Journal of the Faculty of Engineering and Architecture of Gazi University 36:3 (2021) 1551-1564

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University
Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Fast path planning in multi-obstacle environments for mobile robots

Mustafa Yusuf Yıldırım*, Rüştü Akay
Mechatronics Engineering Department, Erciyes University, Kayseri, 38039, Turkey

Highlights:

- Simplifying the problem to speed up path planning algorithms
- Clustering of randomly located a large number of obstacles using clustering algorithms
- Using metaheuristic algorithms to find safe optimum path

Keywords:

- Mobile robot
- Path planning
- Optimization
- Clustering
- Running speed

Article Info:

Research Article
Received: 22.10.2020
Accepted: 21.02.2021

DOI:

10.17341/gazimmfd.802646

Correspondence:

Author: Mustafa Yusuf Yıldırım
e-mail: myyildirim@erciyes.edu.tr
phone: +90 505 478 9296

Graphical/Tabular Abstract

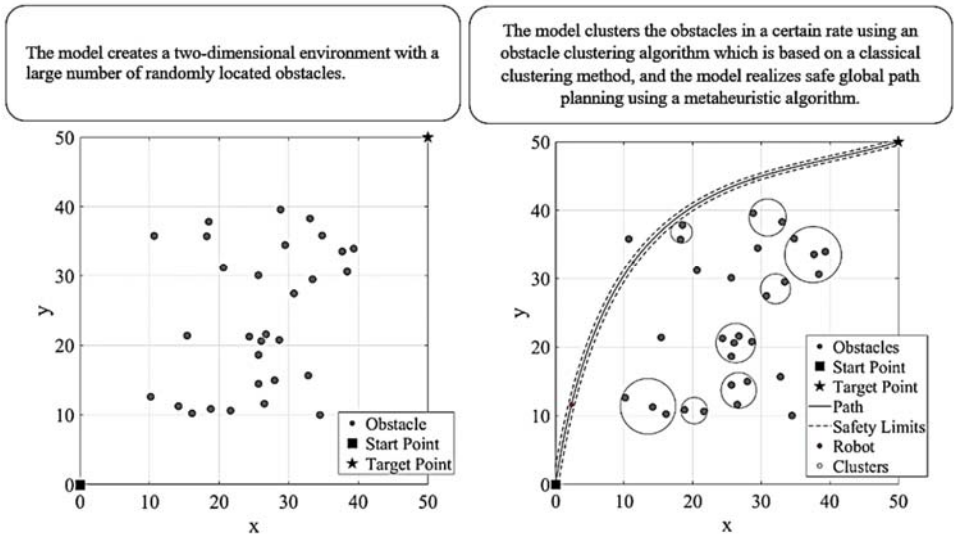


Figure A. Clustering of obstacles to increase the running speed of path planning algorithms

Purpose: In complex environments with a large number of randomly located obstacles, solving the problem of global path planning in acceptable time is an important issue. For this issue, running speeds of path planning algorithms should be increased. However, improvements have been generally made on the algorithm side in the literature. The purpose of this study is to make improvements on the problem side by reducing complexity of environments.

Theory and Methods:
In this paper, environment complexity is reduced by clustering of obstacles. In order to accomplish this, a hybrid model is proposed in which a metaheuristic algorithm and a clustering algorithm are used together is proposed. With a obstacle clustering algorithm that includes a classical clustering algorithm method, the obstacles in original environment are clustered and a new environment is created. The safe optimum path is planned in this new environment with a metaheuristic algorithm.

Results:
A detailed analysis was performed for the proposed model using PSO and k-means clustering algorithms in two different environments with 20 and 30 obstacles. As a result of the detailed analysis, while small increases were observed in the shortest distance, significant reductions in running times were detected. These reductions also largely compensate the small increases in the shortest distance. Additionally, TLBO, ABC, DE, GA and hierarchical clustering algorithms are used to support detailed analysis. When their performances are evaluated, although TLBO and ABC show the best performance in terms of the shortest distance, GA showed the best performance in terms of running time. It can also be said that the hierarchical clustering algorithm runs more efficiently than the k-means method.

Conclusion:
As a result, running speeds of path planning algorithms can be optimized by simplifying the environment, choosing a suitable clustering ratio, a suitable optimization and clustering algorithms.

Journal of the Faculty of Engineering and Architecture of Gazi University 36:3 (2021) 1551-1564

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Mobil robotlar için çok engelli ortamlarda hızlı yol planlama

Mustafa Yusuf Yıldırım*, Rüştü Akay
Erciyes Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü, 38039, Melikgazi Kayseri, Türkiye

ÖNEÇIKANLAR

- Yol planlama algoritmalarını hızlandırmak için problemin basitleştirilmesi
- Rastgele konumlanmış çok sayıda engelin kümeleme algoritmaları ile kümelenmesi
- Güvenli optimum yolun bulunması için metasezgisel algoritmaların kullanılması

Makale Bilgileri
Araştırma Makalesi
Geliş: 22.10.2020
Kabul: 21.02.2021

DOI:
10.17341/gazimmfd.802646

Anahtar Kelimeler:
Mobil robot,
yol planlama, optimizasyon,
kümeleme,
çalışma hızı

ÖZ

Günümüzde mobil robotların hedef noktalara optimum maliyetle ulaşma problemi önemli bir çalışma sahası haline gelmiştir. Optimum maliyet çalışmalarda farklılık göstermekle beraber genel olarak, hedefe ulaşmak için geçen süre, mesafe, harcanan enerji veya bunların bir arada değerlendirildiği değişik kombinasyonlar olabilmektedir. Özellikle çok engelli karmaşık ortamlarda problemlerin çözümünün kabul edilebilir sürelerde gerçekleştirilebilmesi için genelde algoritma tarafında iyileştirmeler yapılmaktadır. Bu çalışmada ise problem tarafında iyileştirmeye odaklanılmıştır. Bu doğrultuda, statik engelli ve iki boyutlu ortamlarda engellerin kümelenmesiyle ortam karmaşıklığının azaltılması ve bu sayede optimizasyon algoritmalarının çalışma hızının artırılması amacıyla, bir metasezgisel algoritma ve bir kümeleme algoritmasının bir arada kullanıldığı hibrit bir model önerilmektedir. Önerilen model için öncelikle detaylı bir analiz gerçekleştirilmiştir. Bu analizde metasezgisel algoritma olarak parçacık sürü optimizasyonu (Particle Swarm Optimization, PSO) ve kümeleme algoritması olarak k-ortalamalar kümeleme algoritması kullanılmıştır. Çeşitli kümeleme oranları ile test edilen bu analiz sonucunda kümeleme oranı arttıkça en kısa mesafeli yol açısından küçük kayıplar elde edilmiş ancak algoritmanın çalışma hızı bu kayıpları fazlasıyla telafi edebilecek seviyede artmıştır. Ayrıca önerilen modelin etkinliği farklı metasezgisel ve kümeleme algoritmaları üzerinde de karşılaştırmalı değerlendirilmiştir. Sonuçlar, rastgele konumlanmış çok sayıda engelin bulunduğu iki boyutlu ortamlar için yol planlama algoritmalarının çalışma hızı önerilen model sayesinde artırılabileceğini göstermiştir.

Fast path planning in multi-obstacle environments for mobile robots

HIGHLIGHTS

- Simplifying the problem to speed up path planning algorithms
- Clustering of randomly located a large number of obstacles using clustering algorithms
- Using metaheuristic algorithms to find safe optimum path

Article Info
Research Article
Received: 22.10.2020
Accepted: 21.02.2021

DOI:
10.17341/gazimmfd.802646

Keywords:
Mobile robot,
path planning,
optimization,
clustering,
running speed

ABSTRACT

Nowadays, the problem of mobile robots to reach target points with optimum cost has become an important field of study. Although the optimum cost varies in studies, in general, time spent, distance, energy spent, or different combinations in which these are evaluated together. Especially in complex environments with a large number of obstacles, improvements are generally made on the algorithm side in order to solve the problems in acceptable time. This study focuses on improvement on the problem side. In this context, a hybrid model is proposed in which a metaheuristic algorithm and a clustering algorithm are used together in order to reduce the complexity of the environment by clustering the obstacles in static and two-dimensional environments and thus increase the running speed of optimization algorithms. For the proposed model, a detailed analysis has been performed first. In this analysis, particle swarm optimization (PSO) as metaheuristic algorithm and k-means clustering algorithm are used as clustering algorithm. As a result of this analysis, which was tested with various clustering rates, as the clustering rate increased, small losses were obtained in terms of shortest distance path, but the running speed of the algorithm increased at a level that could compensate these losses. In addition, effectiveness of the proposed model was evaluated comparatively on different metaheuristic and clustering algorithms. The results show that speed of path planning algorithms can be increased by the proposed model for two-dimensional environments with a large number of randomly located obstacles.

*Sorumlu Yazar / Yazarlar / Corresponding Author / Authors: *myyildirim@erciyes.edu.tr, akay@erciyes.edu.tr / Tel: +90 505 478 9296

1. GİRİŞ (INTRODUCTION)

Başta endüstri olmak üzere birçok alanda robotlar yaygın olarak kullanılmaktadır. Robotların çevresel şartlara uyumlu çalışması da her zaman önem taşımaktadır. Robotların bir çeşidi olan mobil robotların bulunduğu ortamdaki engellere çarpmadan ve ortama zarar vermeden hareket etmesi gerekir. Bunun için literatürde birçok engelden kaçınma algoritmaları geliştirilmiştir. Günümüzde, özellikle mobil robotların engelli ortamlarda yol planlamasında güvenli bir şekilde hareket etmesi oldukça önemli bir araştırma konusu haline gelmiştir. Ortamdaki engellerin karakteristikleri bu konunun kilit noktasıdır. Ortamdaki engellerin sayısı arttıkça robotun güvenli bir şekilde hareket etmesi zorlaşır ve algoritmaların hesaplama maliyeti artar; örneğin, algoritmaların çalışma süresi artabilir. Ayrıca engellerin statik yerine dinamik olması planlamanın daha da zorlaşması anlamına gelmektedir. Ortam karmaşıklığının basitleştirilmesi yol planlama algoritmaları açısından bir avantaj olabilir [1].

Literatürde optimum yol planlama ile ilgili bazı güncel çalışmalar şu şekilde açıklanabilir: Ajeil vd. yol planlama için standart yarasa algoritmasının performansını artırmak için Modified Frequency Bat ve Hybrid Particle Swarm Optimization-Modified Frequency Bat şeklinde isimlendirilen iki algoritma geliştirmişlerdir [2]. Aynı araştırma ekibi, karınca koloni algoritmasına yaşa dayalı bir güncelleme getirerek Aging-Based Ant Colony Optimization algoritması geliştirmişlerdir [3]. Kim vd. standart Analytic Hierarchy Process algoritmasına göre daha iyi sonuçlar üreten Fuzzy Analytic Hierarchy Process tabanlı yeni bir yol planlama yöntemi geliştirmişlerdir [4]. Aynı araştırmacı farklı bir ekiple birlikte, bir mobil robot için A* ve Fuzzy Analytic Hierarchy Process tabanlı yol planlama modülü ile Brain Limbic System tabanlı hareket kontrol modülünden oluşan bir kontrol yapısı geliştirmiştir [5]. Li vd. diğer konseptlerden farklı olarak yol planlama problemini termal uyumluluğu minimize eden ısı transferi kavramına dönüştürmüşlerdir [6]. Dirik vd. gerçek zamanlı mobil robot uygulamalarında yol planlamanın kinematik kontrol yapısı için Visual Servoing ve hibrit algoritmalara dayanan bir yaklaşım geliştirmişlerdir [7]. Zhong vd. geniş ölçekli ortamlarda yol planlamanın zorlukları için A* ve Adaptive Window Approach tabanlı bir yöntem geliştirmişlerdir [8]. Jung vd. yol planlamasında, yerel minimumdan kaçınmak için temaslı engeller arasında bir itici alan oluşturan ızgara tabanlı bir yöntem geliştirmişlerdir [9]. Wu vd. yol planlamanın tamamen gerçek ortamda veya tamamen simülasyon yoluyla gerçekleştirilmesinden farklı olarak mobil robotun gerçek ortama dahil edilen sanal nesnelerle etkileşimini araştırmışlardır [10]. Arango vd. çok hedefli karmaşık ortamlarda güvenli bir yol planlaması için homotopi sürekliliğine dayanan bir yöntem geliştirmişlerdir [11]. Wang vd. dinamik ortamlarda yol planlama için A* algoritmasının kullanıldığı üç boyutlu bir bulut haritası geliştirmişlerdir [12]. Elmi vd. dinamik ve karakteristikleri bilinmeyen ortamlarda yol planlamanın başarısız olmasından dolayı bu probleme Grasshopper algoritması dahil edilerek bir çözüm önermişlerdir [13]. Zhang vd. bilinen ve iç mekan ortamlarda yol planlama için Deep Learning, Ray Tracing, Waiting Rule, Rapidly-Exploring Random Tree algoritmalarına dayanan hibrit bir yaklaşım geliştirmişlerdir [14]. Li vd. yol planlama algoritmalarının yakınsama hızını artırmak ve yerel minimumdan kaçınmak için standart genetik algoritmayı geliştirmişler ve yol planlama problemine uygulamışlardır [15]. Ali vd. yol planlamanın kalitesini artırmak için Ant Colony Optimization ve A* Multi-Directional algoritmalarına dayalı hibrit bir yöntem geliştirmişlerdir [16]. Kıvanç vd. çoklu disiplin yaklaşımlarının literatürdeki yetersizliğinden dolayı haritalama, yerelleştirme, yol planlama ve yol takip algoritmalarının bir arada kullanıldığı bir insansız kara aracı tasarlamışlardır. Bu çalışmada yol planlama için A* algoritması kullanılmıştır [17]. Yılmaz vd. insansız hava araçları için sensörlerin görüş yeteneklerinin yol planlamaya dâhil edildiği ve bu sayede birim sensör kaplama algoritması kullanılarak varılacak hedeflerin kümelendiği bir model önermişlerdir. Hedeflerin kümelenmesiyle toplam uçuş yolu kısaltılmıştır. Önerilen modelde yol planlama, tavlama benzetimi tabanlı Demon algoritması ile gerçekleşmektedir [18].

Literatürde birçok optimum yol planlama algoritması geliştirilmesine rağmen, bu algoritmalar karmaşık ve çok engelli ortamlarda uygulandığında uzun çalışma sürelerine ihtiyaç duymaktadır. Bunun için bu algoritmaların çalışma hızlarının artırılması gerekmektedir. Ancak ihtiyaç duyulan hızlanma genellikle algoritma tarafında çözülmeye çalışılmaktadır. Daha hızlı algoritmalar geliştirilmekte veya mevcut algoritmalar güncellenerek karmaşıklıkları azaltılmaktadır. Bu çalışma ise hızlanmanın problem tarafında çözülmesine odaklanmış ve problemin basitleştirilmesi üzerine durulmuştur. Bunun için engellerin kümelenmesiyle ortam karmaşıklığının azaltılması ve bu sayede yol planlama algoritmalarının çalışma hızlarının artırılması amaçlanmıştır. Bu amaçlar doğrultusunda metasezgisel ve kümeleme algoritmalarının bir arada kullanıldığı hibrit bir model önerilmiştir. Öncelikle PSO ve k-ortalamalar kümeleme algoritmaları ile önerilen modelin detaylı analizi gerçekleştirilmiştir. Daha sonra önerilen modelin etkinliği, öğretme-öğrenme temelli optimizasyon (Teaching-Learning Based Optimization, TLBO), yapay arı koloni algoritması (Artificial Bee Colony, ABC), diferansiyel gelişim algoritması (Differantial Evolution, DE), genetik algoritma (Genetic Algorithm, GA) ve hiyerarşik kümeleme algoritmalarının kullanılması ile karşılaştırmalı değerlendirilmiştir. Önerilen model ile daha kısa sürede optimum yol planlaması gerçekleştirilebileceği söylenebilir.

Çalışmanın bölümlerini şu şekilde sıralamak mümkündür: İkinci bölümde materyal ve yönteme, üçüncü bölümde bulgulara, dördüncü bölümde de sonuca yer verilmiştir.

2. MATERYAL VE YÖNTEM
(MATERIAL AND METHOD)

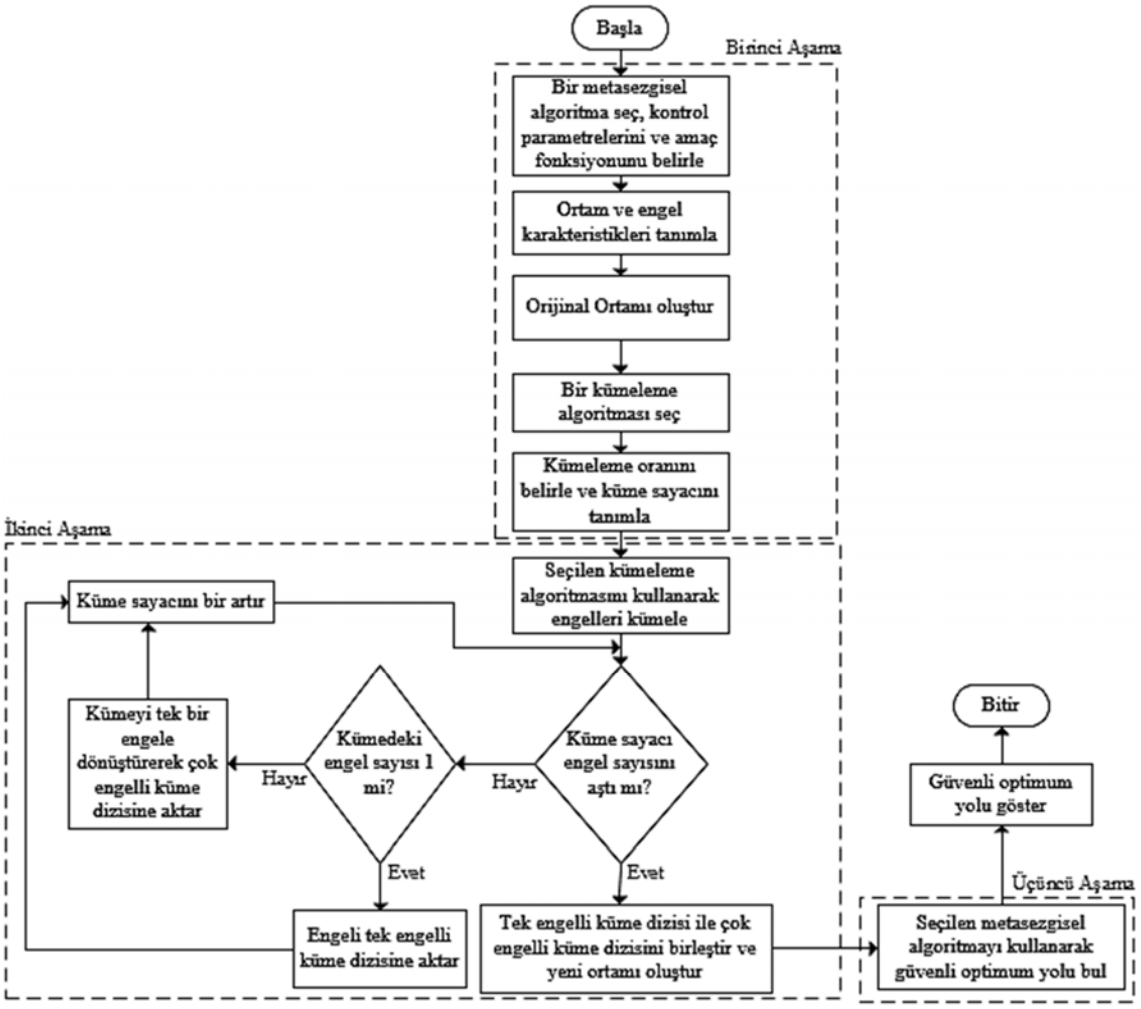
Bu çalışmada engelleri kümeleme yaklaşımının yol planlama algoritmaları üzerindeki etkisi araştırılmıştır. Bu araştırma için metasezgisel ve kümeleme algoritmalarının bir arada kullanıldığı hibrit bir model önerilmektedir. Bu model üç aşamadan oluşmaktadır. İlk aşamada, kullanılan metasezgisel algoritmanın kontrol parametreleri ile kümeleme oranı belirlenmekte, ortam ve engel karakteristikleri tanımlanarak orijinal ortam oluşturulmaktadır. İkinci aşama olarak, bir kümeleme yönteminin kullanıldığı bir engel kümeleme algoritması geliştirilmiştir. Bu algoritma ile orijinal ortamdaki engeller belli sayılarda kümelenmektedir. Kümelenen engeller tek bir engel olarak tanımlanmaktadır. Böylece ortam karmaşıklığı azaltılmakta ve yol planlama için yeni bir ortam oluşturulmaktadır. Üçüncü aşamada, söz konusu metasezgisel algoritmanın kullanıldığı bir yol planlama simülasyonu ile bu yeni ortamda optimum yol planlanmaktadır. Önerilen modelin akış diyagramı Şekil 1'de gösterilmektedir.

2.1. K-Ortalamalar Kümeleme Algoritması
(K-means Clustering Algorithm)

K-ortalama kümeleme algoritması verilerin belli bir kritere göre otomatik olarak gruplandırılması için yaygın olarak kullanılan ve yinelemeli olarak çalışan denetimsiz bir öğrenme tekniğidir. Bu kümeleme yöntemindeki temel prensip, küme sayısının ve küme merkezlerinin belirlenerek birbirine benzerlik gösteren verilerin aynı kümelere yerleştirilmesidir. Küme merkezlerinin birbirine olabildiğince uzak konumlarda belirlenmesi gerekir. Küme sayısının belirlenmesi bu yöntemin kilit noktasıdır. Algoritma öncelikle küme sayısınca rastgele merkezler oluşturur ve her bir verinin küme merkezleri arasındaki mesafe Eş. 1 kullanılarak hesaplanır ve bu mesafe aracılığıyla veriler en yakın kümeye atanır.

$$d_i = \left\| p_i - c_j \right\|^2, i = 1, \dots, n;\ j = 1, \dots, k \quad (1)$$

Burada p_i i. veriyi, c_j j. küme merkezini, d_i ise i. veri ile j. küme merkezi arasındaki Öklid mesafesini temsil etmektedir. n veri sayısı ve k küme sayısıdır. Öklid mesafesi



Şekil 1. Önerilen modelin akış diyagramı (Flowchart of the proposed model)

yerine farklı mesafe denklemleri de kullanılabilir. Veriler en yakın kümelere atandıktan sonra Eş. 2'deki denklem kullanılarak her kümenin ortalaması alınır ve böylece yeni küme merkezleri oluşturulur.

$$c_j = \frac{1}{|c_j|}\sum p \tag{2}$$

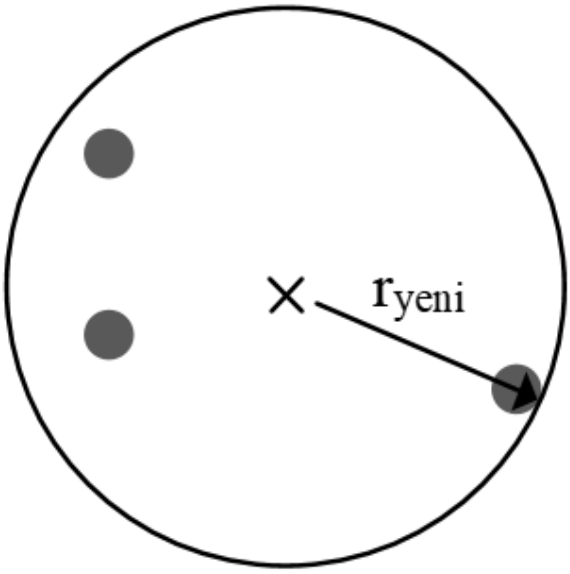
Burada $|c_j|$ j. kümedeki veri sayısı, p j. kümedeki verilerdir. Bu yeni küme merkezleri ve veriler arasındaki mesafe tekrar hesaplanır. Bu yinelemeli süreç tüm verilerin en yakın merkezlere atanmasıyla (durdurma kriteri) son bulur. K-ortalamalar kümeleme algoritmasının sözde kodu Tablo 1'de gösterilmektedir [19-22].

2.2. Engel Kümeleme Algoritması
(Obstacle Clustering Algorithm)

Çalışmada, çeşitli kümeleme algoritmalarının kullanılabildiği bir engel kümeleme algoritması geliştirilmiştir. Engeller dairesel oldukları için gruplandırma da dairesel olarak tasarlanmıştır. Detaylı analizde kümeleme, k-ortalamalar kümeleme algoritması ile gerçekleştirilmektedir. Bu yöntem için küme sayısı gereklidir. Ancak bu çalışma yol planlama algoritmalarının çalışma hızlarını farklı küme sayıları ile analiz etmektedir. Bu sebeple küme sayısını değiştiren bir oran oluşturulmuştur. Çalışmada bu oran % biriminde kümeleme oranı olarak ifade edilmektedir. Bu oran engellerin yüzde kaçının kümeleneceğini ifade etmektedir. Önerilen modelde birbirine yakın engellerin kümelenmesi esas alınmıştır. Her engel kümelenmediği için tek engelli kümeler oluşabilmektedir. Kümeleme oranı arttıkça kümelerin içindeki engel sayısı her zaman artar, ancak küme sayısı belli bir seviyeye kadar artış gösterir. Bu seviyeden daha yüksek oranlarda, tek engelli küme kalmaz ve kümelenecek engel sayısı arttığı için daha büyük kümeler oluşur. Bu durumda küme sayısı azalır ve kümelerde çakışma durumu meydana gelir. Ancak algoritma bu kümeleri yeni statik engeller olarak belirlediği için yol planlama konusunda herhangi bir zorluğa neden olmaz. Bu sayede engel sayısı ve kümeleme oranı kullanılarak k-ortalamalar kümeleme için gerekli olan küme sayısı hesaplanır. Küme sayısının ifadesi Eş. 3'teki gibi gösterilebilir.

$$\text{Küme Sayısı} = \text{Engel Sayısı} \times \text{Kümeleme Oranı (\%)} \tag{3}$$

Birden fazla engelin kümelenmesinde, Şekil 2'de olduğu gibi merkeze en uzak engelin tamamını kapsayacak geçecek şekilde bir daire oluşturulur. Geliştirilen bu algoritma, bu daireleri ve tek engelli kümeleri yeni bir engel dizisine aktarır ve yeni engel dizisini baz alarak çalışır. Bu şekilde engel sayısı azaltılarak ortam karmaşıklığının bir miktar basitleştirilmesi sağlanmaktadır. Bu çalışmadaki engel kümeleme algoritmasının sözde kodu Tablo 2'de gösterilmektedir.



Şekil 2. Birden fazla engelin kümelenmesi [23]
(Clustering of multi-obstacles)

Tablo 1. K-ortalamalar kümeleme algoritmasının sözde kodu (Pseudo code of K-means clustering algorithm)

```
1 :  fonksiyon kümeler = f_K-ORTALAMALAR (veri, k)
2 :  p_i ← veri
3 :  n ← veri sayısı
4 :  k ← küme sayısı
5 :  Rastgele k adet küme merkezi belirle (c_1, c_2, ... , c_j)
6 :  repeat
7 :  for i = 1 : n
8 :  for j = 1 : k
9 :  d_i ← Eşitlik 1'deki denklemi kullanarak p_i ve c_j arasındaki mesafeyi hesapla
10 : end
11 : d_i vektörünün minimumunun ait olduğu kümeyi seç ve p_i verisini o kümeye ata
12 : end
13 : for j = 1 : k
14 : c_j ← Eşitlik 2'deki denklemi kullanarak yeni küme merkezini belirle
15 : end
16 : until durdurma kriteri sağlanana kadar
17 : end fonksiyon
```

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

Tablo 2. K-ortalamalar yöntemi için engel kümeleme algoritmasının sözde kodu
(Pseudo code of obstacle clustering algorithm for k-means method)

```
fonksiyon yeniEngelDizisi = kümeleme (obsNum, obsRad, clusRate, engelDizisi)
obsNum ← engel sayısı
obsRad ← engellerin yarıçapı
clusRate ← kümeleme oranı
dataClus ← engelDizisi
clusNum ← Eşitlik 3'teki denklemi kullanarak küme sayısını hesapla
kümeler ← f_K-ORTALAMALAR (engelDizisi, clusNum)
for i = 1 : clusNum
if size(dataClus(i)) == 1 // tek engelli kümelerin dizisini oluştur
tekEngel ← i. kümedeki tekli engelin konum ve yarıçap bilgisini al
else // çok engelli kümelerin dizisini oluştur
çokEngel.konum ← i. kümedeki engellerin konum bilgilerini al
for k = 1 : size(dataClus(i))
d(k) ← i. küme merkezi ile k. engel arasındaki mesafeyi hesapla
end for
çokEngel.yarıçap ← merkezden en uzak engele olan mesafeye engelin yarıçapını ekle
end if
end for
yeniEngelDizisi = [tekEngel çokEngel]
end fonksiyon
```

2.3. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)

Parçacık sürü optimizasyonu, kuş ve balıkların sosyal davranışlarından esinlenerek bilgisayar simülasyonlarına aktarılmasıyla geliştirilen sürü tabanlı bir optimizasyon algoritmasıdır. 1995 yılında Eberhart ve Kennedy tarafından geliştirilmiştir [24]. Bu algoritmada her bir birey parçacık şeklinde isimlendirilmiştir. Algoritma genellikle nümerik problemler için kullanılsa da literatürde farklı problemlere uyarlanan versiyonları da mevcuttur.

PSO algoritmasının genel çalışma mantığı şu şekildedir: İlk aşamada, parçacıkların ilk konumları rastgele dağılımlı olarak oluşturulur ve ilk hızları sıfır olarak belirlenir. İkinci aşamada, konumların uygunluk değerleri hesaplanır. Bu uygunluk değerlerinin elde edilmesi için amaç fonksiyonu kullanılır. Üçüncü aşamada, ilgili iterasyon için her bir parçacığın en iyi konumu ($x_{best,i}$) ve sürünün o ana kadarki en iyisinin konumu (g_{best}) tespit edilir. Bu tespitin yapılabilmesi için ikinci aşamada hesaplanan uygunluk değerleri kullanılır. Dördüncü aşamada, $x_{best,i}$ ve g_{best} aracılığıyla her bir parçacığın hızı ve konumu güncellenir. Hızın güncellenmesinde bir ağırlık katsayısı (eylemsizlik ağırlığı) kullanılır. Beşinci aşamada, bu güncellemeler ile elde edilen en iyi çözüm hafızada saklanır ve algoritma ikinci aşamaya yönelir. Bu iteratif süreç durdurma kriteri sağlanana kadar devam eder. Eş. 4 ve Eş. 5'te hız ve konum güncelleme denklemlerini, Eş. 6'da ise bu çalışmadaki eylemsizlik ağırlığının her iterasyonda güncellendiği denklem gösterilmektedir. Algoritmanın sözde kodu Tablo 3'te gösterilmektedir.

$$v_i(t+1) = w.v_i(t) + r_1.c_1.\left(x_{best,i} - x_i(t)\right) + r_2.c_2.\left(g_{best} - x_i(t)\right), i = 1, \dots, P \quad (4)$$

$$w = w_{max} - \frac{w_{max} - w_{min}}{maxIt} \quad (6)$$

Burada $v_i(t+1)$ i. parçacığın güncel hızı, w eylemsizlik ağırlığı, $v_i(t)$ i. parçacığın eski hızı, r_1-r_2 (0, 1) arasında üretilen rastgele sayılar, c_1-c_2 öğrenme katsayıları, $x_i(t)$ i. parçacığın eski konumu, P popülasyon boyutu, $x_i(t+1)$ i. parçacığın güncel konumu, w_{min} - w_{max} eylemsizlik ağırlığının minimum ve maksimum sınır değerleri, maxIt algoritmanın maksimum iterasyon sayısıdır [24].

2.4. Amaç Fonksiyonu (Fitness Function)

Bu çalışmada yol planlama, kübik spline interpolasyonu ile gerçekleştirilmiştir. Kübik spline interpolasyonunun uygulanabilmesi için belli noktalara ihtiyaç duyulur. PSO algoritmasındaki çözümlere karşılık gelen bu noktalar parametre noktaları olarak, noktaların sayısı da parametre sayısı (d) olarak ifade edilmektedir. Öncelikle, amaç fonksiyonuna girdi olarak gelen bu parametre noktalarının x ve y konumları, planlanacak yolun başlangıç ve bitiş noktaları ile birlikte farklı satır vektörlerine aktarılır. Bu vektörler n adet arama noktasına sahip kübik spline fonksiyonu ile ayrı ayrı interpole edilir. Bu şekilde planlanan yolun x ve y nokta dizileri elde edilmiş olur, burada x dizisindeki her bir nokta P_{ix}, y dizisindeki her bir nokta P_{iy} olarak ifade edilmektedir. Daha sonra, parametre noktalarının uygunluk değerleri Eş. 7'deki uygunluk denklemi ile hesaplanır. Bu denklem iki kısımdan oluşmaktadır. Birinci kısım robotun gideceği yolun uzunluğunu hesaplarken, ikinci kısım da robot ile engeller arasındaki uygulanabilir mesafeyi (engelden kaçınma kontrolü) hesaplar.

$$\min F(P_i, O_j) = L(P_i).\left[1 + \beta.V(P_i, O_j)\right], \quad i = 1, \dots, n;\ j = 1, \dots, o \quad (7)$$

Tablo 3. PSO algoritmasının sözde kodu (Pseudo code of PSO algorithm)

1:	maxIt ← maksimum iterasyon sayısı
2:	popSize ← popülasyon boyutu
3:	c_1-c_2 ← öğrenme katsayıları
4:	for i = 1 : popSize
5:	x_i ← i. parçacık için rastgele çözüm üret
6:	if $f(x_{best,i}) < f(g_{best})$
7:	$g_{best} \leftarrow x_{best,i}$
8:	end if
9:	end for
10:	repeat
11:	for i = 1 : popSize
12:	v_i ← Eşitlik 4'teki denklemi kullanarak i. parçacığın hızını güncelle
13:	x_i ← Eşitlik 5'teki denklemi kullanarak i. parçacığın konumunu güncelle
14:	if $f(x_i) < f(x_{best,i})$
15:	$x_{best,i} \leftarrow x_i$
16:	if $f(x_{best,i}) < f(g_{best})$
17:	$g_{best} \leftarrow x_{best,i}$
18:	end if
19:	end if
20:	end for
21:	w ← Eşitlik 6'daki denklemi kullanarak eylemsizlik ağırlığını güncelle
22:	until durdurma kriteri sağlanana kadar (maksimum iterasyon sayısı kadar)
23:	g_{best} ve $f(g_{best})$ değerlerini göster

Burada $F(P_i, O_j)$ optimize edilecek amaç fonksiyonu, O_j j. engelin konumu, $L(P_i)$ yol uzunluk denklemi, β engelden kaçınma faktörü, $V(P_i, O_j)$ engelden kaçınma kontrol denklemi, o engel sayısıdır. Yol uzunluk denklemi Eş. 8'de, engelden kaçınma kontrol denklemi Eş. 9'da gösterilmektedir. Engeller dairesel olarak tasarlanmıştır ve a_j engellerin yarıçapını temsil etmektedir [25].

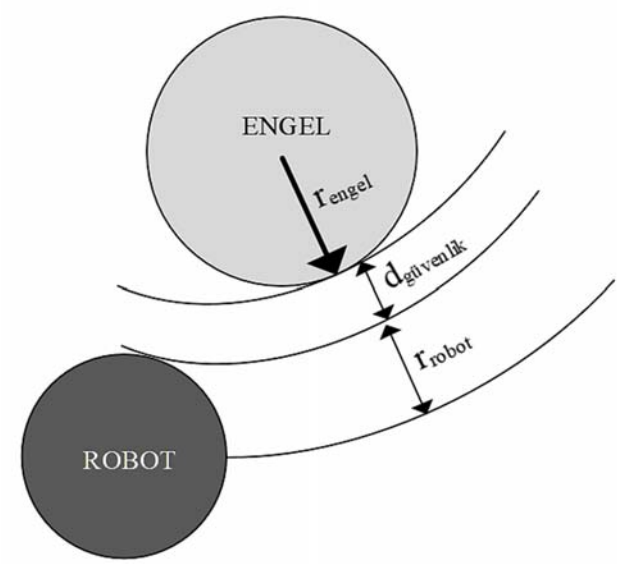
$$L(P_i) = \sum_{i=1}^{n-1} \sqrt{(P_{ix} - P_{(i+1)x})^2 + (P_{iy} - P_{(i+1)y})^2} \quad (8)$$

$$V(P_i, O_j) = \frac{1}{n} \sum_{i=1}^{o} \left[\sum_{j=1}^{n} \max \left(1 - \frac{\sqrt{(P_{ix} - O_{jx})^2 + (P_{iy} - O_{jy})^2}}{a_j} \right) \right] \quad (9)$$

Çalışmadaki mobil robot da engeller gibi dairesel olarak tasarlanmıştır. Engelden kaçınma kontrolünde sadece engelin boyutu değil, aynı zamanda robotun boyutu ve güvenlik mesafesi de hesaba katılmıştır. Güvenlik mesafesi robot ile engel arasındaki boş alanı ifade etmektedir. Engelden kaçınma kontrolünün temsili çizimi Şekil 3'te gösterilmektedir.

3. BULGULAR (RESULTS)

Önerilen model, MATLAB 2019 programlama dilinde kodlanmış ve Windows 10 işletim sistemi, INTEL CORE i7 işlemcisi, 16 GB RAM'e sahip bir bilgisayarda çalıştırılmıştır. Ortam, 50 x 50 cm boyutlu karesel alan olarak tasarlanmıştır. Robotun başlangıç konumu ortam ekranının sol alt köşesi, hedef konumu ise sağ üst köşedir. Engelsiz en kısa mesafe 70,7107 cm'dir. Engel ihlali faktörü ve kübik spline interpolasyonu için n arama noktası 100'dür. Amaç fonksiyonu için parametre noktalarının sınır değerleri x ve y için [0 50] olarak, engel konumlarının sınır değerleri x ve y için [10 40] olarak belirlenmiştir. Engellerin yarıçapı 0,5 cm, robotun yarıçapı 0,3 cm ve güvenlik mesafesi 0,2 cm'dir.



Şekil 3. Engelden kaçınma kontrolünün temsili çizimi (Representative drawing of obstacle avoidance control)

Öncelikle önerilen modelin detaylı analizi gerçekleştirilmiştir. Bu analiz sonucunda yol planlama algoritmalarının yüksek çalışma hızlarına tekabül eden

optimum kümeleme oranları elde edilmiştir. Daha sonra detaylı analizi desteklemek amacıyla model üzerinde farklı metasezgisel ve kümeleme algoritmalarının performansları değerlendirilmiştir.

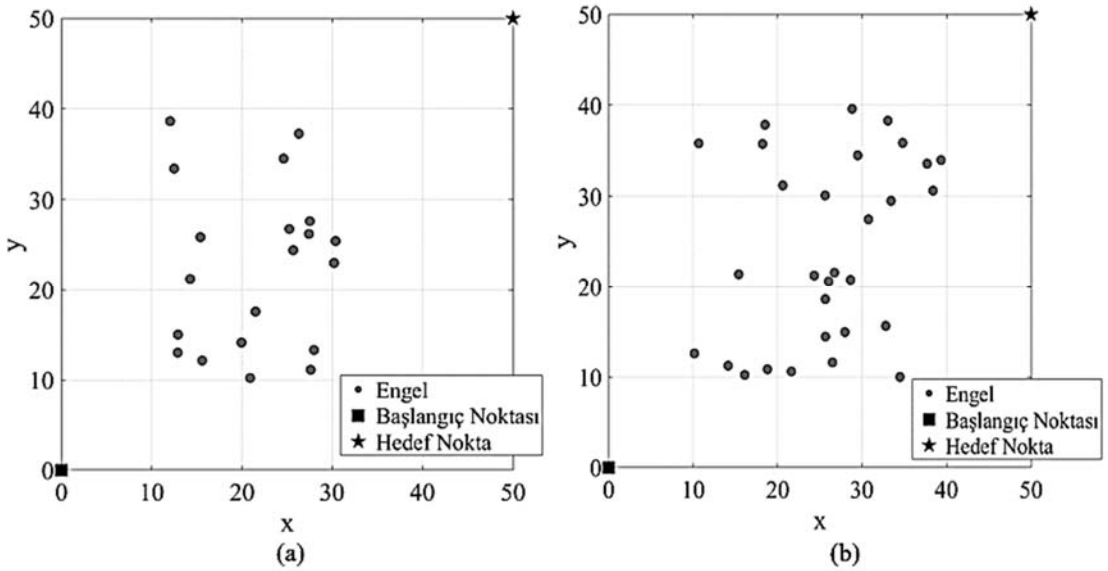
3.1. Önerilen Modelin Detaylı Analizi
(Detailed Analysis of the Proposed Model)

Detaylı analiz için metasezgisel algoritmalardan PSO ve kümeleme yöntemlerinden k-ortalamalar kümeleme algoritması tercih edilmiştir. PSO algoritması, maksimum iterasyon sayısı 50, popülasyon boyutu 20 için çalıştırılmıştır. Öğrenme katsayıları [2, 2], eylemsizlik ağırlığının başlangıç değeri 1, eylemsizlik ağırlığının sınır değerleri [0,1 0,9] olarak belirlenmiştir. PSO algoritmasındaki rasgele sayılar r_1 ve r_2 her iterasyonda farklı üretilir. Parametre sayısı 2, 3, 4, 5, 6 ve 8 için ayrı ayrı denenmiş ve en uygun değerinin 2 olduğu gözlemlenmiştir. Bu sebeple bu çalışmadaki parametre sayısı 2 olarak belirlenmiştir.

Model, engellerin rastgele konumlarda üretildiği ortamlarda değerlendirmeye alınmıştır. Bunun için 20 ve 30 engelli olmak üzere iki farklı ortam oluşturulmuştur. Ancak aynı alan içerisinde engel sayısı arttıkça modelin problemi çözmesi zorlaşmaktadır. Bu çalışmada ek olarak daha fazla engelli ortamlar da oluşturulmuş, ancak 30 engelin üzerindeki ortamlar için orta ve yüksek seviyelerdeki kümeleme oranlarında alan darlığından dolayı model problemi çözememektedir. Bunun için alanın genişletilmesi gerekir, ancak bu çalışmada model sabit alan üzerinde değerlendirmeye alınmıştır. 20 engelin altındaki ortamlar ise değerlendirme için anlamlı olamamaktadır. Bu çalışmada kullanılan bu iki ortam Şekil 4'te gösterilmektedir. Değerlendirme için model kümelemesiz ve kümelemeli olarak çalıştırılmıştır. Kümelemeli çalışmada 9 farklı kümeleme oranı kullanılmıştır. Ayrıca model çalışma süresi bakımından, engel kümeleme algoritmasının (EKA) çalışma süresinin dâhil olduğu ve olmadığı durumlar için ayrı ayrı değerlendirmeye alınmıştır. Değerlendirme sonucunda genel olarak en kısa mesafede küçük artışlar görülürken, çalışma

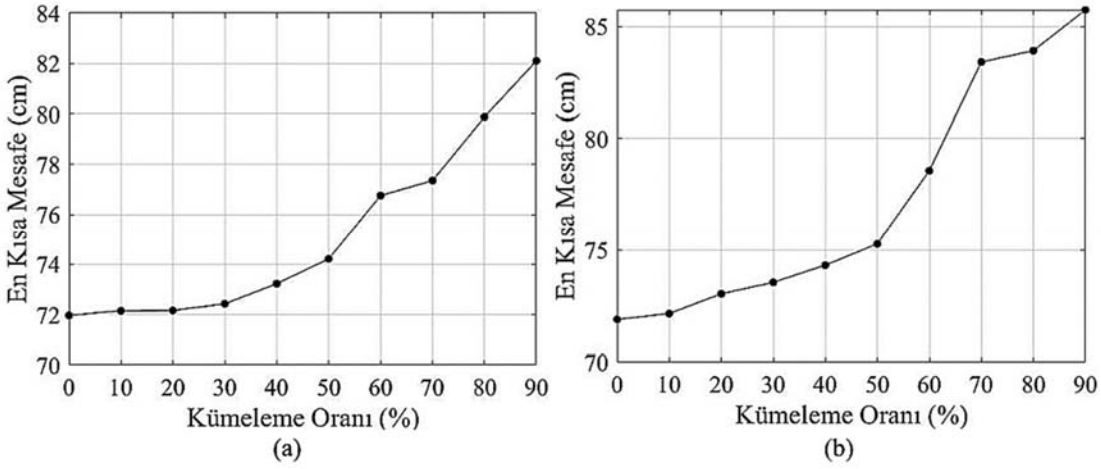
Tablo 4. Her iki ortam için elde edilen ortalama en kısa mesafeler ve bu mesafelerin kümelemesiz çalışmaya göre artış oranları
(The average shortest distances obtained for both environments and the rates of increase of these distances compared to non-clustering operation)

		20 Engelli Ortam		30 Engelli Ortam	
	Kümeleme Oranı	En Kısa Mesafe (cm)	Artış Oranı (%)	En Kısa Mesafe (cm)	Artış Oranı (%)
Kümelemesiz	-	71,9677	-	71,8987	-
Kümelemeli	%10	72,1500	0,25	72,1599	0,36
	%20	72,1635	0,27	73,0464	1,59
	%30	72,4269	0,63	73,5610	2,31
	%40	73,2285	1,75	74,3283	3,37
	%50	74,2230	3,13	75,2897	4,71
	%60	76,7568	6,65	78,5742	9,28
	%70	77,3544	7,48	83,4202	16,02
	%80	79,8786	10,99	83,9273	16,72
	%90	82,1017	14,08	85,7522	19,26

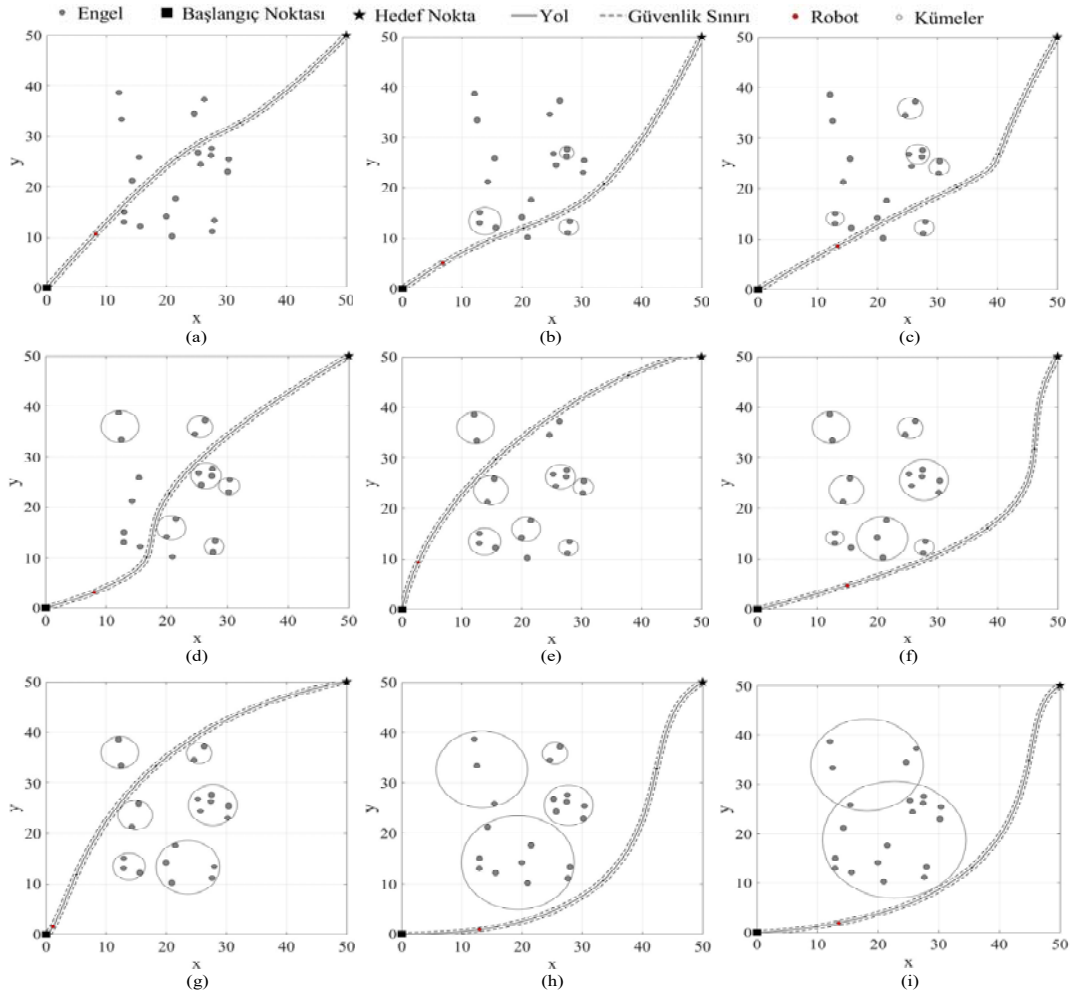


Şekil 4. Modelin test edildiği ortamlar: (a) 20 engelli ortam, (b) 30 engelli ortam
(The environments where the model was evaluated: (a) the environment with 20 obstacles, (b) the environment with 30 obstacles)

sürelerinde kayda değer azalmalar tespit edilmiştir. Her iki ortam için elde edilen ortalama en kısa mesafeler ve bu mesafelerin kümelemesiz çalışmaya göre artış oranları Tablo 4'te gösterilmektedir. Tablo 4'teki bulgular 30 koşmanın ortalamasıdır. Şekil 5, Tablo 4'teki en kısa mesafe değerlerini grafiksel olarak göstermektedir. Şekil 5'te x ekseninde gösterilen kümeleme oranlarındaki 0 değerleri kümelemesiz çalışmayı temsil etmektedir. Tablo 4 ve Şekil 5 genel olarak incelendiğinde, kümeleme oranı arttıkça elde edilen en kısa mesafelerde artış görülmektedir. Bunun nedeni, kümeleme oranının artmasıyla yeni ortamdaki engellerin boyutlarının artması ve bundan dolayı yolların daha kavisli olmasıdır. Ancak Tablo 4'teki artış oranları göz önüne alındığında, yüksek kümeleme oranları dışında telafi veya ihmal edilebilir bir artış mevcuttur. 20 engelli ortam için yüksek kümeleme oranları %80 ve %90 olarak kabul edilirse, diğer 7 kümeleme oranında artış miktarı %10'u geçmemektedir. 30 engelli ortam için yüksek kümeleme oranları %70, %80 ve %90 olarak kabul edilirse, diğer 6 kümeleme oranında artış miktarı yine %10'u geçmemektedir. Çalışmadan örnek bir koşma için 20 engelli ortamda modelin kümelemesiz ve kümelemeli olarak planladığı yollar Şekil 6'da, bu örnek koşma için yakınsama eğrileri Şekil 7'de gösterilmektedir. Şekil 6 genel olarak incelendiğinde, kümeleme oranı arttıkça yolların daha kavisli bir şekilde elde edildiği görülmektedir. Ayrıca Şekil 7'de görüldüğü gibi kümeleme oranı arttıkça elde edilen en kısa mesafelerin optimum değerlerinde artış görülmektedir. Ancak bu artışlar yüzdesel olarak düşük seviyelerdedir. Bazı kümeleme oranlarındaki yakınsama eğrilerinde çakışmalar tespit edildiği için bunların bir kısmı gösterilmemiştir. Örnek bir koşma için 30 engelli ortamda modelin kümelemesiz ve kümelemeli olarak planladığı yollar Şekil 8'de, bu örnek koşma için yakınsama eğrileri Şekil 9'da gösterilmektedir. Şekil 8 genel olarak incelendiğinde, kümeleme oranı arttıkça yolların 20 engelli ortama göre çok daha kavisli bir şekilde elde edildiği görülmektedir. Ayrıca Şekil 9'da görüldüğü gibi kümeleme oranı arttıkça elde edilen en kısa mesafelerin optimum değerlerinde yine artış görülmektedir ve bu artış 20 engelli ortama göre daha fazladır. Ancak bu artışlar da yüzdesel olarak düşük seviyelerdedir. Her iki ortam için elde edilen ortalama çalışma süreleri ve bu sürelerin kümelemesiz çalışmaya göre azalma oranları Tablo 5'te gösterilmektedir. Tablo 5'teki bulgular 30 koşmanın ortalamasıdır. Şekil 10, Tablo 5'teki çalışma süresi değerlerini grafiksel olarak göstermektedir. Şekil 10'da x ekseninde gösterilen kümeleme oranlarındaki 0 değerleri kümelemesiz çalışmayı temsil etmektedir. Tablo 5 ve Şekil 10 genel olarak incelendiğinde, kümeleme oranı arttıkça elde edilen çalışma sürelerinde azalma görülmektedir. Bunun nedeni, kümeleme oranının artmasıyla engel sayısının ve ortam karmaşıklığının azalmasıdır. Ayrıca Tablo 5'teki azalma oranları göz önüne alındığında kayda değer azalma oranları elde edilmiştir ve bu oranlarda süreklilik mevcuttur. 20 engelli ortamdaki azalma oranları 30 engelli ortamdaki oranlara göre daha yüksektir, çünkü ortam karmaşıklığı daha düşük seviyededir. Kümelemeli çalışmalarda EKA'nın dâhil edilmesi çalışma süresini sadece milisaniyelik düzeyde artmasına sebep olduğu için azalma oranları yine kayda değer seviyede kalmaktadır. Çalışmadaki değerlendirme, en kısa mesafeler ile çalışma süreleri arasındaki ilişki üzerinedir. Bu ilişki çalışma sürelerindeki azalma ile en kısa mesafedeki artış arasındaki farktır, kazanç olarak da düşünülebilir. Bu kazanç tanımında en kısa mesafe ve çalışma süresi eşit ağırlığa sahiptir ve kazanç oranları bu şekilde hesaplanmıştır. Her iki ortam için kümeleme oranlarına göre kazanç oranları Tablo 6'da gösterilmektedir. Tablo 6 genel olarak incelendiğinde, kümeleme oranı arttıkça kazanç oranlarında önce bir artış ve daha sonra bir azalma görülmektedir. Orta seviyelerdeki kümeleme oranlarında maksimum kazanç elde edilirken, daha düşük veya daha yüksek kümeleme oranlarında bu kazanç oranları düşmektedir. 20 engelli ortamda %70 kümeleme oranında maksimum kazanç değeri elde edilirken, 30 engelli ortam için bu oran %60'tır. EKA'nın dâhil edilmesi çalışma süresini etkilediği gibi kazanç oranlarını da sadece minimal düzeyde etkilemektedir. Ayrıca model farklı ortamlarda test edilmesine rağmen sonuç değişmemiş ve optimum kümeleme oranının orta seviyelerde olduğu tespit edilmiştir. Engelleri kümelemek en kısa mesafe açısından küçük bir dezavantaj gibi görünse de, çalışma süresi



Şekil 5. Her kümeleme oranı için elde edilen ortalama en kısa mesafeler: (a) 20 engelli ortam, (b) 30 engelli ortam
(Average shortest distances obtained for each clustering ratio: (a) the environment with 20 obstacles, (b) the environment with 30 obstacles)

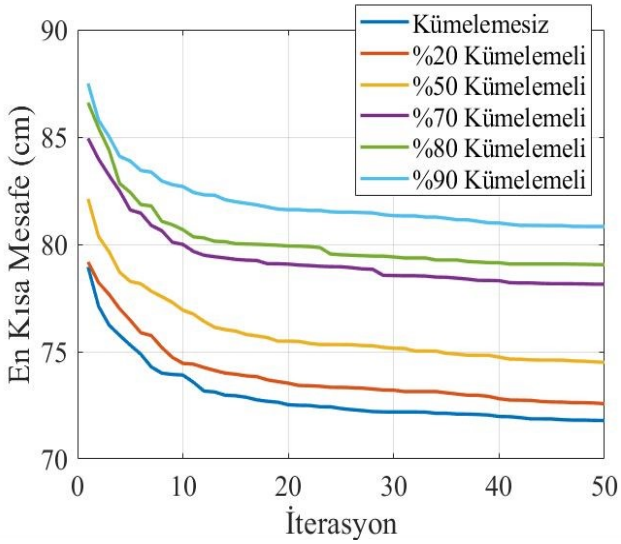


Şekil 6. Örnek koşma için 20 engelli ortamda modelin kümelemesiz ve kümelemeli olarak planladığı yollar: (a) kümelemesiz, (b) %20 kümelemeli, (c) %30 kümelemeli, (d) %40 kümelemeli, (e) %50 kümelemeli, (f) %60 kümelemeli, (g) %70 kümelemeli, (h) %80 kümelemeli, (i) %90 kümelemeli

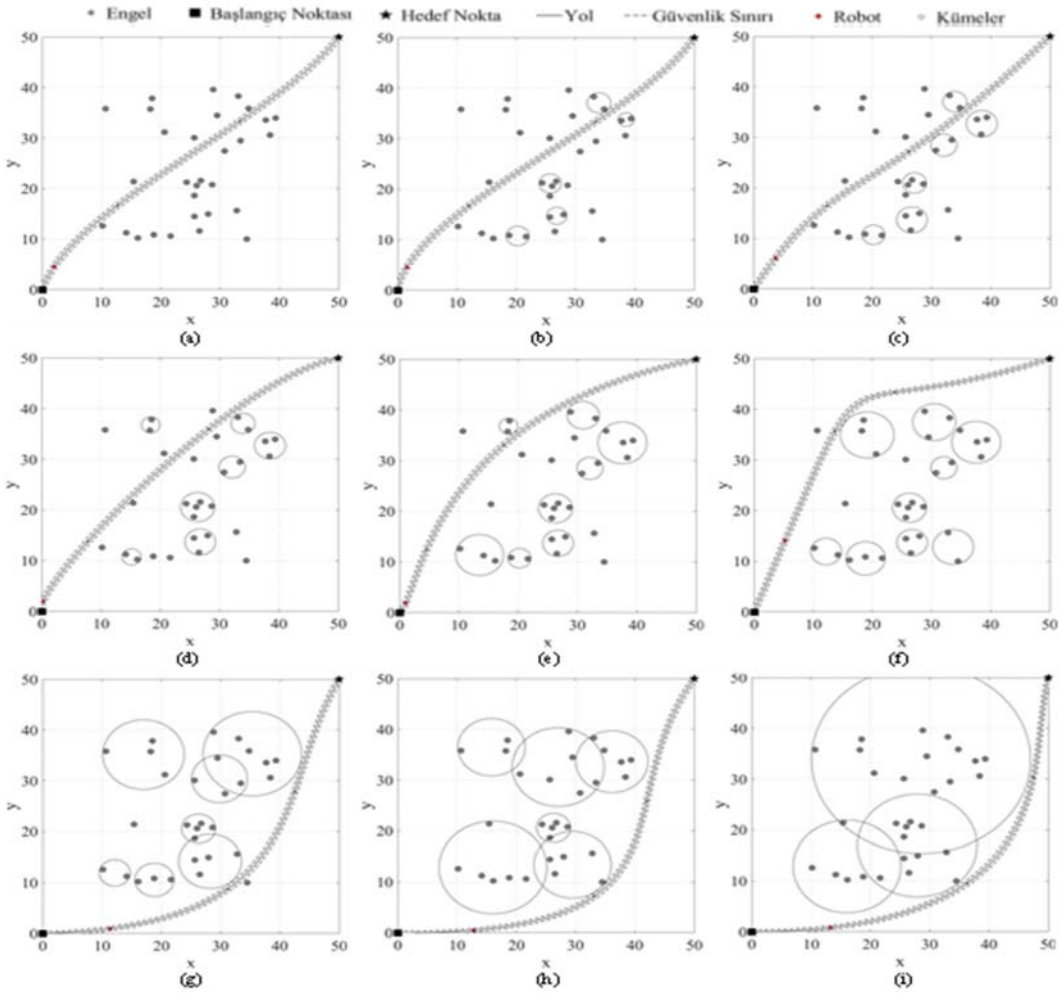
açısından en kısa mesafedeki zararı telafi edebilecek ve buna ek olarak hız konusunda kazanç sağlayacak düzeyde bir avantaj sağlamaktadır.

3.2. Önerilen Modelin Farklı Metasezgisel ve Kümeleme Algoritmalarla Gerçekleştirimi (Implementation of the Proposed Model with Different Metaheuristic and Clustering Algorithms)

Önerilen modelin detaylı analizinde optimum kümeleme oranları elde edilmişti. Modelin etkinliğini göstermek ve analizi desteklemek amacıyla, bu kümeleme oranları ile farklı metasezgisel ve kümeleme algoritmalarının performansları karşılaştırmalı değerlendirilmiştir. Engellerin kümelenmesi için k-ortalamalar yöntemine ek olarak hiyerarşik kümeleme algoritması kullanılmıştır. Metasezgisel algoritmalardan da PSO'ya ek olarak TLBO [26], ABC [27], DE [28] ve GA [29] algoritmaları ele alınmıştır. Bu algoritmaların kontrol parametreleri Tablo 7'de gösterilmektedir. Bu parametrelere ek olarak DE algoritmasında ölçekleme faktörü sabit bir değer yerine belli bir aralıkta her iterasyonda değişen rastgele bir değer olarak belirlenmiştir.

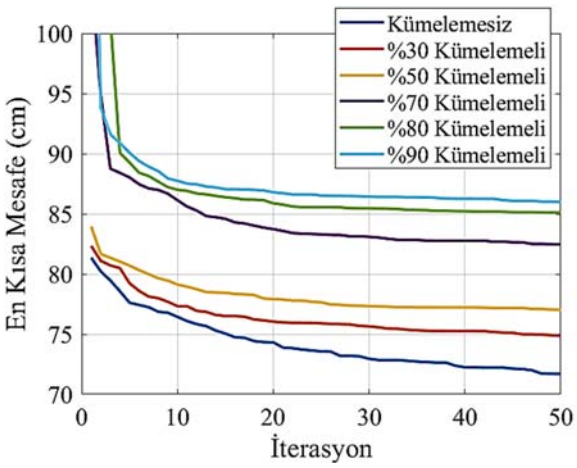


Şekil 7. Örnek koşma için 20 engelli ortamdaki yakınsama eğrileri (Convergence curves in the environment with 20 obstacles for sample running)



Şekil 8. Örnek koşma için 30 engelli ortamda modelin kümelemesiz ve kümelemeli olarak planladığı yollar: (a) kümelemesiz, (b) %20 kümelemeli, (c) %30 kümelemeli, (d) %40 kümelemeli, (e) %50 kümelemeli, (f) %60 kümelemeli, (g) %70 kümelemeli, (h) %80 kümelemeli, (i) %90 kümelemeli

(The paths that the model planned for the environment with 30 obstacles for sample running: (a) non-clustered, (b) 20% clustered, (c) 30% clustered, (d) 40% clustered, (e) 50% clustered, (f) 60% clustered, (g) 70% clustered, (h) 80% clustered, (i) 90% clustered)



Şekil 9. Örnek koşma için 30 engelli ortamdaki yakınsama eğrileri (Convergence curves in the environment with 30 obstacles for sample running)

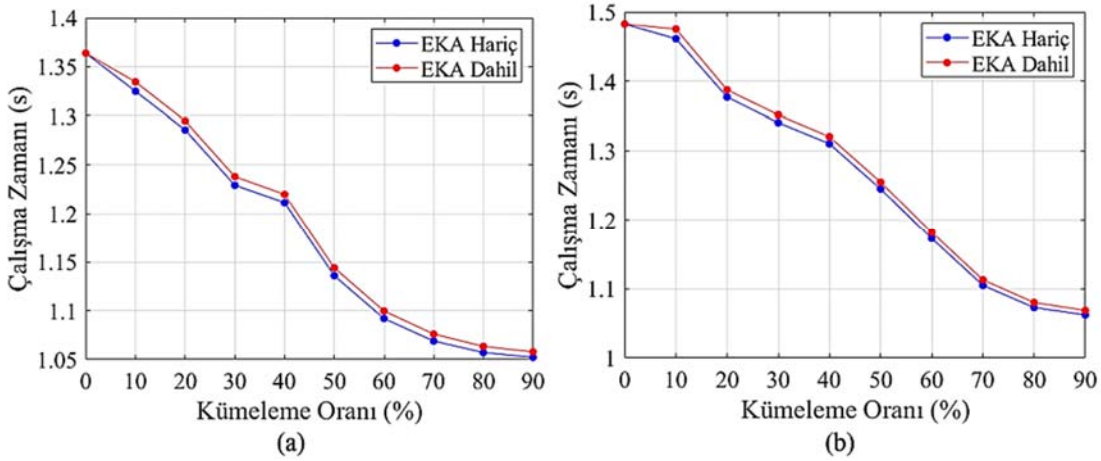
Önerilen modelde bu algoritmalar her iki kümeleme algoritması için ayrı ayrı çalıştırılmıştır. Her iki ortam için optimum kümeleme oranlarında bu algoritmalar tarafından elde edilen en kısa mesafe ve EKA'nın dâhil olduğu çalışma süreleri Tablo 8'de gösterilmektedir.

Tablo 8 kümeleme algoritmaları açısından incelendiğinde, her iki ortam için hiyerarşik kümeleme algoritmasının k-ortalamalar algoritmasına göre hem en kısa mesafe hem de çalışma süresi açısında avantajlı olduğu söylenebilir. Çalışmada ayrıca, hiyerarşik ve k-ortalamalar kümeleme algoritmalarının engellerin kümelenmesi için harcadığı süreler sırasıyla yaklaşık 6 ve 20 milisaniye olarak tespit edilmiştir. Bu değerler de bu avantajı desteklemektedir.

Tablo 8 metasezgisel algoritmalar için en kısa mesafe açısından incelendiğinde, her iki ortam için en iyi performansı TLBO ve ABC algoritmaları göstermiştir. DE,

Tablo 5. Her iki ortam için elde edilen ortalama çalışma süreleri ve bu sürelerin kümelemesiz çalışmaya göre azalma oranları
(The average running times obtained for both environments and the reduction rates of these times compared to non-clustering operation)

		20 Engelli Ortam				30 Engelli Ortam			
		EKA Hariç		EKA Dâhil		EKA Hariç		EKA Dâhil	
	Kümeleme Oranı	Çalışma Süresi (s)	Çalışma Süresi Azalma Oranı (%)	Çalışma Süresi (s)	Çalışma Süresi Azalma Oranı (%)	Çalışma Süresi (s)	Çalışma Süresi Azalma Oranı (%)	Çalışma Süresi (s)	Çalışma Süresi Azalma Oranı (%)
Kümelemesiz	-	1,3637	-	1,3637	-	1,4824	-	1,4824	-
Kümelemeli	%10	1,3248	2,85	1,3344	2,14	1,4613	1,42	1,4753	0,47
	%20	1,2857	5,17	1,2948	5,05	1,3771	7,10	1,3875	6,40
	%30	1,2292	9,86	1,2380	9,21	1,3407	9,55	1,3517	8,81
	%40	1,2114	11,16	1,2198	10,55	1,3104	11,60	1,3205	10,92
	%50	1,1360	16,69	1,1441	16,10	1,2451	16,00	1,2549	15,34
	%60	1,0921	19,91	1,0998	19,35	1,1732	20,85	1,1821	20,25
	%70	1,0690	21,61	1,0762	21,08	1,1050	25,45	1,1130	24,91
	%80	1,0572	22,47	1,0635	22,01	1,0729	27,62	1,0802	27,13
	%90	1,0523	22,83	1,0579	22,42	1,0623	28,33	1,0687	27,90



Şekil 10. Her kümeleme oranı için elde edilen ortalama çalışma süreleri: (a) 20 engelli ortam, (b) 30 engelli ortam.
(Average running times obtained for each clustering ratio: (a) the environment with 20 obstacles, (b) the environment with 30 obstacles)

Tablo 6. Her iki ortam için kümeleme oranlarına göre kazanç oranları (Gain rates relative to clustering rates for both environments)

	Kazanç Oranları (%)			
	20 Engelli Ortam		30 Engelli Ortam	
Kümeleme Oranı	EKA Hariç	EKA Dâhil	EKA Hariç	EKA Dâhil
%10	2,60	1,89	1,06	0,11
%20	4,90	4,78	5,51	4,81
%30	9,23	8,58	7,24	6,50
%40	9,41	8,80	8,23	7,55
%50	13,56	12,97	11,29	10,63
%60	13,26	12,70	11,57	10,97
%70	14,13	13,60	9,43	8,89
%80	11,48	11,02	10,90	10,41
%90	8,75	8,34	9,07	8,64

PSO ve GA algoritmaları daha verimsiz çalışırken, bunlar arasında en kötü performans DE algoritması ile alınmıştır. Çalışma süresi açısından incelendiğinde ise her iki ortam için en hızlı algoritma GA olmuştur.

Tablo 7. TLBO, ABC, DE ve GA'nın kontrol parametreleri (Control parameters of TLBO, ABC, DE and GA)

Kontrol Parametreleri	TLBO	ABC	DE	GA
Maksimum İterasyon Sayısı	50	50	50	50
Popülasyon Boyutu	20	20	20	20
Limit Değeri (ABC)	-	0.5	-	-
Ölçekleme Faktörü (DE)	-	-	[0,2 - 0,8]	-
Çaprazlama Oranı (DE, GA)	-	-	0,2	0,98
Mutasyon Oranı (GA)	-	-	-	0,1

Tablo 8. PSO, TLBO, ABC, DE ve GA algoritmaları ile elde edilen ortalama en kısa mesafe ve çalışma süreleri (Bu bulgular 30 koşmanın ortalamasıdır)
(The average shortest distance and the running times obtained by the PSO, TLBO, ABC, DE and GA algorithms (These results are the average of 30 runs))

	20 Engelli Ortam (%70 Kümeleme Oranı)				30 Engelli Ortam (%60 Kümeleme Oranı)			
	K-Ortalamalar Kümeleme		Hiyerarşik Kümeleme		K-Ortalamalar Kümeleme		Hiyerarşik Kümeleme	
Algoritma	En Kısa Mesafe (cm)	Çalışma Süresi (s) (EKA Dâhil)	En Kısa Mesafe (cm)	Çalışma Süresi (s) (EKA Dâhil)	En Kısa Mesafe (cm)	Çalışma Süresi (s) (EKA Dâhil)	En Kısa Mesafe (cm)	Çalışma Süresi (s) (EKA Dâhil)
PSO	77,3544	1,0762	77,0401	1,1258	78,5742	1,1821	75,4592	1,1532
TLBO	75,4902	1,6022	75,2253	1,5952	74,9568	1,6464	74,2559	1,6125
ABC	75,2478	1,6153	75,2089	1,6094	74,5329	1,7203	74,0870	1,6509
DE	87,3753	1,1767	87,3504	1,1548	90,5944	1,3045	85,7691	1,2823
GA	79,5047	1,0837	80,1133	1,0701	81,9372	1,1468	79,5148	1,0830

5. SONUÇLAR (CONCLUSIONS)

Bu çalışmada statik, iki boyutlu ve engellerin rastgele konumlarda üretildiği ortamlarda ortam karmaşıklığının azaltılması ve bu sayede yol planlama algoritmalarının çalışma hızlarının artırılması amacıyla, metasezgisel ve kümeleme algoritmalarının bir arada kullanıldığı hibrit bir model önerilmiştir. Bu modelde ortam karmaşıklığı engellerin kümelenmesiyle azaltılmıştır. Önerilen model için önce detaylı bir analiz gerçekleştirilmiştir. Bu analizde, çeşitli kümeleme oranları ve iki farklı ortam kullanılarak yüksek çalışma hızlarına tekabül eden optimal kümeleme oranları tespit edilmiştir. Bu analizin sonucunda her iki ortam için, en kısa mesafe bakımından küçük seviyelerde kayıplar gözlemlenmiştir. Ancak çalışma süreleri incelendiğinde, algoritmanın bu kayıpları fazlasıyla telafi ettiği ve çalışma hızı bakımından kazanç elde edildiği söylenebilir. Ayrıca farklı ortamlarda algoritmaların test edilmesi sonucu değiştirmemiştir. Önerilen modelde daha sonra PSO'ya ek olarak TLBO, ABC, DE, GA algoritmaları ve k-ortalamalar kümeleme yöntemine ek olarak hiyerarşik kümeleme algoritması kullanılmış, hem metasezgisel hem de kümeleme algoritmaları açısından değerlendirme yapılmıştır. Bu değerlendirme sonucunda en kısa mesafe açısından en iyi performansı TLBO ve ABC algoritmaları gösterirken en hızlı algoritma GA olmuştur. Ayrıca hiyerarşik kümeleme algoritmasının k-ortalamalar yöntemine göre daha verimli çalıştığı söylenebilir. Sonuç olarak yol planlama algoritmalarının çalışma hızları uygun bir kümeleme oranının seçilerek, ortamın basitleştirilmesiyle başarılı bir şekilde iyileştirilebileceği görülmüştür.
Sonraki çalışmalarda, daha yüksek kazanç oranları elde etmek için farklı metasezgisel algoritmaların geliştirilmiş modelleri önerilebilir. Önerilen modele k-ortalamalar veya hiyerarşik kümeleme algoritmalarından farklı kümeleme yöntemleri dâhil edilebilir.

KAYNAKLAR (REFERENCES)

1. Chen J.L., Liu J.S., Lee W.C., A recursive algorithm for on-line clustering obstacles cluttered in dynamic environments, Journal of Intelligent and Robotic Systems, 33, 209-230, 2002.
2. Ajeil F.H., Ibraheem I.K., Azar A.T., Humaidi A.J., Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment, International Journal of Advanced Robotic Systems, 17 (3), 1-15, 2020.
3. Ajeil F.H., Ibraheem I.K., Azar A.T., Humaidi A.J., Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments, Sensors, 20 (7), 1-26, 2020.
4. Kim C., Kim Y., Yi H., Fuzzy analytic hierarchy process-based mobile robot path planning, Electronics, 9 (2), 1-18, 2020.
5. Kim C., Suh J., Han J.H., Development of a hybrid path planning algorithm and a bio-ınspired control for an omni-wheel mobile robot, Sensors, 20 (15), 1-22, 2020.

6. Li X., Zhao G., Li B., Generating optimal path by level set approach for a mobile robot moving in static/dynamic environments, Applied Mathematical Modelling, 85, 210-230, 2020.

7. Dirik M., Kocamaz A.F., Castillo O., Global path planning and path-following for wheeled mobile robot using a novel control structure based on a vision sensor, International Journal of Fuzzy Systems, 22 (6), 1880-1891, 2020.

8. Zhong X., Tian J., Hu H., Peng X., Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment, Journal of Intelligent & Robotic Systems, 99, 65-77, 2020.

9. Jung J.H., Kim D.H., Local path planning of a mobile robot using a novel grid-based potential method, International Journal of Fuzzy Logic and Intelligent Systems, 20 (1), 26-34, 2020.

10. Wu M., Dai S. L., Yang C., Mixed reality enhanced user ınteractive path planning for omnidirectional mobile robot, Applied Sciences, 10 (3), 1-16, 2020.

11. Arango G.D., Leal H.V., Martinez L.H., Fernandez V.M.J., Jimenez A.H., Ambrosio R.C., Chua J.S., Cholula H.D.C., Mendez S.H., Multiple-target homotopic quasi-complete path planning method for mobile robot using a piecewise linear approach, Sensors, 20 (11), 1-47, 2020.

12. Wang X., Mizukami Y., Tada M., Matsuno F., Navigation of a mobile robot in a dynamic environment using a point cloud map, Artificial Life and Robotics, 1-11, 2020.

13. Elmi Z., Efe M.Ö., Yang C., Online path planning of mobile robot using grasshopper algorithm in a dynamic and unknown environment, Journal of Experimental & Theoretical Artificial Intelligence, 1-19, 2020.

14. Zhang L., Zhang Y., Li Y., Path planning for indoor mobile robot based on deep learning, Optik, 209, 1-17, 2020.

15. Li Y., Huang Z., Xie Y., Path planning of mobile robot based on improved genetic algorithm, 2020 3rd International Conference on Electron Device and Mechanical Engineering (ICEDME), Suzhou, China, 691-695, 1-3 Mayıs, 2020.

16. Ali H., Gong D., Wang M., Dai X., Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment, Frontiers in Neurorobotics, 14, 1-13, 2020.

17. Kıvanç Ö.C., Mungan T.E., Atila B., Tosun G., An integrated approach to development of unmanned ground vehicle: design, analysis, implementation and suggestions, Journal of the Faculty of Engineering and Architecture of Gazi University, 34 (4), 1957-1973, 2019.

18. Yılmaz N., Gencer C.T., Integration of sensor vision capabilities on UAV flight route optimization: A linear model and a heuristic algorithm proposal, Journal of the Faculty of Engineering and Architecture of Gazi University, 34 (4), 1917-1928, 2019.

19. Zhao F., Hung D. L. S., Wu S., K-means clustering-driven detection of time-resolved vortex patterns and cyclic variations inside a direct injection engine, Applied Thermal Engineering, 180, 1-13, 2020.

20. Aytaç E., Unsupervised learning approach in defining the similarity of catchments: Hydrological response unit based k-means clustering, a demonstration on Western Black Sea Region of Turkey, International Soil and Water Conservation Research, 8, 321-331, 2020.

21. Rehman T.U., Mahmud Md.S., Chang Y.K., Jin J., Shin J., Current and future applications of statistical machine learning algorithms for agricultural machine vision systems, Computers and Electronics in Agriculture, 156, 585-605, 2019.

22. Tang Y., Zhou R., Sun G., Di B., Xiong R., A novel cooperative path planning for multirobot persistent coverage in complex environments, IEEE Sensors Journal, 20 (8), 4485-4495, 2020.

23. Peng Y., Qu D., Zhong Y., Xie S., Luo J., The obstacle detection and obstacle avoidance algorithm based on 2-D lidar, International Conference on Information and Automation, Lijiang, China, 1648-1653, 8-10 Ağustos, 2015.

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

25. E. Chołodowicz, D. Figurowski, Mobile robot path planning with obstacle avoidance using particle swarm optimization, Pomiary Automatyka Robotyka, 21 (3), 59-68, 2017.

26. J. Hernandez-Barragan, Mobile robot path planning based on conformal geometric algebra and teaching-learning based optimization, IFAC-PapersOnLine, 51 (13), 338-343, 2018.

27. M. A. Contreras-Cruz, V. Ayala-Ramirez, U. H. Hernandez-Belmonte, Mobile robot path planning using artificial bee colony and evolutionary programming, Applied Soft Computing, 30, 319-328, 2015.

28. A. Zamuda, J. D. H. Sosa, Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures, Applied Soft Computing, 24, 95-108, 2014.

29. C. Lamini, S. Benhlima, A. Elbekri, Genetic algorithm based approach for autonomous mobile robot path planning, Procedia Computer Science, 127, 180-189, 2018.