www.iners.duzce.edu.tr

5. Uluslararası Mühendislik Araştırmaları Sempozyumu
07-09 Mart 2024, Düzce/Türkiye

5th International Engineering Research Symposium
07-09 March 2024, Düzce/Türkiye
ISBN: 978-625-6330-05-4

Otonom Mobil Robotlar İçin Metasezgisel Yöntemler Kullanılarak Yol Planlama Algoritmasının Geliştirilmesi Ve Performanslarının Karşılaştırılması

Yunus Tezel[a], Dr. Suat Karakaya[b]*

[a]Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Bölümü, Kocaeli/Türkiye.
[b]Kocaeli Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü, Kocaeli/Türkiye.
[a*]**Sorumlu Yazar:** yunustezel@gmail.com

[b]suatronik@gmail.com

ÖZET

Son yıllarda mobil robot navigasyonunda kritik bir önem taşıyan yol planlama üzerine çalışmalar yapılmakta ve bu problemi çözmek için metasezgisel algoritmalar kullanılmaktadır. Mobil robotlar için yol planlama, belirli bir hedefe veya göreve ulaşmak amacıyla engellere çarpmadan en uygun rotayı belirleme sürecidir. Bu süreç mobil robotun bir başlangıç noktasından hedef noktaya en az maliyetle ulaşmasını amaçlayan bir optimizasyon problemidir. Yol planlama, metasezgisel algoritmalar kullanılarak gerçekleştirilebilir. Metasezgisel algoritmalar popülasyon tabanlı ve tek çözüm tabanlı olmak üzere ikiye ayrılır. Popülasyon tabanlı algoritmalar kendi içinde sürü tabanlı, fizik tabanlı, evrimsel tabanlı, insan tabanlı ve matematik tabanlı olmak üzere beş farklı alt kategoriye ayrılır. Bu çalışma kapsamında yol planlama için üç farklı metasezgisel algoritma kullanılacaktır. Bu algoritmalar; tek çözüm tabanlı olan Benzetimli Tavlama (Simulated Annealing, SA), popülasyon ve evrimsel tabanlı algoritma olan Genetik Algoritma (Genetic Algorithm, GA), popülasyon ve sürü tabanlı olan Guguk Kuşu Arama (Cuckoo Search, CS) algoritmalarıdır. Bu metasezgisel algoritmaların performanslarının karşılaştırılması için ızgara tabanlı farklı senaryolar içeren haritalar oluşturulmuştur. Böylece farklı yol planlama problemlerinde algoritmaların nasıl bir sonuç ürettiği gözlemlenmiştir. Gözlemlerin sonucunda, bu algoritmaların performansları oluşturulan farklı haritalarda test edilmiş ve karşılaştırmaları yapılmıştır.

Anahtar Kelimeler: *Metasezgisel Algoritmalar, Optimizasyon, Otonom Mobil Robotlar, Yol Planlama*

Development of Path Planning Algorithm For Autonomous Mobile Robots Using Metaheuristic Methods And Comparison Of Their Peformances

Yunus Tezel[a], Dr. Suat Karakaya[b]*

[a]Kocaeli University, Graduate School of Science, Department of Mechatronics Engineering, Kocaeli/Türkiye.
[b]Kocaeli University, Engineering Faculty, Department of Mechatronics Engineering, Kocaeli/Türkiye.
[a*]**Corresponding Author:** yunustezel@gmail.com

[b]suatronik@gmail.com

ABSTRACT

In recent years, studies have been carried out on path planning, which plays a critical role in mobile robot navigation, and metaheuristic algorithms are used to solve this problem. Path planning for mobile robots is the process of determining the optimal route to reach a specific goal or task without hitting any obstacles. This process is an optimization problem that aims to ensure that the mobile robot reaches the target point from a starting point at the least cost. Path planning can be performed using metaheuristic algorithms.

Metaheuristic algorithms are classified into two categories: population-based and single-solution-based. Population-based algorithms are divided into five different subcategories: swarm-based, physic-based, evolutionary-based, human-based and math-based. Within the scope of this study, three different metaheuristic algorithms will be used for path planning. There are Simulated Annealing (SA) which is a single-solution-based algorithm, Genetic Algorithm (GA) which is a population and evolutionary-based algorithm, and Cuckoo Search (CS) which is a population and swarm-based algorithm. Various grid-based maps with specialized scenarios were created to compare the performance of these metaheuristic algorithms. Thus, the kind of results the algorithms produced in different path planning problems was observed. As a result of the observations, the performances of these algorithms were tested and compared on the maps that were initially designed.

Keywords: *Autonomous Mobile Robots, Metaheuristic Algorithms, Optimization, Path Planning*

I. GİRİŞ

Otonom mobil robotların kullanımı Endüstri 4.0 ile birlikte önem kazanmıştır. Otonom mobil robotlar yük taşıma amacıyla fabrikalarda, depolarda ve ofis ortamlarında tercih edilmeye başlanmıştır [1]. Güvenlik ve çevresel faktörler göz önünde bulundurulduğunda mobil robotların tercihinin önemi artmaktadır. Son yıllarda mobil robotlar, önemli insansız görevleri yerine getirmek için askeri, endüstri ve güvenlik gibi farklı alanlarda başarıyla uygulanmıştır [2].

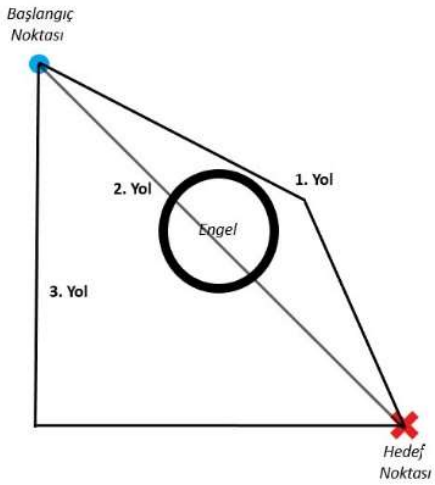
Otonom mobil robot navigasyonu, başlangıç noktasından hedef noktasına engellere çarpmadan ulaşma sürecidir. Bu süreç temel olarak dört bileşenden oluşur [3]:

- Algılama: Mobil robot üzerindeki sensörler aracılığı ile çevre bilgisini toplar.
- Konumlandırma: Mobil robot çalışma uzayındaki konumu tespit eder.
- Kavrama ve yol planlama: Mobil robotun hedef konumuna ulaşması için engellere çarpmadan izleyeceği yolu belirler.
- Hareket kontrol: Mobil robot planlanan yolu gerçekleştirmek için hareketini düzenler.

Otonom mobil robotların navigasyon süreci için gerekli olan bileşenlerden biri de yol planlamadır. 1960'lı yılların ortalarından itibaren yol planlama birçok bilim insanının ilgisini çekmiştir. Mobil robotlar için yapılan iyi bir yol planlama sadece zamandan tasarruf kazandırmaz ayrıca mobil robotun kullanım ömrünü de arttırır ve sermaye yatırımlarını azaltır [4]. Mobil robotlar için yol planlama, belirli bir hedefe veya göreve ulaşmak amacıyla engellere çarpmadan en uygun rotayı belirleme sürecidir. Bu süreç mobil robotun bir başlangıç noktasından hedef noktaya en az maliyetle ulaşmasını amaçlayan bir optimizasyon problemidir. Yol planlama, karmaşık iç mekan ortamlarında mobil bir robotun otonom bir şekilde gezinebilmesi ve keşif yapabilmesi için çözülmesi gereken en temel problemlerden biridir [5].

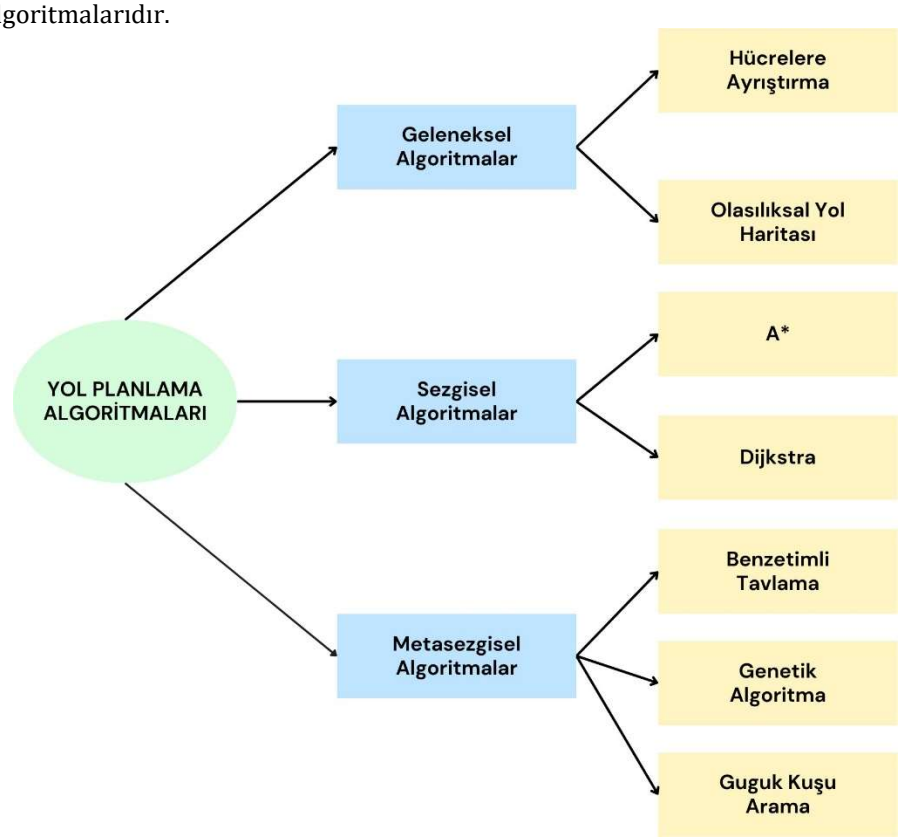
Yukarıda yapılan yol planlama tanımdan anlaşılacağı üzere yol planlama problemi temel olarak engellere çarpmama ve optimum yolu bulma kriterlerini içerir. Bu kriterlere ek olarak algoritmanın çalışma süresi, robotun hedefe ulaşma süresi, robotun harcadığı enerji gibi kriterler eklenebilir. Şekil 1'de bir yol planlama örneği gösterilmiştir. Şekilde üç farklı alternatif yol görülmektedir. 1. Yol'u inceleyecek olursak engele çarpma durumunun olmadığı ve yol uzunluğu açısından uygun bir yol olduğu görülmektedir. 2. Yol aralarındaki en kısa mesafeye sahip olan yol olmasına rağmen engele çarpma durumu olduğu için bu yolun uygun olmadığını söyleyebiliriz. 3. Yol'da ise engele çarpma durumu yoktur ve 1.Yol'la kıyaslandığında engelle arasında daha fazla mesafe bulunduğu için daha emniyetlidir ancak mesafe açısından 1. Yol'a göre 3. Yol daha fazla uzunluğa sahiptir. Buradan anlaşılacağı üzere engele çarpma olmaması bize uygun bir yol verir fakat yol planlama probleminde amacımız problemimize uygun en optimal yolu bulmaya çalışmaktır.

Genel olarak yol planlama, çevreyle ilgili tüm bilgilerin erişilebilir olup olmamasına göre iki kategoriye ayrılabilir. Küresel yol planlamada, çevreye ilişkin tüm bilgiler robot çalışmaya başlamadan önce robot tarafından bilinir. Yerel yol planlamasında ise çevreye ilişkin hemen hemen tüm bilgiler robot tarafından çalışmaya başlamadan önce bilinmemektedir [6]. Yol planlama yaparken çevre içerisinde genellikle iki tür engel vardır: Statik engeller, sabit konumda olan ve hareket etmeyen engellerdir. Statik engellere duvarlar, bariyerler, sütunlar örnek olarak verilebilir. Dinamik engeller, hareket edebilen ve beklenmedik şekilde konumları değişebilen engellerdir. Dinamik engellere insanlar, ortamdaki diğer mobil robotlar, araçlar örnek olarak verilebilir. Bu çalışmada ortamdaki engeller statik olarak kabul edilmiştir ve çevre bilgisi bilindiği için küresel yol planlama yapılmıştır.



Şekil 1 Yol Planlama Örneği

Yol planlama algoritmalarını Şekil 2'deki gibi geleneksel algoritmalar, sezgisel algoritmalar ve metasezgisel algoritmalar olmak üzere üç sınıfa ayırabiliriz [7]. Son yıllarda mobil robot navigasyonunda kritik bir önem taşıyan yol planlama üzerine çalışmalar yapılmakta ve bu problemi çözmek için metasezgisel algoritmalar kullanılmaktadır. Metasezgisel algoritmalar, insan zekasından veya doğadan ilham alır ve geniş anlamda bir stokastik optimizasyon algoritması olarak sınıflandırılabilir. Bu çalışma kapsamında yol planlama için üç farklı metasezgisel algoritma kullanılacaktır. Bu algoritmalar; tek çözüm tabanlı olan Benzetimli Tavlama (Simulated Annealing, SA), popülasyon ve evrimsel tabanlı algoritma olan Genetik Algoritma (Genetic Algorithm, GA), popülasyon ve sürü tabanlı olan Guguk Kuşu Arama (Cuckoo Search, CS) algoritmalarıdır.



Şekil 2 Yol Planlama Algoritmaları

Literatürde metasezgisel algoritmalarla yol planlama üzerine birçok çalışma mevcuttur. Gul ve ark. yol planlama için üç aşamadan oluşan bir yöntem önermişlerdir. Bu yöntem ilk aşamada Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO) ve Gri Kurt Optimizasyon (Grey Wolf Optimizer, GWO) algoritmalarını hibrit bir şekilde kullanarak en kısa yol mesafesini hesaplar, ikinci aşamada PSO-GWO tarafından üretilen uygun olmayan noktalar Yerel Arama (Local Search) yöntemiyle entegre edilerek uygun

noktalara dönüştürülür son aşamada ise engellerden kaçınma çalışması yapılır [8]. Mohanty ve Parhi statik engeller içeren, bilinmeyen veya kısmen bilinen ortamlarda otonom bir mobil robot için Guguk Kuşu Arama algoritmasını kullanarak bir metodoloji tanımlamışlardır. Önerilen algoritma simülasyon ortamında test edilmiş daha sonra doğrulamak için mobil robot üzerinde testler yapılmıştır [9]. Cong ve Ponnambalam, mobil robot yol planlama problemini çözmek için Karınca Kolonisi Optimizasyon (Ant Colony Optimization, ACO) algoritmasını önermişlerdir. Önerilen ACO ile yol planlama algoritmasını literatürden elde ettikleri haritalar üzerinde test etmişlerdir. Söz konusu makale ile kendi çalışmalarını karşılaştırmışlardır [10]. Amar ve Jasim, çeşitli statik ve dinamik engeller içeren bir ortamda otonom bir mobil robotun küresel yol planlama problemi için bir metodoloji geliştirmişlerdir. Parçacık Sürü Optimizasyonu ve Karınca Kolonisi Optimizasyon algoritmalarını kullanarak dinamik bir ortamta simülasyonlar gerçekleştirmişlerdir. Daha sonra PSO ve ACO algoritmalarının hibrit olarak kullanıldığı bir yöntem önermişlerdir. Algoritmaları karşılaştırdıklarında en optimal yol uzunluğuna ve en kısa çalışma süresine sahip olan algoritmanın PSO-ACO hibrit algoritması olduğunu elde etmişlerdir. Bununla birlikte PSO algoritması ACO algoritmasına kıyasla daha iyi performans göstermiştir [11]. Lin ve ark. depoda çalışacak bir mobil robotun yol planlaması için Benzetimli Tavlama algoritmasından ilham alarak PSO-SA hibrit algoritmasını geliştirmişlerdir. Geliştirdikleri algoritmayı, Parçacık Sürü Optimizasyonu, Benzetimli Tavlama, Harmoni Arama (Harmony Search, HS), Ateşböceği Algoritması (Firefly Algorithm, FA), Yapay Arı Kolonisi (Artifical Bee Colony, ABC) ve Genetik Algoritma gibi metasezgisel algoritmalarla karşılaştırmışlardır. PSO-SA hibrit algoritmasının ortalama iterasyon sayısı PSO, SA, HS, FA, ABC ve GA algoritmalarına göre sırası ile %3, %35, %40, %42, %36 ve %44 daha az olarak bulunmuştur. PSO-SA hibrit algoritmasının çalışması süresi her bir algoritma için yine aynı sıra ile %26, %64, %96, %100, %99 ve %99 daha az olarak belirlenmiştir. Sonuç olarak PSO-SA hibrit algoritmasının diğer algoritmalara kıyasla hızlı yakınsamaya, yüksek doğruluğa ve daha az çalışma süresine sahip olduğu tespit edilmiştir [12].

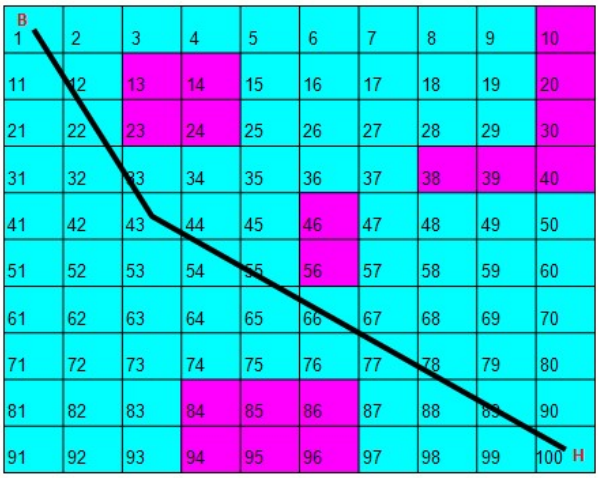
Çalışmanın bölümlerinin sıralaması şu şekildedir: İkinci bölümde materyal metotta, üçüncü bölümde sonuca, dördüncü bölümde tartışmaya ve beşinci bölümde çıkarıma yer verilmiştir.

II. MATERYAL METOT

Bu bölümde ilk önce ızgara tabanlı haritalardan bahsedilip problemin tanımı yapılacaktır. Daha sonra amaç fonksiyonu tanımlanacak ve son olarak çalışmada kullanılan metasezgisel algoritmalar açıklanacaktır.

2.1 Izgara Tabanlı Haritalar

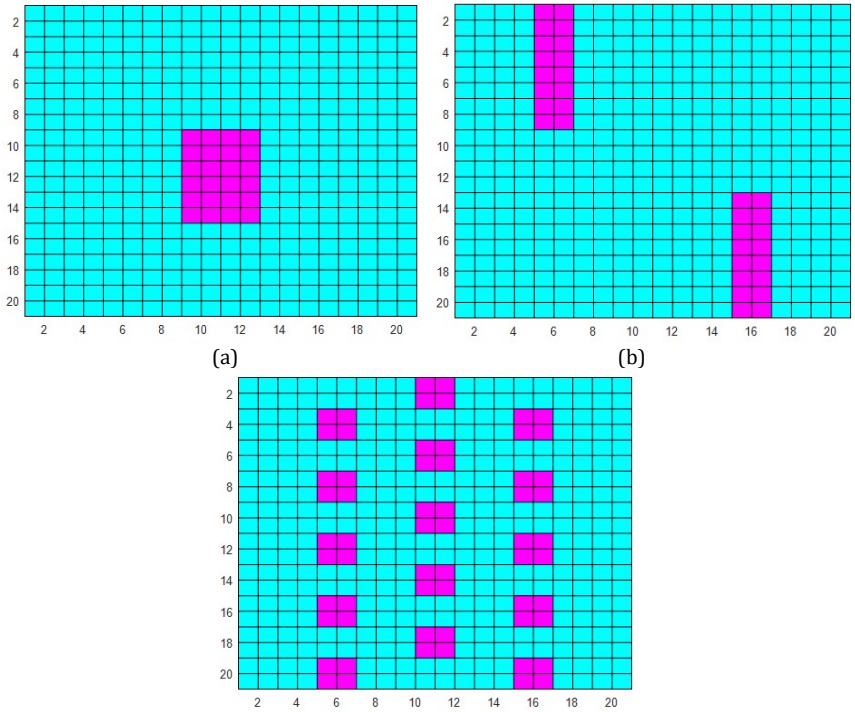
Literatürde nokta tabanlı haritalar, ağ tabanlı haritalar gibi farklı harita türleri bulunmaktadır. Bu çalışmada ızgara tabanlı harita türü kullanılmıştır. Izgara tabanlı haritalar, yol planlama probleminde sıklıkla kullanılan bir yaklaşımdır. Izgara tabanlı haritalar, belirli bir bölgeyi ızgara formunda temsil eder ve genellik 2 boyutlu bir düzlem üzerine yerleştirilir. Çalışma ortamı ızgara olarak adlandırılan belirli boyutlarda hücrelere bölünerek haritalandırma yapılır. Her bir hücre kendisi hakkında bilgiler içerir. Yol planlama problemi için bu bilgi genellikle engel olup olmama durumudur. Engelli bölgeler dolu alanları veya geçilemez bölgeleri temsil ederken, engelsiz bölgeler serbest geçiş alanlarını temsil eder. Şekil 3'te çalışmada kullanılan ızgara tabanlı harita örneği verilmiştir. Şekilde pembe renkli hücreler engelleri, açık mavi renkli hücreler ise serbest geçiş bölgelerini temsil etmektedir. Robotun başlangıç noktası "B" harfi ile işaretlenen hücre, hedef noktası ise "H" harfi ile işaretlenen hücredir. Yol planlama ile elde edilen yol siyah bir çizgi şeklinde gösterilmiştir.



Şekil 3 Izgara Tabanlı Harita Örneği

2.2 Problem Tanımı

Metasezgisel yol planlama algoritmalarını karşılaştırmak için daha önce üzerinde çalışılmış üç farklı tipte harita tasarımı seçilmiştir. Tüm haritalarda robot sol üst hücrede başlamakta ve hedef noktası olan sağ alt noktaya ulaşmayı hedeflemektedir. Şekil 4'te görüldüğü üzere haritalar farklı şekilde engeller içeren 20x20 birim boyutunda tasarlanmıştır. Burada ki amaç algoritmaların hangi harita türlerinde iyi sonuç verdiğini tespit etmektir.



Şekil 4 Izgara Tabanlı Haritalar: (a) Harita 1, (b) Harita 2, (c) Harita 3

2.3 Amaç Fonksiyonu

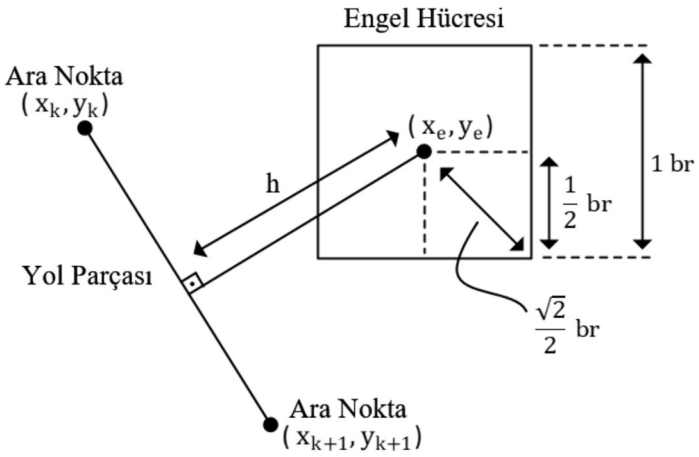
Yol planlama probleminin amacı robotun başlangıç pozisyonu ile hedef nokta arasındaki en uygun yolu bulmaktır. En uygun yol, yolculuk için en kısa, en az zaman alan ve en az enerji gerektiren yol olabilir [13]. Yol planlama probleminde genellikle amaç fonksiyonu engellere çarpmadan hedefe ulaşılabilecek en kısa yol olarak kabul edilir. Bu çalışmada kullanılan amaç fonksiyonu aşağıdaki denklemde verilmiştir:

$$f = \begin{cases} \sum_{i=1}^{n-1} d(p_i, p_{i+1}) + ceza, & engel\ varsa \\ \sum_{i=1}^{n-1} d(p_i, p_{i+1}), & aksi\ durumda \end{cases} \quad (1)$$

$$d(p_i, p_{i+1}) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (2)$$

Burada, f amaç fonksiyonu, p_i oluşturulan çözüm noktalarından i. nokta, n çözümün boyutu, d iki nokta arasındaki uzunluk, x_i ve y_i robotun mevcut konumunun yatay ve dikey pozisyonu, x_{i+1} ve y_{i+1} robotun bir sonraki konumunun yatay ve dikey pozisyonudur.

İki nokta arası her yol parçasının engelle kesişip kesişmediği kontrol edilir. Her yol parçasının kesiştiği engel sayısı kadar amaç fonksiyonuna ceza puanı eklenir. Engel tespiti için her yol parçasının, engelin bulunduğu hücrenin merkezine olan en kısa mesafesi hesaplanır [14]. Engel tespit yöntemi Şekil 5'te gösterildiği gibidir.



Şekil 5 Engel Tespiti [14]

Şekil 5'te görüldüğü üzere yolun engele olan en kısa mesafesini hesaplamak için bir doğrunun bir noktaya olan en kısa mesafe denkleminden yararlanılmıştır. Bu denklem Eşitlik 3'te verilmiştir.

$$h = \frac{|(x_{k+1}-x_k)(y_k-y_e)-(x_k-x_e)(y_{k+1}-y_k)|}{\sqrt{(x_{k+1}-x_k)^2+(y_{k+1}-y_k)^2}} \quad (3)$$

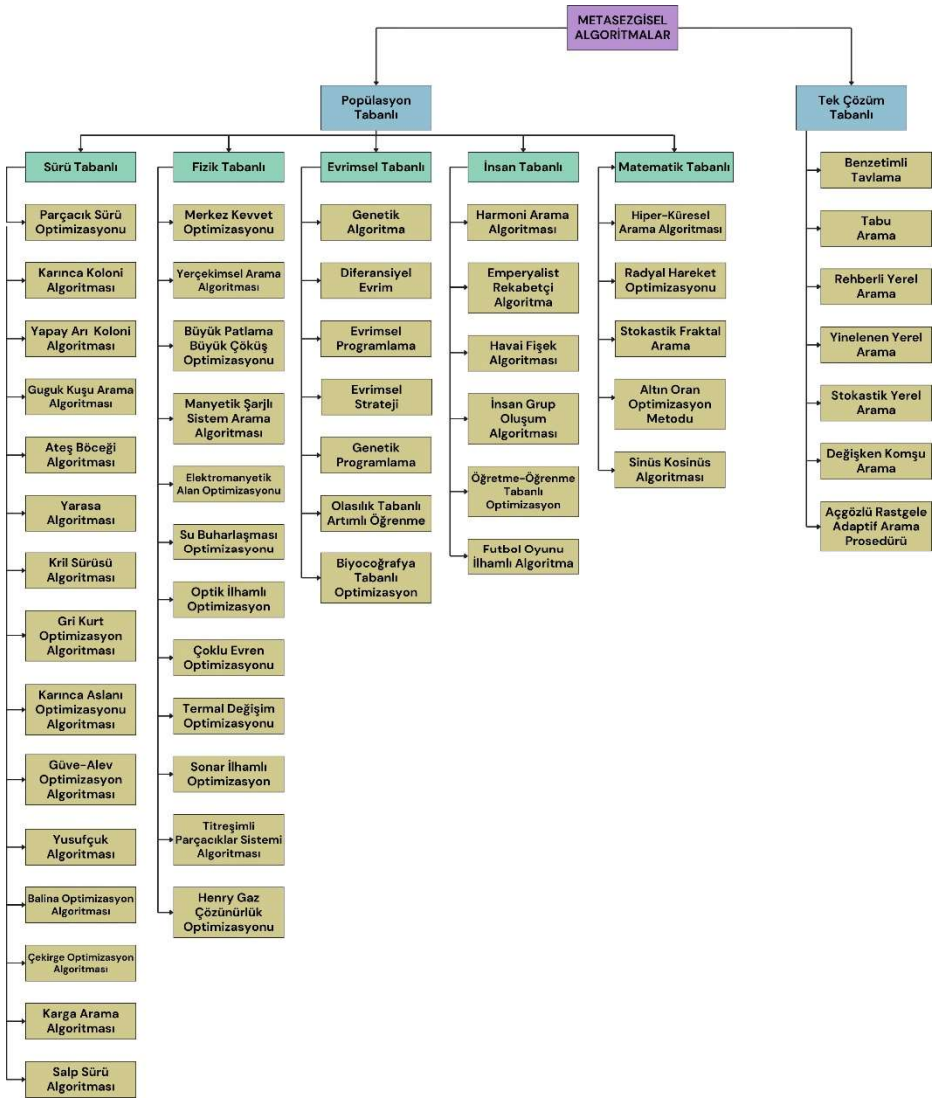
Burada x_k, y_k, x_{k+1} ve y_{k+1} yol parçasının x ve y koordinatlarını, x_e ve y_e engelin bulunduğu hücrenin merkezinin koordinatlarını temsil etmektedir. Engel hücresinin merkez noktasının, engel çevresine olan en uzak mesafesi $\sqrt{2}/2$ birim olduğu Şekil 5'te görülmektedir. Hesaplanan h değeri, $\sqrt{2}/2$'den küçükse yol parçası ile engel kesişmektedir aksi durumda yol parçası engeli kesmemektedir. Ayrıca sadece $\sqrt{2}/2$ değeri arttırılarak robotun engelden daha uzak bir mesafeden güvenli bir şekilde hareket etmesi sağlanabilir.

2.4 Metasezgisel Algoritmalar

Metasezgisel algoritmalar, insan zekasından veya doğadan ilham alır. Bu tür algoritmalar kesin çözümü garanti etmezler sadece kesin çözüme yakın bir çözümü garanti edebilirler [15]. Metasezgisel algoritmalar farklı şekillerde sınıflandırılır. Gabis ve ark. metasezgisel algoritmaları Şekil 6'daki gibi sınıflandırmışlardır. Bu çalışmada tek çözüm tabanlı algoritmalardan Benzetimli Tavlama, popülasyon tabanlı algoritmalardan, evrimsel tabanlı olan Genetik Algoritma ve sürü tabanlı olan Guguk Kuşu Arama algoritması incelenecektir. Buradaki amaç tek çözüm tabanlı ile popülasyon tabanlı metasezgisel algoritmaları aynı zamanda da popülasyon tabanlı algoritmalar içerisinden sürü tabanlı ve evrimsel tabanlı algoritmaları yol planlama problemi için karşılaştırmaktır.

2.4.1 Benzetimli Tavlama

Benzetimli Tavlama algoritması 1983 yılında Kirkpatrick ve arkadaşları tarafından optimizasyon problemlerinin çözümü için geliştirilmiştir. Algoritma, metallerin tavlama işleminden ilham alınarak optimizasyon problemlerine çözüm sunmuştur. Tavlama işleminde, bir metalin sıcaklığı belirli bir maksimum dereceye kadar arttırılır, daha sonra yavaş yavaş soğutulur eğer soğutma işlemi uygun şekilde yapılırsa güçlü kristal yapı elde edilir. Tavlama işlemi iki adımda gerçekleştirilir: Metal istenilen sıcaklığa kadar ısıtılır, belirli bir sıcaklıkta bir süre tutulduktan sonra yavaş yavaş soğutulur [15].

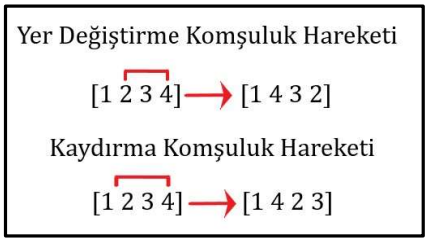


Şekil 6 Metasezgisel Algoritmaların Sınıflandırılması

SA algoritması tek çözüm tabanlı bir algoritmadır ve algoritma rastgele üretilen tek çözüm üzerinden ilerler. Rastgele üretilen çözümler temel olarak üç şekilde aşağıdaki gibi gösterilebilir:

$$\textit{İkili Gösterim} \rightarrow [1\ 0\ 1\ 1\ 0\ 1\ 0\ 1] \quad (4)$$
$$\textit{Tam Sayı Gösterimi} \rightarrow [7\ 3\ 1\ 8\ 13\ 4\ 0\ 41] \quad (5)$$
$$\textit{Gerçel Sayı Gösterimi} \rightarrow [4.7\ 1.2\ 2.4\ 4.8\ 0.7\ 4.4\ 0.1\ 9.1\ ] \quad (6)$$

SA algoritmasında var olan çözüm kullanılarak üretilen yeni çözüm komşu çözüm olarak adlandırılır. Komşu çözüm elde etmek için komşuluk hareketleri kullanılır. Şekil 7'de gösterilen yer değiştirme ve kaydırma komşuluk hareketleri kullanılabilir.



Şekil 7 Komşuluk Hareketleri

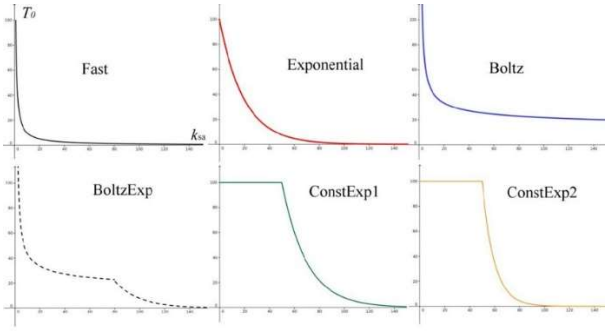
SA algoritmasında yerel minimuma takılmamak için kötü çözümler belirli bir olasılıkla kabul edilebilir. Bu olasılık çözüm ile komşu çözümün maliyetleri arasındaki fark ve sıcaklığa bağlıdır. Olasılık koşulu aşağıdaki denklemlerde verilmiştir.

$$\Delta C = C_i - C_{i-1} \quad (7)$$

$$R < e^{\left(\frac{-\Delta C}{kT}\right)} \quad (8)$$

Burada, ΔC maliyet fonksiyonları arasındaki farkı, C_i komşu çözümü, C_{i-1} var olan çözümü, R sıfır ile bir arasında üretilen rastgele sayıyı, k sabiti ve T sıcaklığı temsil etmektedir. Denklemleri inceleyecek olursak $e^{\left(\frac{-\Delta C}{kT}\right)}$ ifadesi olasılık koşulunu ifade etmektedir eğer rastgele üretilen sayı bu değerden küçük ise komşu çözüm yeni çözüm olarak kabul edilir. Eksponansiyel ifadeyi inceleyecek olursak maliyet fonksiyonları arasındaki fark azaldıkça ve sıcaklık değeri arttıkça komşu çözümün kabul olasılığı artmaktadır.

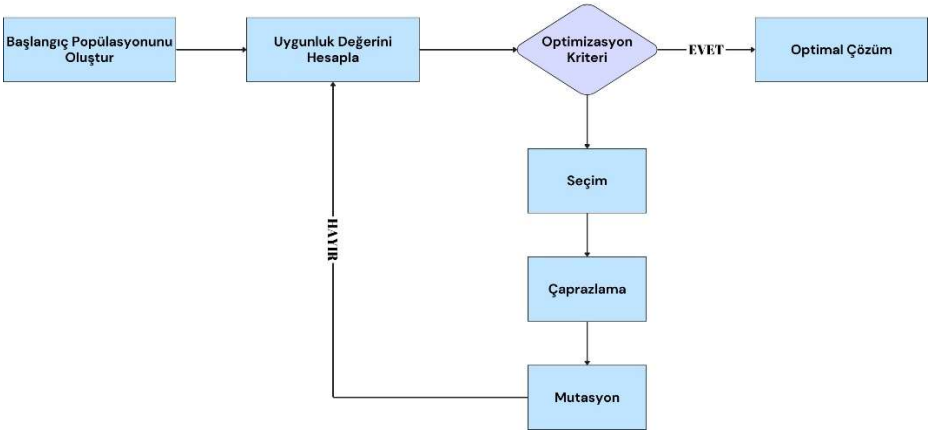
SA algoritmasının performansı sıcaklığa bağlıdır bunun için kullanılan soğutma çizelgesinin seçimi önemlidir. Şekil 8'de kullanılabilecek farkı soğutma çizelgesi örnekleri gösterilmektedir. Ayrıca soğutma işleminde kullanılan başlangıç sıcaklığı, son sıcaklık ve algoritmanın iterasyon sayısının seçimi iyi bir sonuç almak için son derece önemlidir. SA algoritması çözüme arama uzayının rastgele bir noktasından başladığı için başlangıç noktasına bağlıdır.



Şekil 8 Soğutma Çizelgesi Örnekleri [16]

2.4.2 Genetik Algoritma

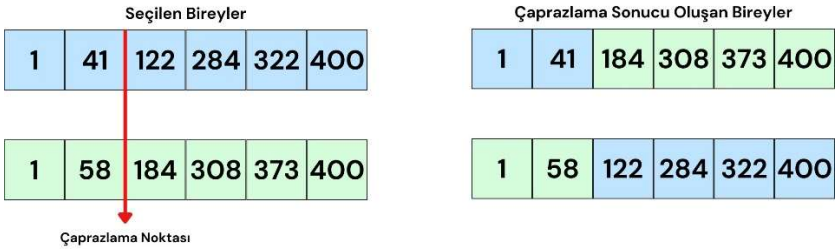
Genetik algoritma 1974 yılında J. Holland ve arkadaşları tarafından geliştirilmiştir. Darwin'in evrim teorisini temel alır. Genetikte olduğu gibi en iyi olanın hayatta kalması veya güçlünün zayıfı yenmesi felsefesine dayanır [17]. Evrim teorisinde yer alan; üreme, çaprazlama ve mutasyon genetik işlemlerini kullanır. Paralel bir arama yöntemi olduğu için çalışma uzayının tümünde arama yapabilir. Genetik algoritmanın akış diyagramı Şekil 9'da gösterildiği gibidir.



Şekil 9 Genetik Algoritma Akış Diyagramı

GA'da bütün çözümler popülasyon, tek bir çözüm kromozom veya birey, çözümün her bir değişkeni ise gen olarak adlandırılır. GA en iyi kromozomların bir sonraki nesle aktarılması prensibine dayanır. Bu da seçim işlemi ile gerçekleştirilir. Bu çalışmada seçim için rulet tekeri yöntemi kullanılmıştır. Bu yöntemde ilk önce tüm kromozomların amaç fonksiyonu değeri hesaplanır, amaç fonksiyon değerleri uygunluk değerlerine dönüştürülür ve uygunluk değeri yüksek olan bireylerin seçilme olasılığı daha yüksek olacak şekilde seçim işlemi yapılır.

Çaprazlama işleminde ise seçilen kromozom çiftlerinin özellikleri birleştirilerek yeni kromozomlar elde edilir. Çalışmada kullanılan tek noktalı çaprazlama işlemi Şekil 10'da gösterilmiştir. Mutasyonda ise amaç bireylerin çeşitliliğini arttırmak ve yerel minimuma takılmayı önlemektir. Her bir gen düşük bir olasılık ile mutasyon işlemine uğrar.



Şekil 10 Tek Noktalı Çaprazlama

2.4.3 Guguk Kuşu Arama

2009 yılında X. Yang ve S. Deb tarafından guguk kuşlarının üremelerinden ilham alınarak geliştirilmiştir. Guguk kuşlarının bazı türleri yumurtalarını konak (yabancı) kuşların yuvalarına bırakır. Dişi guguk kuşu yumurtalarını konak kuşun yumurtalarına şekil ve renk açısından benzeyecek şekilde üretir. Böylece kendi yumurtalarının konak kuş tarafından fark edilmesini engellemeye çalışır. Şekil 11'de guguk kuşunun, konak kuşun yuvasına yerleştirdiği benzer yumurtalar görülmektedir.



Şekil 11 Guguk Kuşunun Yerleştirdiği Yumurtalar [18]

Yeni guguk kuşları üretilirken Lévy uçuşundan faydalanılır çünkü birçok hayvan ve böceğin uçuş davranışları Levy uçuşunun tipik özelliklerini sergiler. Yeni çözümlerin üretilmesi:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Lévy(\lambda) \quad (9)$$

$$Lévy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (10)$$

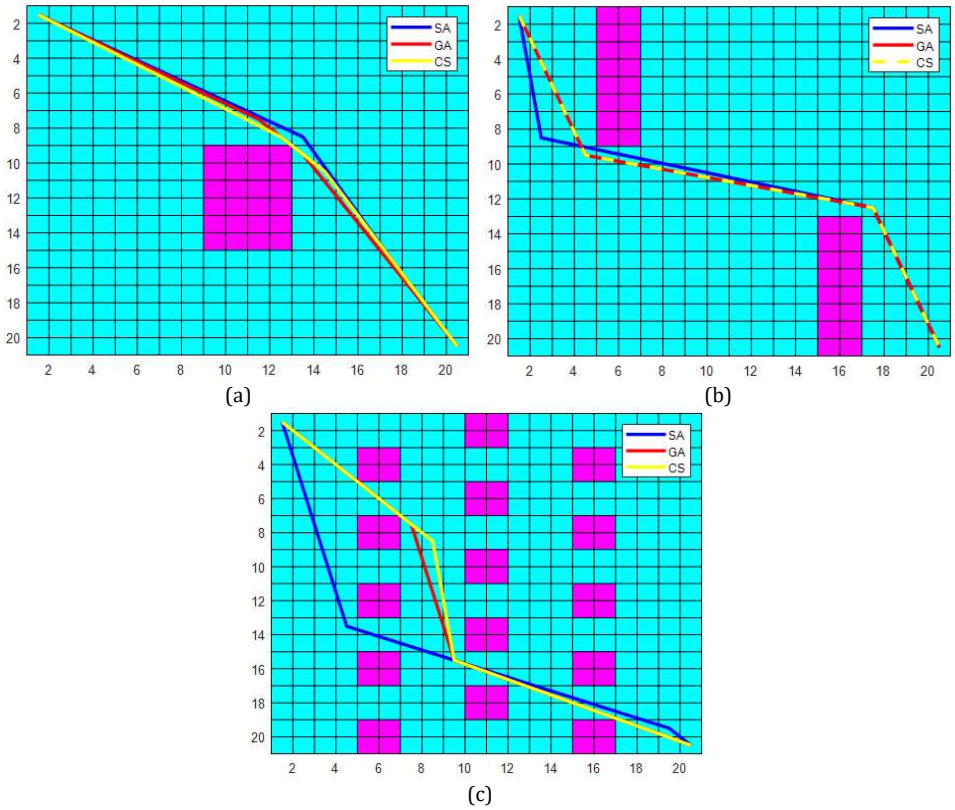
Burada $x_i^{(t+1)}$ yeni çözümü, $x_i^{(t)}$ var olan çözümü, α adım büyüklüğünü (α>0) ve $Lévy(\lambda)$ Lévy uçusunu temsil etmektedir.

Guguk Kuşu Arama algoritmasını basit bir şekilde açıklamak için üç ideal kural kullanılır [19]:

1) Her guguk kuşu her seferde sadece bir yumurta bırakır ve yumurtayı rastgele seçilmiş yuvaya bırakır.
2) Yüksek kaliteli yumurtalara sahip en iyi yuvalar bir sonraki nesle aktarılır.
3) Mevcut ana yuva sayısı sabittir ve guguk kuşu tarafından bırakılan yumurta konak kuş tarafından $p_a \in [0,1]$ olasılığıyla fark edilir. Bu durumda:
    a. Konak kuş, guguk kuşunun yumurtalarını yuvadan atabilir.
    b. Yuvasını terk edip yeni yuva yapabilir.

III. SONUÇLAR

Yol planlama algoritmaları MATLAB 2023b ortamında kodlanmış ve simüle edilmiştir. Kullanılan bilgisayar INTEL CORE i7 işlemciye, 16 GB RAM'e ve Windows 10 Home işletim sistemine sahiptir. Tüm algoritmalarda problem boyutu dört ve ceza puanı 10 olarak alınmıştır. Tüm algoritmalar 50 iterasyon değerinde ve 30 kez çalıştırılarak sonuçlar elde edilmiştir. GA ve CS algoritmalarının popülasyon sayısı 30 olarak alınmıştır. SA'da başlangıç sıcaklığı 1000, soğutma kat sayısı 0.84; GA'da çaprazlama olasılığı 0.95, mutasyon olasılığı 0.4; CS'de yumurtaların fark edilme olasılığı p_a 0.25, Lévy uçusunda λ değeri 3/2 olarak alınmıştır. Algoritmaların 30 kez çalıştırılması sonucunda elde edilen en optimal yollar Şekil 12'de gösterilmiş ve sonuçlar Tablo 1'de verilmiştir.



Şekil 12 Elde Edilen En Optimal Yollar: (a) Harita 1, (b) Harita 2, (c) Harita 3

Tablo 1 Simülasyon Sonucu Elde Edilen Sonuçlar (Ort: Ortalama, Std: Standart Sapma)

Yol Planlama Algoritmaları	Harita 1				Harita 2				Harita 3			
	Maliyet			Çalışma Süresi (sn)	Maliyet			Çalışma Süresi (sn)	Maliyet			Çalışma Süresi (sn)
	Ort.	Std.	En iyi		Ort.	Std.	En iyi		Ort.	Std.	En iyi	
SA	28.772	0.9741	27.7857	0.0001	43.2319	8.8826	31.1392	0.0001	44.4287	11.3214	29.939	0.0002
GA	27.7844	0.1026	27.5287	0.0072	31.5461	0.8241	30.4297	0.0076	29.8145	0.4819	28.8145	0.0084
CS	27.7365	0.0912	27.5287	0.0556	31.4482	0.8763	30.4297	0.0392	29.5977	0.3363	29.0536	0.1121

IV. TARTIŞMA

Şekil 12 incelendiğinde GA ve CS algoritmasının SA'ya göre daha iyi sonuç verdiği görülmektedir. GA ve CS'nin, Harita 1 ve Harita 2'de planladığı yollar yakınken Harita 3'te GA daha iyi sonuç vermiştir. Tablo 1'de görüldüğü üzere SA'nın diğer algoritmalara oranla planladığı yolun uzunluğu fazla olmasına rağmen çalışması süresi açısından diğer algoritmalara göre daha avantajlıdır. Birbirlerine hemen hemen yakın sonuç veren GA ve CS'yi inceleyecek olursak GA süre bakımından daha iyi bir sonuç verirken, CS'nin standart sapma değerinin daha iyi olduğu görülmektir.

Haritalar literatürdeki bir çalışmadan alınmış ve bu çalışmada olduğu gibi algoritmalar 50 iterasyon değerinde ve 30 kez çalıştırılmıştır [14]. Literatürdeki çalışmada matematik tabanlı metasezgisel algoritmalar olan Stokastik Fraktal Algoritması (SFS), Aritmetik Optimizasyon Algoritması (AOA), Sinüs Kosinüs Algoritması (SCA) kullanılmıştır. Yapılan çalışma ile sonuçları karşılaştıracak olursak, GA ve CS algoritmaları AOA ve SCA'dan daha iyi sonuçlar vermiştir. SA Harita 1'de AOA ve SCA'ya yakın, Harita 2'de daha kötü, Harita 3'te ise bu algoritmalardan daha iyi sonuç vermiştir. GA ve CS Harita 1 ve Harita 2'de SFS'ye yakın bir sonuç verirken, Harita 3'te SFS'den daha iyi sonuç vermiştir. SFS, Harita 1 ve Harita 3'te standart sapma olarak GA ve CS'den daha iyi sonuçlar vermiştir. Süre açısından baktığımızda ise SA, GA ve CS'nin bu algoritmalara büyük bir üstünlük kurduğunu söyleyebiliriz.

V. ÇIKARIM

Bu çalışmada üç farklı zorlukta engeller içeren harita tasarımları kullanılarak SA, GA ve CS metasezgisellerinin yol planlama performansları incelenmiştir. GA ve CS yolun maliyeti açısından SA'dan daha iyi bir sonuç vermiştir. SA algoritması ise GA ve CS'den daha hızlı çalışmaktadır. Daha sonra bu algoritmalar literatürdeki matematik tabanlı metasezgisellerin incelendiği çalışma ile karşılaştırılmıştır. Karşılaştırma sonucunda, GA ve CS'nin AOA ve SCA algoritmalarından daha iyi, SFS algoritmasına da yakın bir sonuç verdiği görülmüştür. SA, GA ve CS algoritmaları bu matematiksel tabanlı algoritmalarından daha hızlı çalışmaktadır.

KAYNAKLAR

[1] M. Eyüboğlu ve G. ATALI, "Çok Yönlü İşbirlikçi Mobil Robotlar için Eş Zamanlı Hareket Planlama Algoritmasının Geliştirilmesi ve Uygulaması", Yüksek Lisans Tezi, Sakarya Uygulamalı Bilimler Enstitüsü, Sakarya, Türkiye, Nisan 2022.

[2] Z. Zexuan, W. Fangxiao, H. Shan, S. Yiwen, "Global Path Planning of Mobile Robots Using a Memetic Algorithm", *International Journal of Systems Science*, 46(11):1982–1993, 2015.

[3] Z. Garip, "Mobil Robotların Yol Planlaması için Meta-Sezgisel Hibrit Algoritmalar Geliştirilmesi ve Uygulanması", Doktora Tezi, Sakarya Üniversitesi, Sakarya, Türkiye, Ağustos 2018.

[4] H. Zhang, W. Lin, ve A. Chen, "Path Planning for the Mobile Robot: A Review", *Symmetry*, c10(10), 450, 2018, doi: 10.3390/sym10100450.

[5] M.A. Contreras-Cruz, V. Ayala-Ramirez, U.H. Hernandez-Belmonte, "Mobile robot path planning using artificial bee colony and evolutionary programming", *Appl. Soft Comput.*, 30, 319–328, 2015.

[6] H.V. Hoang, H.A. Sang, C.C. Tae, "Dyna-Q-based vector direction for path planning problem of autonomous mobile robots in unknown environments", *Adv. Robot*, 27, 159–173, 2013.

[7] Y. Xu, Q. Li, X. Xu, J. Yang, ve Y. Chen, "Research Progress of Nature-Inspired Metaheuristic Algorithms in Mobile Robot Path Planning", *Electronics*, c. 12, sy 15, s. 3263, Tem. 2023, doi: 10.3390/electronics12153263.

[8] F. Gul, W. Rahiman, S. S. N. Alhady, A. Ali, I. Mir, ve A. Jalil, "Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming", *J Ambient Intell Human Comput*, c. 12, sy 7, ss. 7873-7890, Tem. 2021, doi: 10.1007/s12652-020-02514-w.

[9] P. K. Mohanty ve D. R. Parhi, "Optimal path planning for a mobile robot using cuckoo search algorithm", Journal of Experimental & Theoretical Artificial Intelligence, c. 28, sy 1-2, ss. 35-52, Mar. 2016, doi: 10.1080/0952813X.2014.971442.

[10] Yee Zi Cong ve S. G. Ponnambalam, "Mobile robot path planning using ant colony optimization", *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Singapore: IEEE, Tem. 2009, ss. 851-856. doi: 10.1109/AIM.2009.5229903.

[11] L. B. Amar ve W. M. Jasim, "Hybrid metaheuristic approach for robot path planning in dynamic environment", *Bulletin EEI*, c. 10, sy 4, ss. 2152-2162, Ağu. 2021, doi: 10.11591/eei.v10i4.2836.

[12] S. Lin, A. Liu, J. Wang, ve X. Kong, "An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse", *Journal of Computational Science*, c. 67, s. 101938, Mar. 2023, doi: 10.1016/j.jocs.2022.101938.

[13] A. Tuncer ve M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm", *Computers & Electrical Engineering*, c. 38, sy 6, ss. 1564-1572, Kas. 2012, doi: 10.1016/j.compeleceng.2012.06.016.

[14] M. Y. Yildirim ve R. Akay, "Izgara Bazlı Yol Planlama için Matematik Tabanlı Metasezgisellerin Karşılaştırılması", *European Journal of Science and Technology*, Ocak 2022, doi: 10.31590/ejosat.1039899.

[15] D. Karaboğa, "Yapay Zeka Optimizasyon Algoritmaları", Nobel Akademik Yayıncılık, Ankara, 7. Basım, Kasım 2020.

[16] G. Velleda Gonzales, E. Domingues Dos Santos, L. Ramos Emmendorfer, L. André Isoldi, L. A. Oliveira Rocha, ve E. Da Silva Diaz Estrada, "A Comparative Study of Simulated Annealing with different Cooling Schedules for Geometric Optimization of a Heat Transfer Problem According to Constructal Design", *Scientia Plena*, c. 11, sy 8, Ağu. 2015, doi: 10.14808/sci.plena.2015.081321.

[17] G. Bektaş, S. M. Nigdeli, M. Yücel ve E. Kayabekir, "Yapay Zeka Optimizasyon Algoritmaları ve Mühendislik Uygulamaları", Seçkin Yayıncılık, Ankara, Mart 2021.

[18] Guguk Kuşu (Cuckoo) Arama/Optimizasyon Algoritması. Available from: https://medium.com/@seckindikbayir/guguk-ku%C5%9Fu-arama-optimizasyon-algoritmas%C4%B1-94049efea5b2 [erişim 4 Mart, 2024]

[19] X.-S. Yang ve Suash Deb, "Cuckoo Search via Lévy flights", *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India: IEEE, 2009, ss. 210-214. doi: 10.1109/NABIC.2009.5393690.