
Araştırma Makalesi / Research Article

Üç boyutlu bir arama yüzeyi için mobil robotların yol planlaması

Gonca OZMEN KOCA^{*1}, Şengül DOĞAN²

¹Fırat Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü, Elazığ, Türkiye

²Fırat Üniversitesi Teknoloji Fakültesi Adli Bilişim Mühendisliği Bölümü, Elazığ, Türkiye

Öz

Mobil robotların yol planlaması, tek/çok katlı kapalı alan endüstriyel robotik navigasyon uygulamalarında hayati bir öneme sahiptir. Bu çalışmada, çok katlı ve her kat için birden çok katlar arası geçişin mümkün olduğu endüstriyel bir binada mobil robotların yönlendirilmesi için graf arama algoritmasına dayanan bir optimum yol planlaması önerilmiştir. Yol planlaması için başlangıç ve varış noktalarının yer aldığı iki katı içeren 3 boyutlu kübik yüzey baz alınarak mesafe hesaplanmıştır. Dijkstra graf arama algoritması, engellerden sakınarak en kısa yolun bulunması için belirlenen yüzeylerde başlangıç noktasından tüm noktalara olan mesafeleri hesapladığından dolayı çalışmada bu algoritma tercih edilmiştir. Çalışmada mobil robotun yönlendirilmesi için farklı durumları içeren iki farklı senaryo oluşturulmuştur. Bu senaryolar hedef noktasının mobil robotun bulunduğu kattaki başlangıç noktası ile aynı katta ve farklı katta olması durumlarını içermektedir. Makalede, MATLAB ortamında elde edilen en kısa yolu gösteren benzetim sonuçları sunulmuştur.

Anahtar kelimeler: Yol planlama, Optimum yol, Graf arama algoritması, Dijkstra algoritması

Three dimensional searching surface path planning of mobile robots

Abstract

Route planning of mobile robots has vital importance in the industrial robotic navigation applications of single / multi-floors closed area. In this study, it is proposed to apply an optimal route planning based on the graph search algorithm for orienting mobile robots in multiple floors industrial building where multiple exits are possible for each floor. For the route planning, the distance is calculated on the basis of a 3-dimensional cubic surface containing two floors, in which the starting and destination points are located. Dijkstra graph searching algorithm is preferred since it calculates the distances from starting point to all points in specified surfaces in order to create shortest route avoiding obstacles. In the study, two different scenarios including different situations are created for orienting the mobile robot. These scenarios include the situation of the destination point on the same and on the different floor with the starting point of the mobile robot. Simulation results of the shortest routes are presented in the paper using MATLAB environment.

Keywords: Route planning, Optimum route, Graph searching algorithm, Dijkstra algorithms.

1. Giriş

Birçok araştırma alanında navigasyon ve en kısa yol optimizasyonu problemlerinin çözümünde kullanılan graf arama algoritmaları robotik sistemlerde de çoğunlukla tercih edilmektedir. Bu arama algoritmalarından olan Dijkstra algoritması sınırları belirli kısıtlı alanlarda, optimum yolun bulunması konusunda iyi bir performansla sahiptir [1-3].

Dobrilovic vd. [3] depolarda iç taşıma yolunun optimizasyonu için Dijkstra algoritmasını kullanmışlardır. Oluşturdukları model ile giriş/çıkış ve ağırlıkları taşımak, kaldırmak ve istif etmek için kullanılan iş makinalarına göre iki farklı durum için optimum yolun bulunması hedeflenmektedir. Bu çalışma ile üretkenliği arttırıp, işletme maliyetini düşürmek amaçlanmıştır. Chen vd. [4] tarafından, doğal afet, ani terör saldırıları veya nükleer kazalarda insanların güvenli bölgelere araçlarla sevk

*Sorumlu yazar: gozmen@firat.edu.tr

Geliş Tarihi: 22.10.2018, Kabul Tarihi: 17.01.2019

edilmesinde yol ağı gözlemlenerek, kavşaklardaki gecikmeler, araçların sabah, akşam ve rutin zamanlarda tahliye hızı dikkate alınarak Dijkstra algoritmasına dayalı bir araç tahliye modeli geliştirilmiştir. Garcia vd. [5] insansız hava araçlarının tehlikeli hava koşullarının bulunduğu bölgelerden sakınarak en kısa yol eğrisinin hesaplanması için birkaç graf arama algoritmasının avantaj ve dezavantajlarını tartışmışlardır. Bu çalışma benzetim ve CATEC deney seti üzerinde deneysel olarak gerçekleştirilmiştir. Bu çalışmada ayrıca arama yapılacak hava sahasının ayrıklaştırılmasında bağlantı katsayısının önemi vurgulanmıştır. Engellerin bulunduğu serbest bir alanda Öklid en kısa yol probleminin Dijkstra ve A* algoritmasına uyarlanması Yershov ve LaValle [6] tarafından gerçekleştirilmiştir.

Jiang vd. [7] yaptıkları çalışmada, SDN topolojisinden elde edilmiş bir graf için kenar ağırlıklarının yanında düğüm ağırlıklarını da dikkate alan genişletilmiş bir Dijkstra en kısa yol algoritması uygulamasına yer vermiştir. Bu çalışmada, ağ üzerinden her alıcı için alıcıdan-sunucuya ve sunucudan-alıcıya en kısa yol üretmek amacıyla üç farklı Dijkstra algoritması denenmiştir. Birim-ağırlıklı Dijkstra ve orijinal Dijkstra algoritmalarıyla kıyaslandığında genişletilmiş Dijkstra algoritmasıyla daha iyi sonuçlar elde edilmiştir.

Bilgi alışverişi veri iletim sürecini optimize etmek için en kısa yol algoritmalarına ihtiyaç duyar. Cihazın işlem gücü ve hafıza sınırlamalarından dolayı en kısa yol bulma algoritmalarının sadeliği önemli bir konu olmuştur. Djojo ve Karyono yaptıkları çalışmada [8], A*, Floyd-Warshall ve Dijkstra algoritmalarını; hesaplama yükü, süre ve hafıza kullanımı açısından karşılaştırmışlardır. A* algoritması her düğümü kontrol etmediği için hesaplama yükü ve süresi açısından en avantajlı olanıdır. Hafıza kullanımı açısından en avantajlı olanı bu algoritmalar içerisinde en az değişken tutucu gerektiren Dijkstra algoritmasıdır. Kang vd. [9] tarafından mobil robotlar için yol planlaması açısından orijinal Dijkstra, geliştirilmiş Dijkstra ve parçacık sürü optimizasyonunun kullanıldığı geliştirilmiş Dijkstra algoritmaları denenerak karşılaştırılmıştır. En iyi sonuçlara çalışmayı yapan ekibin önerdiği parçacık sürü optimizasyonunun kullanıldığı geliştirilmiş Dijkstra algoritması ile ulaşılmıştır.

Fadzli vd. [10] yaptıkları çalışmada, GPS koordinatlarının ve pusula yönlendirmesinin güvenilir olmadığı kapalı ortam uygulamalarında robot yol planlaması için Dijkstra algoritmasına dayalı daha kapsamlı veri yapısı sağlayan çok katmanlı bir sözlük kullanmıştır. Bu çalışmada, önerilen yapı geleneksel algoritma yapısından farklı olarak yolun uzunluğunun yanında zorluk derecesini de göz önünde bulundurmaktadır. Geleneksel Dijkstra algoritmasıyla kıyaslandığında, önerilen yöntemin dönüşlerde daha az bir toplam açığa sahip olduğu görülmüştür. Buda daha kısa bir yol anlamına gelmektedir. Zhang ve Wei [11], gömülü coğrafi bilgi sistemi için optimize edilmiş bir Dijkstra algoritması geliştirmiştir ve bu optimizasyon ile en kısa yolun bulunmasında iyileşmeler olduğu gözlenmiştir. Ancak bu alanda yapılan çalışma bir başlangıç niteliğindedir. Gelecekte trafik ışıkları gibi farklı faktörler dikkate alınarak çalışmanın geliştirilmesi hedeflenmektedir.

Chao ve Hongxia çalışmalarında [12], yol aramadaki işlem hızını artırmak ve hesaplama süresini azaltmak için Dijkstra algoritmasını geliştirilmeyi hedeflemişlerdir. Önerilen yöntem ile gerçek ulaşım yolunun seçiminde daha kapsamlı bir referans çerçeve oluşturmak için farklı türde ağırlık değerleri kullanılmıştır. Bu yöntem ile öncelikli bölgeler belirlenerek anlamsız aramalardan kurtulup arama verimliliğini arttırarak operasyon süresini azaltmak amaçlanmıştır. Ancak haritada başlangıç noktası ile bitiş noktasının diyagonal olduğu durumlarda önerilen yöntemdeki öncelikli bölgeler ile klasik algoritmadaki bölgeler büyük oranda farklılık göstermez. Bu konuda çalışmayı gerçekleştiren ekibin de çalışmada belirttiği gibi verimliliğin arttırılması için algoritmanın geliştirilmesi gerekmektedir.

Graf arama algoritmaları içinde sınırlı alanlarda en kısa yolu bulurken, Dijkstra algoritması kullanılarak aramanın doğruluğu ve mevcut alternatifler içinde en kısa yolun bulunması açısından iyi bir performans elde edilir. Bu çalışmada, çok katlı her katta birden çok katlar arası geçişin mümkün olduğu endüstriyel bir binada mobil robotların kat içi ve katlar arasında taşıma yaparken en kısa yolu bulması için Dijkstra algoritmasına dayalı 3 boyutlu bir yol planlama sistemi gerçekleştirilecek ve sistem performansı benzetim çalışmalarıyla sunulacaktır. Çalışmada;

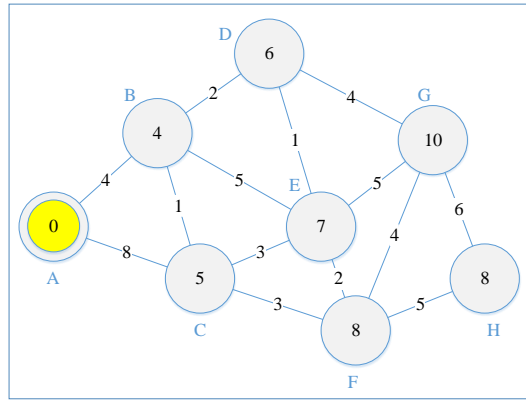
- Dijkstra graf arama algoritmasının çalışma mantığı gereği belirlenen yüzeylerde başlangıç noktasından tüm geçiş noktalarına olan mesafeyi hesaplaması nedeni le birden çok katlar arası geçişin mümkün olduğu çok katlı bir binada en kısa yol planlaması Dijkstra graf arama algoritması kullanılarak yapılmıştır.

- Burada diğer arama algoritmalarından farklı olarak tüm geçişlere olan mesafeler bir bütün olarak hesaplanmıştır. Bu da Dijkstra graf arama algoritmasının hesaplama yükü dezavantajını bu sistem için avantaja çevirmiştir.

Çalışmanın 2. bölümünde en kısa yol problemi tanımlanmıştır. Dijkstra algoritmasına dayalı 3 boyutlu yol planlama sistemi bölüm 3'de ayrıntılandırılmıştır. Bu bölümde ayrıca Dijkstra algoritmasına ait işlem basamakları anlatılmıştır. 4. bölümde benzetim örnekleri ile sistem performansı ortaya konulmuştur. Son bölümde sonuçlar irdelenmiş ve gelecek çalışmalar için öneriler sunulmuştur.

2. En kısa yol problemi

En kısa yol problemi, belirli bir maliyet ile mevcut yüzeyde belirlenen noktalar arasında bir yolun varlığının tespit edilmesidir. En kısa yol probleminin çözümü için Dijkstra, Bellman ve Ford, A* gibi algoritmalar geliştirilmiştir. Bu algoritmalar genel yapısı ile aynı amacı gütmekle beraber çözüm olarak belirli farklılıklar ile ayrılabilirler. Örneğin Dijkstra algoritması, yol tespit edilirken bulunan yüzeyde tüm noktalara ulaşım mesafesi hesaplayabildiği gibi A* algoritması sadece belirlenen iki nokta arasındaki mesafe hesabı ile ilgilenmektedir. Belirlenen yüzey için A noktasından diğer tüm noktalara olan uzaklık Şekil 1'de gösterilmiştir [13- 15].



Şekil 1. En Kısa Yol Probleminin Şematik Gösterimi

Şekil 1'de belirlenen noktalar ve uzaklıklar için hesaplanan komşuluk matrisi Şekil 2'de verilmiştir [13].

A	A	B	C	D	E	F	G	H
A	-1	4	8	-1	-1	-1	-1	-1
B	4	-1	1	2	5	-1	-1	-1
C	8	1	-1	-1	3	3	-1	-1
D	-1	2	-1	-1	1	-1	4	-1
E	-1	5	3	1	-1	2	5	-1
F	-1	-1	3	-1	2	-1	4	5
G	-1	-1	-1	4	5	4	-1	6
H	-1	-1	-1	-1	-1	5	6	-1

Şekil 2. Komşuluk Matrisi

Şekil 2'de verilen matriste belirlenen noktaların direk komşuluğunun olmadığı noktalar "-1" ile temsil edilmektedir. Şekil 1 ve 2 dikkate alındığında, en kısa yol algoritmalarının çalışma mantığına göre başlangıç noktası A'dan varış noktası olan E'ye gidiş ile ilgili mevcut güzergahlar (A-B-E, A-C-E, A-B-C-E, ...) içerisinde en kısa mesafeye sahip yol tercih edilmelidir. Aynı şekilde A noktasından C noktasına gitmek için algoritma tarafından farklı varyasyonlar denenerek en kısa mesafe bulunmalıdır.

A'dan C'ye gidiş için;

1. yol: A ile C arasındaki direk yol kullanılabilir ve A'dan C'ye uzunluk Şekil 1'e göre 8 birimdir.

2. yol: A-B-C yolu kullanılır ise A-B arası 4 birim, B-C arası 1 birim olduğu için toplam mesafe $4+1$ birim olacaktır.

Dolayısıyla en kısa yol algoritması 1. yol yerine 2. yolu tercih etmelidir.

3. Dijkstra algoritmasına dayalı yol planlama sistemi

3.1. Dijkstra algoritması

Dijkstra algoritması belirli bir alanda (A_{ij}) başlangıç düğümünden ($b(i, j)$) bu alanda yer alan bütün düğümlere ($A_{ij} \rightarrow i \in \{0, \dots, m\}, j \in \{0, \dots, n\}$) olan en kısa yolun elde edilmesi için kullanılan bir graf arama algoritmasıdır. Belirlenmiş alan düğüm adı verdiğimiz küçük gridlere bölünür. Başlangıç düğümünden başlanarak tüm düğümlere olan en kısa mesafeler ($S_r \rightarrow b(i, j)$ ' den r . düğüme olan en kısa mesafe $r \in \{0, \dots, (m \times n)\}$) hesaplanmak üzere geliştirilen bu algorithmada denklem (1)'de sunulduğu gibi bir etiketleme yapısı kullanılır.

$$[S_t, r] = [(S_r + p_{rt}), (r)] \quad (1)$$

Burada p_{rt} , r . düğüm ile t . düğüm arasındaki mesafeyi göstermektedir. Ayrıca düğümler geçici ve kalıcı olmak üzere de etiketlenir. Bu algoritma için işlem adımlarını gösteren sözde kod Tablo 1'de verilmiştir [14, 15].

Tablo 1. Dijkstra algoritmasının sözde kodu

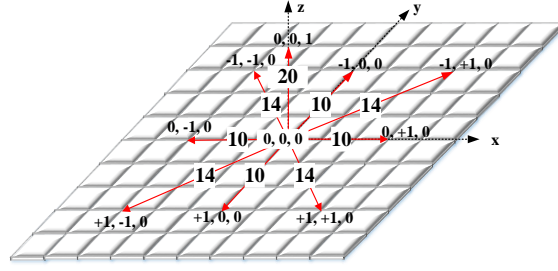
1/	$b(i, j)$ ekle \rightarrow kalıcı_liste;
2/	$r=1$;
3/	{ Döngü {
3/1/1	Etiketleme yap $[S_t, r] = [(S_r + p_{rt}), (r)]$; (r . düğümün tüm t komşuları için)
3/1/2	t düğümlerini ekle \rightarrow geçici_liste;
3/1/3	Döngü bitir; }
3/2	Eğer { t . düğüm önceden etiketlenmişse ($[S_t, v]$)
3/2/1	Eğer { $(S_k + p_{kt}) < S_t$ { $[S_t, v] \leftarrow [(S_r + p_{rt}), (r)]$; } }
3/3	Eğer { $(\forall A_{ij} \in$ kalıcı_liste) {bitir}
3/4	$r = \min(\text{geçici_liste})$; }

Bu algorithmada tüm düğümler için erişilme maliyeti hesaplanarak en kısa yolun bulunması garanti edilmiş olur. Daha iyi bir erişim maliyetinin olmadığı yani daha kısa bir yolun bulunmadığı durumda düğüm kalıcı listeye eklenir. Tüm düğümler kalıcı listeye eklendiğinde program durdurulur [14-17].

3.2. Üç boyutlu yol planlama sistemi

Bu çalışmanın hedefi N katlı bir binada her kat için katlar arası geçiş noktalarının ($G_s \leftarrow \{s=1, 2, \dots, g\} \leftarrow \{g=4\}$) varlığında kat içinde ve katlar arasında gezebilen bir mobil robot için yol planlama sistemi oluşturmaktır. Katlar arası geçişlerde dikkate alındığında başlangıç noktasının ($b(i, j, k)$) bulunduğu kat ile hedef noktanın ($h(i, j, k)$) bulunduğu kat alt veya üst yüzeyleri oluşturmak üzere 3 boyutlu kübik yüzey üzerinde arama yapılmaktadır.

Hesaplamalarda, Şekil 3'de gösterildiği gibi bulunulan noktadaki indis (i, j, k) komşu noktadaki indisler (kom_i, kom_j, kom_k) ile ifade edilirse, farklı durumlar için mesafe değerleri denklem (2) ile hesaplanır.



Şekil 3. Her bir noktanın komşularına olan mesafelerinin şematik gösterimi

$$\left\{ \begin{array}{l} (i = kom_i \vee j = kom_j) \rightarrow p_rt = 10 \\ (i \neq kom_i \vee j \neq kom_j) \rightarrow p_rt = 14 \\ (i = kom_i \wedge j = kom_j \wedge k = kom_k) \rightarrow p_rt = 20 \end{array} \right\} \quad (2)$$

Denklem (2)'de p_rt gidilecek mesafeyi temsil etmektedir. Oluşturulan 3 boyutlu yol planlama sisteminde aramalar Tablo 2'de sözde kodu verilen hesaplama adımları ile gerçekleştirilmektedir.

Tablo 2. Yol planlama sisteminin mesafe hesaplama adımlarının sözde kodu

1/	$r \leftarrow b(i, j, k), t \leftarrow h(i, j, k);$
2/	Eğer $\{ b(...,k) \neq h(...,k) \}$ {
2/1	$k \leftarrow b(...,k);$
2/2	$hesapla \rightarrow p_rG_s \forall G_s \in A_ijk;$
2/3	$k \leftarrow h(...,k);$
2/4	$hesapla \rightarrow p_G_{st} \forall G_s \in A_ijk;$
2/5	$p_rt = p_rG_s + p_G_{st} + (h(...,k) - b(...,k)) * 20;$
2/6	$min(p_rt); \}$
2/1	$\{ hesapla \rightarrow p_rt \forall r \wedge t \in A_ijk \{ k \leftarrow b(...,k) \}$

Mobil robotun başlangıç noktası ile ulaşılmak istenen hedef noktanın aynı katta olması durumunda, bulunulan kat numarası k katsayısını gösterir ve kat içinde r başlangıç düğümünden t hedef düğümüne mesafe denklem (2) ve Tablo 2'de $\{ b(...,k) = h(...,k) \}$ şartına karşılık gelen 2/1 nolu işlem adımı dikkate alınarak hesaplanır. Mobil robotun bulunduğu kat ile hedef noktanın bulunduğu katın farklı olması durumunda ise, mesafe hesaplaması Tablo 2'de $\{ b(...,k) \neq h(...,k) \}$ şartına karşılık gelen 2/5 nolu işlem adımı dikkate alınarak ve tüm geçişlere ait olasılıklar değerlendirilerek hesaplanır.

4. Benzetim çalışmaları ve sonuçlar

Bu çalışmada, mobil robotun bir başlangıç noktasından kat içinde veya katlar arasındaki bir hedef noktaya en kısa yoldan engellerden sakınarak ulaşması için Dijkstra algoritmasına dayalı 3 boyutlu bir yapı oluşturulmuştur. Çalışmada aynı katta ve farklı kattaki hedeflere ulaşmak için farklı rasgele değerler kullanılarak iki senaryo oluşturulmuş ve algoritmanın en kısa yol bulma performansı elde edilmiştir.

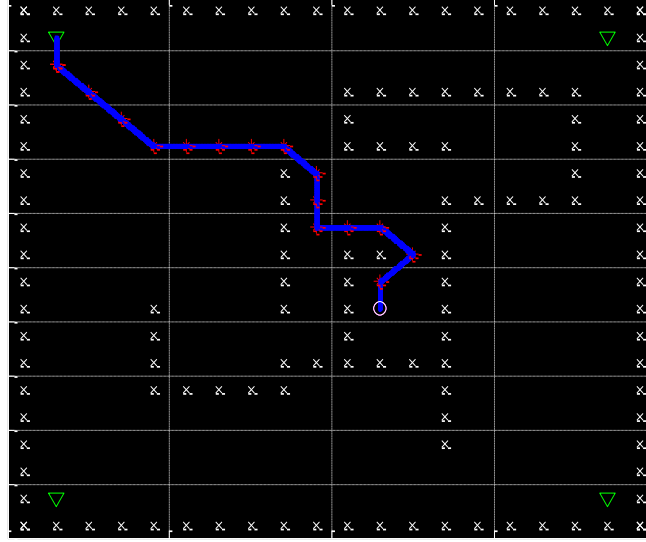
1. Senaryo: Bu senaryoda ulaşılmak istenen hedef, başlangıç noktası ile aynı kattadır. Tablo 3'de 1. Senaryo için verilen rasgele seçilmiş engel, başlangıç ve hedef noktaları için elde edilen güzergâh Şekil 4'de sunulmuştur. Oluşturulan senaryoda başlangıç noktasından güzergâhlar $\{ \forall G_s \in A_{ijk} \}$ için Tablo 4'de verilmiştir.

Tablo 3. 1. Senaryo için mobil robotun bulunduğu katta rasgele seçilmiş engel, başlangıç ve hedef noktalarının koordinatları

	Mobil robotun olduğu kattaki engeller	Başlangıç noktası	Hedef noktası
Koordinatlar	(11:14,5) (14,6:9) (13,9:13) (6:11,9) (10:11,11) (9,11:12) (3,11-18) (4:7,18) (7,14:17) (8:16,14)	(11,12)	(19,2) (19,19) (2,2) (2,19)

Tablo 4. 1. Senaryo için mobil robotun bulunduğu katta başlangıç noktasından tüm geçiş noktalarına izlenecek güzergâh koordinatları

İzlenecek Güzergâhlar				
Adım No	1.Geçiş	2.Geçiş	3.Geçiş	4.Geçiş
1	11,12	11,12	11,12	11,12
2	12,11	12,11	10,12	10,12
3	12,10	12,10	9,13	9,13
4	12,9	12,9	8,12	8,13
5	12,8	12,8	8,11	7,13
6	12,7	12,7	8,10	6,13
7	12,6	12,6	7,10	6,12
8	11,6	11,6	6,10	6,11
9	10,5	10,5	5,9	5,10
10	11,4	11,4	5,8	4,10
11	12,4	12,4	5,7	3,10
12	13,4	13,4	5,6	2,11
13	14,4	14,4	5,5	2,12
14	15,4	15,5	4,4	2,13
15	16,4	15,6	3,3	2,14
16	17,4	15,7	2,2	2,15
17	18,3	15,8		2,16
18	19,2	15,9		2,17
19		15,10		2,18
20		15,11		2,19
21		15,12		
22		15,13		
23		16,13		
24		17,14		
25		17,15		
26		17,16		
27		17,17		
28		18,18		
29		19,19		



Şekil 4. 1. Senaryo için en kısa yol güzergâhı

Şekilde “yeşil ∇ ” ile simgelenen sol alt köşe, sağ alt köşe, sol üst köşe ve sağ üst köşe sırasıyla 1, 2, 3, 4 nolu geçiş noktalarını göstermektedir. “Beyaz x” ile sınırlar ve engel noktaları simgelenmiştir. Mobil robotun başlangıcı “beyaz o” ile en kısa yol güzergâhı üzerinde uğranan her nokta “kırmızı x” ile gösterilmiştir. En kısa yol güzergâhı “mavi” ile verilmiştir. Başlangıç noktasından her bir geçiş noktasına mesafeler sırasıyla 1, 2, 3, 4 nolu geçiş noktaları için 190, 308, 174, 202 olarak hesaplanmıştır. Bu durumda rasgele seçilen değerlerle oluşturulan 1. Senaryo için 3 nolu geçiş noktasının kullanılması en kısa yolu oluşturur.

2. *Senaryo*: Bu senaryoda ulaşılacak hedef, başlangıç noktasının bulunduğu katın bir alt katında bulunmaktadır. Bu senaryoda en kısa yol hesaplaması üç boyutta gerçekleştirilmiştir. Dijkstra algoritması, bulunulan katta başlangıç noktasından tüm noktalara olan mesafeleri hesapladığı için tüm geçiş noktalarına (her katta 4 geçiş) olan mesafeler, katlar arası mesafe ve hedef noktanın bulunduğu katta tüm geçiş noktalarından hedef noktaya olan mesafeler dikkate alınarak üç boyutlu kübik yüzeyde hesaplanır. Mesafeler hesaplanırken aynı kat yüzeyi için x - y düzleminde Şekil 3’de verilen mesafeler dikkate alınırken katlar arası için z düzleminde her kat arası mesafe 20 birim olacak şekilde belirlenmiştir.

Tablo 5’de 2. *Senaryo* için verilen engel, başlangıç ve hedef noktaları için elde edilen güzergâh mobil robotun bulunduğu ve hedef noktanın yer aldığı bir alt kat için sırasıyla Şekil 5.a ve b’de sunulmuştur. 2. *Senaryo*’da başlangıç noktasından hedef noktasına güzergâhlar mobil robotun bulunduğu kat ve hedef noktanın bulunduğu için sırasıyla Tablo 6 ve 7’de verilmiştir.

Tablo 5. 2. *Senaryo* için rasgele seçilmiş engel, başlangıç ve hedef noktalarının koordinatları

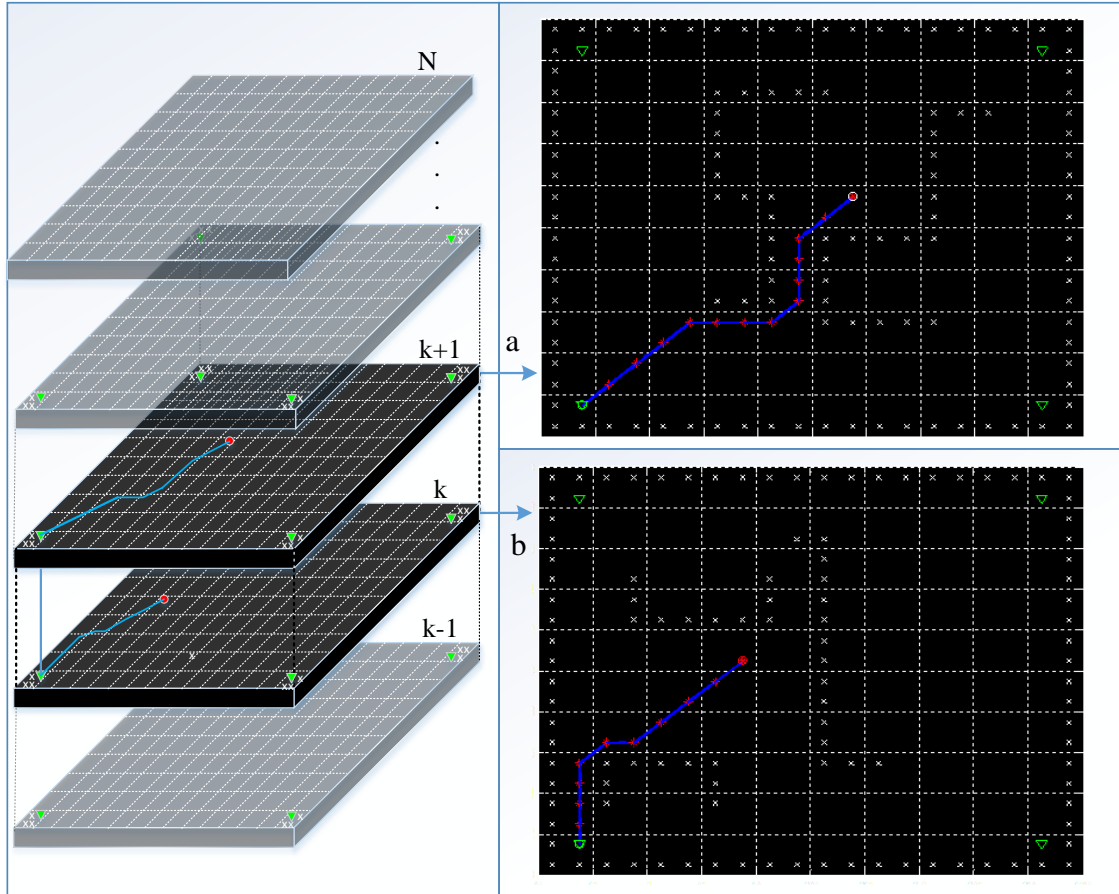
	Mobil robotun olduğu kattaki engeller	Hedef noktanın olduğu kattaki engeller	Başlangıç noktası	Hedef noktası
Koordinatlar	(4:9,7) (4,8-11) (9,8:9) (10:14,9) (14,7:8) (11:15,11) (15,12:15) (11,12:15) (5:10,15) (5,16:17)	(6:8,6) (8,7:9) (6:7,9) (15:17,3) (15,4:7) (16:17,7)	(9,12)	(10,8)

Tablo 6. 2. *Senaryo* için mobil robotun bulunduğu katta başlangıç noktasından geçiş noktalarına izlenecek güzergâh koordinatları

İzlenecek Güzergâhlar				
Adım No	1.Geçiş	2.Geçiş	3.Geçiş	4.Geçiş
1	9,12	9,12	9,12	9,12
2	10,11	10,11	8,12	8,13
3	11,10	11,10	7,12	7,14
4	12,10	12,10	6,12	6,14
5	13,10	13,10	5,12	5,14
6	14,10	14,10	4,12	4,15
7	15,9	15,10	3,11	4,16
8	15,8	16,11	3,10	4,17
9	15,7	16,12	3,9	3,18
10	15,6	16,13	3,8	2,19
11	16,5	16,14	3,7	
12	17,4	16,15	3,6	
13	18,3	16,16	3,5	
14	19,2	17,17	3,4	
15		18,18	3,3	
16		19,19	2,2	

Tablo 7. 2. *Senaryo* için hedef noktanın bulunduğu katta geçiş noktalarından hedef noktasına izlenecek güzergâh koordinatları

İzlenecek Güzergâhlar				
Adım No	1.Geçiş	2.Geçiş	3.Geçiş	4.Geçiş
1	10,8	10,8	10,8	10,8
2	11,7	11,9	9,7	9,9
3	12,6	12,10	9,6	8,10
4	13,5	13,10	9,5	7,10
5	14,4	14,10	9,4	6,10
6	14,3	15,10	8,3	5,10
7	15,2	16,11	7,2	4,9
8	16,2	16,12	6,2	3,10
9	17,2	16,13	5,2	3,11
10	18,2	16,14	4,2	3,12
11	19,2	16,15	3,2	3,13
12		16,16	2,2	3,14
13		17,17		3,15
14		18,18		3,16
15		19,19		3,17
16				3,18
17				2,19



Şekil 5. a. 2. Senaryo için, mobil robotun bulunduğu katta en kısa yol güzergâhı b. hedef noktanın bulunduğu katta en kısa yol güzergâhı

Şekil 5.a’da mobil robotun bulunduğu katta 1, 2, 3, 4 nolu geçiş noktalarına olan mesafeler Dijkstra arama algoritması tabanlı yapı kullanılarak sırasıyla 158, 174, 158, 110 birim olarak elde edilmiştir. Şekil 5.b’ de hedef noktasının bulunduğu katta 1, 2, 3, 4 nolu geçiş noktalarından hedef noktaya olan mesafeler sırasıyla 120, 160, 122, 180 birim bulunmuştur. Hedef noktanın mobil robotun bulunduğu katın bir alt katında olduğu varsayılarak katlar arası mesafe de dikkate alınmıştır. Buna göre mobil robotun hedef noktaya ulaşmak için 1, 2, 3, 4 nolu geçişlerden her birini kullanması durumunda kat edeceği mesafeler sırasıyla 298, 354, 300, 310 birim olarak elde edilmiştir. Bu durumda rasgele seçilen değerlerle oluşturulan 2. Senaryo için 1 nolu geçiş noktasının kullanılması en kısa yolu oluşturur.

5. Sonuçlar ve tartışma

Bu çalışmada graf arama algoritmalarından Dijkstra algoritması bir mobil robotun kat içinde veya başka bir katta istenilen hedef noktasına engellerden sakınarak en kısa yoldan ulaşmasını sağlamak amacıyla kullanılmıştır. Dijkstra arama algoritması, başlangıç noktasından başlar ve noktanın etrafındaki 8 komşu için mesafe değerlerini hesaplayarak tüm noktaların giriş noktasına olan uzaklıklarını tespit eder. Bu yönüyle çalışmada katlar arası harekette her kat için 4 adet geçiş noktası olduğundan ve en kısa yolun elde edilmesinde her katta tüm geçişlere olan mesafelerin bulunması gerektiğinden Dijkstra algoritması tercih edilmiştir.

Farklı durumları içeren başlangıç, hedef ve engel noktaları için rasgele değerlerin kullanıldığı iki farklı senaryo oluşturulmuştur. 1. Senaryo bir kat içinde rasgele engellerin varlığında mobil robotun bir hedef noktasına en kısa yoldan gitmesini hedefler. Burada hedef noktası engellerin varlığında en kısa mesafeye sahip geçiş noktasıdır. Algoritma hangi geçişin hedef olacağını her birine olan mesafeleri hesaplayarak karar verir. 2. Senaryo’ da mobil robotun bulunduğu katın dışında başka bir kattaki hedef noktasına engellerin varlığında en kısa yoldan ulaşması test edilmiştir. Mesafeler mobil robotun

bulunduğu katta geçişlere olan mesafeler, katlar arası mesafeler ve hedef noktasının bulunduğu katta geçiş noktalarından hedef noktasına olan mesafeler dikkate alınarak hesaplanmıştır.

Belirli alanlarda doğruluğu yüksek sonuç üreten Dijkstra algoritması çalışmada elde edilen en kısa yolların güvenilirliğini artırmıştır. Dijkstra algoritması ile tüm noktalara olan mesafeler hesaplandığı için hesaplama süresinin uzun olması dezavantajı bulunmasına rağmen bu çalışmada belirli bir alan uygulaması gerçekleştirilmiştir ve geçiş noktalarının birden fazla olması her birine olan mesafelerin hesaplanması gerekliliğinden dolayı bu dezavantajı büyük oranda ortadan kaldırmıştır. Gelecek çalışmalarda Dijkstra algoritmasının geçiş noktalarının birden fazla olması durumunda hesaplama süresinin azaltılmasına yönelik iyileştirmeler hedeflenmektedir.

Kaynaklar

- [1] Galán-García J.L., Aguilera-Venegas G., Galán-García M.Á., Rodríguez-Cielos P. 2015. A new Probabilistic Extension of Dijkstra's Algorithm to simulate more realistic traffic flow in a smart city. *Applied Mathematics and Computation*, 267: 780-789.
- [2] Deng Y., Chen Y., Zhang Y., Mahadevan S. 2012. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, 12 (3): 1231-1237.
- [3] Dobrilovic D., Jevtic V., Beker I., Stojanov Z. 2012. Shortest-path based model for warehouse inner transportation optimization. In *Applied Computational Intelligence and Informatics (SACI)*, 7th IEEE International Symposium. 63-68.
- [4] Chen Y.Z., Shen S.F., Chen T., Yang R. 2014. Path optimization study for vehicles evacuation based on Dijkstra algorithm. *Procedia Engineering*, 71: 159-165.
- [5] Garcia M., Viguria, A., Ollero A. 2013. Dynamic graph-search algorithm for global path planning in presence of hazardous weather. *Journal of Intelligent & Robotic Systems*, 1-11.
- [6] Yershov D.S., LaValle S.M. 2011. Simplicial Dijkstra and A* algorithms for optimal feedback planning. In *Intelligent Robots and Systems (IROS)*, IEEE/RSJ International Conference. 3862-3867.
- [7] Jiang J.R., Huang H.W., Liao J.H., Chen S.Y. 2014. Extending Dijkstra's shortest path algorithm for software defined networking. In *Network Operations and Management Symposium (APNOMS)* IEEE, 16th Asia-Pacific, 1-4.
- [8] Djojo M.A., Karyono K. 2013. Computational load analysis of Dijkstra, A*, and Floyd-Warshall algorithms in mesh network. In *Robotics, Biomimetics, and Intelligent Computational Systems (ROBIONETICS)*, IEEE International Conference, 104-108.
- [9] Kang H.I., Lee B., Kim K. 2008. Path planning algorithm using the particle swarm optimization and the improved Dijkstra algorithm. In *Computational Intelligence and Industrial Application, PACIA'08. Pacific-Asia Workshop IEEE*, 2, 1002-1004.
- [10] Fadzli S.A., Abdulkadir S.I., Makhtar M., Jamal A.A. 2015. Robotic indoor path planning using dijkstra's algorithm with multi-layer dictionaries. In *Information Science and Security (ICISS)*, 2nd International Conference, 1-4.
- [11] Zhang D., Wei Z., Kim J.H., Tang S. 2010. An optimized Dijkstra algorithm for Embedded-GIS. In *Computer Design and Applications (ICDDA)*, International Conference, 1: 147-150.
- [12] Yin C., Wang H. 2010. Developed Dijkstra shortest path search algorithm and simulation. In *Computer Design and Applications (ICDDA)*, International Conference, 1: 116-119.
- [13] http://www.zafercomert.com/Medya/2015_05_11_2_121_69f9a888.pdf#viewer.action=download (Erişim tarihi: 01.03.2016).
- [14] Dijkstra E.W. 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1 (1): 269-271.
- [15] Boğar E., Beyhan S. 2016. A Hybrid Genetic Algorithm for Mobile Robot Shortest Path Problem, *International Journal of Intelligent Systems and Applications in Engineering*, 4: 264-267.
- [16] Wang H., Yu Y., Yuan Q. 2011. Application of Dijkstra algorithm in robot path-planning. In *Mechanic Automation and Control Engineering (MACE)*, Second International Conference IEEE, 1067-1069.
- [17] Ferone D., Festa P., Guerriero F., Laganà D. 2016. The constrained shortest path tour problem. *Computers & Operations Research*, 74: 64-77.