SETSCI Conference Indexing System,
Volume 3 (2018), 1034-1039

ISAS 2018-Winter
2nd International Symposium on Innovative Approaches in Scientific Studies
November 30 – December 2, 2018, Samsun, Turkey

YAPAY ARI KOLONİ ALGORİTMALARININ SÜRÜ ROBOT HEDEF BULMA PROBLEMİNE UYGULANMASI

M. Akif FINDIKLI[1*+], A. Fatih KOCAMAZ[2]

[1]Bilgisayar Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, Türkiye
[2]Bilgisayar Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, Türkiye
[*]Sorumlu Yazar: m.akif.findikli@inonu.edu.tr
[+]Konuşmacı: m.akif.findikli@inonu.edu.tr
Sunum/Makale Tipi: Sözlü / Tam Metin

Özet— Bu çalışmada Yapay Arı Koloni optimizasyon algoritmalarının üç varyantı sürü robotların hedef bulma probleminde kullanılmıştır. Yapay arı kolonisi optimizasyon algoritmalarının Unity 3D ortamında simülasyonunu gerçekleştiren yazılım geliştirilmiştir. Sürü robotların hedef bulma problemine daha iyi çözümler üretebilmesi için algoritmalara yeni modifikasyonlar önerilmiştir. Ayrıca çalışmada sabit bir hedefin bulunması problemi yanında pozisyonu zamanla değişen hedef bulma problemi de incelenmiştir. Geliştirilen sürü robot simülasyon uygulamasının, birçok farklı algoritmanın üzerinde test edilerek sonuçlarının karşılaştırılabileceği bir platform olması hedeflenmiştir.

Anahtar Kelimeler— Sürü robot, sürü optimizasyon algoritması, yapay arı kolonisi algoritması, hedef bulma, dinamik hedef.

Abstract— In this study, we utilized three version of artificial bee colony optimization algorithms for target locating problem in swarm robotics. A software, which implements simulations of artificial bee colony optimization algorithms in Unity 3D environments is developed. We suggested some modifications to these algorithms so that they can offer better solutions to problem of target locating in swarm robotics. Besides, we also addressed locating problem of dynamic targets apart from static ones. We developed swarm robotics simulation software to create a platform that can be used for testing different algorithms, and compare their results.

Index Terms— Swarm robotics, swarm optimization algorithms, artificial bee colony algorithm, source locating, dynamic source.

I. GİRİŞ

Günümüzde, robotlar ve robotik uygulamalar yaygın bir şekilde kullanılmaya başlanmıştır. Robotikleşmeye doğru artan bu eğilim, daha verimli uygulama ve hizmet ihtiyacından doğmaktadır. Robotik sistemlerinin daha az hata ile sürekli çalışabilme kabiliyetleri bir çok uygulamada önemli avantajlar sunmaktadır. Çoğu uygulamada kullanılan robotların otonom hale gelmesi hem kaynaktan hem de zamandan tasarruf sağladığı görülmüştür [1-2]. Sürü zekâsı konusunda son zamanda çalışmalar yapılmış ve saha problemleri çözümünde avantaj sağlayabildiği görülmüştür [3]. Otonom robotların kolektif bir şekilde çalışarak problemleri çözmeleri; birbirleri ile bilgi alışverişinde bulunabilmeleri ve ortak karar süreçlerinden faydalanabilmeleri nedeni ile tekli robot uygulamalarına göre çok daha verimli ve etkin olmaktadır [4]

Sürü zekâsının başarılı uygulamaları sürü tabanlı optimizasyon yöntemlerinde görülmüştür [5-7]. Bu yöntemler metasezgisel yöntemler olarak adlandırılmış, pratik uygulamalarda sürü tabanlı metasezgisel yöntemlerin avantajları gösterilmiştir. Analitik yöntemler ile çözülemeyecek derecede karmaşık optimizasyon problemleri sürü zekâsının avantajlarını kullanan metasezgisel optimizasyon yöntemleri ile çözülebilmiştir [8-9].

Çalışmada, sürü tabanlı metasezgisel yöntemlerin ortaya koyduğu sürü zekâsı çözümlerinin robotikte uygulanmasına yönelik araştırma yapılmıştır. Bu amaçla Yapay Arı Kolonisi algoritması (Artifical Bee Colony Algorithm - ABC) ve varyantlarının sürü robot hareketlerine uygulanabilmesi ve adapte edilebilmesi incelenmiştir. ABC algoritması Karaboğa'nın 2005 yılında, bal arılarının besin bulma davranışlarından esinlenerek geliştirilmiştir [5]. ABC algoritması verilen nümerik problemlerin global veya lokal minimumuna ulaşabilmesi için, gözcü, kâşif ve işçi arı davranışlarını kullanabilen bir sürü zekâsını modellemektedir. Bu model, kaynak bulma problemi gibi sürü robotik (swarm robotics) alanındaki bir çok problemin çözümüne uygundur.

Bu çalışmada, ABC algortiması ve varyantları incelenmiş ve bu algoritmalar çoklu-robot hedef bulma problemine uygulanmıştır. Gerçekçi robot hareketlerine uyum için algoritmalarda adaptasyonlar yapılmıştır. Örneğin, ABC algoritmasında nümerik çözümü kolaylaştırmak için tasarlanan bireylerin yeni olası çözüme ışınlanma hareketleri gerçek robot hareketlerinde mümkün olmadığı için ışınlanma hareketi robotların doğrusal hareketi ile değiştirilerek modellenmiştir. Algoritmada arı modellerinin hareket serbestliği gerçek robot hareket serbestliğine uyumu için arttırılmıştır.

II. METOT

Çalışmalar, Unity 3D ortamında geliştirilen çoklu robot simülasyon ortamında gerçekleştirilmiştir. Bu ortamda sabit ve hareketli hedef için hedef bulma testleri ABC algoritması ve varyantları için ayrı ayrı gerçekleştirilmiş ve sonuçlar mukayese edilerek sunulmuştur.

A. ABC Algoritması

Yapay Arı Kolonisi algoritması literatürde ilk defa KARABOĞA tarafından öne sürülmüştür [5]. ABC algoritması arı sürülerinin besin bulmak için kullandığı davranış biçimini, verilen problemin optimum çözümünü bulmaya benzeterek modellemektedir.

Arılar bölgedeki besin kaynaklarını bulmak için ilk olarak rasgele bir şekilde alana dağılırlar. Daha sonra bulunan kaynaklarda yiyecek miktarının azalması ile ya da diğer arılardan alınan daha kaliteli besin bölgesi bilgileri ile başka besin kaynaklarına yönelirler. Algoritmanın adımları Şekil 1'de görülebilir.

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

UNTIL *(gereksinimler tamamlanana kadar)*

Şekil 1. ABC Algoritmasının Adımları

En başta popülasyondaki bütün arılar, alanda rasgele bölgelere Eşitlik 1'deki gibi dağılırlar.

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

Burada x_i arı bireylerini, j ise problemin parametrelerini belirtmektedir. Her bir bireye parametrelerinin minimum ve maksimum değerleri arasında rasgele değerler atanarak arıların alana dağılması sağlanır.

Ardından *İşçi Arı Fazı (İAF)* başlar, bu fazda tüm arılar bulundukları konumun etrafında daha kaliteli bir çözümü Eşitlik 2'deki denklem ile bulmaya çalışırlar, bulabilirlerse bu yeni çözümü hafızalarına alırlar. Bulamazlarsa yeni sonuç üretememe sayaçlarını bir arttırırlar. Bu denklemde V_{ij}, i arısının j. parametresi için yeni bir çözümü ifade eder. Yeni çözüm, hali hazırda olan parametre değerinin rasgele bir arının parametresine (x_{kj}) bağlı olarak oluşturduğu yeni değerdir.

$$V_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

Daha sonra bulunan çözümün uygunluğu Eşitlik 3 ile hesaplanır. Bu uygunluk değeri mevcut olan uygunluk değerinden daha iyi ise, bu yeni çözüm hafızaya alınır. (Açgözlü Seleksiyon)

$$fitness_i = \begin{cases} 1/(1+fi), & fi \geq 0 \\ 1 + abs(fi), & fi < 0 \end{cases} \quad (3)$$

Ardından *Gözcü Arı Fazı (GAF)* başlar. Bu fazda arıların buldukları çözümlerin uygunluklarına göre arılara yeni olası çözümlerin yerleri bildirilir. Bunun için her çözümün olasılık değeri Eşitlik 4'teki gibi hesaplanır.

$$p_i = \frac{fitness_i}{\sum_{j=i}^{SN} fitness_i} \quad (4)$$

Daha sonra gözcü arılar çözümlerin olasılık değerlerine göre dağılırlar.

Arılar her zaman daha iyi bir yeni çözüm bulamayabilirler. Çevrim tamamlanmadan önce her arının yeni çözüm üretememe sayacı kontrol edilir ve belirlenen değeri aşmış ise bu arı kâşif arı olarak görevlendirilir. Böylece *Kâşif Arı Fazı (KAF)* başlar. Kâşif olan arılar alanda tamamen rasgele yeni bir konuma giderek aramalarına bu noktada devam ederler. Bunun sonucunda ilk iterasyon tamamlanır ve tekrar İAF'ye dönülür.

B. qABC Algoritması

Karaboğa ve Görkemli tarafından 2013 yılında öne sürülmüştür [10]. Bu algoritmada gözcü arılar işçi arılardan farklı bir şekilde besin araması işlemi yapmaktadır. Bir gözcü arı işleyeceği besin kaynağını önceden tanımlı bir menzilin içindeki komşu arılardan aldığı bilgiye göre seçmektedir.

$$md_m = \frac{\sum_{i=1}^{SN} d(m,j)}{SN-1} \quad (5)$$

Eşitlik 5'teki j o an kontrol edilen arıyı, $d(m,j)$ j. arı ile gözcü arı arasındaki Öklid uzaklığını, md_m ise tüm uzaklıkların ortalamasını ifade etmektedir.

Eğer o an kontrol edilen arı, gözcü arıya ortalama uzaklığın önceden belirlenmiş bir menzil katsayısı (r) ile çarpımından daha yakın ise, o an kontrol edilen arı gözcü arının komşusudur denilebilir. Bu işlem Şekil 2'de gösterilen pseudo code ile gerçekleştirilmektedir.

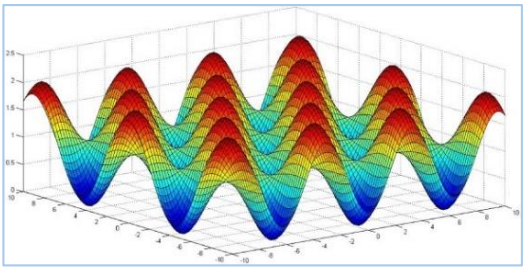
if $d(m,j)r \times md_m$ ***then*** x_j ***is a neighbor of*** x_m, ***else not***

Şekil 2. qABC Algoritmasının Komşu Belirleme İşlemi [10]

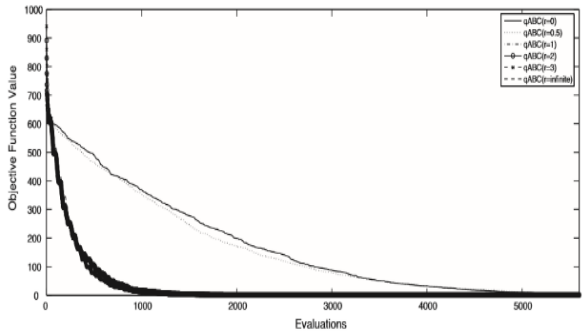
Komşular belirlendikten sonra da gözcü arılar yeni çözümlerini komşularındaki uygunluk değeri en iyi olan arıdan aldıkları bilgiye göre üretirler.

Karaboğa ve Görkemli ABC algoritmasına önerdikleri bu modifikasyonu birçok *benchmark* fonksiyonlarında ABC algoritması ile mukayese edip modifikasyonun daha iyi performans sergilediğini göstermişlerdir [10]. *Benchmark* fonksiyonlarından *Griewank* fonksiyonu (Eşitlik 6 – Şekil 3) için elde ettikleri sonuçlarda qABC algoritmasının ABC algoritmasına göre çok daha az adımda optimum noktaya yaklaştığı Şekil 4'teki grafikte görülmektedir.

$$f(x) = \frac{1}{4000}\left(\sum_{i=1}^{D} x_i^2\right) - \left(\prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1 \quad (6)$$



Şekil 3 Griewank Fonksiyonunun Grafiği [10]



Şekil 4. Griewank Fonksiyonu İçin ABC ve qABC Algoritmalarının Performanslarının Karşılaştırılması [10]

C. mABC Algoritması

Babayiğit ve Özdemir tarafından 2014 yılında öne sürülmüştür [11]. ABC algoritmasından iki farklı özelliği bulunmaktadır.

ABC algoritmasında gözcü arılar kaynakları seçme işlemini kaynakların uygunluk değeri ile doğru orantılı bir şekilde gerçekleştirmektedir. Bu durum en iyi sonucun her zaman ön planda olmasına, kâşif arıların ise öneminin düşmesine sebep olmuştur. Düşük kalitedeki kaynaklara da şans verilebilmesi için uygunluk fonksiyonu Eşitlik 7'deki gibi güncellenmiştir.

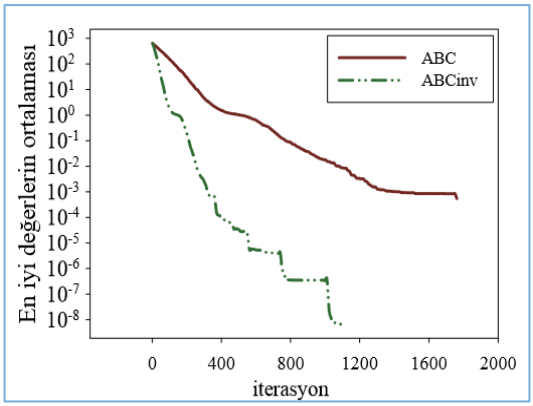
$$p_i = (1 + fit_i)\exp(-fit_i) \tag{7}$$

Ayrıca ABC algoritmasında gözcü arılar yeni komşu bulurken tek parametre değiştirirken, mABC algoritmasında ise yeni çözümde ilk paremetreye ek olarak ikinci bir parametre daha Eşitlik 8'deki denkleme göre değiştirilmektedir. İlk parametreden farklı olarak ikinci parametre global minimum değere sahip bireyin parametresinin komşuluğunda aranmaktadır. Böylece bulunan global minimum çözümü kaybetmeden kâşif arılardan da faydalanılması sağlanmıştır.

$$V_{i,n} = x_{best,n} + \varphi_1(x_{k,n} - x_{m,n}) \tag{8}$$

Eşitlik 8'de x_{best} global minimumu, x_m ise x_k ve gözcünün kendisinden farklı yeni bir arıdır.

Şekil 5'teki grafikte Babayiğit ve Özdemir'in önerdikleri algoritmayı ABC algoritması ile *Griewank* fonksiyonu üzerindeki mukayeseleri görülmektedir. mABC algoritması da ABC'ye göre oldukça başarılı sonuçlar elde etmiştir [11].



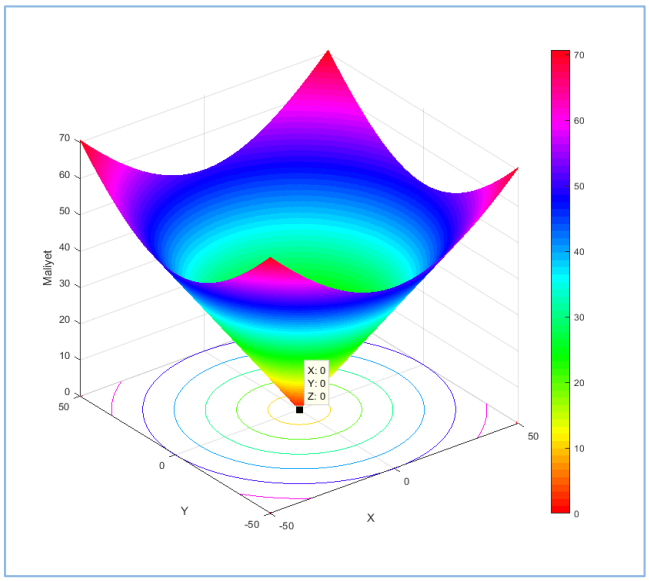
Şekil 5. Griewank Fonksiyonu İçin ABC ve mABC Algoritmalarının Performanslarının Karşılaştırılması [11]

III. Robotik Uygulamaları İçin Yapılan Adaptasyonlar

ABC algoritmasında besinin kalitesi optimize edilmeye çalışılan fonksiyonların maliyetleri ile hesaplanmaktadır. Sürü robotlar için çözümün kalitesi hedefe yakınlıktır. Bu bilgi sensörlerden alınan sinyal şiddeti, koku şiddeti gibi özellikler ile ölçülebilir. Simülasyon ortamında maliyet fonksiyonunun kolayca benzetiminin yapılabilmesi için maliyet, robot ile bulunması gereken hedef arasındaki Eşitlik 9'daki *Öklid Uzaklığı* ile ifade edilmiştir.

$$f(x) = \sqrt{x^2 + y^2} \tag{9}$$

Hedefin başlangıç pozisyonu $x = 0,\ y = 0$ olarak alınmıştır. Şekil 6'daki grafikte maliyet fonksiyonun sahaya dağılımı görülmektedir.



Şekil 6. (a) Maliyet Fonksiyonunun Değişimi
(b) Maliyet Fonksiyonunun Değişiminin Renk Grafiği

Bu çalışmada yapılan temel modifikasyonlar şu şekilde özetlenebilir;

- Hareket iki boyutlu ortamda x ve y parametrelerinin değişimi ile gerçekleştiği için, ABC algoritmasının tek parametre değiştirme özelliği iki parametreye çıkarılmıştır. Böylece robotlara diyagonal hareket etme özelliği kazandırılmıştır.

- ABC algoritmasında bireylerin yeni buldukları çözüme ışınlanma olayı fiziksel ortamda mümkün olamayacağı için ışınlanma olayı iki nokta arasında, robotun önceden belirlenmiş sabit bir hızı ile lineer hareketi olarak gerçekleştirilmiştir.

- Fiziksel ortamda hedefin sabit olma zorunluluğu olmadığı için, bu robotik uygulamasında hedef dinamik olarak hareket ettirilebilir hale getirilmiştir.

IV. SİMÜLASYON

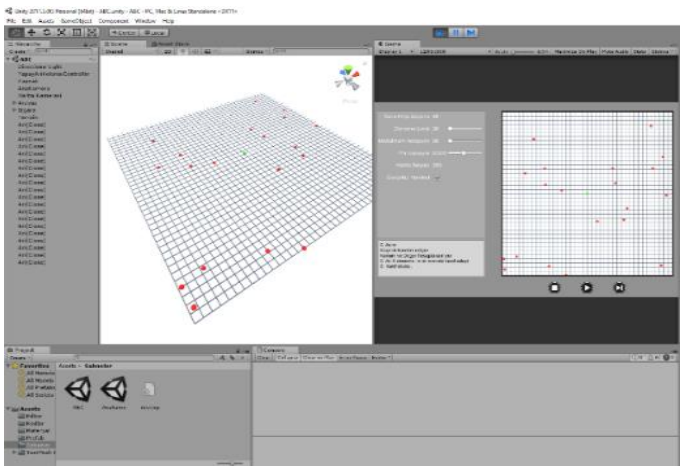
A. Simülasyon Ortamı

Unity 3D bir oyun geliştirme motoru olarak kullanıcılara önceden tanımlı yerçekimi, hareket, kuvvet gibi fiziksel kurallar kütüphanesi sunmaktadır [12].

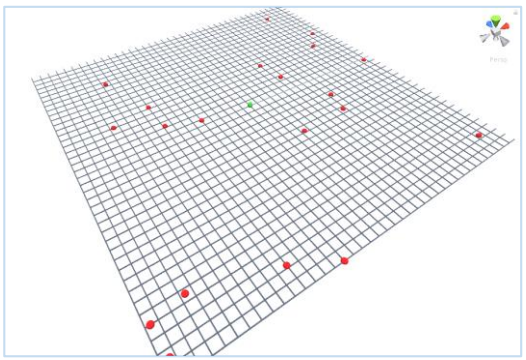
Unity'nin arayüzünde sahneyi hem kullanıcı hem de programcı gözünden görmek mümkündür. Şekil 7'de görülen program arayüzünün sol tarafında sahnede bulunan objeleri, sol alt tarafında sahnede kullanılan materyaller ve kodları, sağ alt tarafında ise konsol ekranı görülebilir. Bu alanlar kişiselleştirilebilir.

Şekil 8'de görüldüğü üzere sürü robotların gezeceği alanda robot hareketlerinin daha net görülebilmesi için alan ızgaralanmıştır. Robotlar kırmızı küreler, bulunması gereken hedef ise yeşil bir küre ile temsil edilmiştir.

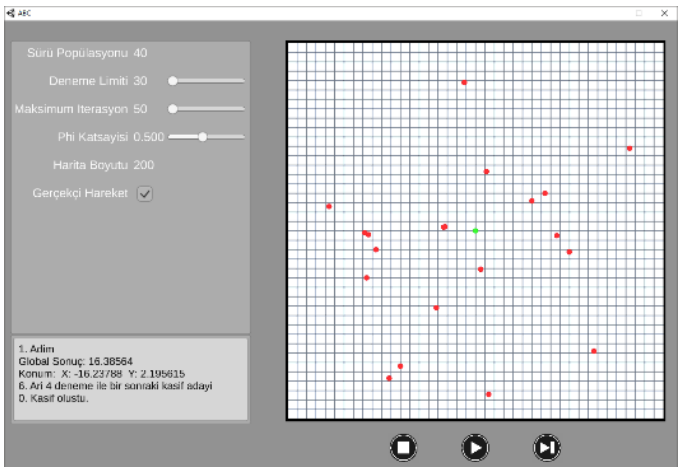
Programın arayüzünde kullanıcılar simülasyonun başlangıç değerlerini Şekil 9'da görülen arayüzün sol tarafındaki panelden ayarlayabilmektedirler. Sahnenin altında bulunan başlat, durdur ve adım adım ilerle seçenekleri ile de simülasyonlarını kontrol edebilmektedirler. Simülasyonun anlık sonuçları ise, sol alt taraftaki konsol panelinden izlenebilmektedir. Simülasyon tamamlandığında ise kullanıcının, verilerini analiz edebilmesi için her iterasyondaki global minimuma sahip robotun maliyet, uygunluk, pozisyon gibi özellikleri bir text dosyasında sunulmaktadır. Kullanıcı, simülasyonu hızlandırmak için robotların gerçekçi hareketini iptal edebilme ya da robotları hızlandırma yeteneğine sahiptir. Ayrıca kullanıcı klavyesindeki yön tuşlarını kullanarak hedefin pozisyonunu anlık olarak değiştirebilmektedir.



Şekil 7. Unity 3D Arayüzü



Şekil 8. Simülasyon Ortamı



Şekil 9. Simülasyon Uygulamasının Arayüzü

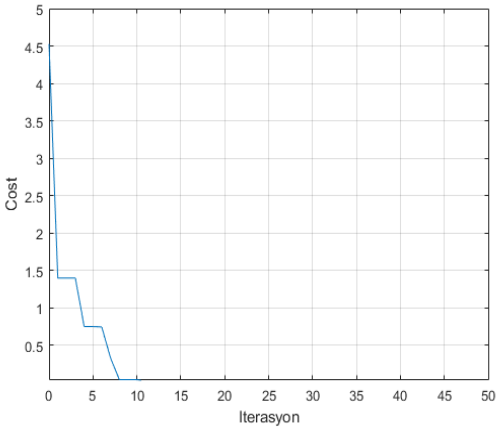
B. Simülasyon Sonuçları

Sürü robotlar ile kullanılabilmesi için adapte ettiğimiz ABC algoritmasının simülasyonu için aşağıdaki parametre değerleri kullanılmıştır.

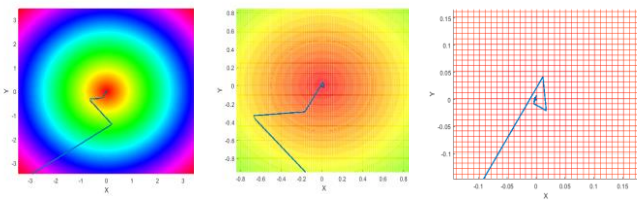
Başlangıç değerleri;

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

Şekil 10'daki grafikte her iterasyondaki popülasyonda bulunan global minimuma sahip robotun maliyet değerinin iterasyona bağlı değişimi görülmektedir. Yapılan simülasyon sonucunda robotların hedefe oldukça hızlı yaklaşabildiği gözlemlenmiştir.



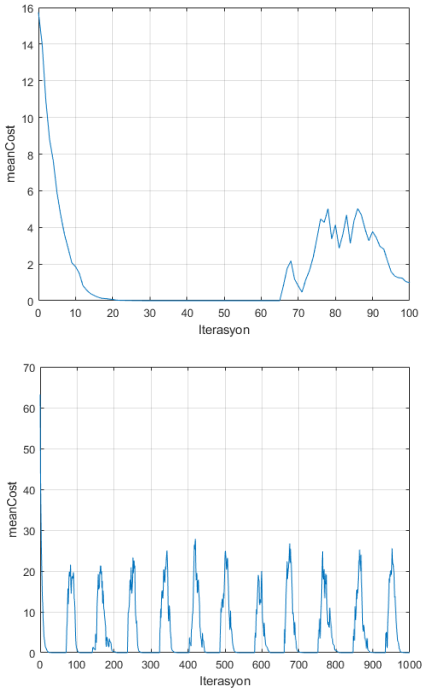
Şekil 10. Maliyetin İterasyona Bağlı Değişim Grafiği



(a) (b) (c)

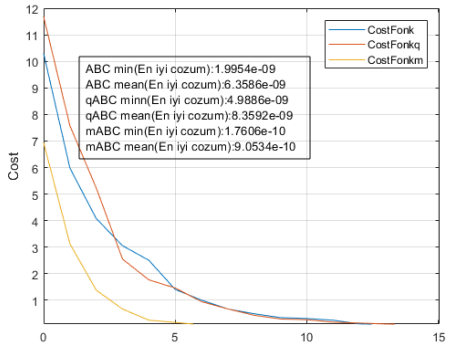
Şekil 11. (a) Robotların Hedefe Yaklaşmaları
(b) Yaklaşımın 4 Kez Büyütülmüş Hali
(c) Yaklaşımın 20 Kez Büyütülmüş Hali

Her iterasyondaki global minimuma sahip robotların hedefe $(x = 0, y = 0)$ yaklaşması x ve y parametrelerinin değişimi ile Şekil 11'de görselleştirilmiştir. Yaklaşma hareketinin anlaşılabilmesi için grafik kademeli olarak büyütülmüştür. Her bir nokta, bir sonraki iterasyondaki global minimum çözüme sahip robotun pozisyonunu göstermektedir.



Şekil 12. (a) Ortalama Maliyetin 100 İterasyondaki Değişimi
(b) Ortalama Maliyetin 1000 İterasyondaki Değişimi

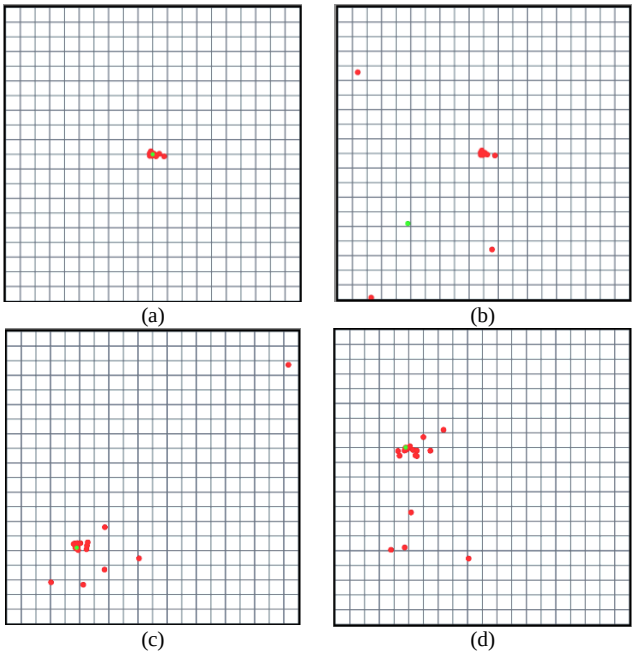
Şekil 12 (a)'daki grafikte her iterasyonda sürüdeki tüm robotların maliyetlerinin ortalaması görülmektedir. 65. iterasyonda ortalama maliyet yükselmeye başlamaktadır. Bu bölgede, gözcü robotlar daha iyi sonuç bulamadıkları için kâşif robota dönüşüp alanda rasgele bir yere gitmeye başlamıştır. Bu da ortalama maliyeti arttırmıştır. Bu fenomen iterasyon sayısı Şekil 12 (b)'de görülen grafikteki gibi arttırıldığında daha rahat gözlemlenebilmektedir. Bu fenomenin tek ve sabit hedef için katkısı olmasa da, dinamik hedef ve çoklu hedeflerde kritik bir önemi vardır. Eğer robotlar hedefe çok yaklaştıkları zaman, hedef uzak bir noktaya hareket ederse, robotlar hedefin yeni konumuna yaklaşmak yerine hedefe en yakın robotun etrafında gezineceklerdir. Kâşif robotun bu olumsuz etkiyi ortadan kaldırmada önemli bir rol oynayabildiği gözlemlenmiştir. Çoklu hedef senaryosunda tek bir hedefe odaklanan robotları diğer hedeflere yönlendirme işi de kâşif robotlar sayesinde sağlanabilmektedir.



Şekil 13 ABC, qABC ve mABC Algoritmalarının Performanslarının Karşılaştırılması

ABC algoritması simülasyonda çalıştırıldıktan sonra ABC algoritması için önerilen diğer algoritmalar da bu algoritmaya uygulanan adaptasyonlar ile simüle edilmiş, Şekil 13'teki grafikte iterasyona bağlı maliyetleri paylaşılmıştır.

ABC algoritmasının simülasyonu, pozisyonu simülasyon süresince değişebilen bir hedef için tekrar çalıştırılmıştır. Hedef en başta Şekil 14 (a)'da görülebileceği gibi hareket ettirilmemiş ve robotların hedefe toplanması sağlanmıştır. Daha sonra hedefin Şekil 14 (b)'deki gibi yeri değiştirilmiş, ve robotların kâşif robotlar oluşmaya başlayana kadar gruplarını terk etmedikleri gözlemlenmiştir. İlk kâşif robotlardan gelen yeni bilgilerle beraber hedefe doğru Şekil 14 (c)'deki gibi hareket etmeye başlamışlardır. Hedef her pozisyon değiştirdiğinde bu işlemin tekrar ettiği Şekil 14 (d)'de gözlemlenebilir.



Şekil 14 (a) Robotların hedefi bulması
(b) Hedef hareket ettirildi ve kâşifler oluştu
(c) Robotların hedefi tekrar bulması
(d) Hareket eden kaynağı takip hareketi

V. SONUÇLAR

Bu çalışmanın sonunda sürü robotların hedef bulma davranışını metasezgisel yöntemlerden olan ABC algoritması ve varyantları ile benzetiminin yapılabildiği bir simülasyon uygulaması geliştirilmiştir. Bu uygulamada, sabit hedef bulma davranışı benzetimi yapılabildiği gibi pozisyonu zamana bağlı değişebilen hedeflerin de bulunma davranışı incelenebilmiş ve başarılı sonuçlar elde edilmiştir.

İncelenen ABC algoritması ve bu algoritma için daha önceden önerilen qABC ve mABC adaptasyonlarına robotik alanında kullanılabilmesi için bazı adaptasyonlar önerilmiş, bu öneriler simülasyon uygulamasına başarıyla entegre edilmiştir. Daha sonra bu algoritmalar aynı başlangıç değerleri ile simülasyona tabi tutulup performansları mukayese edilmiştir.

Çalışmanın devamında simülasyon uygulaması için çoklu hedef bulma davranışları, robot hareketinin daha gerçekçi ve verimli modellenmesi, asenkron iteratif yöntemler gibi yeni özellikler geliştirilebilir.

KAYNAKÇA

[1] O. Parlaktuna, "Otonom Robot Sistemleri için Eğitim Araçları Elektrik-Elektronik Mühendisliği."

[2] A. Howell, E. Way, R. McGrann, and R. Woods, "Autonomous robots as a generic teaching tool," Proc. - Front. Educ. Conf. FIE, pp. 17–21, 2006.

[3] C. G. Cihan ERCAN, "İnsansız Hava Sistemleri Rota Planlaması Dinamik Çözüm Metotlari Ve Literatür Araştirmasi," Selcuk Univ. J. Eng. Sci. Tech., vol. 1, no. 2, pp. 51–72, 2013.

[4] Y.U. Cao, AS. Fukunaga, and A.B. Khang. "Cooperative mobile robotics: Antecedents and directions." Autonomous Robots, 4, 1997.

[5] D. Karaboga, "An Idea Based On Honey Bee Swarm For Numerıcal Optımızatıon,Technıcal Report-Tr06",Erciyes University, Engineering Faculty, Computer Engineering Department 2005

[6] D. E. GOLDBERG, et al. "Genetic algorithms with sharing for multimodal function optimization." In: Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms. Hillsdale, NJ: Lawrence Erlbaum, pp. 41-49, 1987.

[7] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization", 1995 IEEE International Conference on Neural Networks, 4, 1942-1948, 1995.

[8] D. Saini and N. Saini, "A Study of Load Flow Analysis Using Particle Swarm Optimization," vol. 5, no. 1, pp. 125–131, 2015.

[9] F. S. Abu-Mouti and M. E. El-Hawary, "Optimal distributed generation allocation and sizing in distribution systems via artificial bee colony algorithm," IEEE Trans. Power Deliv., vol. 26, no. 4, pp. 2090–2101, 2011.

[10] D. Karaboga and B. Gorkemli, "A quick artificial bee colony (qABC) algorithm and its performance on optimization problems," Appl. Soft Comput. J., vol. 23, pp. 227–238, 2014.

[11] B. Babayiğit, R. Özdemir, B. M. Bölümü, and E. Üniversitesi, "Modifiye Yapay Arı Koloni Algoritması ile Nümerik Fonksiyon Optimizasyonu Modified Artificial Bee Colony Algorithm for Numerical Function Optimization," pp. 618–622, 2012.

[12] https://unity3d.com/ - Erişim Tarihi: 18.10.2018