe giden yolda optimum robot hareketi üzerine çalışma yapmışlardır. Çalışmalarını engel sayılarının farklı olduğu iki farklı deney senaryosunda gerçekleştirmişlerdir [6].

Liang ve Lee, Yapay Arı Kolonisi Optimizasyonu tekniğini taban alan Etkili Yapay Arı Kolonisi Optimizasyonu tekniği ile çoklu mobil robotlarla lokal navigasyon teknikleri yardımıyla çarpışmanın engellendiği bir ortam için bir çalışma sunmuşlardır. Önerdikleri çalışma kapsamında mobil robotların uyması gereken üç kuraldan bahsedilmiştir: hedefe ulaşma, engellerle çarpışmadan kaçınma ve robotlarla çarpışmadan kaçınma. Bu çalışmanın performansını değerlendirmek için çeşitli matematiksel fonksiyonlar test edilmiştir. Test fonksiyonlarının sonuçları değerlendirildiğinde optimal sonuca ulaşmada Etkili Yapay Arı Kolonisi Optimizasyon tekniğinin YAKO, PSO teknikleri ile rekabetçi olduğu görülmüştür [7].

Sürü zekası davranışı baz alınarak geliştirilen çoklu mobil robot hareketi planlaması çalışmaları incelendiğinde çoğunlukla iki boyutlu koordinat sisteminde Parçacık Sürü Optimizasyonu , Yapay Arı Kolonisi Optimizasyonu ve Karınca Kolonisi Optimizasyonu teknikleri ve bu tekniklerin karıştırılmış versiyonları üzerine olduğu görülmüştür [1–7]. Bunun haricinde diğer optimizasyon teknikleri (Ateş Böceği

Algoritması Optimizasyonu, Bakteriyel Besin Arama Optimizasyonu, Guguk Kuşu Arama Optimizasyonu, Sıçrayan Kurbağa Optimizasyonu ve Mirket Klan Algoritması Optimizasyonu) incelediğinde sürü zekası davranışsal yapılarının tek yöneticili veya yönetici vasfına sahip olmayan üyelerden oluştuğu anlaşılmıştır [10–14].

Sharma, Bansal ve Jadon tarafından önerilen Örümcek Maymun Optimizasyonu tekniğinde ise ortamdaki besin arama, rekabet vb. durumlarda sürünün alt gruplara bölüdüğü ve bu alt grupların hepsi yönetici görevine sahip birer dişi lider tarafından yönetildiği görülmüştür [15].

1.2 Tezin Amacı

Çoklu mobil robotların arama kurtarma çalışmaları gibi açık alanda gerçekleşen geniş çaplı faaliyetleri neticesinde robotların arama alanını eksiksiz bir şekilde taraması, bu taramayı yaparken sahip oldukları sınırlı batarya enerjisini verimli şekilde kullanmaları gerekmektedir. Aynı zamanda veri aktarımının kesintisiz ve doğru bir şekilde devam edebilmesi için arama alanında birbirlerini kaybetmemeleri, iletişimlerinin kesilmemesi oldukça önemlidir.

Literatür özetinde bahsedilen tek yöneticinin bulunduğu çoklu mobil robot uygulamalarında, robot sürüsünün tek bir merkezden yönetilmesi sebebiyle ulaşılmak istenen hedefe varış süresinde ve tüketilen enerjide verimsizlik yaşanması söz konusudur. Çok geniş ve dinamik ortamlarda tek yöneticili sistemler yetersiz kalabilmektedir. Ayrıca geniş arama alanlarında mobil robotların yönetici robottan limitlerin dışında bir mesafeye gitmesi durumunda veri kaybı meydana gelebilir.

Bu çalışmada çok yöneticili çoklu mobil robot hareketi planlaması kapsamında değerlendirilen optimizasyon teknikleri arasındaki farklar ve üstünlükler belirtilmiştir. İncelenen teknikler arasında Örümcek Maymun Optimizasyonu tekniğinin birden fazla dişi yönetici lidere sahip olması ile hedeflenen metodolojiye uygun olduğu görülmüştür.

Çok yöneticili sistemin benimsenmesinin yanında arama alanının üç boyutlu uzayda sağlanması mobil robotların birbirlerinin konumlarını ve aralarındaki mesafeyi bilmeleri açısından önemlidir. Mobil robotların sahip oldukları konumlar küre denklemleri ile ifade edilirken veri aktarımının kesintisiz bir şekilde gerçekleşmesi ve birbirlerini kaybetmemeleri için robotlar arasındaki mesafeyi tanımlayan alınan sinyalin gücü göstergesi (RSSI-Received Signal Strength Indication) bilgisinden yararlanılmıştır. Küre denklemlerinden oluşturulan matrisler yardımıyla önceden bilinen hedef noktanın doğruluğu tespit edilmeye çalışılmıştır. Doğrulama sonucu

eksenel boyutta bir sapma ile karşılaşıldığında mobil robotlar uydu üzerinden aldıkları GPS verisi ile matematiksel olarak hesaplanan konum arasındaki farkı hata olarak tanımlar ve bu hatayı birbirlerine komutlar vererek mobil robotun istenen konuma gelmesi için kalibre eder.

Bu sayede çoklu mobil robotların belirlenen optimum yol boyunca birbirlerini arama alanında kaybetmeden çok yöneticili olarak senkronize hareketi sırasında birbirine göre üç boyutlu uzayda konum belirlemesi ve belirlenen konumdaki hatanın kalibrasyonu sağlanmıştır.

1.3 Hipotez

Literatür özetinde bahsedilen bilgiler doğrultusunda bu tez çalışmasında üç boyutlu uzayda çok yöneticili bir yöntem ile mobil robotların ortak hareketi ve sürekli kalibrasyon ile optimum konum belirlenmesi sağlanmıştır. Konum belirleme ve hata kalibrasyonu için üretilen matematiksel modellerin ve taban alınan sürü zekası davranışının dağınık ortamlarda ve üç boyutlu uzaydaki çalışmalara ışık tutması hedeflenmiştir.

1.4 Tezin organizasyonu

Bölüm 2'nin ilk yarısında sürü zekası ve incelenen optimizasyon tekniklerine yer verilmiştir. Akabinde önerilen çok yöneticili mobil robot hareketi planlaması ve bazı konum belirleme tekniklerinden bahsedilmiştir. Bölüm 3'te üç boyutlu uzayda konumun matematiksel olarak modellenmesi ve hata kalibrasyonuna yer verilmiştir. Bölüm 4'te önerilen çok yöneticili mobil robot hareketi ile elde ettiğimiz kazanımlardan bahsedilmiştir.

Sürü zekası, 1993 yılında Gerardo Beni ve Jing Wang tarafından önerilmiş olup robotik sistemler-yapay zeka çatısı altında merkezi bir kontrolün olmadığı ve bireysel zeka yerine kolektif zekayı gerektiren problemlere karşı ortak davranış sergileyen ve kendi içerisinde organize olabilme yeteneğine sahip doğal veya yapay sistemler için kullanılan bir yaklaşımdır [16]. Geçtiğimiz yıllar içerisinde çeşitli türlerdeki sürülerin davranışlarından esinlenilerek çok sayıda sürü zekası tabanlı optimizasyon yöntemi önerilmiştir [10–15, 17–19]. Bu bölümde Tablo 2.1’de verilen optimizasyon türleri sırası ile incelenmiştir. Ardından sürü davranışları birbirleri ile hiyerarşik açıdan kıyaslanmıştır. Çalışmada önerilen çok yöneticili çoklu mobil robot hareketi ile buna uygun optimizasyon tekniği ilişkilendirilmiştir.

2.1 Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu (PSO), balık ve kuş sürülerinin sosyal davranışlarından esinlenilerek 1995 yılında Eberhart ve Kennedy tarafından geliştirmiştir. PSO algoritmasında hedefe ulaşmak için sürü içerisinde liderlik etmekle görevli herhangi bir lider bulunmamaktadır. Bir kuş sürüsü yiyecek aramaya çıktığında herhangi bir lidere ihtiyaç duymadan besin kaynağına en yakın üye ile birlikte yoluna devam etmektedir. Sürü içerisindeki üyelerin birbirleri ile olan iletişimi sayesinde istenen hedefe ulaşılmış olunur. PSO algoritması bir grup parçacıklardan oluşmuş olup her bir parçacık potansiyel bir çözümü ifade etmektedir [17]. Eberhart ve Kennedy tarafından önerilen bu algoritmaya göre her bir parçacığın hızı birbiri ile ilişkilidir ve sürüdeki tüm parçacıklar aşağıda verilen 3 basit kurala uygun davranış göstermesi gerekmektedir;

- Çarpışmadan kaçınma
- Hız eşleştirme
- Besin merkezde olacak şekilde alana dağılma

Bu 3 kurala ilave olarak her parçacık 3 davranışa sahiptir;

- Bölgesel en iyi pozisyon belirleme
- Global en iyi pozisyon belirleme
- Parçacık bir sonraki pozisyona ilerlediğinde belirlediği bu iki pozisyonu referans olarak gösterme

Başlangıçta parçacıklar boş bir alanda rasgele dağılmış pozisyonda bulunmaktadır. Arama alanındaki her bir parçacık hedefe ulaşacağı hızı hesaplamak için uygunluk fonksiyonuna sahiptir ve kendi hız ve konum bilgisini kendisine ait konum vektöründe saklı tutmaktadır. Bir parçacık harekete geçtiğinde ilk olarak hız bilgisini, bölgesel ve global en iyi pozisyonuna ve şimdiki hızına göre güncellemektedir. Akabinde parçacık güncellenen hız bilgisine göre kendi pozisyonunu güncellemektedir [17].

Arama alanında çözüm olarak da tanımlanan her parçacık hedefe doğru olan hızını belirlemek için uygunluk fonksiyonuna sahiptir ve her parçacık kendisinin bir sonraki hareketini hesaplamak için bir vektör ile temsil edilir. Böylece uygunluk fonksiyonu aday çözüm vektörü yerini alır ve gerçek bir değer üretir. Sonrasında fonksiyonun çıktısı arama alanındaki diğer çıktılarla karşılaştırılır. Her bir parçacık (çözüm) dört parametreye sahiptir; arama alanındaki mevcut konum, mevcut uygunluk değeri, mevcut hız ve hafıza. Her yinelemede parçacıkların hızı ve pozisyonu mevcut hız olarak güncellenirken, belirli iterasyona kadar uygunluk değeri hesaplanan parçacıkların en iyi pozisyonu da hedefe en yakın partiküllerin pozisyonunu gösteren global en iyi pozisyon dikkate alınarak güncellenir. N boyutlu arama alanında i. parçacığın konum ve hız kümesi aşağıdaki denklemlerde verilmiştir [17].

$$x_i = [x_{i,1} \quad x_{i,2} \quad \dots \quad x_{i,n}] \quad (2.1)$$

$$v_i = [v_{i,1} \quad v_{i,2} \quad \dots \quad v_{i,n}]; \quad i = \{1, 2, \dots, m\} \quad (2.2)$$

Parçacıklar sürü içinde basit kurallara uymakla beraber parçacıkların davranışını yönlendiren merkezi bir kontrol sistemi bulunmamaktadır. Aşağıda verilen (2.3) ve (2.4) denklemleri ile her bir iterasyonda i. parçacığının t anındaki hızı (v_i) ve pozisyonu (x_i) güncellenir. Bir parçacığın, çözümü aradığı süre boyunca kendisinin çözüme en çok yaklaştığı o andaki en iyi durum p_i olarak adlandırılırken, sürü içerisinde arama boyunca çözüme en çok yaklaşan parçacığın o andaki durumuna ise p_g denir [17].

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [p_i(t) - x_i(t)] + c_2 r_2 [p_g(t) - x_i(t)] \quad (2.3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.4)$$

Parametreler:

ω : Eylemsizlik faktörü

$p_i(t)$: i. parçacığın t anındaki en iyi konumu

$p_g(t)$: Tüm parçacıkların deneyimi ile belirlenen en iyi konum (global en iyi konum)

c_1 : Parçacığın hızlandırma katsayısı

c_2 : Sürünün hızlandırma katsayısı

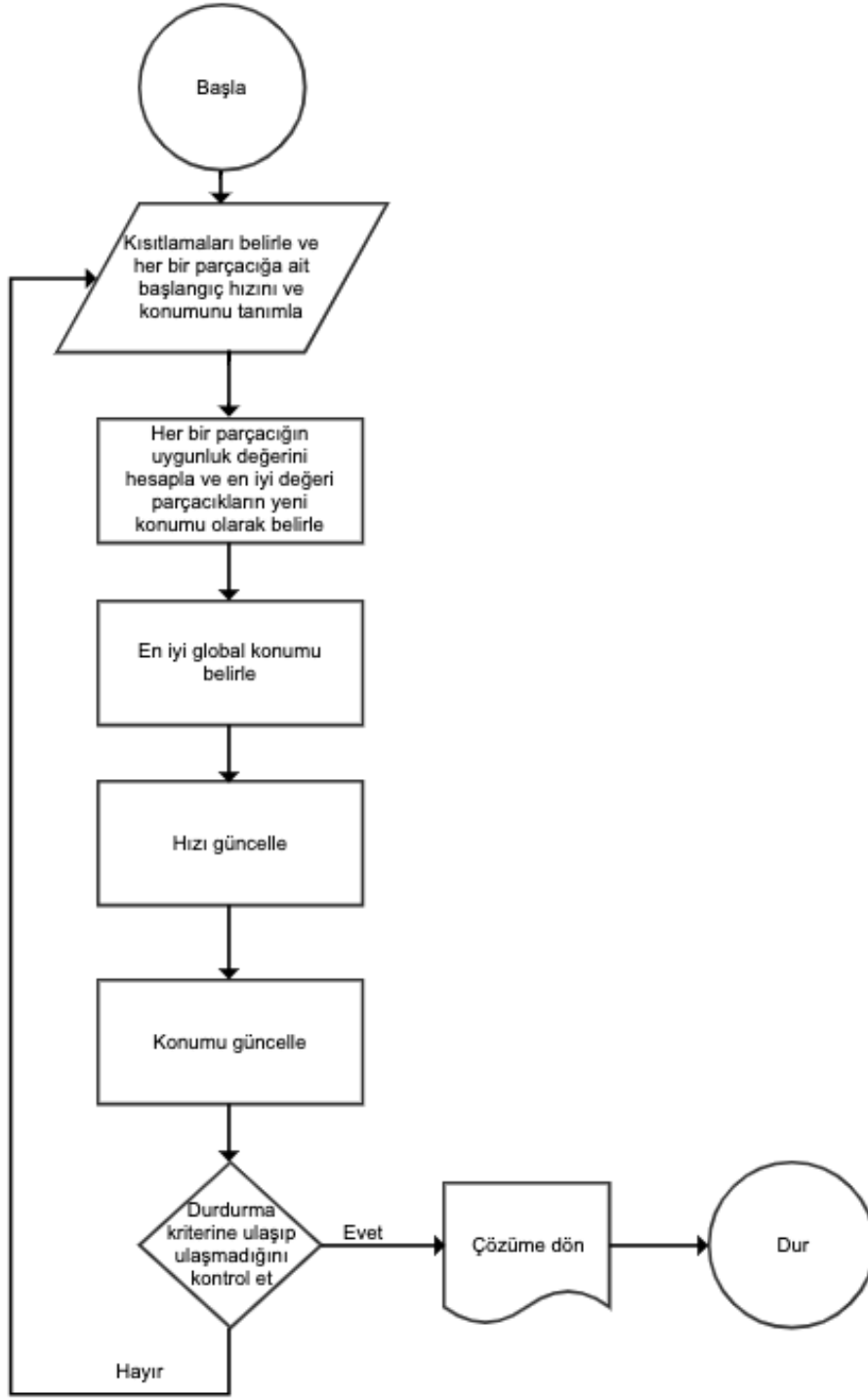
r_1 ve r_2 : 0 ve 1 arasında bağımsız rasgele değişkenler

PSO tekniği üç ana faktörü içerir: eylemsizlik, hafıza ve iş birliği. Bu faktörler sürünün davranışını karakterize eder ve böylece parçacıklar uygunluk fonksiyonunun kalitesini artırmak için bilgi toplar ve paylaşırlar [17].

Şekil 2.1'de görüldüğü gibi PSO algoritması problemin sınırlarını ve ilgili parametreleri tanımlamakla başlar. Daha sonra uygunluk fonksiyonu mevcut sonuçların iyiliğini ölçmek amacıyla her bir parçacık için değerlendirilir. Daha sonra diğer parçacıklarla etkileşime girerek hız ve konum değerleri sırasıyla verilen denkleme göre değiştirilir. Bu şekilde sonlandırma kriterini yerine getirene kadar süreç tekrarlanır. Algoritmanın sonlandırma kriterleri mevcut pozisyon ile hedef arasındaki minimum mesafeye ulaşmak veya minimum yineleme sayısına ulaşmak olarak belirlenebilir. Bunun yanı sıra önceden belirlenmiş yineleme sınırlarında beklenmeyen durumlar için alternatif durdurma kriterleri eklenebilir [17]. Şekil 2.1'de PSO tekniğinin genel akış şeması verilmiştir.

2.2 Karınca Kolonisi Optimizasyonu

Karınca Kolonisi Optimizasyonu (KKO), 1992 yılında Marco Dorigo tarafından öne sürülen popülasyon tabanlı sürü zekası algoritmalarından biridir. KKO, stigmergy olarak da bilinen endirekt iletişim konseptine dayanır. Bu konseptin ilkesi ortamda izler bırakıp bir sonraki hareket için koloniyi tetiklemektir. Bir karınca hareket etmeye başladığında veya bir görevi yerine getirirken arkalarında feromon denilen bir maddeyi bırakırlar. Karıncalar tarafından ortama bırakılan bu maddenin miktarı diğer karıncaların aktivitelerini değiştirir ve bu karıncalar feromon salgılayan karıncayı



Şekil 2.1 PSO Algoritması akış diyagramı [20]

takip eder. Bu optimizasyon yöntemine örnek verilecek uygulamalardan bir gezgin satıcı problemidir. Karınca kolonilerinden oluşan n sayıda şehrin bulunduğu bir gezgin satıcı probleminde karıncalar izledikleri güzergah üzerinde sanal feromon izi bırakırlar ve feromon seviyesi zamana göre değişim göstermektedir. Bir karınca sonraki adıma geçmek için bir güzergaha karar verirken bunu yol üzerinde bulunan feromon miktarına göre yapar [18].

Her karınca alanda rasgele seçilen noktalardan hareket etmeye başlar. Başlangıç noktasından hedefe olan bağlantı çizgisi bir yol olarak tanımlanır. Her yol feromon seviyesi ve buna bağlı sezgisel değerle sınıflandırılır. Böylece herhangi bir yol için bu parametrelerin daha yüksek olması karıncanın o yolu seçme olasılığını yükseltir. Geriye kalan karıncalar besin kaynağına ulaşmak için yolda biriken feromonu kullanır. Her bir karınca besin kaynağına olan seyahatini tamamlayana kadar bu olay tekrarlanır ve karıncalar tarafından izlenen her yoldaki feromon seviyesi güncellenir. Sonuç olarak her karınca bir çözüm sağlar. Bu tekniğin temel özelliği, besin kaynağına giden rotadaki feromon seviyesine bağlı olduğundan biriken feromon seviyesi yükseldikçe çözümün optimalitesi de artar. Besin kaynağına ulaşmanın zorluğuna göre feromon maddesinde artış olur [18].

Bu optimizasyon yönteminin en çok kullanıldığı uygulamalar gezgin satıcı problemleridir;

$$p_{mn}^k(t) = \left\{ \frac{[\tau_{mn}(t)]^\alpha [\eta_{mn}(t)]^\beta}{\sum [\tau_{mn}(t)]^\alpha [\eta_{mn}(t)]^\beta} \right\}, n \in J_k(m) \quad (2.5)$$

$$p_{mn}^k(t) = 0, m \notin J_k(m) \quad (2.6)$$

Parametreler:

$J_k(m)$: Şehir m 'deki k karıncasının tüm şehirlerdeki birleşimi.

$\tau_{mn}(t)$: m ve n şehirleri arasındaki feromon miktarı.

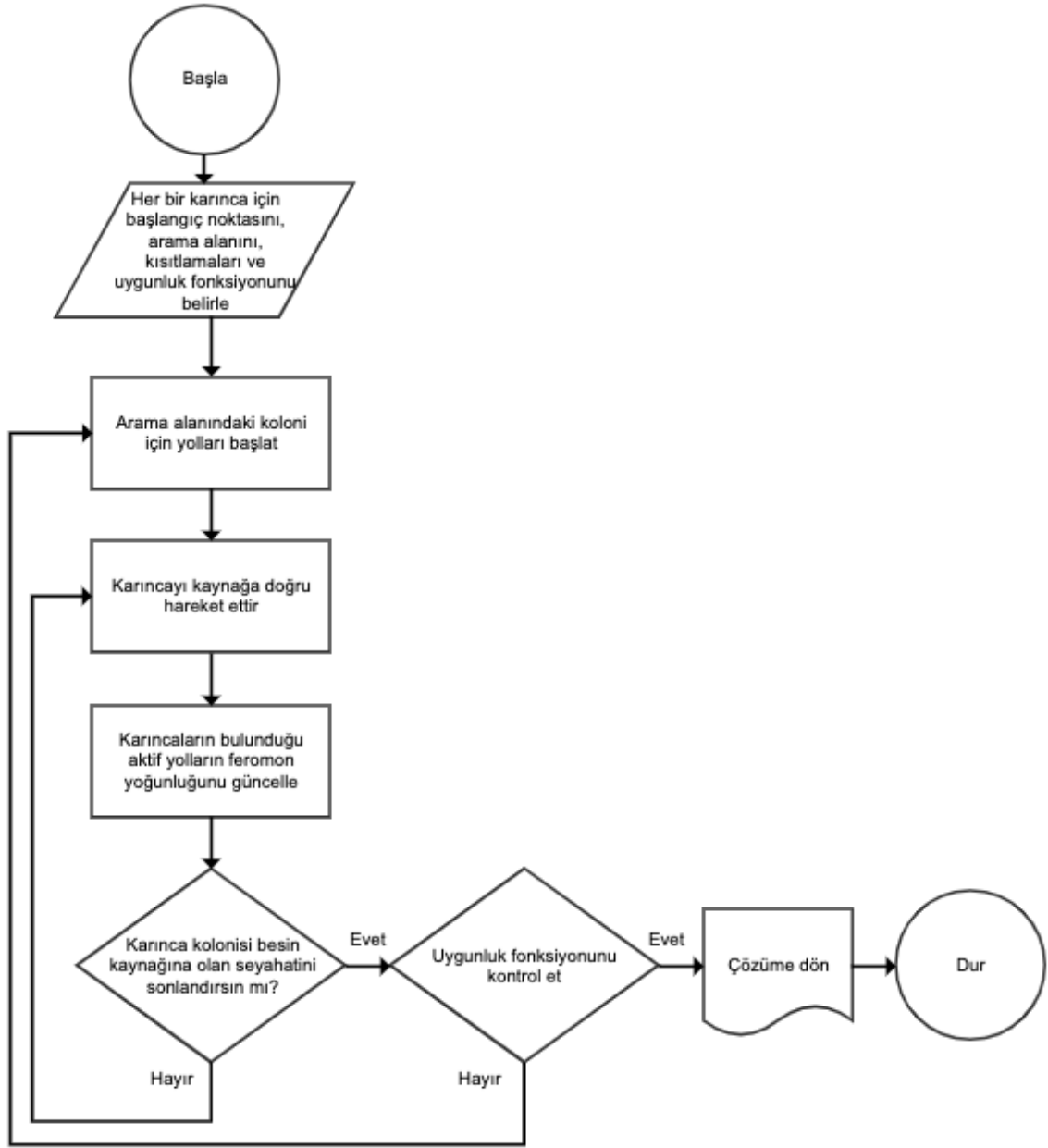
α ve β : Feromon izinin kontrolü için ayarlanabilir parametrelerdir.

$\eta_{mn}(t)$: Var olan çözümlere m , n kenarlarının eklenme istenilebilirliğinin sezgisel bilgisidir.

Gezgin satıcı problemlerinde bu bilgi şurdan alınır: $\eta_{mn}(t) = \frac{1}{d_{mn}(t)}$

$d_{mn}(t)$: m şehirden n şehrine olan uzaklığının ters mesafesidir.

Şekil 2.2’de karınca kolonisi optimizasyonu tekniğinin genel akış şeması verilmiştir.



Şekil 2.2 KKO Algoritması akış diyagramı [20]

2.3 Yapay Arı Kolonisi Optimizasyonu

Yapay Arı Kolonisi Optimizasyonu (YAKO) yaklaşımı Karaboğa tarafından arıların kovanlarında toplanma ilkelerini açıklamak amacıyla önerilmiştir. Yiyecek ararken farklı görevlerdeki arılar zengin nektarlı besin kaynaklarını araştırmak ve kullanmak için iş birliği halinde çalışırlar. Bu yaklaşıma göre koloni içinde üç tür arı vardır; çalışan arılar, gözcüler ve kaşifler. Her besin kaynağına ulaşmakla görevli bir arı bulunmaktadır [19]. Kaşif arılar, arama alanında rasgele bir arama yaparak ilk besin kaynaklarını bulmakla görevlendirilir. Sonrasında bu ilk olarak keşfedilen besin

kaynaklarından faydalanmak için çalışan arılar gönderilir ve gönderilen arılardan her biri bir besinle görevlendirilir. Çalışan arılar arama alanındaki besin kaynağından faydalanırken bir yandan da etrafta daha iyi besin kaynağı bulmak için alan araştırması yapar. Daha iyi bir besin kaynağı bulunursa çalışan arılardan biri bu yeni keşfedilen besin kaynağına yönelir. Bu süreçte çalışan bütün arılar işlerini tamamladıktan sonra kovanlarına geri dönerler ve besin kaynaklarından topladıkları bilgiyi kovan içinde sallanma hareketinde olan gözcü arılarla paylaşırlar [19]. Gözcü arılar çalışan arıları takip etmeyi ve dolayısıyla çalışan arılara karşılık gelen besin kaynaklarından faydalanmayı olasılığa dayalı olarak seçeceklerdir. Bu olasılık besin kaynaklarının zenginliği ile hesaplanmaktadır. Bir defa gözcü arı çalışan bir arıyı seçtiğinde seçilen bu arı çalışan arı olarak kolonideki görevini tekrar eder [19]. Rasgele seçilen arama alanındaki besin kaynaklarının bulunması ve faydalanılmasının belirli sayıda tekrarından sonra besin kaynağının tüketilmesi durumunda terk edilir. Bu durumda o arama alanında çalışan arı, kaşif arı rolünü üstlenir ve terkedilen besin kaynağının yerine yenisi bulur. Bu algoritma 4 aşamaya ayrılabilir:

- Başlatma fazı
- Çalışan arı fazı
- Gözcü arı fazı
- Kaşif arı fazı

Yapay gözcü arı, (2.7) ile hesaplanan besin kaynağı bulma olasılığı p_i değerine bağlı olarak bir besin kaynağı seçmektedir:

$$p_i = \frac{fit_i}{\sum_{n=1}^{BS} fit_n} \quad (2.7)$$

BS : Çalışan arı sayısına veya izleyen arı sayısına eşit besin kaynağı sayısıdır.

fit_i : i . pozisyondaki çözümün uygunluk (fitness) değeri

p_i : i . pozisyondaki çözüme ait uygunluk değerinin keşfedilen tüm besin kaynakların bulunduğu pozisyonundaki toplam uygunluk değerine oranı

Bellekte başlangıç olarak aday bir besin kaynağı üretmek için (2.8)'de verilen ifadeyi kullanır:

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (2.8)$$

Parametreler:

k ve j : $k \in \{1, 2, \dots, BS\}$ ve $j \in \{1, 2, \dots, D\}$ rasgele olarak seçilen indisler olup $k \neq i$ şartı sağlanmalıdır.

$\phi_{ij} : [-1, 1]$ arasında rasgele bir sayıdır. Komşu besin kaynaklarının x_{ij} etrafındaki üretimini kontrol eder ve iki gıda pozisyonunun bir arı tarafından görsel olarak karşılaştırılmasını temsil eder.

(2.8)'de verildiği gibi x_{ij} ve x_{kj} arasındaki fark azaldıkça x_{ij} pozisyonunun değerinde de sapma azalacaktır. Bu durumda arama alanındaki optimum çözüm bulma süreci hızlanacaktır.

Nektarın tüketildiği besin kaynağı kaşifler tarafından terk edilerek yeni bir besin kaynağı ile değiştirilir. YAKO algoritmasında bu davranış rasgele bir konum üreterek ve onu terkedilmiş olanla değiştirerek simüle edilir. YAKO algoritmasında önceden belirlenen sayıda pozisyon geliştirilemezse kaşif arıların besin kaynağını terk ettiği varsayılır. Önceden belirlenen döngü sayısının değeri, YAKO algoritmasının iptal edilmesi için "limit" olarak adlandırılan önemli bir kontrol parametresidir [19].

Terk edilen kaynağın x_i ve $j \in \{1, 2, \dots, D\}$ olduğunu varsayalım. Sonrasında kaşif arı x_i ile değiştirilecek yeni bir besin kaynağı keşfeder. Bu işlem (2.9)'de ifade edilmiştir.

$$x_i^j = x_{min}^j + rand[0, 1](x_{max}^j - x_{min}^j) \quad (2.9)$$

Her aday besin kaynağı pozisyonu v_{ij} hesaplandıktan ve sonra yapay arı tarafından değerlendirildikten sonra, kaynağın performansı bir önceki ile karşılaştırılır. Yeni besin kaynağının performansı eski kaynağın performansına (nektar kalitesi) eşit veya daha iyi ise hafızadaki eski olanla değiştirilir. Aksi halde eski kaynak hafızada tutulur [19].

- Denklem (2.7)'yi kullanarak olasılıksal seçim süreci (başlatma fazı)
- İzleyici arıların yeni besin kaynakları bulmak için yaptıkları keşifler (gözcü arı fazı)
- Denklem (2.8)'de verildiği gibi çiçeklerin (besin kaynakların) rengi, şekli ve kokusu gibi görsel bilgilere bağlı olarak çalışan ve gözcü arıların rasgele arama alanındaki yerel bir olasılıksal seçim süreci (gözcü arı fazı).
- Denklem (2.9)'da verildiği gibi kaşif arılar tarafından gerçekleştirilen rasgele seçim süreci (kaşif arı fazı).

YAKO üç kontrol parametresine dayanmaktadır. Bunlar; besin kaynağı sayısı, limit değeri ve maksimum döngü sayısı. Çalışan arıların kolonideki oranı, yeni keşfedilen bir besin kaynağının ne kadar çabuk bulup kullandığının bir ölçüsüdür. Benzer şekilde çalışan arıların oranı zor optimizasyon problemlerinde kaliteli çözümlerin keşfedilme hızını da ifade eder. Arı kolonisinin hayatta kalması ve ilerlemesi en iyi gıda kaynaklarının hızlı keşfedilmesine ve verimli kullanılmasına bağlıdır. Zorlu mühendislik problemlerinin başarılı çözümü, özellikle gerçek zamanlı olarak çözülmesi gereken problemler için nispeten hızlı ve iyi çözümler bulmaya bağlıdır. Nitelikli bir besin arama faaliyetinde arama ve besin kaynağının kullanılması birlikte yürütülmelidir. YAKO algoritmasında, gözcü ve çalışan arılar arama alanındaki arama faaliyetinden sorumlu iken kaşif arılar bu faaliyeti kontrol eder. Şekil 2.3' te YAKO tekniğinin genel akış şeması verilmiştir [19].

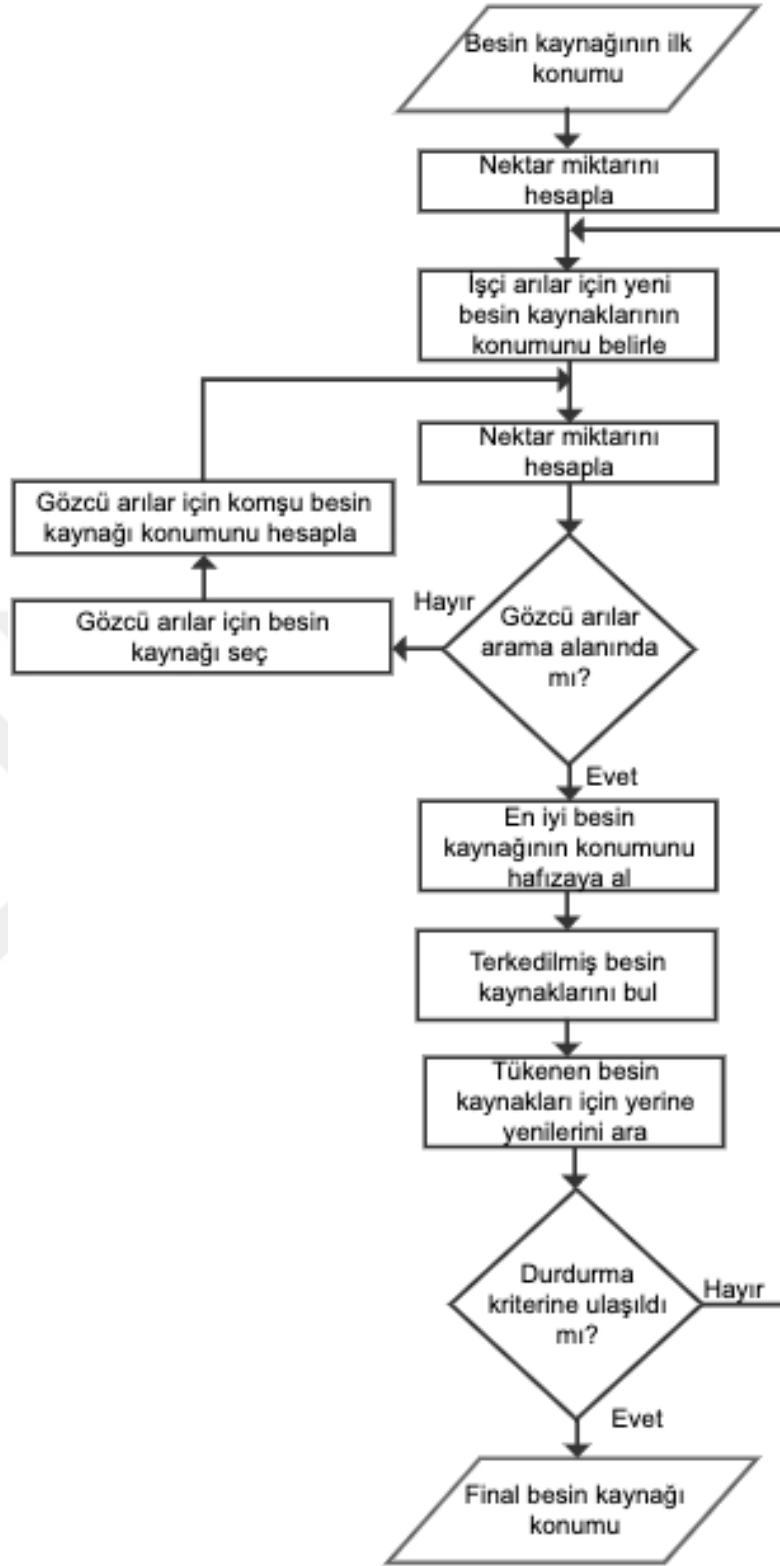
2.4 Ateş Böceği Algoritması Optimizasyonu

Ateş böceği algoritması ilk olarak 2007 yıllarında Yang tarafından ateş böceklerinin yanıp sönmeye düzenleri ve davranışları temel alınarak geliştirilmiştir [10]. Temel olarak 3 kurala dayanır:

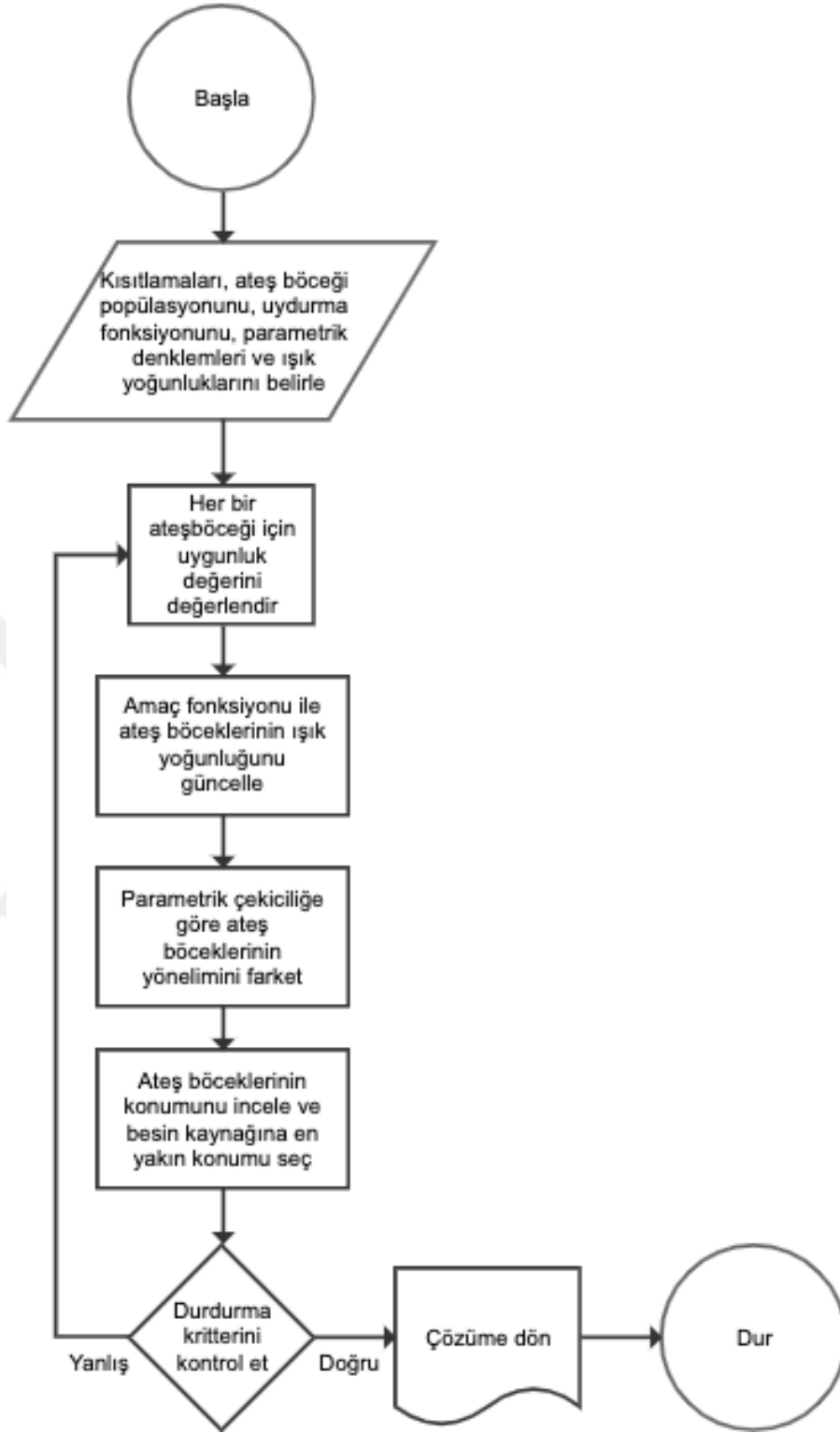
- Ateş böcekleri tek eşeyli olduklarından cinsiyetlerinden bağımsız olarak bir ateş böceği diğeri tarafından çekilecektir.
- Çekicilik oranı parlaklıkla orantılıdır ve birbirlerinden uzaklaştıkça bu oran azalır. Dolayısıyla ateş böceklerinin eğilimi daha parlak olana doğru olacaktır. Herhangi bir ateş böceğinin diğeriyle daha parlak olmadığı durumlarda (parlaklık miktarları arasında fark yoksa) rasgele hareket ederler.
- Bir ateş böceğinin parlaklığı uygunluk fonksiyonu ile belirlenir.

Uygunluk fonksiyonu ateş böceği popülasyonunun yanıp sönen ışık özellikleri ile ilişkilidir. Işık yoğunluğunun fiziksel prensibi göz önüne alındığında, ışık yoğunluğu alanın karesi ile ters orantılıdır. Böylece bu prensip herhangi iki ateşböceği arasındaki mesafeye uygun bir fonksiyon tanımlamayı mümkün kılar. Uydurma (fitting) fonksiyonunun optimizasyonu için popülasyondaki bireyler sistematik veya rasgele hareketlerle popülasyonda hareket etmeye zorlanır. Bu şekilde ateş böceklerinin daha parlak yanıp sönen daha çekici olanlara doğru hareket etmesi sağlanır. Şekil 2.4'te ateş böceği algoritması tekniğinin genel akış şeması verilmiştir [10].

Ateş böceği i 'nin daha parlak olan bir diğeri j 'ye hareketi şu formülle



Şekil 2.3 YAKO Algoritması akış diyagramı [20]



Şekil 2.4 ABAO akış diyagramı [20]

hesaplanmaktadır.

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha \varepsilon_i^t \quad (2.10)$$

Parametreler:

β_0 : $r = 0$ mesafesinde iken ateş böceğinin sahip olduğu çekicilik

α : Rasgele sıralama parametresi

ε_i^t : t anında Gauss veya homojen dağılımdan çizilebilen rasgele sayıların bir vektörü

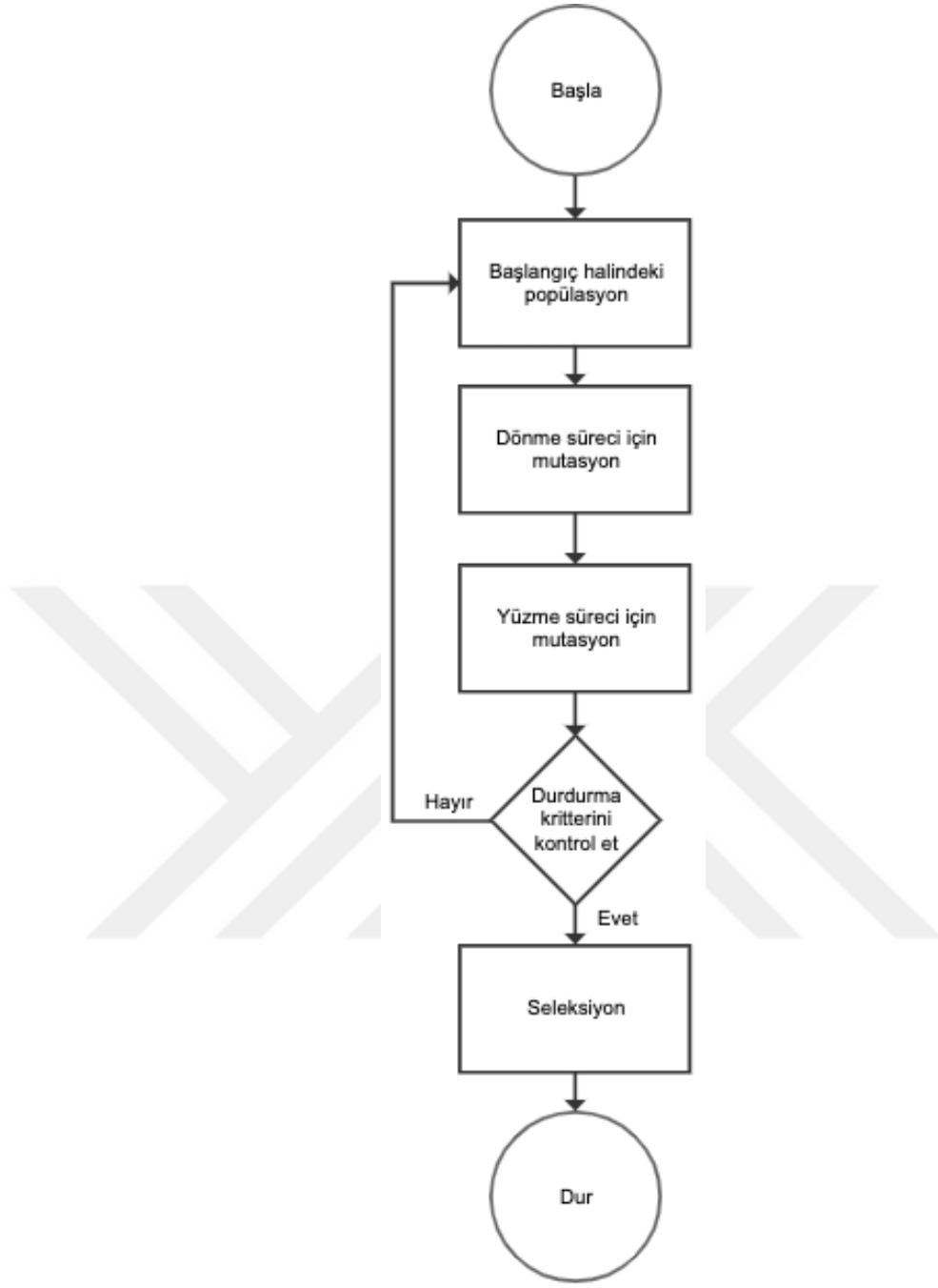
2.5 Bakteriyel Besin Arama Optimizasyonu

E.coli ve M.Xanthus bakterilerinin davranışlarından esinlenen bu algoritma 2002 yılında Passino tarafından öne sürülmüştür. Bu bakteriler birim zamandaki enerjiyi en verimli şekilde kullanarak besin arama işlevini yerine getirirler. E. coli bakterisi besin kaynağına ulaştığında ortamda bulunan 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 [11]. Bakterilerin besin arama bölgesindeki davranışı aşağıda maddeler halinde verilmiştir;

- Bakteriler arama alanında daima besin yoğunluğunun daha fazla olduğu alana doğru ilerlerler. Yeterince besin kaynağına sahip bakteriler daha uzun ömre sahiptir ve iki eşit parçaya bölünür. Daha az besin bölgesindeki bakteriler de dağılır ve ölür.
- Besin açısından daha zengin bölgede bulunan bakteriler kimyasal feromon ile diğerlerine doğru çekilirler. Şekil 2.5'da Bakteriyel Besin Arama Optimizasyonu tekniğinin genel akış şeması verilmiştir.

2.6 Guguk Kuşu Arama Optimizasyonu

Guguk Kuşu Arama Optimizasyonu (GKAO), 2009 yılında Yang ve Deb tarafından önerilen bu algoritma guguk kuşlarının yuvalarındaki diğer ev sahibi guguk kuşları ile birlikte yumurtaları üzerinde oturma davranışından esinlenilerek ortaya konulmuştur. Her bir guguk kuşu rasgele seçilen bir yuvada belirli bir zaman süresince yumurtaları üzerinde oturur. Yüksek kaliteli yumurtaları olan en iyi yuvalar seçilerek bir sonraki



Şekil 2.5 BBAO akış diyagramı [20]

nesillere taşınır. En kötü yuva, daha iyisi bulunmak üzere guguk kuşları tarafından terk edilir ve arama davranışı içerisine girerler [12].

Guguk kuşu yavruları büyüdüğünde ve yumurtlama zamanı geldiğinde kuş mevcut bulunduğu ortamdan yumurtalarını yabancı kuşların yumurtalarına daha çok benzetebileceği türlere sahip ve yavruları için daha çok yiyecek kaynağına sahip yeni ve daha iyi bir ortama göç ederler. Guguk kuşlarının farklı alanlarda grup oluşturmasının ardından en iyi üretime sahip olan topluluk diğer guguk kuşlarının göç etmesi için hedef nokta olarak seçilir. Guguk kuşu grupları en iyi ortamı bulana kadar göç olayını devam ettirirler [12].

Guguk kuşları saldırgan üreme stratejileri ile diğer kuşlardan ayrılmaktadırlar. Bazı türleri yumurtalarını uygun gördükleri farklı kuşların yuvalarına bırakıp yuva içerisindeki bir yumurtayı dışarı atmaktadırlar. Fark edilmeyecek derecede benzerliğe sahip yumurtaları sayesinde yuva sahibi kuşun kuluçka parazitliğine katlanmasına mecbur bırakırlar [12].

Her guguk kuşu her seferinde bir yumurta bırakır ve onu rastgele bir yuvaya atar; en iyi yuvalar (en yüksek kalitede yumurta içeren) yumurtayı bir sonraki kuşağa aktarır. Ev sahibi yuvalarının sayısı sabittir ve her zaman guguk kuşunun ürettiği yumurtanın ev sahibi tarafından keşfedilme olasılığı vardır. Keşfedildiği durumda ev sahibi kuş yumurtayı dışarı atar. Esasen bu kurallar en iyi yumurtaların nesilden nesile hayatta kalmasını sağlayan optimizasyon algoritması için bir seçim süreci sağlar [12].

2.7 Sıçrayan Kurbağa Optimizasyonu

PSO' ye ait hız parametresinin kullanılmadığı Correa ve Freitas tarafından önerilen sezgi-üstü (meta-heuristic) yöntemlerden biridir. Ayrık uzayda sürekliliğin olmadığı durumlarda hareket, hız ve atalet kavramı anlamını yitirmektedir. Bu parametrelerin olmadığı durumlarda SKO devreye girer ve çekiciliği en iyi pozisyonda tutar. Hız ve atalet parametreleri yerine partiküllerin hareketinde rasgele bir bileşen olduğu varsayılır. Bu rasgele bileşen kurbağaların zıplama şekli olarak belirlenmiştir. Partikülün hızı ve konumunun güncellenmesi Correa ve Freitas önerilen yöntemle göre partikülün en iyi konumunun çekiciliği, en iyi global konumun çekiciliği veya kendi sosyal ortamındaki en iyi konumunun çekiciliği ile yönlendirilmesinin olasılıkları olarak yorumlanır [13].

2.8 Mirket Klan Algoritması Optimizasyonu

Yeni bir sürü zekası algoritması olarak öne sürülen Mirket Klan Algoritması Optimizasyonu (MKAO), mirket hayvanlarının klan içindeki davranışlarından esinlenilerek 2018 yılında öne sürülmüştür [14].

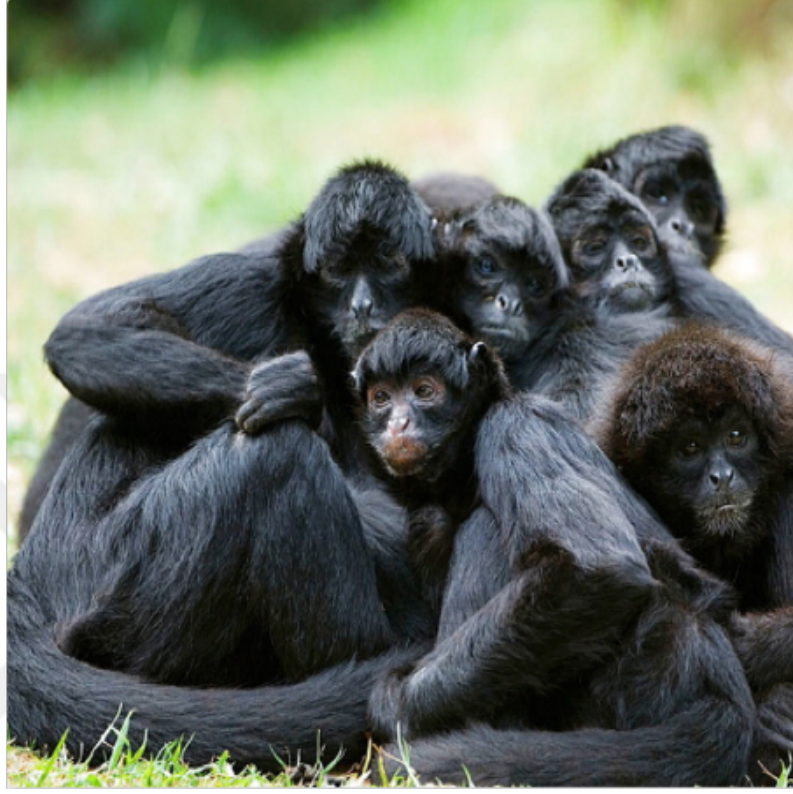


Şekil 2.6 Mirket Hayvanı

Kolonilerinde 5'ten 30'a kadar mirketle yaşayan bu hayvanlar nöbetçi, besin avcısı ve bebek bakıcısı görevleri ile sınıflandırılmıştır. Koloni içerisinde en az bir adet mirket, kolonilerini dışarıdan gelebilecek tehlikelere karşı korumak için nöbetçi olarak görevlendirilir. Nöbetçi mirket veya mirketler tehlikeli bir durumla karşılaştığında havlama davranışında bulunarak kolonideki diğer mirketleri uyarır. Bu uyarı sinyalini alan mirketler yuvalarına koşar. Nöbetçi mirketler tehlike geçene kadar diğer mirketlerin yuvadan çıkmasını engellemek için havlamaya devam ederler. Koloni içerisindeki genç üyeler tehlike anında bebek bakıcısı görevini üstlenen mirket tarafından güvenli bölgeye yerleştirilir. Mirketler besin arama davranışı sırasında birbirleri ile görsel ve işitsel temasını sürdürürler. Bu algoritmada koloninin en iyileri nöbetçi olarak seçilirken geri kalanı besin avcısı ve bebek bakıcısı olarak 2 farklı gruba ayrılır. Besin avcısı grubundaki en kötü performansa sahip mirket bebek bakıcı grubuna aktarılır. Bebek bakıcısı grubundaki en kötü performansa sahip birey ise gruptan çıkarılır ve yerine başka bir mirket dahil edilir. Besin avcısı grubundaki en iyi birey ise nöbetçi görevini üstlenen mirket ile değiştirilir. Her mirket kolonisi hem dişi ve hem de erkek lidere sahiptir [14].

2.9 Örümcek Maymun Optimizasyonu

Örümcek Maymun Optimizasyonu (ÖMO), 2014 yılında örümcek maymunların avlanma davranışından esinlenilerek önerilen diğer sürü zekası tabanlı algoritmalarından biridir [15].



Şekil 2.7 Örümcek Maymunu

Örümcek maymunlar bölünme-birleşme sosyal yapısında kategorize edilirler. Bu sosyal yapıya göre örümcek maymunlar besin kıtlığı veya besine ulaşılabilirlik açısından büyük gruplardan küçük gruplara ayrılırlar. Bu optimizasyon yönteminde grubu yönetmek ve besin kaynağı bulmakla görevli dişi bir lider bulunmaktadır. Dişi lider yeterli besin kaynağı bulamadığı durumlarda grubu alt gruplara böler ve bu alt grupların liderleri yine dişi üyelerden oluşmaktadır. Alt gruplar besin arama davranışına devam ederken bulunan besin kaynaklarından haberdar olmak ve belirli çevresel sınırlar içinde kalmak amacıyla birbirleri ile iletişim halinde olurlar [15].

Örümcek maymunların avlanma davranışı araştırmacılara bölünme-birleşme (fission-fusion) sosyal yapısına dayalı yeni bir stokastik optimizasyon tekniğinin gelişmesinde ilham olmuştur. Bölünme-birleşme sosyal yapısının özellikleri aşağıda maddeler halinde verilmiştir [15]:

- Belirtilen sosyal yapıya sahip hayvanlar 40-50 bireyden oluşan gruplarda yaşamakla beraber bu yapıya sahip sürüler grup üyeleri arasındaki avlanma

rekabetini azaltmak ve besin aramak için grubu alt gruplara böler.

- Grubu yöneten dışı lider (global lider) aynı zamanda besin kaynağı aramakla da yükümlüdür. Yiyecek bulamadığı durumlarda grubu 3-8 bireyden oluşan küçük alt gruplara böler.
- Oluşan bu küçük alt gruplar yine dışı bir lider (bölgesel lider) tarafından yönetilmektedir. Küçük alt gruplardaki liderler avlanma için belirlenen rotaların hangisinin daha uygun olduğuna karar vermekle görevlidir.
- Alt grupların üyeleri belirli arama sınırları içerisinde kalmak ve besin kaynağının ulaşılabilirliğine bağlı olarak kendi içlerinde ve birbirleri ile sürekli iletişimde kalırlar.

Geliştirilen bu stratejiye göre bu sosyal yapıya sahip hayvanların avlanma davranışı Şekil 2.8'de açıklanmaktadır. Bu dört aşama istenilen sonuç elde edilene kadar birbirini tekrar eder. Bu optimizasyon yönteminde iki adet kontrol parametresi vardır. Bunlardan biri global lider limiti diğeri ise bölgesel lider limit olarak adlandırılır. Bölgesel lider limiti alt grubun besin arama davranışının kesintiye uğramaması için kullanılır. Eğer bölgesel lider tanımlanan süre içerisinde konumunu güncellemezse alt grup besin arama davranışına farklı yönlerde devam eder. Buradaki tanımlanan süre bölgesel lider limiti kontrol parametresi olarak adlandırılır. Global lider limiti de aynı amaçla global lider için kullanılır. Global lider tanımlanan süre içerisinde konumunu güncellemezse grubu alt gruplara böler. Örümcek Maymun Optimizasyonu yönteminde sürünün kendine ait organizasyon yeteneğine sahip olması ve iş bölümünde bulunabilmeleri nedeniyle sürü zekası algoritmalarından biri olma özelliği kazanmaktadır [15].

2.9.1 Örümcek Maymun Optimizasyonu Algoritması

ÖMO algoritması 6 fazdan oluşmaktadır. Bunlar lokal lider fazı, global lider öğrenme fazı, lokal lider öğrenme fazı, global lider öğrenme fazı, lokal lider karar fazı ve global lider karar fazıdır [15].

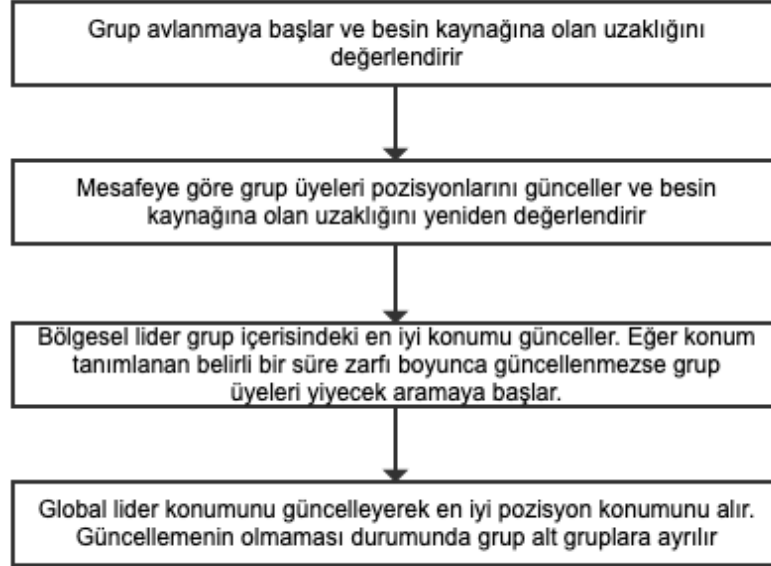
2.9.1.1 Popülasyonun Başlangıcı

ÖMO, her bir örümcek maymununun OM_z ($z = 1, 2, \dots, n$) d eksenel vektör ile gösterildiği N tane örümcek maymununun olduğu homojen dağılmış bir popülasyon üretir. Bu bölümde örümcek maymunu kısaca OM şeklinde ifade edeceğiz [15].

$$OM_{zj} = OM_{minj} + U(0, 1) \times (OM_{maxj} - OM_{minj}) \quad (2.11)$$

Tablo 2.1 Üyeler ve Sürü Zekası Algoritmaları

Üye	Sürü Zekası Algoritması
Parçacık	Parçacık Sürü Optimizasyonu
Karınca	Karınca Kolonisi Optimizasyonu
Arı	Yapay Arı Kolonisi Optimizasyonu
Ateş böceği	Ateş Böceği Algoritması Optimizasyonu
Bakteri	Bakteriyel Besin Arama Optimizasyonu
Guguk Kuşu	Guguk Kuşu Arama Optimizasyonu
Kurbağa	Sıçrayan Kurbağa Optimizasyonu
Mirket	Mirket Klan Algoritması Optimizasyonu
Maymun	Örümcek Maymun Optimizasyonu



Şekil 2.8 Örümcek Maymun Sürüsü Avlanma Davranışı

Parametreler:

d : Optimizasyon problemindeki deęişkenlerin sayısıdır.

OM_z : Popülasyondaki z . örümcek maymundur.

$U(0, 1)$: 0 ve 1 arasında rasgele bir sayıdır.

OM_{maxj} ve OM_{minj} : OM_z 'ye ait j yönündeki sınırlarıdır.

j : $j \in \{1, 2, .. D\}$

2.9.1.2 Lokal Lider Fazı

Bu aşamada isminden de anlaşıldığı gibi lokal (bölgesel) grup liderinin konum belirleme davranışı anlatılmaktadır [15].

- Her bir örümcek maymun lokal liderin ve aynı zamanda lokal grup üyelerinin deneyimleri bilgisi ile konumunu deęiştirir.
- Yeni konumun fitness deęeri hesaplanır.
- Yeni konumun eski konuma göre fitness deęeri daha yüksek ise örümcek maymun konumunu yeni konum olarak günceller.

K. lokal grubun üyesi olan z . örümcek maymun için konum güncelleme denklemi;

$$OM_{yenzj} = OM_{minj} + U(0, 1) \times (LL_{kj} - OM_{minj}) + U(-1, 1) \times (OM_{rj} - OM_{zj}) \quad (2.12)$$

Parametreler:

LL_{kj} : k . lokal grup liderinin j . eksenindeki konumudur.

OM_{yenzj} : z . örümcek maymunun j . eksenindeki güncel konumudur.

OM_{zj} : z . örümcek maymunun j . eksenindeki konumudur.

OM_{rj} : k .grup içerisinde $r \neq j$ şartı ile rasgele seçilen r . maymunun j . eksenindeki konumudur.

2.9.1.3 Global Lider Fazı

Lokal lider fazının tamamlanmasından sonra global lider fazı başlar. Bu fazda tüm örümcek maymunlar konumlarını lokal grup üyeleri ve global liderlerin deneyimlerine göre güncellerler [15].

$$OM_{yeni z_j} = OM_{z_j} + U(0, 1) \times (GL_j - OM_{z_j}) + U(-1, 1) \times (OM_{r_j} - OM_{z_j}) \quad (2.13)$$

Parametreler:

GL_j : Global grup liderinin j . eksenindeki konumudur.

$OM_{yeni z_j}$: z . örümcek maymununun j . eksenindeki konumudur.

OM_{r_j} : k . grup içerisinde $r \neq j$ şartı ile rasgele seçilen r . maymunun j . eksenindeki konumudur.

Bu aşamada güncelleme $olasilik_z$ dahilinde yapılır.

$$olasilik_z = 0,9 \times \frac{fitness_z}{maxfitness_z} + 0,1 \quad (2.14)$$

Yeni üretilen konumun fitness(uygunluk) değeri hesaplanır ve eski olanla karşılaştırılır. Olasılığı yüksek olan tercih edilir.

Parametreler:

$maxfitness_z$: Grubun en büyük uygunluk değeridir.

$fitness_z$: z . maymununun uygunluk değeri

2.9.1.4 Global Lider Öğrenme Fazı

Popülasyondaki en yüksek uygunluk değerine sahip örümcek maymununun konumu global liderin güncel konumu olarak seçilir [15].

2.9.1.5 Lokal Lider Öğrenme Fazı

Gruptaki en iyi fitness değerine sahip örümcek maymununun konumu lokal liderin güncel konumu olarak seçilir [15].

2.9.1.6 Global Lider Karar Fazı

Global bölgelerdeki duraksamalardan kaçınmak amacıyla kullanılır. Bu durumda popülasyon gruplara ayrılır [15].

2.9.1.7 Lokal Lider Karar Fazı

Lokal bölgelerdeki duraksamalardan kaçınmak amacıyla kullanılır. Bu durumda popülasyon alt gruplara ayrılır [15].

2.10 Çok Yöneticili Mobil Robotların Sürü Hareketi

Üç boyutlu uzayda, önerilen çok yöneticili robotların hareketi sürü zekasına dayandırılarak yeni bir sürü hareketi metodu öne sürülmüştür. Çok yöneticili bir sürü hareketi planlaması ile aynı mantıkta I^2 Haberleşme Protokolü bulunmaktadır. I^2 protokolü düşük bant genişliğine sahip, kısa mesafeli bir senkron haberleşme protokolüdür. Haberleşme için iki hat kullanılır bunlar SCL ve SDA hatlarıdır. SCL (Serial Clock) veri senkronizasyonunu sağlayan hat iken SDA (Serial Data) ise verilerin iletiği hattır. I^2 veriyolu çok yöneticili olmasından dolayı birden fazla sayıda data transferini başlatabilme özelliğindeki tümdevre, veriyoluna bağlanabilir [21].

Çalışmada bahsedilen çok yöneticili mobil robotların sürü hareketi planlanırken şu varsayımlardan yararlanılmıştır. Amacımız hareket boyunca optimum yolun belirlenmesi olduğu için sürünün minimum 4 adet mobilden oluşması gerekmektedir. Robotlar sürüdeki görevine göre ikişer adet yönetici (kaşif) ve takipçi robot olarak ayrılmaktadır. Belirlenen robot sayısı ve hiyerarşisi dışında bir adet yönetici (kaşif) robotun olması durumunda takipçi robotlar optimum yolu takip etmektense yönetici (kaşif) robot nereye giderse oraya yönlenecektir. Bu yüzden popülasyonda en az iki yönetici (kaşif) robotun bulunması gerekmektedir. Benzer şekilde üç şeritli bir otoyolda en hızlı giden şeridi bulmak için üç farklı arabanın her bir şeridi ayrı ayrı deneyimlemesi gerekiyorsa, yönetici (kaşif) robotların da optimum yolu bulmak için alternatif yollar adedinde olması gerekmektedir. Üç şeritli bir yolu tek veya iki adet yönetici (kaşif) robotun deneyimlemesi optimum yolun bulunması açısından yetersizdir. Dört adet yönetici robotun kullanılması da bu perspektifte gereksiz olacaktır. İki adet takipçi robotun kullanılma nedenine değinecek olursak, iki mobilden biri konumunu kaybederse çevresindeki robotlara yayın (broadcast) yaparak konum bilgisi ister. Bu durumda konumunu kaybeden robot her bir robotla haberleşerek üç robotun konum bilgisini yani RSSI bilgisini alır.

2.10.1 Çok Yöneticili Mobil Robotların Sürü İçerisindeki Hiyerarşik Dağılım

Sürü içerisindeki mobil robotlardan hangisinin yönetici hangisinin takipçi olarak seçileceği belirlenmiştir. Belirlenen bu dağılım yapısında yönetici robotlar hedefe giden alternatif yollarda en önde bulunan robotlardır. Sürü içerisindeki robotlardan kaç tanesinin yönetici olacağı ise (2.15)'te verildiği şekli ile ifade edilmiştir. (2.15)' e göre en az hedefe giden alternatif yollar adedinde yönetici vasfına sahip mobil robot olmalıdır.

$$\text{Alternatif yol sayısı} \geq \text{Yönetici robot sayısı} \quad (2.15)$$

Yönetici robot sayısının alternatif yol sayısından fazla olması durumunda yönetici vasfına sahip olabilecek robotlar kendi aralarında kıyaslanır ve hareket kabiliyeti, sahip olunan bataryaların yüksek olması vb. açılardan değerlendirilerek biri diğerine göre daha üstün olan robot yönetici vasfı ile sürü içerisinde hareket eder.

Bu vasıfların aynı olması durumunda yönetici robotlar grup halinde hareket ederler.

2.10.2 Mobil Robotların Ortam Sınırlarının Belirlenmesi

Önerilen sürü hareketi planlamasında hedefe giden her güzergah için yönetici ve takipçi robotlar tayin etmiştik. Sürü hareketinin planlandığı ortamda sınırların belli olduğu varsayılarak Şekil 2.9'de olduğu gibi d birimlik alanın k adedinde şeritlere bölelim. Bu durumda d birimlik alanı taraması gereken yönetici (kaşif) ve takipçi mobil robot sayısı (2.16)'da verildiği üzere formülize edilmiştir.

$$i = d / (k + 1) \quad (2.16)$$

Parametreler:

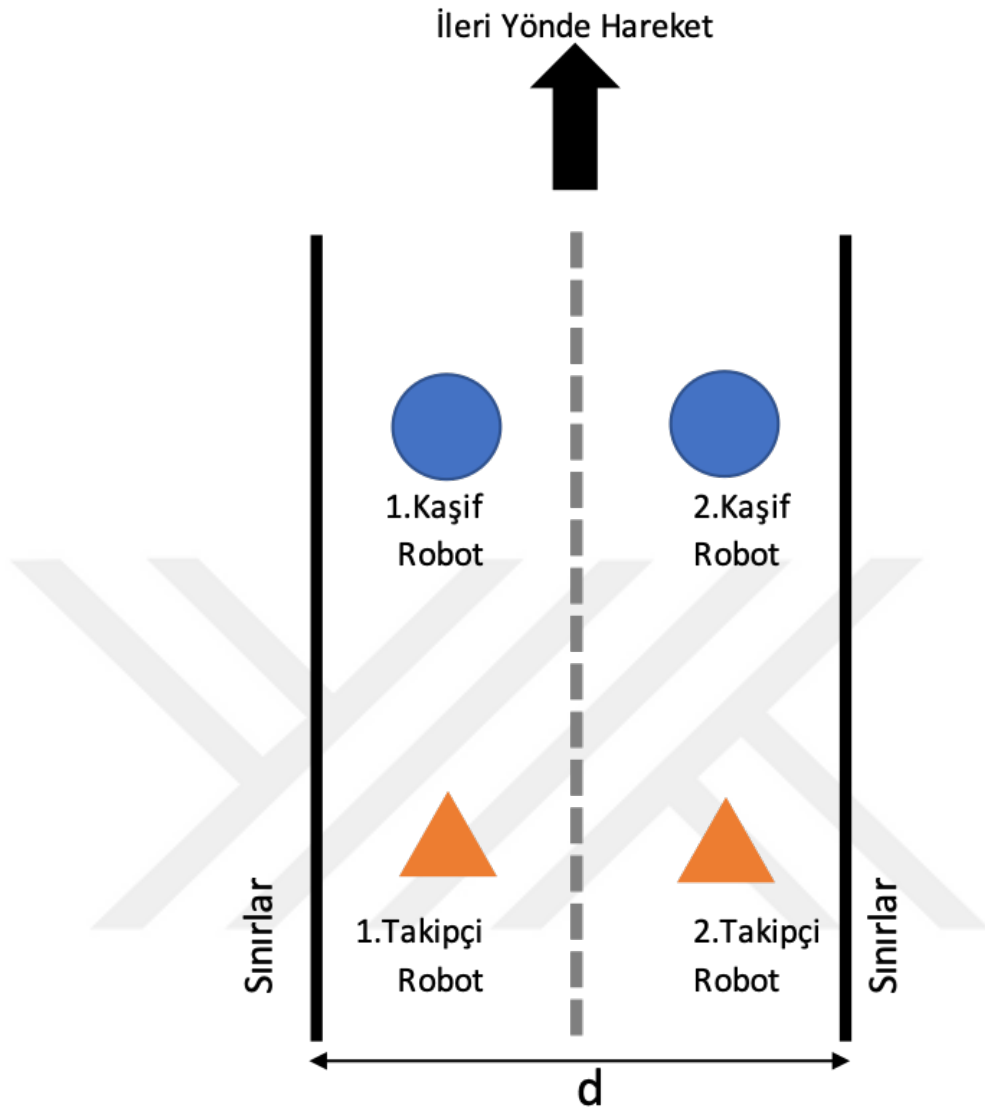
i : Robot etiketi

d : Ortam sınırları belirli olan sürü hareket alanının genişliği

k : Alanın kaç bölündüğünü ifade eden şerit sayısı

2.11 Optimizasyon Yöntemlerinin Karşılaştırılması

Bu çalışmada incelenen optimizasyon yöntemlerinin önerilen birden fazla yöneticili mobil robotların hareketine uygunluğu Tablo 2.2'de kıyaslanmıştır. Kıyaslama sonucu



Şekil 2.9 Mobil Robot Ortam Sınırları

önerilen mobil robot hareketine en uygun sürü hareketinin Örümcek Maymun Optimizasyonu olduğu anlaşılmıştır.

Bu çalışmada planlanan mobil robotların sürü hareketi ile ÖMO algoritması arasındaki ilişki Tablo 2.3'te verilmiştir:



Tablo 2.2 İncelenen Algoritmaların hiyerarşi bakımından karşılaştırması

Algoritma	PSO	KKO	YAKO	ABA	BBA	GKA	SKA	MKA	ÖMO
Hiyerarşi	-	-	Gözcü, işçi ve kaşif	-	-	-	-	Nöbetçi, besin avcısı ve bebek bakıcısı	Dişi liderler, takipçiler
Yönetici	-	-	-	-	-	-	-	-	Birden fazla

Tablo 2.3 ÖMO algoritması ve Çok Yöneticili Mobil Robotların Sürü Hareketi İlişkilendirilmesi

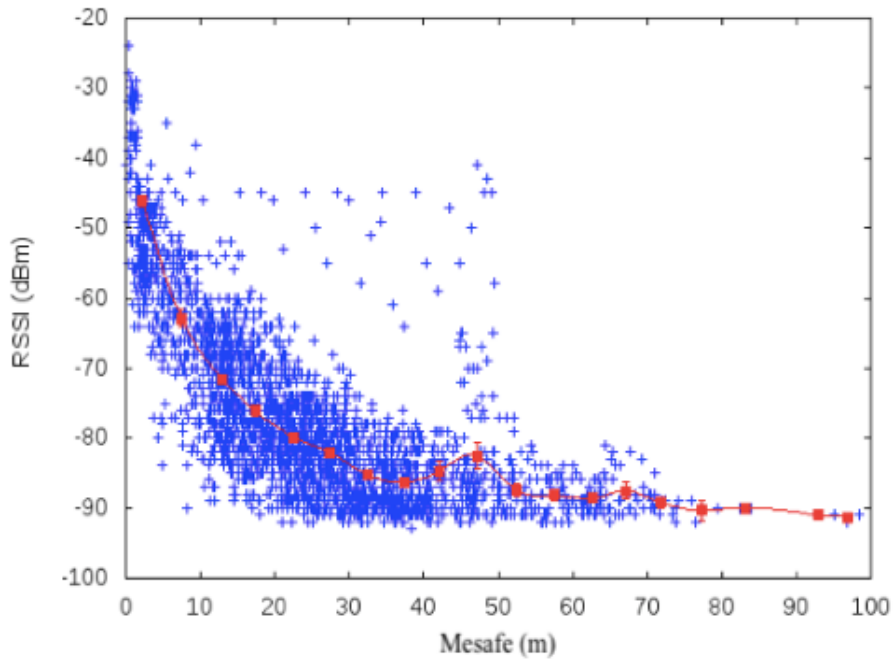
Esinlenen Davranışlar	Mobil Robotların Davranışı
Besin aramak için küçük alt gruplara bölünme	Mobil robotların sınırları belirli olan ortam içerisinde hedeflenen göreve ulaşabilmesi için gruplara ayrılarak farklı rotaları deneyimlemesi
Her alt grubun bir tane dişi liderinin bulunması	Ayrılan her robot grubuna ait bir tane yönetici (kaşif) robotun bulunması ve diğer takipçi robotların yönetici (kaşif), robotu izlemesi
Oluşan bu küçük alt gruplar yine dişi bir lider (bölgesel lider) tarafından yönetilmektedir. Küçük alt gruplardaki liderler avlanma için belirlenen rotaların hangisinin daha uygun olduğuna karar vermekle görevlidir.	İzlenen rotaların kıyaslanarak hedefe ulaşmada en kısa ve hızlı olanının seçilmesi
Alt grupların üyeleri belirli arama sınırları içerisinde kalmak ve besin kaynağının ulaşılabilirliğine bağlı olarak kendi içlerinde ve birbirleri ile sürekli iletişimde kalırlar.	Mobil robot gruplarının konum bilgisi ve data aktarımı için birbirleri ile sürekli iletişimde kalması

2.12 Konum Belirleme Teknikleri

Mobil robotların kontrolü, arama alanındaki dağılımı ve birbiri ile senkronize olabilmeleri konuları açısından oldukça önem taşımaktadır. Çalışmanın bu bölümünde mobil robot uygulamalarında kullanılan bazı konum belirleme teknikleri hakkında bilgi verilmiştir.

2.12.1 Alınan Sinyalin Gücü Göstergesi (RSSI)

Kapalı mekan uygulamalarında geniş çapta kullanılan en basit yaklaşımlardan biridir. Genellikle desibel-miliwatt cinsinden alıcı kanaldan alınan sinyalin gücünü ifade eder. RSSI, alıcı ve verici arasındaki mesafeyi tahmin ederken yüksek RSSI değeri alıcı ve verici arasındaki mesafenin kısa olduğunu ifade eder [22], [23] ve [24]. RSSI alınan



Şekil 2.10 RSSI ve mesafe arasındaki ilişki [25]

sinyalin gücünü belirten bir gösterge olup, göstergenin ilişkili ölçüm değeri çip üreticisi firma tarafından belirlenir. Atheros 0 – 60 değerini kullanırken, Cisco 0 – 100 aralığını kullanmaktadır. Genel olarak tüm üreticiler 0'a en yakın olan değer her zaman daha iyi sinyal olduğunu göstermektedir [26]. Bu çalışmada RSSI bilgisi mobil robotların ortam içerisinde birbirlerini kaybetmemeleri ve sürekli olarak iletişimde kalmalarını sağlamak için kullanılmıştır. Mobil robotlar iletişimin kesintisiz olarak sağlandığı belirli bir mesafenin dışına çıktıklarında sahip oldukları RSSI bilgisi ile birbirlerini belli sınırlar içerisinde kalmak için uyarırlar.

2.12.2 Parmak İzi Analizi

Bu tekniğe dayalı konum belirleme yönteminde analizi sağlayacak veriler için ortam veya çevresel koşullara ait özelliklerin araştırılması gerekmektedir. Başlangıçta çevrimdışı iken RSSI ölçümleri toplanır. İkinci aşamada ise gerçek eş zamanlı olarak çevrimiçi ölçümler alınır ve her iki ölçüm değerleri kıyaslanır. Çeşitli algoritmalar (olasılıksal) metotlar, yapay sinir ağları, en yakın komşu metodu ve destek vektör makineleri vb.) yardımıyla sinyalin ait olduğu konum tahmin edilir. Daha sonra sinyalin gerçekte ait olduğu konum ve tahmin edilen konumun birbirleriyle kıyaslanarak kullanılan algoritmanın doğruluğu belirlenir [27].

2.12.3 GPS (Global Positioning System)

1960'lı yıllardan beri kullanılan ve Amerikan Ordusu tarafından geliştirilen GPS sistemi günümüzde halen işlevini tam olarak yerine getirmekle birlikte optimum konum belirleme sistemi için belirlenen kriterleri (küresel kapsama alanı, sürekli ve tüm hava koşullarında çalışma, hızlı değişen platformlara hizmet verebilme yeteneği ve yüksek doğruluk) karşılamaktadır. Bu sistem uygun ekipmanlarla birlikte kullanıcılara üç boyutlu (3B) konum ve hız bilgisi sağlamaktadır. GPS sistemi en çok navigasyon uygulamalarında kullanılmakla birlikte sağladığı frekans ve zaman doğruluğu nedeniyle telekomünikasyon uygulamalarında da kullanılmaktadır [28].

ÜÇ BOYUTLU UZAYDA KONUMUNUN MATEMATİKSEL OLARAK MODELLENMESİ VE HATA KALIBRASYONU

Bu bölümde ilk alt başlıkta 3 Boyutlu (3B) uzayda koordinat sistemi ile konumun belirlenmesi konusuna değinilmiştir. Robot sürüsüne dahil olan her bir i. robot küre denklemleri ile ifade edilerek matrisler oluşturulmuştur. İkinci alt başlıkta ise matematiksel olarak belirlenen konumun içerdiği veya içerebileceği hata miktarının kalibrasyonu üzerinde çalışılmıştır. Kalibrasyon için sözde ters matrisinden yararlanılmıştır.

3.1 Üç Boyutlu Uzayda Sisteminde Konumun Matematiksel Olarak Modellenmesi

3B uzayda herhangi bir noktaya ait konumun belirlenebilmesi için üç farklı noktaya ihtiyaç duyulmaktadır. Bu üç noktanın kesiştiği nokta konumu belirlenmek istenen bir noktanın pozisyonunu belirtir.

Örnek olarak Şekil 3.1'de üç adet küre 3B uzayda çakıştırılmıştır. MATLAB kodları Ek-A'da Şekil A.1'de verildiği gibidir.

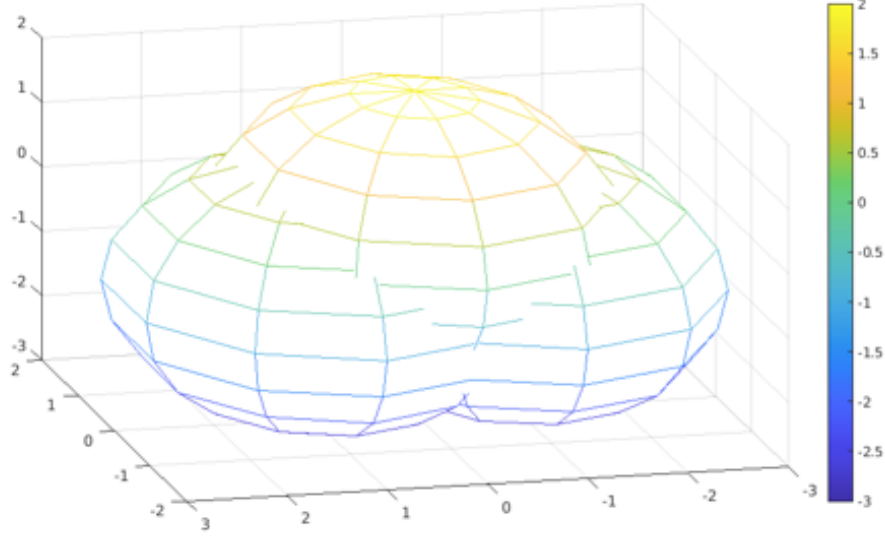
Küre Denklemi:

Merkezi $C(h, k, l)$ ve yarıçapı r olan kürenin denklemi;

$$(x - h)^2 + (y - k)^2 + (z - l)^2 = r^2 \quad (3.1)$$

Üç Boyutlu Uzayda İki Nokta Arasındaki Mesafe:

$P_1(x_1, y_1, z_1)$ ve $P_2(x_2, y_2, z_2)$ 3B uzayda tanımlı iki nokta olsun. Bu iki nokta arasındaki mesafe (3.2)'de verilmiştir:



Şekil 3.1 MATLAB yardımıyla üç adet kürenin çakıştırılması

$$|P_1P_2| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (3.2)$$

3.2 Hatanın Kalibrasyonu

Bu kısımda mobil robotlar için önerilen yöntem ile bilinen konum ve GPS verisinden öğrendiği gerçek konum arasındaki hata miktarını üretilen matematiksel modellemeye yararlanarak kalibre eder. Gerçek konum mobil robotun matematiksel olarak hatalardan arındırılmış konumunu temsil eder. Bilinen konum ise mobil robotun matematiksel olarak hataları da içerdiği, hatalarla ile olan toplam konumudur.

Matematiksel modellemede kullanılan her bir değişkenin terimsel gösterimi aşağıda verilmiştir.

Terimlerin gösterimi

Robotun gerçek konumu: G

Robotun bilinen konumu: B

Durum etiketi: D

Yönetici (Kaşif) robot: Y

Takipçi robot: T

Mobil Robot etiketi: $i=1, \dots, n$.

Hata: ϵ

Yukarıda verilen terimsel gösterim kullanılarak her bir mobil robotun üç boyutlu uzaydaki gerçek ve bilinen konumu aşağıdaki gibi nitelendirilir.

${}^D_G X_i(t)$ = i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki X eksenine göre gerçek konumu,

${}^D_G Y_i(t)$ = i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Y eksenine göre gerçek konumu,

${}^D_G Z_i(t)$ = i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Z eksenine göre gerçek konumu,

${}^D_B X_i(t)$ = i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki X eksenine göre bilinen konumu,

${}^D_B Y_i(t)$ = i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Y eksenine göre bilinen konumu,

${}^D_B Z_i(t)$ = i. robotun popülasyon içerisinde sahip olduğu role göre t anındaki Z eksenine göre bilinen konumu,

Mobil robot etiketi i ile gösterilen mobil robotların üç boyutlu eksenlerdeki hata tanımı her bir eksen için ayrıca tanımlanmıştır.

$\epsilon_{X_i}(t)$ = i. robotun t anındaki X eksenine göre hatası,

$\epsilon_{Y_i}(t)$ = i. robotun t anındaki Y eksenine göre hatası,

$\epsilon_{Z_i}(t)$ = i. robotun t anındaki Z eksenine göre hatası,

Bilinen ve gerçek konumun nitelendirilmesiyle hata tanımı anlamlandırılır. Bilinen konumdan gerçek konumun çıkarılmasıyla mobil robotların üç boyutlu uzayda sahip olduğu gerçek konum modellenir.

$${}^D_G X_i(t) = {}^D_B X_i(t) - \epsilon_{X_i}(t) \quad (3.3)$$

$${}^D_G Y_i(t) = {}^D_B Y_i(t) - \epsilon_{Y_i}(t) \quad (3.4)$$

$${}^D_G Z_i(t) = {}^D_B Z_i(t) - \epsilon_{Z_i}(t) \quad (3.5)$$

3B Gerçek Pozisyon

Üç boyutlu uzayda mobil robotların sahip olduğu gerçek pozisyon ${}^D P_i(t)$ ifadesi ile tanımlanıp 3 boyutlu uzayda (3.6) ve (3.7) denklemlerinde verildiği şekli ile gösterilmiştir.

$${}^D P_i(t) = ({}^D X_i(t), {}^D Y_i(t), {}^D Z_i(t)) \quad (3.6)$$

veya

$${}^D P_i(t) = {}^D (X_i(t), Y_i(t), Z_i(t)) \quad (3.7)$$

3B Bilinen Pozisyon

Üç boyutlu uzayda mobil robotların bilinen pozisyonu ${}^B P_i(t)$ ifadesi ile tanımlanıp 3 boyutlu uzayda (3.8) ve (3.9) denklemlerinde verildiği şekli ile gösterilmiştir.

$${}^B P_i(t) = ({}^B X_i(t), {}^B Y_i(t), {}^B Z_i(t)) \quad (3.8)$$

veya

$${}^B P_i(t) = {}^B (X_i(t), Y_i(t), Z_i(t)) \quad (3.9)$$

Verilen bu matematiksel ifadelerle göre iki robot arasındaki bilinen mesafe (3.10)'daki gibi ifade edilir.

Mobil robotlardan biri a diğeri ise b olsun;

$${}^D |P_a(t) P_b(t)| = {}^D (\sqrt{(X_a(t) - X_b(t))^2 + (Y_a(t) - Y_b(t))^2 + (Z_a(t) - Z_b(t))^2}) \quad (3.10)$$

Aynı şekilde iki mobil robot arasındaki gerçek mesafe ise (3.11) ile ifade edilmiştir;

$${}^D |P_a(t) P_b(t)| = {}^D (\sqrt{(X_a(t) - X_b(t))^2 + (Y_a(t) - Y_b(t))^2 + (Z_a(t) - Z_b(t))^2}) \quad (3.11)$$

İki mobil robot arasındaki bilinen ve gerçek mesafe arasındaki her bir ekseni kapsayan toplam hatalar ϵ_T olarak ifade edilmiştir.

$$\epsilon_T = \sqrt{\epsilon_{X_i}(t)^2 + \epsilon_{Y_i}(t)^2 + \epsilon_{Z_i}(t)^2} \quad (3.12)$$

Sonuç olarak bu kısımda bir mobil robotun üç boyutlu uzayda gerçek ve bilinen pozisyonu modellenmiştir. Ayrıca bu iki pozisyonun çıkarımı ile aksenel boyutta kalibre edilmesi gereken hata ifadesi modellenmiştir.

3.3 Sözde Ters Matris

Bir matrisin tersinin alınabilmesi için iki önemli şart bulunmaktadır;

- Kare matris olma şartı
- Determinantının sıfır olmaması şartı

Determinantın sıfır çıkması koşulunda matrisin tersi alınmak istendiğinde sonuç sonsuz çıkmaktadır. Bu durumda sözde ters matrisi (pseudo inverse matrix) yöntemi yardımıyla matrisin tersi alınır. A^+ ifadesi ile gösterilir.

Bir A matrisinin sütunları lineer olarak bağımsız ise $A^T.A$ sonucu oluşan matris tersinir matristir. Bu bilgi doğrultusunda aşağıdaki verilen sözde ters matrisi formülü (3.13) oluşturulmuştur.

$$A^+ = (A^T.A)^{-1}.A^T \quad (3.13)$$

Denklem (3.14)' te verilen A^+,A matrisinin soldan tersidir.

$$A^+.A = E \quad (3.14)$$

Eğer A matrisinin satırları lineer olarak bağımsız ise sözde ters matrisi formülü (3.15)'deki gibi oluşturulmuştur.

$$A^+ = A^T.(A.A^T)^{-1} \quad (3.15)$$

Denklem (3.16)' da verilen A^+,A matrisinin sağdan tersidir.

$$A.A^+ = E \quad (3.16)$$

Eğer A matrisinin hem satırları hem de sütunları lineer olarak bağımsız ise A matrisi tersinirdir ve sözde ters matrisi matrisin tersine eşittir [29], [30]. Sözde ters matris konusu bu bölümde örnek bir soru çözümü ile açıklanmıştır.

Örnek: 3B uzayda Tablo 3.1'de verilen üç noktanın konumları ve bir hedefe olan uzaklıkları aşağıda verilmiştir. Hedef noktanın $D(11, 12, 8)$ olduğu bilinmektedir.

Tablo 3.1 Verilen üç robotun konumu ve hedefe olan mesafeleri

Nokta	Hedefe Olan Mesafe
A(2,16,3)	$\sqrt{122}$
B(18,8,12)	$\sqrt{101}$
C(15,1,10)	$\sqrt{141}$

Verilen bu üç noktanın küre denklemlerini yazalım;

$$|P_A P_D| = \sqrt{(x-2)^2 + (y-16)^2 + (z-3)^2} = \sqrt{122} \quad (3.17)$$

$$|P_B P_D| = \sqrt{(x-18)^2 + (y-8)^2 + (z-12)^2} = \sqrt{101} \quad (3.18)$$

$$|P_C P_D| = \sqrt{(x-15)^2 + (y-1)^2 + (z-10)^2} = \sqrt{141} \quad (3.19)$$

Denklem (3.17)'yi düzenleyelim;

$$x^2 - 4x + y^2 - 32y + z^2 - 6z = -147 \quad (3.20)$$

Denklem (3.18)'i düzenleyelim;

$$x^2 - 36x + y^2 - 36y + z^2 - 24z = -691 \quad (3.21)$$

Denklem (3.19)'u düzenleyelim;

$$x^2 - 30x + y^2 - 2y + z^2 - 20z = -185 \quad (3.22)$$

Matris oluşturabilmek için x^2 , y^2 , z^2 'li ifadelerin denklemlerden elimine edilmesi gerekmektedir. (3.20), (3.21), (3.22) no'lu denklemler alt alta yazılır;

$$x^2 - 4x + y^2 - 32y + z^2 - 6z = -147$$

$$x^2 - 36x + y^2 - 36y + z^2 - 24z = -691$$

$$x^2 - 30x + y^2 - 2y + z^2 - 20z = -185$$

(3.21) no'lu denklem -1 katsayısı ile çarpılıp (3.20) no'lu denklemle toplanır. Yapılan işlem sonucunda elde edilen denklem (3.23)'te verilmiştir.

$$32x + 4y + 18z = 544 \quad (3.23)$$

(3.19) no'lu denklem -1 katsayısı ile çarpılıp (3.20) no'lu denklemle toplanır. Yapılan

işlem sonucunda elde edilen denklem (3.24)'te verilmiştir.

$$26x - 30y + 14z = 38 \quad (3.24)$$

(3.21) no'lu denklem -1 katsayısı ile çarpılıp (3.22) no'lu denklemle toplanır. Yapılan işlem sonucunda elde edilen denklem (3.25)'te verilmiştir.

$$6x + 34y + 4z = 506 \quad (3.25)$$

Elde ettiğimiz denklemleri alt alta yazıldıktan sonra matris oluşturulur;

$$32x + 4y + 18z = 544$$

$$26x - 30y + 14z = 38$$

$$6x + 34y + 4z = 506$$

$$\begin{vmatrix} 32 & 4 & 18 \\ 26 & -30 & 14 \\ 6 & 34 & 4 \end{vmatrix} \times \begin{vmatrix} x \\ y \\ z \end{vmatrix} = \begin{vmatrix} 544 \\ 38 \\ 506 \end{vmatrix} \quad (3.26)$$

veya

$$J \times K = L \quad (3.27)$$

Örnek çalışmada hedef noktasının (11, 12, 8) olduğu bilgisi verilmişti. Doğrulanmak için K matrisi MATLAB yardımıyla bulunur.

MATLAB' de yapılan işlemler sonucu J matrisinin determinantı 0 olduğundan tersinin sonsuz çıkmaktığı görülmektedir. Bu işleme ait MATLAB kodları Ek-A'da Şekil A.2'de verilmiştir.

Bu durumda MATLAB' de sözde ters matris komutu kullanılarak (3.28)'de verildiği üzere J matrisinin tersinin alınması sağlanmıştır. Bu işleme ait MATLAB kodları Ek-A'da Şekil A.3'te verilmiştir.

$$K = pinv(J) \times L \quad (3.28)$$

İşlem sonucu bulunması gereken değerler (11, 12, 8) olarak başlangıçta verilmişti.

Buradan hareketle hata payları değerlendirilir;

$$\Delta x = 0,6875 \rightarrow \text{sapma deęeri } (\delta) = \%6,25$$

$$\Delta y = 0,0231 \rightarrow \text{sapma deęeri } (\delta) = \%0,2$$

$$\Delta z = 1,1273 \rightarrow \text{sapma deęeri } (\delta) = \%15,34$$

Matematiksel olarak yapılan işlemler sonucunda hedef noktanın eksenlere göre farklı sapmalarla sonuçlandığı gözlenmiştir.

Bu örnekte referans üç nokta kullanılarak hedef noktanın üç boyutlu uzaydaki konumu belirlenmiştir. Referans noktaların sayısı artırılarak konum tayinindeki sapma değerleri azaltılabilir.



4 Sonuçlar ve Öneriler

Önerilen çok yöneticili mobil robot hareketi ile robot sürüsünün tek merkezden yönetilmesinin yerine yönetici sayısının birden fazla tayin edilmesiyle enerji ve zaman kaybında optimizasyon yapılmıştır. Yapılan literatür incelemeleri neticesinde Örümcek Maymun Optimizasyonu algoritmasının geniş ve dağınık ortamlardaki hedef tespiti için diğer algoritmalara göre daha uygun ve kullanışlı olduğu tespit edilmiştir. Örümcek Maymun Optimizasyonu algoritmasındaki çok yöneticili hiyerarşik yapı mobil robotların sürü hareketine uyarlanarak daha etkili bir arama yöntemi elde edilmiştir. Üç boyutlu uzayda konum tespitinde matrislerden faydalanılırken determinantın sıfır çıkması durumunda işleme devam edilebilmesi için sözde ters matrisinden faydalanılmıştır. Sözde ters matrisinden yararlanılarak olası en küçük hatalar ile konum hesabı yapılabilmektedir. Üretilen matematiksel modeller ile robotların gerçek konumu ve bilinen konum arasındaki hatalar tanımlanarak GPS verisi ve matemaitsel modeller üzerinden elde edilen sonuçlar kıyaslanarak hata miktarı kalibre edilmiştir. Mobil robotların hareket planlaması ve her bir robotun erişim noktası gibi de davranması gerekliliği belirtilmiştir. Robotun sadece konumunu değil aynı zamanda gövde pozisyonunu da bilmesi üç boyutlu uzayda hareket planlamasında önemlidir.

Önerilen çok yöneticili mobil robot sürü hareketinde ortamdaki gürültü dahil edilmemiştir. Gelecekte gürültü faktörü de dahil edilerek Örümcek Maymun Optimizasyonu tabanlı çok yöneticili mobil robot hareketi planlamasının simülasyonu ve prototipi üzerine bir çalışma yapılması planlanmaktadır.

Şekil A.1'de rasgele çakıştırılan üç küreye ait MATLAB kodu verilmiştir.

```
>> [x, y, z]=sphere(10);  
>> mesh(2*x, 2*y, 2*z)  
>> hold on  
>> mesh(2*x, 2*y-1, 2*z-1)  
>> mesh(2*x, 2*y+1, 2*z-1)
```

Şekil A.1 Çakıştırılan üç küreye ait MATLAB kodu

Çalışmada verilen örnek soru üzerinden matrisin tersi alındığında sonsuza gittiği MATLAB yardımıyla Şekil A.2'de verilmiştir.

```
>> J=[32, 4, 18; 26, -30, 14; 6, 34, 4]  
  
J =  
  
    32     4    18  
    26   -30    14  
     6    34     4  
  
>> inv(J)  
Warning: Matrix is singular to working precision.  
  
ans =  
  
    Inf    Inf    Inf  
    Inf    Inf    Inf  
    Inf    Inf    Inf  
  
fx >> |
```

Şekil A.2 J matrisi tersinin MATLAB yardımıyla bulunması

J matrisinin tersinin sonsuz çıkması neticesinde sözde ters matrisi kullanılmıştır. Bu işleme ait MATLAB kodları Şekil A.3'te verilmiştir.

```
>> K=pinv(J)*L
```

```
K =
```

```
11.6875
```

```
12.0231
```

```
6.7727
```

Şekil A.3 K matrisinin MATLAB programında sözde ters matris komutu kullanılarak bulunması

- [1] P. Bhattacharjee, P. Rakshit, I. Goswami, A. Konar, and A. K. Nagar, "Multi-robot path-planning using artificial bee colony optimization algorithm," in *2011 Third World Congress on Nature and Biologically Inspired Computing*, IEEE, 2011, pp. 219–224.
- [2] D. Banerjee, C. G. Majumder, S. Roy, A. Chakraborty, A. Konar, and R. Janarthanan, "Multi-robot motion planning amidst dynamic obstacle," in *2011 International Conference on Recent Trends in Information Systems*, IEEE, 2011, pp. 191–196.
- [3] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Computing*, vol. 21, no. 4, pp. 949–964, 2017.
- [4] M. Brand and X.-H. Yu, "Autonomous robot path optimization using firefly algorithm," in *2013 International Conference on Machine Learning and Cybernetics*, IEEE, vol. 3, 2013, pp. 1028–1032.
- [5] T. Guan-Zheng, H. Huan, and A. Sloman, "Ant colony system algorithm for real-time globally optimal path planning of mobile robots," *Acta automatica sinica*, vol. 33, no. 3, pp. 279–285, 2007.
- [6] C. A. Sierakowski and L. dos Santos Coelho, "Path planning optimization for mobile robots based on bacteria colony approach," in *Applied soft computing technologies: The challenge of complexity*, Springer, 2006, pp. 187–198.
- [7] J.-H. Liang and C.-H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm," *Advances in Engineering Software*, vol. 79, pp. 47–56, 2015.
- [8] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.
- [9] M. Bakhshipour, M. J. Ghadi, and F. Namdari, "Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach," *Applied Soft Computing*, vol. 57, pp. 708–726, 2017.
- [10] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
- [11] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE control systems magazine*, vol. 22, no. 3, pp. 52–67, 2002.
- [12] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, 2009, pp. 210–214.

- [13] J. A. M. Garcia F Javier Martinez ve Pérez, “Jumping frogs optimization: A new swarm method for discrete optimization,” *Documentos de Trabajo del DEIOC*, vol. 3, 2008.
- [14] A. T. S. Al-Obaidi, H. S. Abdullah, and Z. O. Ahmed, “Meerkat clan algorithm: A new swarm intelligence algorithm,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 1, pp. 354–360, 2018.
- [15] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, “Spider monkey optimization algorithm for numerical optimization,” *Memetic computing*, vol. 6, no. 1, pp. 31–47, 2014.
- [16] J. Beni Gerardo ve Wang, “Swarm intelligence in cellular robotic systems,” in *Robots and biological systems: towards a new bionics?* Springer, 1993, pp. 703–712.
- [17] J. Eberhart Russell ve Kennedy, “Particle swarm optimization,” in *Proceedings of the IEEE international conference on neural networks*, Citeseer, vol. 4, 1995, pp. 1942–1948.
- [18] L. M. Dorigo Marco ve Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [19] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-tr06, Erciyes university, engineering faculty, computer ..., Tech. Rep., 2005.
- [20] Y. Eren, İ. B. Küçükdemiral, and İ. Üstoğlu, “Introduction to optimization,” in *Optimization in Renewable Energy Systems*, Elsevier, 2017, pp. 27–74.
- [21] F. Leens, “An introduction to 1 2 c and sp1 protocols,” *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8–13, 2009.
- [22] Z. Yang, Z. Zhou, and Y. Liu, “From rss1 to cs1: Indoor localization via channel response,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, p. 25, 2013.
- [23] P. Castro, P. Chiu, T. Kremenek, and R. Muntz, “A probabilistic room location service for wireless networked environments,” in *International conference on ubiquitous computing*, Springer, 2001, pp. 18–34.
- [24] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, “Practical robust localization over large-scale 802.11 wireless networks,” in *Proceedings of the 10th annual international conference on Mobile computing and networking*, ACM, 2004, pp. 70–84.
- [25] M. Soleimanifar, X. Shen, M. Lu, and I. Nikolaidis, “Applying received signal strength based methods for indoor positioning and tracking in construction applications,” *Canadian Journal of Civil Engineering*, vol. 41, no. 8, pp. 703–716, 2014.
- [26] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” *IEEE Communications Surveys & Tutorials*, 2019.
- [27] M. L. Rodrigues, L. F. M. Vieira, and M. F. Campos, “Fingerprinting-based radio localization in indoor environments using multiple wireless technologies,” in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, 2011, pp. 1203–1207.

- [28] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.
- [29] L. Jódar, A. Law, A. Rezazadeh, J. Weston, and G. Wu, “Computations for the moore penrose and other generalized inverses,” *Congressus Numerantium*, pp. 57–57, 1991.
- [30] C. R. Rao, S. K. Mitra, *et al.*, “Generalized inverse of a matrix and its applications,” in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*, The Regents of the University of California, 1972.



Tezden Üretilmiş Yayınlar

İletişim Bilgileri: zulal.tosunoglu@arcelik.com

Konferans Bildirisi

1. Üç Boyutlu Uzayda Çok Yöneticili Mobil Robotlar ile Sürü Hareketi Planlaması
- 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT 2019)