



## Sürü Robotların Hareket Planlamada Kullanılması

Ebubekir Yaşar<sup>1\*</sup>

<sup>1\*</sup> Tokat Gaziosmanpaşa Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Makine Bölümü, Tokat, Türkiye, (ORCID: 0000-0002-0780-893X), ebubekir.yasar@gop.edu.tr

(İlk Geliş Tarihi 06 Temmuz 2020 ve Kabul Tarihi 06 Ekim 2020)

(DOI: 10.31590/ejosat.763444)

**ATIF/REFERENCE:** Yaşar, E. (2020). Sürü Robotların Hareket Planlamada Kullanılması. *Avrupa Bilim ve Teknoloji Dergisi*, (20), 24-29.

### Özet

Sürü robotlar birçok çalışmada karmaşık görevlerin çözümünde başarı ile kullanılmıştır. Sürünün ortak bir görevi yerine getirmesi aralarındaki iletişime bağlıdır. Bu çalışmada, farklı olarak sürü robotlar hareket planlama çalışmasında kullanılmışlardır. Sürü robotlar yardımıyla ana görev robotunun, engellere çarpmadan, başlangıç noktasından, bitiş noktasına doğru en kısa mesafede seyahatini sağlayacak bir yol bulunmuştur. Bu çalışmaya *Trafik Polisi Algoritması (TPA)* ismi verilmiştir. Algoritmaya göre sürüye ait robot bireylerin, yapılandırma alanının hemen her yerinde ortamın sınırları içerisinde belirli bir düzende yayılmaları sağlanmıştır. Bu rasgele yayılım düzenine sahip sürü robot üyeleri, birbiriyle haberleşme mesafesi kadar yakına konumlanmaktadır. Hedef noktaya en yakın üye yoksa üyeler önceden belirlenen kadar mesafe, rastgele hareket etmektedir. Hedef noktayı gören robot yakınındaki diğer komşu robotlara, bildirimde bulunarak mesafe ve yönelim açısını iletmektedir. Tüm robotlar, görebildiği komşu robotlara mesafe ve yönelim bilgisini aktararak bu bilgiler sonunda başlangıç noktasındaki ana görev robotuna iletilir. Başlangıç noktasındaki ana görev robotu, kendisine iletilen düğümler arasında en kısa yolu bulmak için arama yöntemlerinden Dijkstra algoritmasını kullanarak en kısa mesafeyi bulmuştur. Geliştirilen algoritma başlangıçta sanal ortamda denenmiş ve uygulaması ileriki çalışmalarda yapılacaktır.

**Anahtar Kelimeler:** Sürü robotlar, Hareket Planlama, Mobil robotlar.

## Using Swarm Robots in Motion Planning

### Abstract

Swarm robots have been used successfully in many studies to solve complex tasks. The common task of swarm robots depends on the communication between them. In this study, differently, swarm robots were used in the motion planning study. With the help of swarm robots, a path was found that would allow the main mission robot to travel the shortest distance from the starting point to the end point without hitting the obstacles. This study was named Traffic Police Algorithm (TPA). According to the algorithm, robotic individuals belonging to the swarm are provided to spread in a certain order within the boundaries of the environment almost everywhere in the configuration space. The swarm robot members with this random propagation pattern are positioned as close to each other as far as communication distance. If there is no member closest to the target point, the members move randomly at a predetermined distance. When a robot sees the target point, it transmits the distance and orientation angle by notifying other neighboring robots nearby. All robots transmit distance and orientation information to neighboring robots that they can see, and this information is finally transmitted to the main task robot at the starting point. The main task robot at the starting point found the shortest distance using the Dijkstra algorithm, one of the search methods, to find the shortest path among the nodes transmitted to it. The developed algorithm was initially tested in a virtual environment and its implementation will be done in future studies.

**Keywords:** Swarm robots, Motion planning, Mobil robots.

\* Sorumlu Yazar: ebubekir.yasar@gop.edu.tr

## 1. Giriş

Robotlar günlük hayatımızda verilen görevleri hızlı, hatasız ve kolay bir şekilde yapmasıyla iş gücüne olan katkısı giderek artmaktadır. Tek bir robot yerine sürü robotların kullanılmaya başladığı uygulamalar yaygınlaşmaktadır. Robotların verilen görevleri eksiksiz ve hızlı yapması özellikle otonom hareket kabiliyetine bağlıdır. Sürü robot mantığı birden fazla robotun aynı görevin bir parçasını icra edebilmesi ilkesine dayanan bir yaklaşımdır. Sürüdeki bireyler, karşılaştıkları sorunları, kendi başlarına çözmeleri gerekmektedir. Günümüzde artan bir şekilde, birçok farklı alanda sürü robot uygulamaları üzerine çalışılmaktadır. Robotikte nihai hedeflerden birisi otonom robotlar yapmaktır. Böyle robotlar üst düzey görevleri kabul edecekler ve görevlerini insan müdahalesi olmadan yerine getireceklerdir. Günlük hayatta, yapılması gereken görevlerde insan faktörünün dezavantajlarını azaltmak, işlem süresini kısaltmak, zor, karmaşık işlemlerin üstesinden gelmek veya iş güvenliğini sağlamak için insan müdahalesi olmadan hareket eden akıllı robot çalışmaları kaçınılmaz olmuştur. Sürü robotlar özellikle doğadaki canlıların hayatlarını devam ettirmek için gösterdiği dayanışmayı örnek alır. Özellikle karanlık ve girilmesi zor alanlarda (deprem yıkıntıları, mağaralar, yer altı yapıları... gibi) sürü robotların, arama kurtarma veya operatör görevini icra eden ana robota, gideceği yolu bulmada yardımcı olması yapılacak görevi hızlandıracaktır. Bu amaçla birçok yol planlama tekniği kullanılmaktadır (Jaradat M. A., Garibeh M. H. & Feilat E. A. 2012). Yol planlama çalışması robotun engellere çarpmadan hareket edebileceği ve başlangıç pozisyonundan, bitiş noktasına doğru en kısa bir biçimde seyahatini sağlayacak bir yolu arama gayretidir. Hareket planlamanın hedefi fiziksel nesnelere uzaysal düzenlemesidir denilebilir (J.C. Latombe, 1991). Diğer taraftan robot hareket planlama çalışmaları ise dinamik yapıyı ve sınırlandırmaları göz ardı ederek yalnızca robot nakline ve gerekli dönüşlere odaklanır (Steven M. LaValle, 2006).

Literatürde ilk planlama çalışması 1969 yılında Nilsson tarafından Shakey isimli mobil bir robot sistemi üzerinde icra edilmiştir. Nilsson, otonom hareketi sağlaması amacıyla önemli iki görevi (visibility graph yöntemini, A\* algoritması) robot üzerinde beraberce uygulayarak bu alandaki ilk çalışmayı yapmıştır. İlk olarak 1966 yapımına başlanan Shakey isimli tekerlekli mobil robot Charles Rosen tarafından 1972 yılında tamamlanmış ve planlama yeteneklerine sahip ilk robot olarak tarihe geçmiştir (Bertram Raphael, 2012). Robotun hareket ettiği alanda robot dışında gerek yapıya ait, gerekse ortamda hareketli veya sabit çeşitli engeller olabilir. Robotun hareketi sırasında bu engellere çarpmadan hareket etmesi önemlidir. Çünkü her robotun uzayda kapladığı bir herhangi bir geometriye sahip hacmi vardır. Buradan hareketle 1977 yılında Udupa, engelden sakınma algoritmaları için robotun nokta boyuta küçültülmesi fikrini ortaya atmıştır. Lozano Perez ve Wesley bu fikirden yararlanarak engellere değmeden robotların yol planlamasını yapmak için bu fikri sistematik hale getirmişlerdir (Lozano Perez T. ve Wesley M.A., 1979). Bu gayretleri yapılandırma alanı kavramına yol açmıştır. Yapılandırma alanında robot, serbestlik derecelerini tanımlayan parametre uzayında tek bir nokta olarak temsil edilmesini ifade eder (J.C. Latombe, 1999). Yapılan çalışma robotun yapılandırma alanının çıkarılmasına da önemli avantajlar sağlayacaktır. 1983 yılında Andrews ve Hogan ve 1985 yılında Khatib (Khatib O., 1986), potansiyel alan adını verdikleri bir yöntem geliştirmişlerdir. Buna göre robotun çalıştığı ortamı potansiyel alana sahip bir ortam gibi kabul etmişlerdir. Bu

potansiyel alanda varış noktası çekici engeller ve başlangıç noktası itici kuvvetlere sahiptir. 1987 yılında B. Chazelle, yapılandırma alanını belirli boyutlarda hücrelere (approximate decomposition cells) ayırarak engellere değmeyen hücreler üzerinden hedefe giden yolu aramıştır. 1991 yılında F. Aurenhammer Voronoi Diyagram ismini verdiği yöntemle yapılandırma alanını en yakın komşu kuralına göre ayırır (F. Aurenhammer, 1991). 1994 yılında ise Kavraki ve arkadaşları (Kavraki, L. E., P. Svestka, J-C. Latombe & M. Overmars, 1996), olasılık temelli yol (PRM) metodunu, 1998 yılında LaValle ve Kuffner (LaValle, S. M., & Kuffner, J. J., 2001) rasgele ağaç yapıda hızlı arama (RRT) algoritmaları yardımıyla başlangıç noktasını bitiş noktasına birleştiren bir yol aramışlardır.

Hareket planlamada anahtar kavramlardan birisi yapılandırma alanı kavramıdır. Bir konfigürasyon robotun duruşunu açıklar ve konfigürasyon alanı C ise olası tüm konfigürasyonlar kümesidir. Örneğin iki boyutlu bir düzlem üzerinde nokta boyutunda bir robotun yapılandırması, (x, y) gibi iki parametre ile ifade edilir. Aynı şekilde iki boyutlu bir düzlemde iki boyutlu bir robotun yapılandırması, (x, y,  $\theta$ ) gibi üç parametre ile temsil edilir. Engellerle çarpışmayı önleyen yapılandırma alanına serbest yapılandırma alanı ( $C_{free}$ ) denir. Robotun tüm hareketleri serbest yapılandırma alanı içerisinde olmak zorundadır. Serbest yapılandırma alanını hesaplamak zor fakat herhangi bir noktanın serbest yapılandırma alanı içerisinde olup olmadığını belirlemek daha kolaydır (Wikipedia, 2020). Yapılandırma q ile yapılandırma alanı ise Q ile gösterilir.

Hareket planlama çalışmalarında yakın tarihte 2014 yılında Yıldırım Ş. ve Yaşar E. geliştirdikleri akıllı nesnelere algoritması ile olaya farklı bir bakış açısı getirmişlerdir. Bu algoritma nesne tabanlı programlama tekniklerini kullanmaktadır. Çalışma alanına yerleştirilen nesnelere altında engel olup olmadığını algılayan, birbirlerini görebilme yeteneğine sahip nesne tabanlı program mimarisinde oluşturulmuş yazılım tabanlı akıllı nesnelere oluşmaktadır (Yıldırım Ş, Yaşar E., 2015).

Bu çalışmada ana iş yürütücü robotun hareket planlamasına yardımcı olmak amacıyla sürü robotlar kullanılacaktır. Sürü robotlar basit görevleri topluca hareket ederek icra edebilen düşük enerji harcayan robot topluluğudur. Birlikte yaşayan canlıların hayatta kalmak için gösterdikleri dayanışmayı taklit eder (Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M., 2013). Bu dayanışma düşman tehdidi, hayat mücadelesi veya beslenme amaçlı olabilir. Sürü robotlarda bireyler karşılaştıkları problemlerle kendi başlarına başa çıkmak zorundadırlar. Aralarında özellikle yakın bireyler arasında kesintisiz bir iletişim olması zorunludur. Sürü halinde yaşayan canlıların problemlere esnek ve ölçeklenebilir şekilde yaklaşmaları onların problem çözebilme kabiliyetlerini artırmaktadır. Sürü robotların basit olması, az sayıda görevi etkin bir şekilde yapabilecek şekilde tasarlanmaları gerekmektedir (Shlyakhov, N. E., Vatamaniuk, I. V. & Ronzhin, A. L., 2017). Sürü bireylerinin en temel görevi yakın olan bireylerin birbiriyle iletişim içinde olmaları ve birbirine haberleşmeden kopmayacak şekilde yakın olmalarıdır. Ayrıca sürüdeki robotların çevre ile etkileşimi güçlü, bağımsız hareket edebilme ve karşılaşılan sorunların üstesinden gelebilme yetisine sahip olmalıdır (O. Soysal & Sahin, 2005). Bireylerin özerkliği sürünün en iyi yolu bulmasında kilit rol oynamaktadır. Çünkü her bir birey bulduğu çözümü en iyi çözümlerle karşılaştırarak daha iyisini bulduğunda durumunu güncellemektedir. Sürü robotlarda, sürülerdeki her robot, yani birey, gruba göre hareket ederken

gerçekleştirilecek görevi yerine getirecektir. Bireyler grupla birlikte hareket ederken, grubun en iyi çözümü bulmasına yardımcı olacak davranışlar göstermesi gerekmektedir.

Çalışmamıza en yakın çalışmalardan birisi olan PSO(Particle Swarm Optimization) tekniği, robotları sürü halinde istenen noktaya sevkeden bir yaklaşımdır. PSO 1995 yılında James Kennedy ve Russel Eberhart tarafından önerilmiştir. Bu yaklaşımda sürüde hareket eden bazı hayvanların hareketlerinin, yiyecek bulma gibi temel ihtiyaçlarını karşılarken sergiledikleri davranışlardan ilham alan bir yaklaşımla, sürünün içindeki diğer bireyleri etkilemekte ve sürünün amacına daha kolay ulaşması şeklinde bir algoritmaya sahiptir(J. Kennedy & R. Eberhart, 1995).

Hareket planlama çalışmaları gerçek zamanlı ve gerçek zamanlı olmayan olarak iki ayrılmaktadır. Bu çalışmada gerçek zamanlı olmayan yol planlama çalışması yapılacaktır. Gerçek zamanlı olmayan yol planlama çalışmasında yol planlama çalışması önceden elde edilen verilerle yapılır ve sonrasında sonuçlar robota iletilir. Bu çalışmada çalışma ortamına bırakılan sürüye ait robotlar birbirinden belirli uzaklıklarda ve çalışma ortamının her tarafına mümkün olduğunca homojen şekilde dağılmış bir yapıda dizilirler. Sürü robotların karakteristik özelliklerinden haberleşme ve yardımlaşma özellikleri sayesinde yakın robotların birbirine hedefi haber vermesi üzerine bir algoritma kurulmuştur. Yol planlaması yapıldığında, çalışma alanında ayrıca bir serbest yapılandırma alanı hesabı yapmadan hızlı bir şekilde yol planlaması yapılmaktadır. Bu çalışmadaki sürü robot hareketleri, sürü robotların ekseri hareket tarzı olan bir yiyeceğe veya hedef noktaya doğru hareket şeklinde değil, o anki bulunduğu konumda kendi görevini icra ederek bir problemin çözümünde üstüne düşen görevi icra etmeye çalışan bireylerden oluşmaktadır. Burada her bir robot birbirini görebilecek sensörlere, yakınındaki robotlara kendisine gelen verileri aktaracak ekipmanlarla donatılmıştır. Robotların görevi en kısa yol arama algoritmalarına gerekli olan değerleri(düğüm değerlerini) elde etmektir. Bu amaçla Dijkstra algoritması kullanılmıştır. Dijkstra algoritması (SPF algoritması), bir grafikteki düğümler arasındaki, örneğin yol ağlarını temsil edebilecek en kısa yolları bulmak için kullanılan bir algoritmadır. 1956 yılında bilgisayar bilimcisi Edsger W. Dijkstra tarafından tasarlandı ve üç yıl sonra yayınlandı(Philip L. Frana & Thomas J. Misa, 2010). Algoritma birçok varyantta bulunur. Dijkstra'nın orijinal algoritması verilen iki düğüm arasındaki en kısa yolu buldu, ancak daha yaygın bir varyant tek bir düğümü "kaynak" düğümü olarak düzeltir ve kaynaktan grafikteki diğer tüm düğümlere en kısa yolları bulur ve en kısa yol ağacını üretir(Wikipedia, 2020).

## 2. Materyal ve Metot

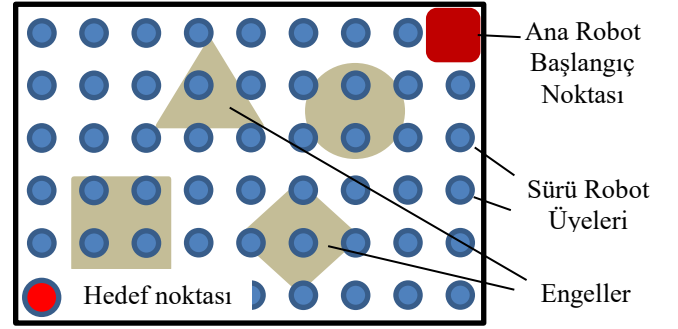
### 2.1. Algoritmaya Giriş ve Temel Kavramlar

Bu çalışmada gerçek zamanlı olayın bir hareket planlama algoritması gerçekleştirilmiştir. Ana işleyici robota, hedef başlangıç noktasından hedef noktaya ulaşması noktasında yardımcı olmak amacıyla sürü robotlar kullanılacaktır. Bu çalışmada sürü robotlar sanal olup aslında programlama dilinde oluşturulan nesnelere ibarettir. Nesne tabanlı programlama teknikleri ile geliştirilen robotlar temel işlevleri yerine getiren arka planda kodlara sahiptir. Nesnelere, nesne tabanlı programlama(Object-Oriented Programming=OOP) içerisinde, bir geliştirici tarafından oluşturulan soyut bir veri türüdür. Birden fazla özellik ve yöntem içerebilir ve hatta başka nesnelere de

içerebilir. Çoğu programlama dilinde, nesnelere sınıf olarak tanımlanır. Nesnelere programlamaya yapılandırılmış bir yaklaşım sağlar. Yazılım nesnelere kavramsal olarak gerçek dünyadaki nesnelere benzer: onlar da durumları ile ilgili davranışlara sahiptir. Bir nesne durumunu alanlara (bazı programlama dillerindeki değişkenler) depolar ve davranışını yöntemlerle (bazı programlama dillerindeki fonksiyonlar) ortaya koyar. Yöntemler(metotlar) bir nesnenin iç durumunda çalışır ve nesneden nesneye iletişim için birincil mekanizma görevi görür(Oracle, 2020). Her bir nesne bu çalışmada sanal robot gibi davranarak çalışma alanına ait resm üzerinde kendisine verilen görevleri yapabilecek uygulamaya(metotlara) sahiptir.

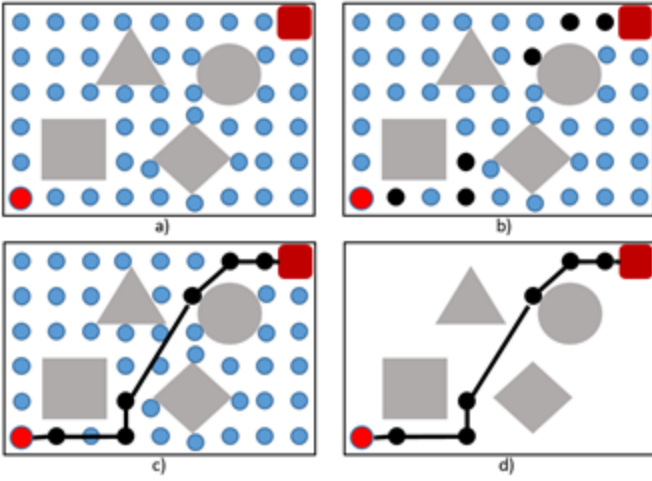
#### 2.1.1. Trafik Polisi Algoritması

Çalışma alanına yerleştirilen sürü robotların, hedefin yerini birbirlerine haber vermesi esasına dayanan algoritmaya Trafik Polisi Algoritması ismi verilmiştir. Bu algoritmaya göre; çalışma alanına belirli aralıklarla, matris formda, ızgara şeklinde sürü robotlar Şekil 1'deki gibi yerleştirilmiştir. Robot sayıları isteğe göre ayarlanabilmektedir. Yol planlamada çözünürlüğü artırmak için daha fazla robot kullanılabilir. Robot sayısı gerekenden daha az sayıda ise engellerden dolayı birbirini göremeyen robotlar çözümün boş küme olmasına neden olacaktır. Engellerin çok ve küçük olduğu ortamlarda robot sayısını fazla tutmak çözümün boş küme olarak dönmelerini engelleyecektir.



Şekil 1. Yapılandırma alanı ve sürü robotlarının yapılandırma alanına yerleştirilmesi

Çalışma ortamına yerleştirilen robotlar, birbirleri ile sadece görerek haberleşmektedirler. Birbirlerine yakın olsalarda, birbirini görmeyen robotlar birbiri ile haberleşmemektedir. Engelden sakınma algoritmalarının en önemli parametresi birbirini gören noktalar kümesinin elde edilmesidir. Dolayısı ile RF tipi bir haberleşme tekniği kullanılamaz. Çünkü RF tabanlı haberleşme tekniğinde alıcı ve vericinin birbirini görmesine gerek yoktur. Haberleşme mesafesi içerisinde herhangi bir noktada olmaları yeterlidir. Çalışma alanındaki bir robotun kesit alanının engelin üzerine %50 den fazlası denk gelmesi durumunda her bir robot kendini yok ederek hesaplamaya katılmamaktadır. %50 den daha az kısmı engelle kesişen robotlar engelden en yakın boşluğa doğru kayarak düşmektedir(yanında durmaktadır). Robotların boyutları ve geometrisi ana robota göre seçilerek serbest yapılandırma alanı hesabı kolaylaştırılmıştır. Serbest yapılandırma alanı sürü robotların konumlanması ve birbirlerini görmesi süreçlerinde dolaylı olarak hesaplanmış olmaktadır. Sürü robot üyelerinin birbirini görmesi demek; en az kendi boyutlarında boş bir alanın aralarında var olduğu anlamına gelmektedir. Robotların, boyutları ve geometrisi ana robota göre seçilebilmektedir. Şekil 2'de algoritmanın tüm adımları görsel olarak gösterilmiştir.

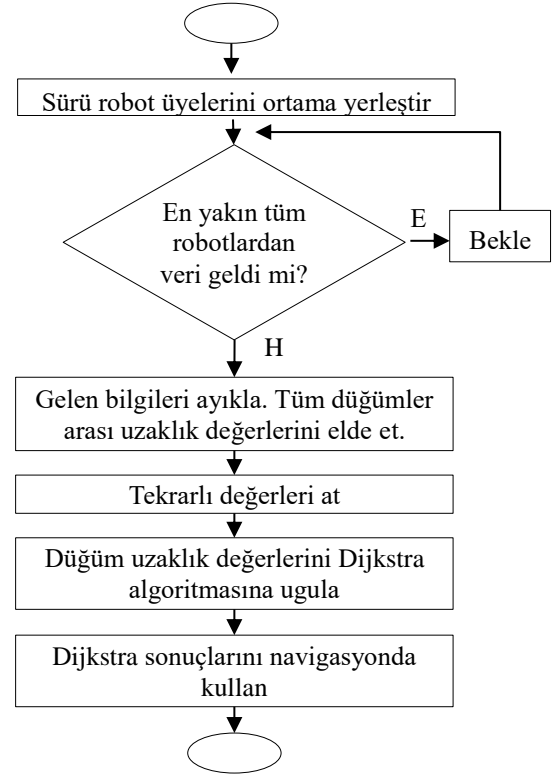


Şekil 2. Algoritma adımlarının görsel olarak gösterilmesi  
a) engel üzerindeki robotların pozisyon almaları. b) Hedefi gören robotun bu bilgiyi komşu robotlarla paylaşması sonucu elde edilen en kısa mesafedeki robot üyelerinin belirlenmesi. c), d) başlangıç noktasından hedef noktaya olası en kısa mesafenin belirlenmesi

Yapılandırma alanına Şekil 1a'deki gibi düzgün ızgara biçiminde yerleştirilen sürü robot üyelerinden engel üzerinde %50 den fazla kesişime sahip robotlar yok edilirler. Diğer durumda robotlar en yakın noktaya kayarak engelin yanına konumlanırlar. Sonrasında hedefi gören robot(renginden tanınmaktadır) kendisine yakın robotlara uzaklık ve yönelim verilerini iletecektir. Bu veriyi alan robot, daha önce haberleşmediği ve görebildiği yakın komşusuna önceki robottan aldığı verilerin üzerine aralarındaki mesafe ve yönelim bilgilerini de ekleyerek komşusuna iletir. Bu şekilde tüm veriler başlangıç pozisyonundaki ana iş robotuna iletilecektir. Ana iş robotuna gelen veriler birçok kanaldan geldiği için tekrarlı bilgiler olacaktır. Ana iş robotu bunları ayıklayarak elde ettiği birbirini gören düğüm çiftlerini arama algoritması yardımıyla en kısa yol bilgisine dönüştürür. Bu navigasyon bilgisi robotun hedefe gitmesinde robot tarafından kullanılacaktır.

## 2.2. Ana İşleyici Robota Ait Görevler

Ana işleyici robot ilk olarak istenen sayıda sürü robotları ortama dağıtarak belirli bir düzende olmalarını sağlar. Sonrasında kendisine en yakın robotlardan gelecek bilgileri bekler. Ana işi görecektir robot, sahadan gelen verileri, komşu robotlardan alarak bunlardan birbirini gören düğüm noktalarını ayıklar. Robot tekrarlı çiftleri atarak elde kalan tekil düğüm çiftlerini Dijkstra arama algoritmasına girdi olarak verip en kısa mesafeyi bulacaktır. Ana robot, en kısa mesafe hangi düğümler arası gerçekleşirse o düğümler arası uzaklık ve yönelim bilgisini navigasyonda kullanarak hedef noktaya doğru ilerleyecek ekipmana sahiptir. Robot, Dijkstra algoritmasının belirlediği noktalardan tek tek geçerek hedefe ulaşır. Ana işleyici robota ait görev akış çizelgesi Şekil 3'te verilmiştir. Sonraki çalışmalarda elde edilen bu yolu düzgunleştirerek daha hızlı bir yol bulma çalışması yapılacaktır.



Şekil 3. Yapılandırma alanı ve sürü robotların yapılandırma alanına yerleştirilmesi

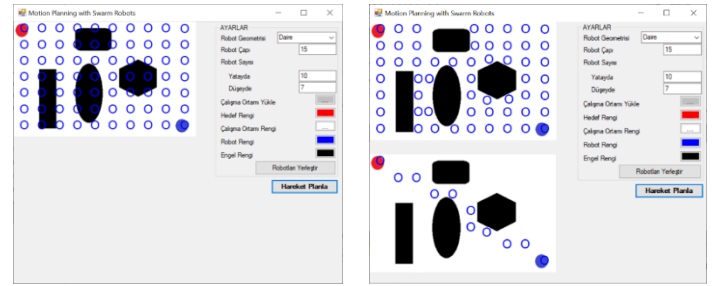
## 2.3. Sürü Robot Özellikleri ve Görevleri

Kullanılacak robotlar birbiriyle haberleşebilme kabiliyetinde ekipmanlara sahiptirler. Her bir robot birbiri ile optik olarak haberleşmektedirler. Optik haberleşme birbirlerini görmesi zorunluluğundan kaynaklanmaktadır. Dolayısı ile 45 derecelik açılarla robot etrafına yerleştirilen alıcı ve verici çiftinin bulunduğu 8 adet göz hangi yönelimle sinyal geldiğini algılayarak, hangi komşu robottan veri geldiğini hafızaya alır. Kendisine veri gönderen robotla daha haberleşilmez ve o port kapatılır. Robot ayrıca yine 45 derecelik açı ile yerleştirilmiş 8 adet uzaklık sensöründen oluşmaktadır. Her bir komşusu ile haberleşebilmesi açısından Her bir robot kendisine gelen uzaklık ve yönelim(açı) bilgilerini kendi 8 adet robotla iletişim kuracaktır. Robotların, birbirlerini, engelleri, başlangıç noktasını ve hedef noktayı tanımlaması ayrı tonda renk özellikleri ile olacaktır. Dolayısı ile her bir robotun etrafında 45 derecelik açı ile 8 adet renk sensörü bulunmaktadır. Robotun altında ise papatya şeklinde dizilmiş 8 adet renk sensörü engelleri tanımlama açısından yerleştirilmiştir. Bu sensörlerden 4( $\leq$  %50) ve daha azı engel rengini algıladığında robot engel olmayan yere doğru kayacaktır. Bu kayma işlemi tüm renk sensörleri ortam rengini algılayana kadar devam edecektir. Eğer robotun alt yüzey alanının %50 den fazlası engelle kesişiyorsa robot kendini imha ederek hesaplamalara dahil olmamaktadır. Sürü robotlara ait akış diyagramı Şekil 4'te verilmiştir.



### 3. Araştırma Sonuçları ve Tartışma

Geliştirilen algoritma C# dilinde kodlanarak uygulama gerçekleştirilmiştir. Hernekadar robotlarda sensörlerden bahsedilse de offline çalışmada bu yapılar tamamen aynı işlevi gören yazılım parçaları ile sağlanmıştır. Hazırlanan program, verilen tüm farklı çalışma alanlarında, farklı geometriye sahip engellerin bulunduğu yapılandırma alanlarında başlangıç noktasından, hedef noktaya bir yol bulmuştur. Bulunan bu yol en kısa yoldur. Nesnelere ayrı bir sınıf yapı içerisinde tanımlanarak kendisine ait metod, olay ve özellikler kontrol şeklinde C# uygulamasına eklenmiştir. Çalışma ortamı olarak resim editörleri ile hazırlanan bmp tabanlı 24 bit resim, hazırlanan programa girdi olarak uygulanabilmektedir. Dolayısı ile istenen çalışma ortamını denemek ve kullanmak kolaylaştırılmıştır. Çalışma ortamına ait renk bilgileri hazır olarak program üzerinden girilebilmektedir. Programın ekran çıktılarına ait resimler Şekil 5'te gösterilmiştir.



Şekil 5. Sürü robotlarla hareket planlamaya ait hazırlanan uygulamanın ekran görüntüleri

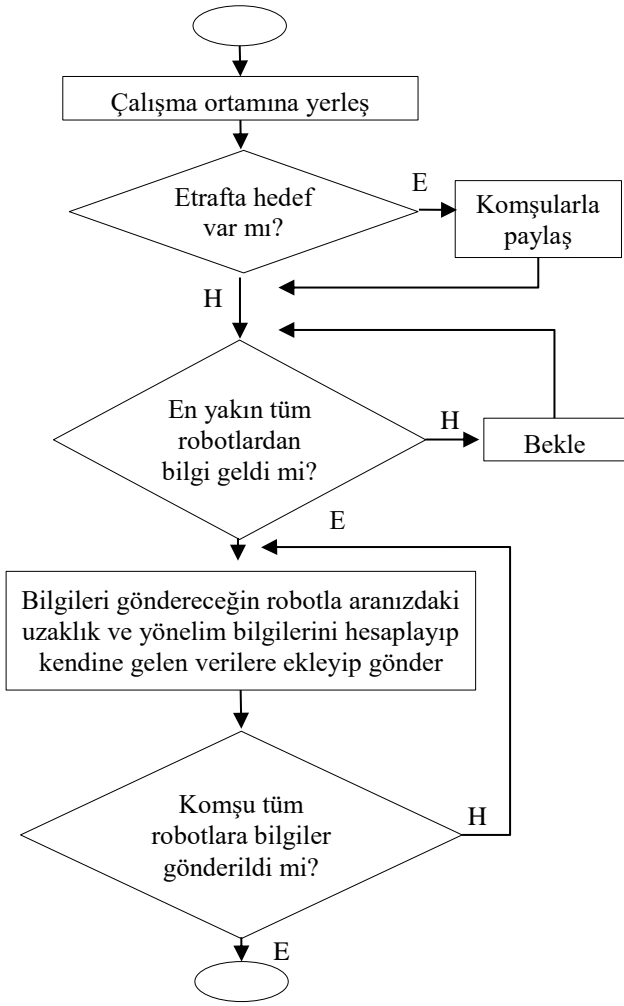
Sürüdeki birey sayısına bağlı olarak algoritmanın sonuç verme süresi uzayıp kısalabilmektedir. Diğer taraftan robot sayısı yetmeyecek miktarda seçildiğinde çözüm boş küme olabilmektedir. Çok fazla girintili, keskin kenarlara sahip engellerin yer aldığı çalışma ortamında daha fazla robotla çalışmak gerekliliği görülmüştür. Algoritmanın ayrıca labirent çözme becerisi de test edilmiştir.

Literatürdeki birçok algoritma ve yöntemden daha basit ve esnek yapıda olduğu gözlemlenmiştir. Örneğin yol düzgünleştirme tekniklerini uygulamak, belirli alanlara kısıtlama getirmek gibi yol planlama çalışmalarının tamamlayıcı unsurları rahatça uygulanabilmektedir. Literatürdeki yağın algoritmalarla karşılaştırıldığında aşağıdaki farklar karşımıza çıkmaktadır.

Trafik Polisi Algoritması yapı olarak ızgara tabanlı tekniklerine benzer bir algoritma yapısına sahiptir. Çünkü ızgara şeklinde tüm çalışma alanı birbirini gören robotlarla işgal edilmektedir. Yapıya kolayca yeni eklemeler ve düzenlemeler yapmak ve daha zeki yapılarla desteklemek mümkündür. Rasgele ağaç yapıdaki gibi bir rastgelelik olmadığından çözüm için fazla zamana ihtiyaç yoktur. Engellerin geometrisini bilmeye dayalı yol planlama çalışmalarındaki karmaşık işlemler ise bu algoritmada yer almamıştır.

### 4. Sonuç

Trafik Polisi Algoritması(TPA) yardımıyla literatürde bulunan hareket planlama tekniklerine nazaran, yazılım tabanlı bir yöntemle daha kolay, akıllı, ölçeklenebilen, esnek ve basit bir şekilde engellerden uzak bir yol bulunmuştur. Sürü robotlarla oluşturulan hareket planlama algoritması, denenmiş ve onlarca çalışma alanı içerisinde engellerin arasından en kısa yolu bulmuştur. Bu algoritmanın en büyük avantajı engellerin geometrisi, sayısı gibi parametrelerin ön hesaplaması



Şekil 4. Sürü robotlarda çalışan yazılıma ait algoritmanın akış diyagramı

Robotlar arası uzaklık koordinat düzleminde iki nokta arasındaki uzaklık formülüne göre hesaplanmıştır. Düzlemde iki  $A(x_1, y_1)$  ile  $B(x_2, y_2)$  noktaları arasındaki uzaklık Denklem 1'e göre hesaplanmıştır.

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

$x_1$  ve  $x_2$  noktaları 2D düzlemde iki robota ait pozisyonların x eksenine ait değerleri,  $y_1$  ve  $y_2$  değerleri ise iki robotun y eksen değerleridir.  $|AB|$  değeri ise iki robot arası uzaklığı temsil eder.

### 2.4. Çalışma Ortamı

Çalışma ortamında engeller, sürü robot üyeleri, başlangıç noktası, bitiş noktası ve ortam ayrı ayrı renklendirilmiştir. Her bir robot üzerindeki renk sensörü sayesinde çevre ile etkileşime geçerek ve verilen görevleri yapacaklardır. Dolayısı ile renk seçiminde birbirinden uzak tonların seçilmesi görüntünün sensörler tarafından rahatça karıştırmaksızın algılanmasını sağlayacaktır.

yapılmaksızın uygulanabiliyor olmasıdır. Uygulamalarda küçük ve çok fazla girinti çıkıntıya sahip olan engellerde çözüm kümesinin boş dönme riskine karşı daha fazla robot kullanılması gerekmektedir. En kısa yol bulduktan sonra hat üzerinde düzleştirme çalışmaları yapmak daha kolay olabilecektir. Yapılandırma alanı ile ilgili ayrıca hesap yapmaya ve yapılandırma kümesini bulmaya gerek yoktur.

İleriki çalışmalarda bu algoritmanın gerçek bir uygulaması yapılacaktır. Gerçek uygulamada belki robotlar benzer özelliklerde olacak ama ortama sürü robotları düzgün ızgara şeklinde bırakmak mümkün olmayacaktır. Bu yüzden robotların uygun pozisyonu almaları için farklı yöntemler geliştirilecek ve denenecektir.

## Kaynakça

- Bertram Raphael. (2012). IEEE Global History Network. *Institute of Electrical and Electronics Engineers*.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41. <https://doi.org/10.1007/s11721-012-0075-2>.
- Dijkstra's algorithm*. 30 Haziran 2020 tarihinde [https://en.wikipedia.org/wiki/Dijkstra's\\_algorithm](https://en.wikipedia.org/wiki/Dijkstra's_algorithm) adresinden alındı.
- F. Aurenhammer. (1991). "Voronoi diagrams—A survey of a fundamental geometric data structure". *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405. <https://doi.org/10.1145/116873.116880>.
- Philip L. Frana & Thomas J. Misa. (August 2010). "An Interview with Edsger W. Dijkstra". *Communications of the ACM*. 53 (8): 41–47. <https://doi.org/10.1145/1787234.1787249>.
- J. Kennedy, R. Eberhart. (1995). "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942–1948 vol.4. <https://doi.org/10.1109/ICNN.1995.488968>.
- Jaradat M. A., Garibeh M. H., Feilat E. A. (2012). "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field" *Springer-Verlag Soft Comput* 16:153–164. <https://doi.org/10.1007/s00500-011-0742-z>.
- J.C. Latombe. (1991). Robot Motion Planning. *Kluwer Academic Publishers*. <https://doi.org/10.1007/978-1-4615-4022-9>.
- J.C. Latombe. (1999). Motion Planning: A Journey of Robots Molecules Digital Actors and Other Artifacts, *The International Journal of Robotics Research* 30: 846–894. <https://doi.org/10.1177/02783649922067753>.
- Kavraki, L. E., P. Svestka, J-C. Latombe, ve M. Overmars. (1996). "Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces", *IEEE Transactions on Robotics and Automation*, vol. 12, issue 4, no. 4, pp. 566–580. <https://doi.org/10.1109/70.508439>.
- Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1), 90–98. <https://doi.org/10.1177/027836498600500106>.
- LaValle, S. M., & Kuffner, J. J. (2001). Randomized Kinodynamic Planning. *The International Journal of Robotics Research*, 20(5), 378–400. <https://doi.org/10.1177/02783640122067453>.
- Lozano Perez T. ve Wesley M.A. (1979). An Algorithm for Planning Collision Free Paths Among Polyhedral Obstacles, *Comm. ACM* 22(10):560–570. <https://doi.org/10.1145/359156.359164>.

- Motion Planning*. 30 Haziran 2020 tarihinde [https://en.wikipedia.org/wiki/Motion\\_planning](https://en.wikipedia.org/wiki/Motion_planning) adresinden alındı.
- Shlyakhov, N. E., Vatamaniuk, I. V, & Ronzhin, A. L. (2017). Survey of Methods and Algorithms of Robot Swarm Aggregation. *Journal of Physics: Conference Series*, 803. <https://doi.org/10.1088/1742-6596/803/1/012146>.
- Soysal, O., & Sahin, E. (2005). Probabilistic aggregation strategies in swarm robotic systems. İçinde *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.* (ss. 325–332). IEEE. <https://doi.org/10.1109/SIS.2005.1501639>.
- Steven M. LaValle. (2006). Planning Algorithms. *Cambridge University Press*. <https://doi.org/10.1017/CBO9780511546877>.
- Yıldırım Ş, Yaşar E. (2015). "An algorithm of avoiding obstacles with intelligent objects". *Recent Innovations in Mechatronics (RIIM)* Vol. 2. No. 1-2. <https://doi.org/10.17667/riim.2015.1-2/14>.
- Yıldırım Ş, Yaşar E. (2015) Development of an obstacle-avoidance algorithm for snake like robots, *Measurement Journal*, 68-73. <https://doi.org/10.1016/j.measurement.2015.04.021>.
- What Is an Object?* 30 Haziran 2020 tarihinde <https://docs.oracle.com/javase/tutorial/java/concepts/object.html> adresinden alındı.