Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 201-213, Ekim 2019
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 201-213, October 2019
Copyright © 2019 EJOSAT
Research Article

Parçacık Sürü Optimizasyon Algoritması Kullanılarak Optimum Robot Yolu Planlama

Mehmet Beşkirli[1*], Mehmet Fatih Tefek[2]

[1] Şırnak Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Şırnak, Türkiye (ORCID: 0000-0002-4842-3817)
[2] Kırşehir Ahi Evran Üniversitesi, Kaman Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Kırşehir, Türkiye (ORCID: 0000-0003-3390-4201)

(This publication has been presented orally at HORA congress.)

(First received 1 August 2019 and in final form 24 October 2019)

(**DOI:** 10.31590/ejosat.637832)

ATIF/REFERENCE: Beşkirli, M. & Tefek, M. F. (2019). Parçacık Sürü Optimizasyon Algoritması Kullanılarak Optimum Robot Yolu Planlama. *European Journal of Science and Technology*, (Special Issue), 201-213.

Öz

Robot yolu planlama problemi robotik ve otomasyon alanı için önemli problemlerden bir tanesidir. Robotların yüksek çalışma hızı, kontrol sistemlerinden aşırı performans gerektirdiği için robot hareketinin doğruluğu ve yol planlaması önem arz etmektedir. Robot yol planlama işleminde, bir başlangıç noktasından son noktaya kadar robotun var olan engellere takılmadan en kısa bir şekilde geometrik bir yol çizerek varış noktasına ulaşması amaçlanır. Robot yol planlama problemi arama yapılan alan uzayında birçok yol seçeneğinin bulunması ve bu yollar arasında en kısa mesafenin karar verilmeye çalışılması nedeniyle zor problemler sınıfına girmektedir. Klasik robot yolu planlama yöntemleri problem karmaşıklaştıkça çözüm bulmakta zorlanmaktadır. Bundan dolayı son yıllarda robotik alanında yol planlama probleminin optimum çözümü için sezgisel yöntemlerin önemi artmaktadır. Robot yolu planlama problemi için literatürde birçok sezgisel algoritma probleminin farklı uygulamaları için kullanılmıştır. Bu çalışmada başlangıç noktasında yer alan bir robotun varış noktasına gidene kadar karşılaşacağı engellere çarpmadan en kısa yolu kullanacak şekilde bitiş noktasına ulaşması için Parçacık Sürü Optimizasyon (PSO) algoritması kullanılarak yol planlama işleminin simülasyonu yapılmıştır. Başlangıç noktası sabit A(0,0) olan ve üç farklı bitiş noktalarına B(4,6), C(6,8) ve D(8,10) göre PSO algoritması ile en kısa robot yolu hesaplanmıştır. Aynı zamanda her bir farklı varış noktası için çalışmada engellerin konumları da değiştirilerek simülasyon işlemi yapılmıştır. Bu şekilde üç farklı konumda robot yolu planlaması çözülmeye çalışılmıştır. Çalışmada kullanılan engeller daire şeklinde olduğundan başlangıç ve bitiş noktaları arasındaki mesafeyi bulmak için bir nokta ve bir doğruya uzaklığının matematiksel formülü kullanılmış ve bu şekilde dairesel engellerden kaçınılmaya çalışılmıştır. PSO algoritması ile yapılan robot yolu planlama problem çözümü her bir durum için tablolar ve grafikler ile gösterilmiştir. PSO ile yapılan çalışma sonuçlarına göre üç farklı durumda robot yolunun en kısa hesaplamaları bulunmuştur. Bu şekilde PSO algoritması çözümlerinin robot yolu planlaması için uygulanabilir olduğu gösterilmiştir.

Anahtar Kelimeler: Parçacık Sürü Optimizasyonu, Optimum Robot Yolu Planlama, Robot yolu Simülasyonu.

Optimal Robot Path Planning using Particle Swarm Optimization Algorithm

Abstract

The problem of robot path planning is one of the major problems in the field of robotics and automation. Since the high working speed of the robots requires extreme performance from the control systems, accuracy of the robot movement and path planning is important. In the robot path planning process, from a starting point to the end point, the robot is intended to reach the destination by drawing a geometric path as soon as possible without getting stuck on the existing obstacles. The robot path planning problem is classified as difficult due to the fact that there are many path options in the searched space space and the shortest distance between these paths is decided. Classical robot path planning methods have difficulty finding solutions as the problem becomes more complex. Therefore, in

[*] Sorumlu Yazar: Şırnak Üniversitesi, Müendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Şırnak, Türkiye (ORCID: 0000-0002-4842-3817), mehmetbes@sirnak.edu.tr

recent years, the importance of heuristic methods for optimum solution of the path planning problem in the field of robotics has been increasing. In the literature, many heuristic algorithms have been used for different applications of the problem for robot path planning problem. In this study, the path planning process is simulated by using Particle Swarm Optimization (PSO) algorithm in order to reach the end point in order to use the shortest path without hitting the obstacles encountered by a robot at the starting point until it reaches the destination. The shortest robot path was calculated by using PSO algorithm according to three different endpoints B (4,6), C (6,8) and D (8,10), whose starting point is fixed A (0,0). Simulations were also performed for each different destination by changing the position of the obstacles in the study. In this way, robot path planning was tried to be solved in three different positions. Since the obstacles used in the study are circular, the mathematical formula of the distance between a point and a line was used to find the distance between the starting and ending points and thus, the circular obstacles were tried to be avoided. The robot path planning problem solving with PSO algorithm is shown with tables and graphs for each case. According to the results of the study with PSO, the shortest calculations of the robot path were found in three different cases. In this way, it is shown that PSO algorithm solutions are applicable for robot path planning.

Keywords: Particle Swarm Optimization, Optimal Robot Path Planning, Robot Path Simulation

1. Giriş

Günümüzde pek çok insan etkinliği yerini robotlara bırakmaktadır. Endüstriyel uygulamalar için standart robotlardan uzay araştırmaları gibi karmaşık görevler için otonom robotlara kadar her çeşit robot sistemi kullanılmaktadır. Buradan robotların çok yönlülüğü ve esnekliği farklı sektörlere göre kullanım alanlarının görevlerini yerine getirmelerine izin verdiği görülmektedir. Robot birbirine eklemlerle birbirine bağlanmış birkaç sert gövdeden oluşan mekanik bir yapı olarak tanımlanabilir (Siciliano vd., 2010). Robotlar içerinde farklı hareketler için farklı eklemler bulunmaktadır. Bu hareketliliğe uygun bir el becerisi için uygun bir bilek gibi örnek olarak verilebilir. Bu sebeplerden dolayı yol planlama problemi robotikte artan bir öneme sahiptir. Yol planlama işlemi bir başlangıç noktasında varış noktasına kadar önceden belirlenmiş veya belirlenmemiş engellere çarpmadan bir geometrik yol oluşturmaktadır.

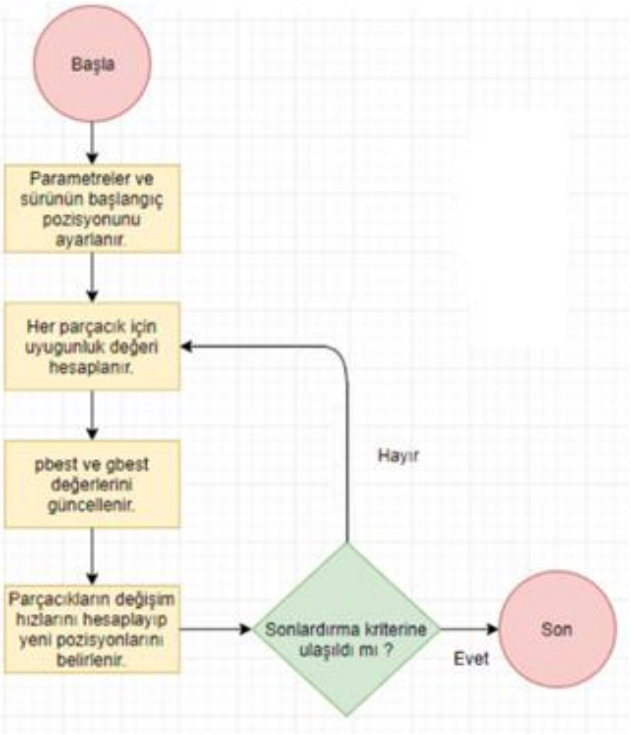
Robotların kontrol sistem girişinin hareket planlaması için genel olarak özel bir modül tarafından üretilen hareket yasası ile verilmektedir. Bu tür hareket planlama modülü, robot hakkında önceden verilen çevre bilgisini kullanarak çevrimiçi veya çevrimdışı çalışabilir. Bu durumda robot hareketini izlemek ve etkinleştirmek için uygun sensörler kullanılmalıdır ki gerçek zamanlı olarak hareketleri ayarlamak için kontrol sistemi devreye girecektir. Ayrıca robotun kontrol edilmesi, engellere çarpmamasını sağlamak için eklemlere uygulanan kuvvet ve torkun iyi ayarlanması gerekmektedir. Tek bir bağlantı kolunun hareketi diğer bağlantıları etkilediğinden bu sistem, problemi çok karmaşık hale gelebilir (Ayari ve Bouamama, 2017).

Robotun en kısa yolu bulma problemi ile ilgi literatürde farklı farklı problemler bulunmaktadır. Doctor vd. (2004) PSO algoritması kullanarak toplu robotik arama işlemi gerçekleştirmişler ve tekli ve çoklu sayıda hedef araması gerçekleştirmişlerdir. Pugh vd. (2006) PSO algoritması üzerinde bazı değişiklikler gerçekleştirerek çoklu robot arama senaryoları üzerine yoğunlaşmışlardır. Hereford (2006) PSO algoritmasında bazı değişikliler yaparak elde ettikleri ve adını dağıtılmış PSO verdikleri algoritma ile bir robotun mevcut pozisyonuna ve mevcut ölçüme dayana bir çalışma gerçekleştirmişlerdir. Chakraborty vd. (2008) tarafından diferansiyel evrim algoritması kullanılarak çoklu robot yolu planlama problemi için bir çalışma gerçekleştirmişlerdir ve sonuçların rekabetçi olduğunu söylemişlerdir. Grandi vd. (2013) tarafından PSO ve Konsensüs algoritmalarına dayalı robot kümelerinin kontrolüne yönelik bir karma teknik sunmuşlardır. Buradaki amaçları bir robot grubunun istenilen bir formasyonda tutmalarıdır. Ayrıca önerilen robot tekniğini sunmak için simülasyonda gerçekleştirmişlerdir. Zhang vd. (2013) tarafından robotların yangın, kara mayınları, savaş alnındaki düşmanlar gibi durumlar için PSO'ya dayanan çok amaçlı bir yol planlama algoritması önermişlerdir. Darvishzadeh ve Bhanu (2014) PSO algoritmasını çoklu robot sistemi için modifeye ederek MPSO'yu robotların kat ettiği toplam yolu optimize etmek için önermişlerdir. Bu çalışmada üç farklı yol ve üç farklı engel kombinasyonu ile oluşan bütün uygulamalar PSO algoritması ile çözülmüş ve elde edilen sonuçlar tablolarda sunulmuştur. Elde edilen simülasyon sonuçları ile yakınsama grafikleri ise şekiller ile ayrı ayrı gösterilmiştir.

2. Materyal ve Metot

2.1. Parçacık Sürü Optimizasyonu

Parçacık sürü optimizasyonu (PSO) 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiş popülasyon temelli sezgisel bir optimizasyon tekniğidir (Kennedy ve Eberhart, 1995). PSO'nun temelinin sosyolojik esinlenme olduğu söylenebilir. Çünkü algoritmanın orijinal fikri, kuşların sürü halinde toplanmasıyla ilişkilendirilmiş sosyolojik davranışlarına dayanmaktadır. Bu algoritma, Kuş, balık ve hayvan sürülerinin bir "bilgi paylaşma" yaklaşımı uygulayarak çevrelerine adapte olabilme, zengin yiyecek kaynağı bulabilme ve avcılardan kaçabilme yeteneklerinden esinlenerek tasarlanmıştır. PSO'da her bir parçacık bir kuşu ifade eder ve her parçacık bir çözüm sunmaktadır. Tüm parçacıkların uygunluk fonksiyonu ile bulanan uygunluk değerleri ulunmaktadır. Parçacıklar, kuşların uçuşlarını yönlendiren hız bilgisine benzer bir bilgiye sahiptir. PSO rastgele üretilmiş belirli sayıda çözümle (parçacıkla) başlatılır ve parçacıklar güncellenerek en uygun çözüm değeri araştırılır. Parçacıkların her biri, parçacığın en iyi kendi çözümü (pbest) ve tüm parçacıkların en iyi çözümü (gbest) kullanılarak parçacıklar güncellenir ve bu değerler hafızada saklanır. Şekil 1'de PSO'nun akış şeması verilmiştir (Tefek ve Uğuz, 2016).



Şekil 1. PSO'nun Akış Şeması

Şekil1'deki akış şemasında da gösterildiği gibi; PSO'da öncelikle çözümü arayacak sürü ve gerekli parametreler belirlenir. Uygunluk fonksiyonu yardımıyla parçacıkların çözüme yakınlığı ölçülür ve bu değerlere göre pbest ve gbest değerleri güncellenir. Daha sonra değişim hızı fonksiyonu ile her parçacığın yapacağı hareket belirlenir ve yeni durumları ayarlanır. Tekrar uygunluk fonksiyonu ile çözüme ne kadar yaklaşıldığı kontrol edilir. Bu döngü istenilen şartlara ulaşılıncaya kadar tekrarlanır.

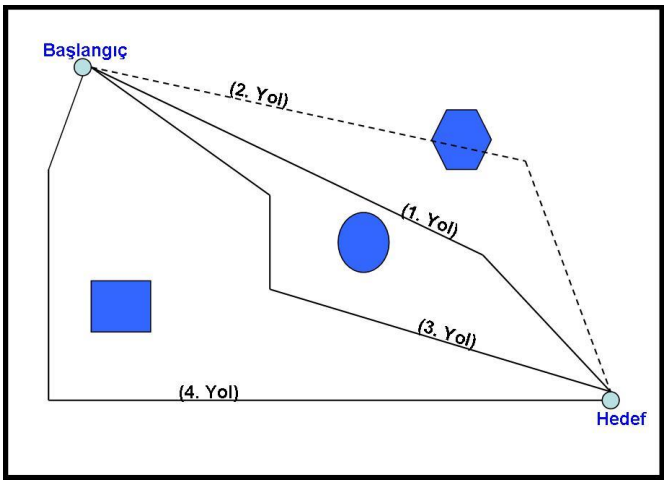
Parçacıkların kendi en iyi çözüme ulaşması ve global en iyi çözüme yönelebilmesi için Denklem 1'deki formül kullanılmaktadır. Böylece bu formül ile parçacığı çözümü en iyi parçacığın ve kendi en iyi durumun yakınlarında aramaya itmektedir (Abdelbar vd., 2005).

$$v_{i+1} = v_i + c_1 * rand_1 * (pbest - x) + c_2 * rand_2 * (gbest - x) \quad (1)$$

PSO için Denklem 1'de verilen parçacıkların değişim hız formülünde, x, parçacık değeri, v, parçacığın değişim hızı, c1, c2, sabit değerler, rand1, rand2, rastgele üretilen değerler [0,1] aralığında, pbest, parçacığın çözüme en çok yaklaştığı durum, gbest, tüm parçacıklar arasında çözüme en çok yaklaşılan durum olarak tanımlanmıştır (Chander vd., 2011).

2.2. Robot Yolu Planlama

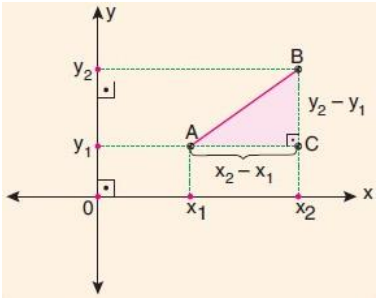
Yol planlama problemi, belirli bir alan içeresinde istenilen bir noktadan hedeflenen başka bir noktaya, alan içeresindeki engellere çarpma olmaksızın optimal uzunluktaki yolu bulmayı hedefleyen bir problem türü olarak tanımlanmaktadır. Yol planlama problemleri işlemsel olarak çok karmaşık problemler olarak ifade edilebilirler. Bunun nedeni ise yol planlama işlemi yapılacak alan içerisinde birçok alternatif yol seçeneğinin bulunması ve bu yollar arasında engellere çarpmaksızın optimum yolun bulunmaya çalışılmasının amaçlanmasıdır. Yol planlamasında ilk hedef planlanan yolun alan içeresinde bulunan engeller ile temas probleminin halledilmesidir. Şekil 2'de buna uygun bir örnek gösterilmiştir (Suvaydan, 2011).



Şekil 2. Yol planlamasına Örnek Bir Çizim

Yukarıdaki şekle göre; 1. Yol engellerle teması olmadığı ve optimum yol uzunluğu açısından da uygun bir yol olduğu görülmektedir. 2. Yol engellerden birisi ile direk olarak temas halinde olduğundan bu yolun uygun olmadığı görülmektedir. 3. Yol ise engellere çarpma olmaksızın 1. Yola göre engellerden uzaklık mesafesine göre daha emniyetli bir yol olduğu görülmektedir. Burada dikkat edilmesi gereken nokta engele çarpma olmamasına rağmen yol planlama probleminin tam olarak çözümü için optimal yol kriterinin de sağlanmış olmasıdır. 4.Yol her ne kadar engellere çarpma açısından uygun bir yol olduğu görülse de optimal yol açısından 1. ve 3. yollara göre uygun olmayan bir yol olarak görülmektedir. Sonuç olarak 1. ve 3. yollar bu yol planlama problemi için birbirine çok yakın optimal çözümler içerisinde yer almaktadır.

Bu problemde engellerden sakınmak için belirli yöntemler kullanılmaktadır. Bu çalışmada bulunan engeller daire şeklindedir. Başlangıç ve bitiş noktaları arasında optimal yolu bulmak için de bir noktanın bir doğruya olan uzaklığı formülünden yararlanılmıştır. Bir noktadan başka bir noktaya her geçişte engellerden sakınma için bu değer hesaplanacak ve dairenin belirlenmiş olan yarıçapı baz alınarak gerekli değerlendirme yapılmıştır. Dik koordinat sisteminde $A(x_1, y_1)$ ve $B(x_2, y_2)$ noktaları arasındaki uzaklık ABC dik üçgeninde Pisagor bağıntısı uygulanarak hesaplanabilmektedir (Boğar, 2016). Şekil 3'de bu bağıntı verilmiştir (Eokultv, 2019).



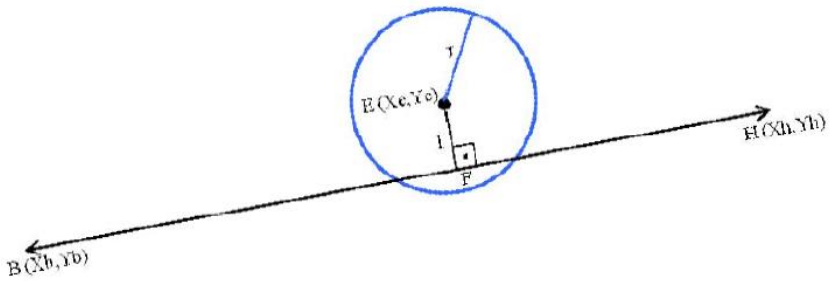
Şekil 3. İki Nokta Arasındaki Uzaklık Hesabı (Pisagor Bağıntısı)

Şekil 3'de verilen bağıntıya göre $|AC| = (x_2 - x_1)$ ve $|BC| = (y_2 - y_1)$ olduğundan buradan Denklem 2'de yer alan formül ortaya çıkmaktadır.

$$|AB|^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

$$|AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

İki nokta arasına çizilen bir doğruya dairenin orijininden olan uzaklık hesaplanacak ve bu değer daire yarıçapı ile karşılaştırılmaktadır. Bun ile ilgili bağıntı Şekil 4'te gösterilmiş olup Denklem 3'de bununla ilgi formülasyon verilmiştir (Suvaydan, 2011).



Şekil 4. İki Noktanın Orijine Olan Uzaklığı

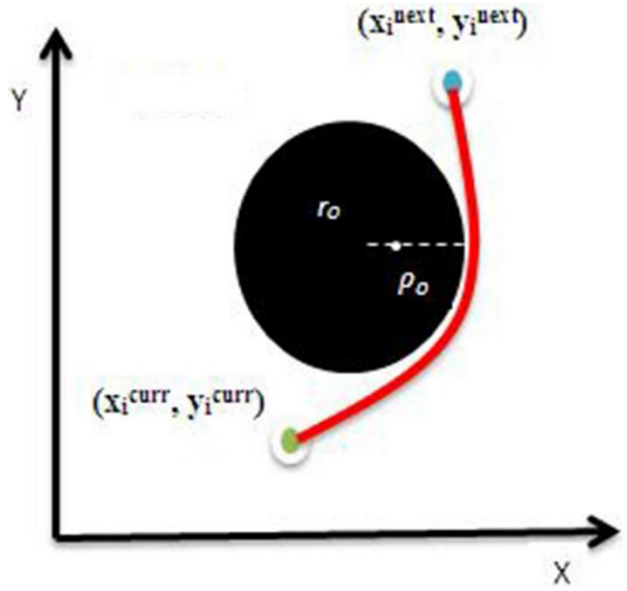
$$|EF| = l = \frac{|(X_e - X_b)(Y_h - Y_b) + (Y_b - Y_e)(X_h - X_b)|}{\sqrt{(X_h - X_b)^2 + (Y_h - Y_b)^2}} \quad (3)$$

Bu işlemler ile ilgi yapılan kodlama aşağıda verilmiştir.

```
L=sum(sqrt(dx.^2+dy.^2));% noktadan noktaya toplam mesafe

nobs = numel(xobs); % Engellerin sayısı
Violation = 0;
for k=1:nobs
    d=sqrt((xx-xobs(k)).^2+(yy-yobs(k)).^2);% Her bir engelin çapının koordinatlar arası mesafe hesabı
    v=max(1-d/robs(k),0);% Yol üzerindeki her bir engelin çapı ve doğruya uzaklığı ceza değeri
    Violation=Violation+mean(v);% ihlaller (engele çarpma vb. gibi)
end
```

Robotik engellerden kaçınma algoritmaları literatürde yer edinmiş çalışmalardandır. Engelden kaçınma yaklaşımları, bir robotun başlangıç noktasından bir bilinen noktaya, bilinen engellerin konum ve şekillerine göre bir yol bulmayı amaçlamaktadır. Bu yöntem iki işlemde gerçekleşmektedir. Birincisi yörüngenin uzunluğunu değerlendirirken, ikincisi ise yolun güvenliğini sağlarken yapılan işlemdir. Daha sonra ise Robot ile nesne arasındaki çarpışmaları tespit edebilmek için Şekil 5'teki yöntem Ayari ve Bouamama (2017) tarafından geliştirilmiş. Bunla ilgili formüller ise Denklem 4 ve 5'te verilmiştir.



Şekil 5. Robot ile Nesnenin Çarpışma Tespiti

Şekil 5'te (x, y) robot pozisyonun ifade ederken, (r_{ox}, r_{oy}) ise nesnenin yani engelin pozisyonun ifade etmektedir. Denklem 4'de ise F_o, bir engel fonksiyonun ifade etmektedir.

$$F_o = \sqrt{(x - r_{ox})^2 + \left(y - r_{oy}\right)^2} \qquad (4)$$

Daha sonra, Denklem 5'te tanımlanan δ_{ij} mantıksal (boolean) fonksiyonun tanımlanır. Burada x, robotu temsil ederken, j ise ortamda bulunan jth engeli temsil etmektedir.

$$\delta_{ij} = \begin{cases} 1, & eğer\ F_{oij} \leq \varepsilon \\ 0, & aksi\ takdirde \end{cases} \qquad (5)$$

3. Bulgular

Bu çalışmada başlangıç noktası sabit A(0,0) olan ve üç farklı bitiş noktalarına B(4,6), C(6,8) ve D(8,10) olan durumlar 3 başlık altında Durum 1, Durum 2 ve Durum 3 olarak incelenmiştir. Bu durumlar içerisinde üç farklı konum kullanılmış ve PSO algoritması uygulanarak en kısa robot yolu hesaplanmıştır. Her bir durum için ayrı ayrı sunulan sonuçlar tablolar halinde verilmiştir. Aynı zamanda çalışmada her bir farklı varış noktası için farklı engellerin konumları da simüle edilerek şekiller halinde gösterilmiştir. En kısa robot yolunun bulunması için algoritma her bir durum çalışması için 30 kez işe koşmuştur.

3.1. Durum 1

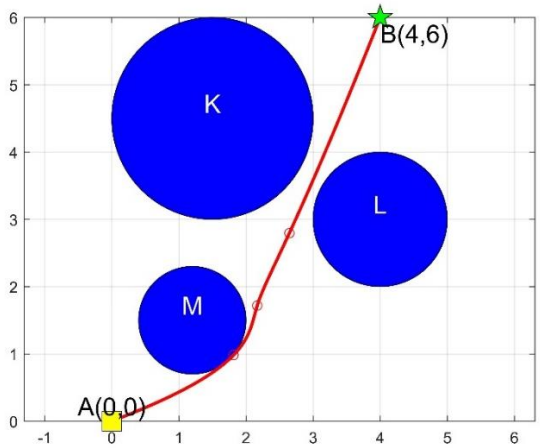
K, L ve M engellerinin koordinatları ve yarı çapları Tablo 1'de verilmiştir. Bu engeller, A(0,0) başlangıç noktası olup B(4,6), C(6,8) ve D(8,10) noktaları bitiş noktaları olup ayrı ayrı en kısa robot yolu sonuçları elde edilmiştir. Ayrıca bu sonuçlar tablo halinde sunulmuş ve sonuçların yol şemaları şekiller ile gösterilmiştir.

Tablo 1. Durum 1 İçin K, L ve M Engellerinin (X, Y) Koordinatları ve R Çapları

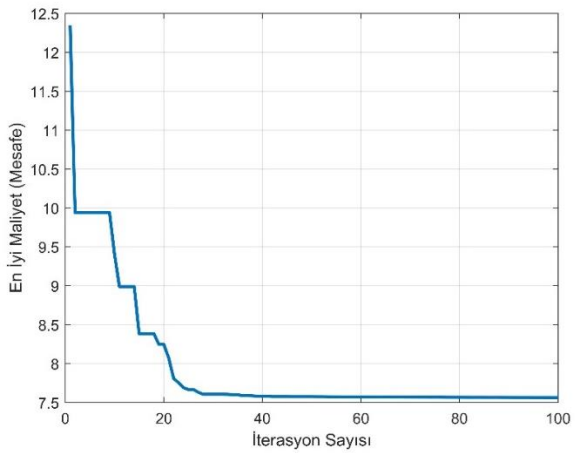
	K	L	M
X Ekseni	1.5	4.0	1.2
Y Ekseni	4.5	3.0	1.5
R Çap	1.5	1.0	0.8

3.1.1. A (0,0) noktasında bulunan bir robotun B (4,6) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(4-0)^2 + (6-0)^2} = 7.21\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 7.5637 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 6 ve 7'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 2'de verilmiştir.



Şekil 6. Robotun İzlediği En Kısa Yol



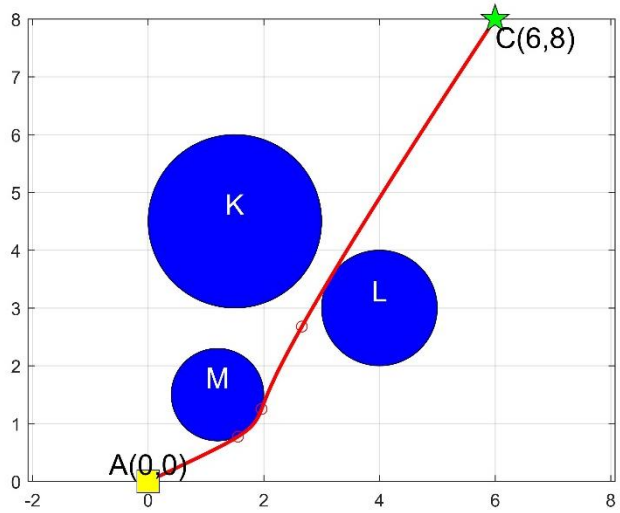
Şekil 7. Yakınsama Grafiği

Tablo 2. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

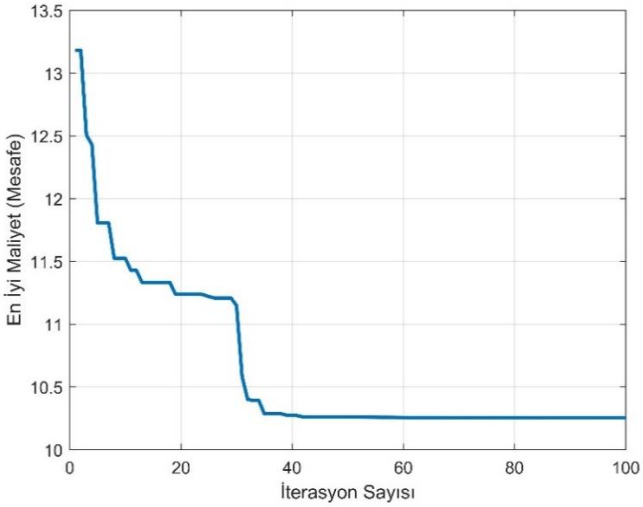
Minimum	Ortalama	Maksimum	Standart Sapma
7.5461	7.6814	9.3872	0.428

3.1.2. A (0,0) noktasında bulunan bir robotun C (6, 8) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(6-0)^2+(8-0)^2} = 10\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 10.2541 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 8 ve 9'da gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 3'te verilmiştir.



Şekil 8. Robotun İzlediği En Kısa Yol



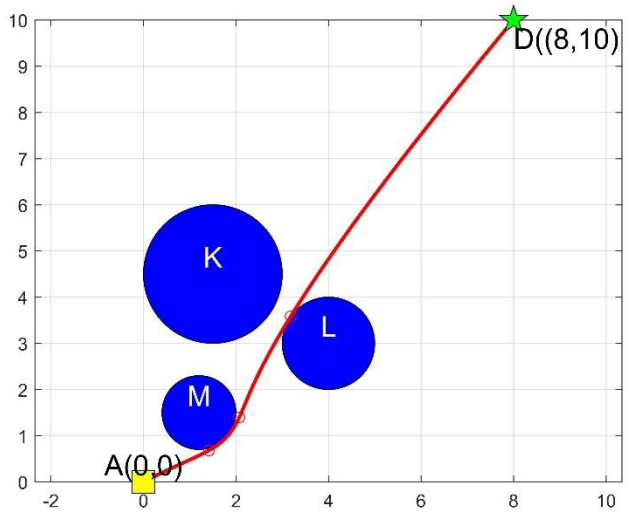
Şekil 9. Yakınsama Grafiği

Tablo 3. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

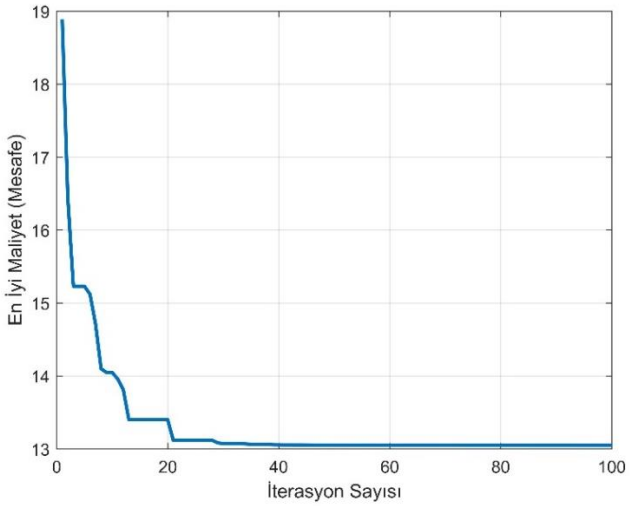
Minimum	Ortalama	Maksimum	Standart Sapma
10.2541	10.56	11.1308	1.86

3.1.3. A (0,0) noktasında bulunan bir robotun D (8, 10) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(8-0)^2+(10-0)^2} = 12.8063\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 13.0473 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 10 ve 11'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 4'te verilmiştir.



Şekil 10. Robotun İzlediği En Kısa Yol



Şekil 11. Yakınsama Grafiği

Tablo 4. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

Minimum	Ortalama	Maksimum	Standart Sapma
13.0473	13.4081	13.5190	1.086

3.2. Durum 2

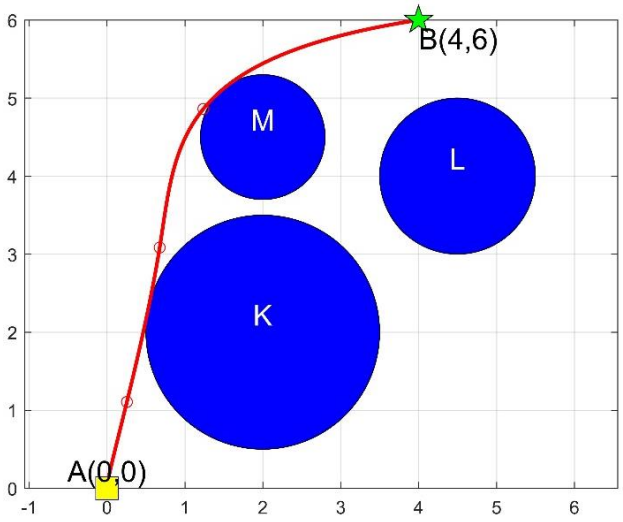
K, L ve M engellerinin koordinatları ve yarı çapları Tablo 5'te verilmiştir. Bu engeller, A(0,0) başlangıç noktası olup B(4,6), C(6,8) ve D(8,10) noktaları bitiş noktaları olup ayrı ayrı en kısa robot yolu sonuçları elde edilmiştir. Ayrıca bu sonuçlar tablo halinde sunulmuş ve sonuçların yol şemaları şekiller ile gösterilmiştir.

Tablo 5. Durum 2 İçin K, L ve M Engellerinin (X, Y) Koordinatları ve R Çapları

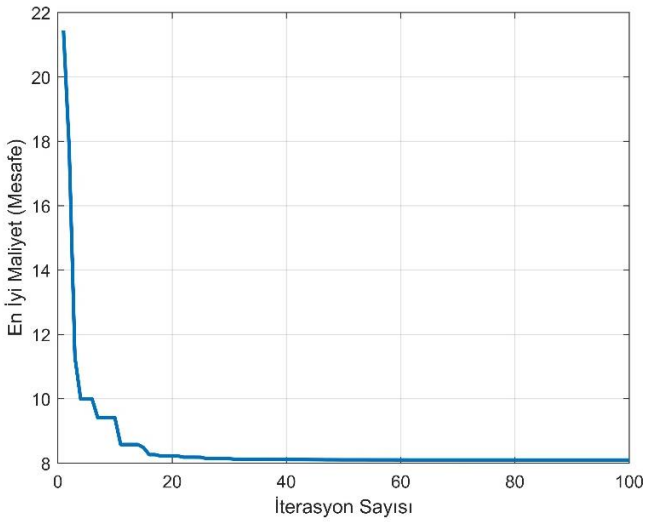
	K	L	M
X Ekseni	2.0	4.5	2.0
Y Ekseni	2.0	4.0	4.5
R Çap	1.5	1.0	0.8

3.2.1. A (0,0) noktasında bulunan bir robotun B (4,6) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(4-0)^2+(6-0)^2} = 7.21\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 7.7286 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 11 ve 12'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 6'da verilmiştir.



Şekil 12. Robotun İzlediği En Kısa Yol



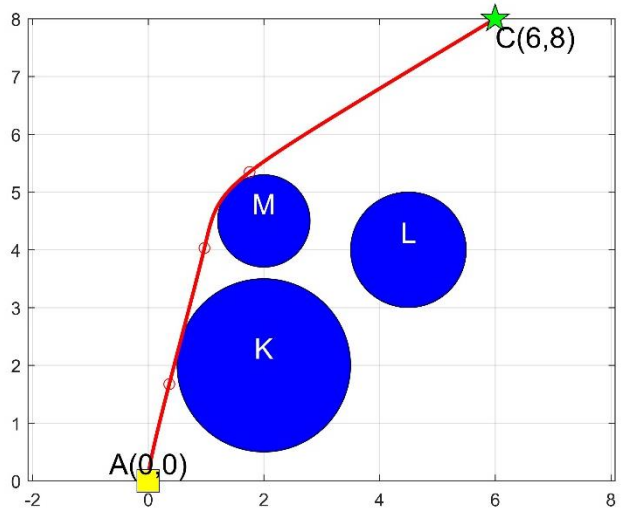
Şekil 13. Yakınsama Grafiği

Tablo 6. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

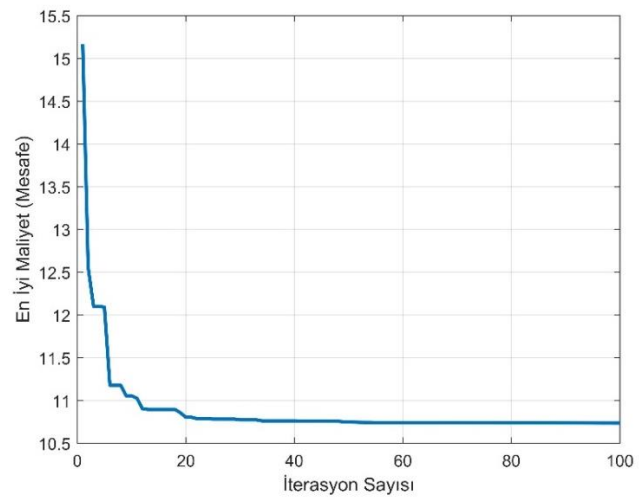
Minimum	Ortalama	Maksimum	Standart Sapma
7.7286	8.045	8.5103	1.763

3.2.2. A (0,0) noktasında bulunan bir robotun C (6, 8) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(6-0)^2+(8-0)^2} = 10\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 10.7058 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 14 ve 15'te gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 7'de verilmiştir.



Şekil 14. Robotun İzlediği En Kısa Yol



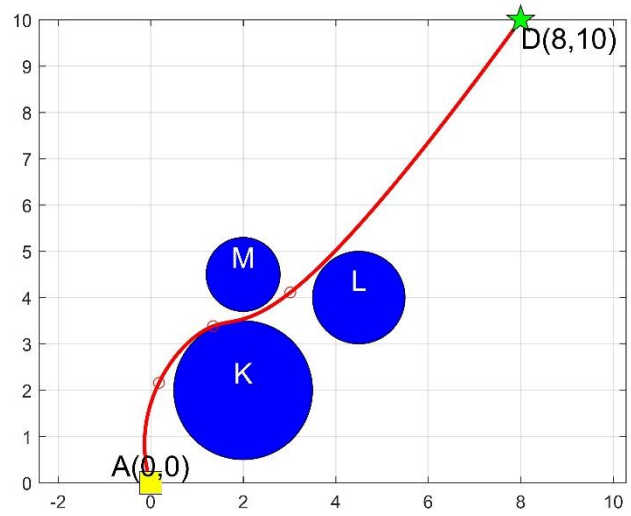
Şekil 15. Yakınsama Grafiği

Tablo 7. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

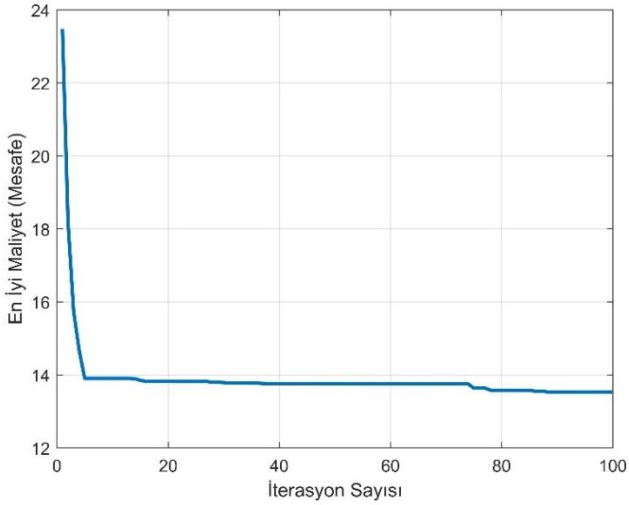
Minimum	Ortalama	Maksimum	Standart Sapma
10.7058	11.1564	11.3456	2.8154

3.2.3. A (0,0) noktasında bulunan bir robotun D (8, 10) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(8-0)^2+(10-0)^2} = 12.8063\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 13.5240 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 16 ve 17'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 8'de verilmiştir.



Şekil 16. Robotun İzlediği En Kısa Yol



Şekil 17. Yakınsama Grafiği

Tablo 8. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

Minimum	Ortalama	Maksimum	Standart Sapma
13.5240	13.7120	13.9525	2.625

3.3. Durum 3

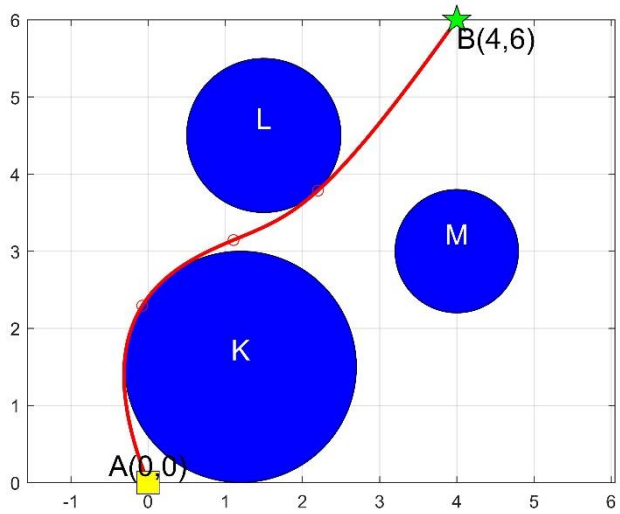
K, L ve M engellerinin koordinatları ve yarı çapları Tablo 9'da verilmiştir. Bu engeller, A(0,0) başlangıç noktası olup B(4,6), C(6,8) ve D(8,10) noktaları bitiş noktaları olup ayrı ayrı en kısa robot yolu sonuçları elde edilmiştir. Ayrıca bu sonuçlar tablo halinde sunulmuş ve sonuçların yol şemaları şekiller ile gösterilmiştir.

Tablo 9. Durum 3 için K, L ve M Engellerinin (X, Y) Koordinatları ve R Çapları

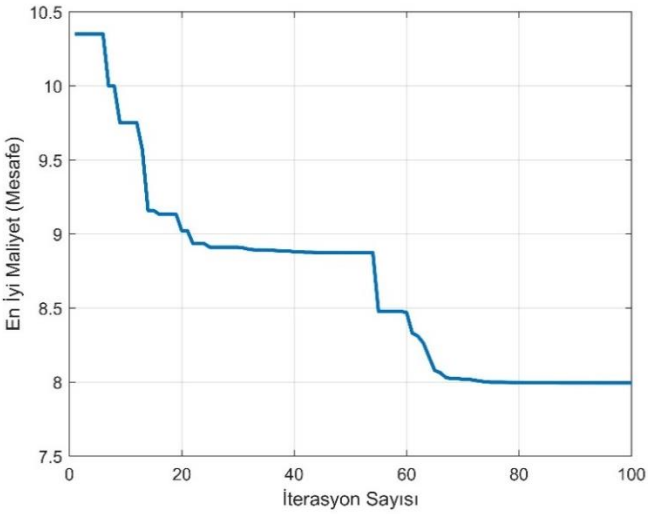
	K	L	M
X Ekseni	1.2	1.5	4.0
Y Ekseni	1.5	4.5	3.0
R Çap	1.5	1.0	0.8

3.3.1. A (0,0) noktasında bulunan bir robotun B (4,6) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(4-0)^2+(6-0)^2} = 7.21\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 7.5637 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 18 ve 19'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 10'da verilmiştir.



Şekil 18. Robotun İzlediği En Kısa Yol



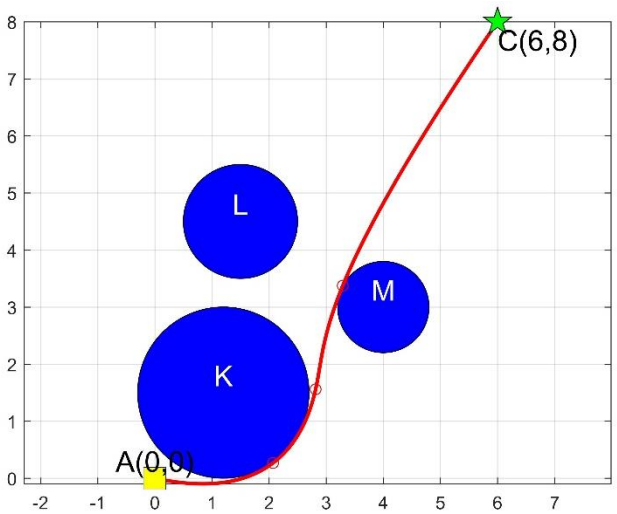
Şekil 19. Yakınsama Grafiği

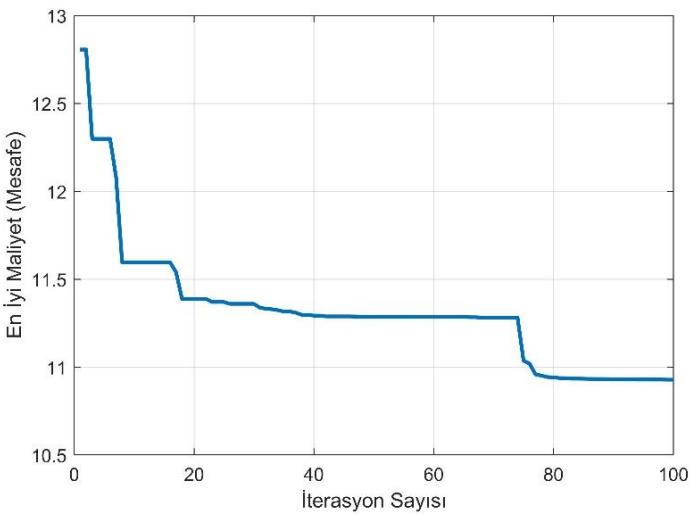
Tablo 10. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

Minimum	Ortalama	Maksimum	Standart Sapma
7.9933	8.0018	8.2473	1.051

3.3.2. A (0,0) noktasında bulunan bir robotun C (6, 8) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(6-0)^2+(8-0)^2} = 10\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 10.92 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 20 ve 21'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 11'de verilmiştir.





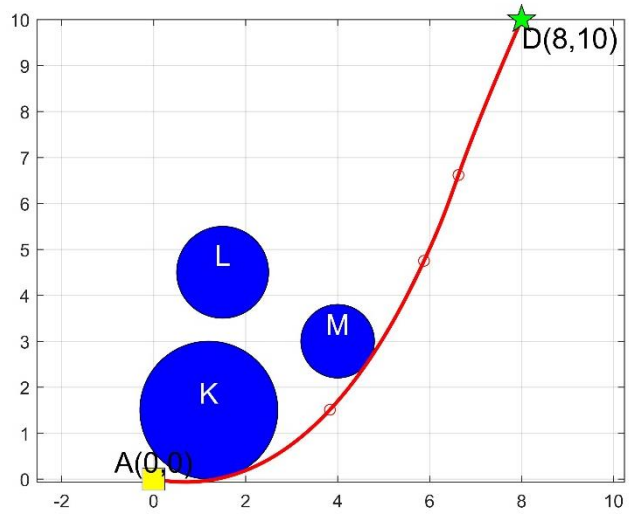
Şekil 20. Robotun İzlediği En Kısa Yol *Şekil 21. Yakınsama Grafiği*

Tablo 11. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

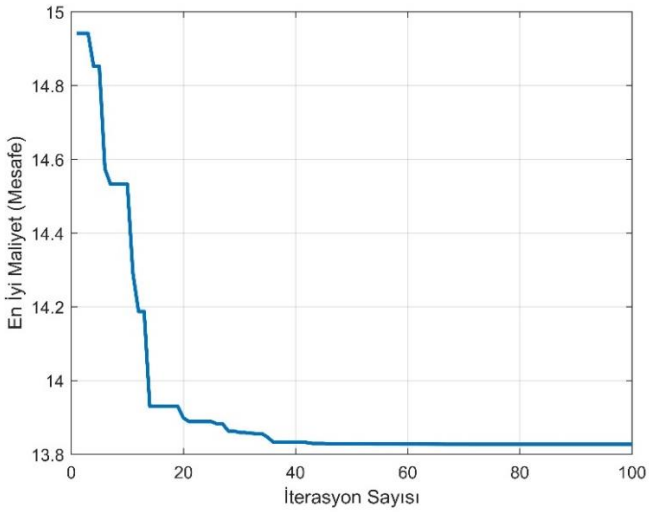
Minimum	Ortalama	Maksimum	Standart Sapma
10.92	11.2516	11.5633	2.015

3.3.3. A (0,0) noktasında bulunan bir robotun D (8, 10) noktasına göre hareketi

En kısa mesafe öklit mesafesi olduğundan başlangıç ve varış noktaları arasında engeller olmasaydı aradaki mesafe, $\sqrt{(8-0)^2+(10-0)^2} = 12.8063\ cm$. Engeller dahil edildiğinde Durum1 için PSO algoritması en kısa robot yolunu 13.5240 cm olarak bulmuştur. Robotun izlediği en kısa yol ile yakınsama grafiği sırasıyla Şekil 22 ve 23'de gösterilmiştir. Ayrıca elde edilen sonuçlar Tablo 12'de verilmiştir.



Şekil 22. Robotun İzlediği En Kısa Yol



Şekil 23. Yakınsama Grafiği

Tablo 12. PSO Algoritmasının En Kısa Yol İçin Elde Edilen Sonuçları

Minimum	Ortalama	Maksimum	Standart Sapma
13.5240	13.7120	13.9525	2.625

4. Sonuçlar

Robot yolu planlama problemi engelleri dikkate alarak en kısa yolu hesaplamaya odaklanmaktadır. Bu problem için birçok farklı çalışma ve yöntem bulunmaktadır. Ancak bu çalışmada PSO algoritması kullanılarak bu problem çözülmüştür. PSO algoritması ile bir robotun başlangıç konumu aynı ancak varış konumları farlı üç konum için en kısa yol bulunmuştur. Bu yollar içerisinde ise üç farlı engel konumlandırılmıştır. Elde edilen sonuçlar üç durum olacak şekilde tablolara ve grafikler halinde gösterilmiştir. PSO algoritması çözümlerinim robot yolu planlaması için uygulanabilir olduğu görülmüştür. Bu çalışma sonucunda, PSO'nun tarafımızca geliştirilmiş modifiye hali ile robot yolu planlaması çalışması yapılmasına zemin hazırlamıştır.

Kaynaklar

Abdelbar, A. M., Abdelshahid, S., & Wunsch, D. C. (2005). *Fuzzy PSO: a generalization of particle swarm optimization.* Paper presented at the Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.

Ayari, A., & Bouamama, S. (2017). A new multiple robot path planning algorithm: dynamic distributed particle swarm optimization. *Robotics and biomimetics, 4*(1), 8.

Boğar, E. (2016). *Tek ve Çok Amaçlı Robot Yol Planlama Problemi için Hibrit Bir Optimizasyon Yöntemi.* (Yüksek Lisans), Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Denizli.

Chakraborty, J., Konar, A., Chakraborty, U. K., & Jain, L. C. (2008). *Distributed cooperative multi-robot path planning using differential evolution.* Paper presented at the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence).

Chander, A., Chatterjee, A., & Siarry, P. (2011). A new social and momentum component adaptive PSO algorithm for image segmentation. *Expert Systems with Applications, 38*(5), 4998-5004.

Darvishzadeh, A., & Bhanu, B. (2014). *Distributed multi-robot search in the real-world using modified particle swarm optimization.* Paper presented at the Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation.

Doctor, S., Venayagamoorthy, G. K., & Gudise, V. G. (2004). *Optimal PSO for collective robotic search applications.* Paper presented at the Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753).

Eokultv. (2019). İki Nokta Arasındaki Uzaklık. Retrieved from https://www.eokultv.com/iki-nokta-arasindaki-uzaklik/3025

Grandi, R., Falconi, R., & Melchiorri, C. (2013). *Coordination and control of autonomous mobile robot groups using a hybrid technique based on particle swarm optimization and consensus.* Paper presented at the 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO).

Hereford, J. M. (2006). *A distributed particle swarm optimization algorithm for swarm robotic applications.* Paper presented at the 2006 IEEE International Conference on Evolutionary Computation.

Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization (PSO).* Paper presented at the Proc. IEEE International Conference on Neural Networks, Perth, Australia.

Pugh, J., Segapelli, L., & Martinoli, A. (2006). *Applying aspects of multi-robot search to particle swarm optimization.* Paper presented at the International Workshop on Ant Colony Optimization and Swarm Intelligence.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: modelling, planning and control*: Springer Science & Business Media.

Suvaydan, F. (2011). *Mobil robotlar için yol planlama problemi ve karınca kolonisi ile yol planlama problemlerinin optimal çözümü.* (Yüksek Lisans), Düzce Üniversitesi, Fen Bilimleri Enstitüsü, Düzce.

Tefek, M. F., & Uğuz, H. (2016). *Investigation of Fuel Cost and Emission Effects of Wind Energy Into Power Systems by Using Gsa, Tlbo and Pso Algorithms*. Paper presented at the 8 th International Ege Energy Syposium, Afyonkarahisar, Turkey.

Zhang, Y., Gong, D.-W., & Zhang, J.-H. (2013). Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing, 103*, 172-185.