Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 598-603, Nisan 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 598-603, April 2020

Research Article

Çok Katlı ve Tam Otomatik Otoparklarda Depolama ve Geri İade Süreçlerinde D* Lite Algoritması ile Yol Planlama*

Hatice Demirci[1†], Erhan Akın[2], Müştak Ağrıklı[3]

[1] Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0002-5018-5205)
[2] Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0001-6476-9255)
[3] Otomatik Otopark Sistemleri San. ve Tic. A.Ş., Türkiye, İstanbul, Türkiye (ORCID: 0000-0000-0000-0000)

(Konferans Tarihi: 5-7 Mart 2020)

(**DOI:** 10.31590/ejosat.araconf72)

ATIF/REFERENCE: Demirci, H., Akın, E., & Ağrıklı, M. (2020). Çok Katlı ve Tam Otomatik Otoparklarda Depolama ve Geri İade Süreçlerinde D* Lite Algoritması ile Yol Planlama. *Avrupa Bilim ve Teknoloji Dergisi*, (Özel Sayı), 598-603.

Öz

Bu çalışmada, çok katlı tam otomatik otoparklarda depolama ve geri iade süreçlerinde en kısa yol probleminin çözümü için D* Lite algoritması optimize edilerek zaman ve enerji maliyetinin minimize edilmesi amaçlanmıştır. Çok katlı tam otomatik otoparklarda park etme veya geri verme esnasında park hücrelerinin ve taşıma asansörlerinin durum ve konumunun sürekli değişmesi nedeniyle sabit bir harita üzerinden yol planlaması yapılamamaktadır. Çalışmada dinamik koşullara uygun olarak tasarlanmış sezgisel bir algoritma ile bu sorun aşılmaya çalışılacaktır. Çok katlı tam otomatik bir otoparkta ilgili katın asansör yükle/boşalt bölümündeki bir araç park edilmek üzere kendisi için belirlenmiş olan park yerine taşınırken veya araç geri verilmek üzere park bölmesinden bulunduğu katın asansör yükle/boşalt bölmesine taşınırken kullanılacak olan yol bulma algoritması olarak D* Lite algoritması önerilmektedir. Park etme veya geri alma işlemi sırasında belirlenmiş olan en uygun rota, eş zamanlı olarak devam etmekte olan diğer işlemler nedeniyle engellerle karşılaşacağından bu engellerin kaldırılması için bu çalışmada A* algoritması ile boş hücrelerin yer değişimi sağlanmaktadır.

Anahtar Kelimeler: Tam otomatik otopark, En kısa yol problemi, Dinamik sezgisel algoritmalar, D*Lite algoritması, Optimizasyon.

Path Planning with D* Lite Algorithm for Fully-automated and Multi-Story Parking

Abstract

Abstract This study aims to minimize time and energy costs by optimizing the D * Lite algorithm for the solution of the shortest path problem in storage and retrieval processes in multi-storey fully automatic parking structure. In a multi-storey fully automatic parking structure, parking can not be planned on a fixed map due to the constantly changing status and location of the parking cells and elevators storage or retrieval. This problem will be tried to overcome with a heuristic algorithm designed in accordance with dynamic conditions. In a multi-storey fully automatic parking, the D * Lite algorithm is recommended for the pathfinding algorithm to be used when moving a vehicle to the parking space designated for it, or when moving the vehicle from the parking lot to the lift loading/unloading compartment of the floor where it is located to be returned. The most appropriate route determined during the parking or retrieval process may face obstacles due to other ongoing operations simultaneously. To remove these obstacles, in this study, empty cells are replaced by the A* algorithm.

Keywords: Fully automatic parking, Shortest path problem, Dynamic heuristic algorithms, D* Lite algorithm, Optimization.

* Bu makale ***International Conference on Access to Recent Advances in Engineering and Digitalization (ARACONF 2020)*** de sunulmuştur.
† Sorumlu Yazar

1. Giriş

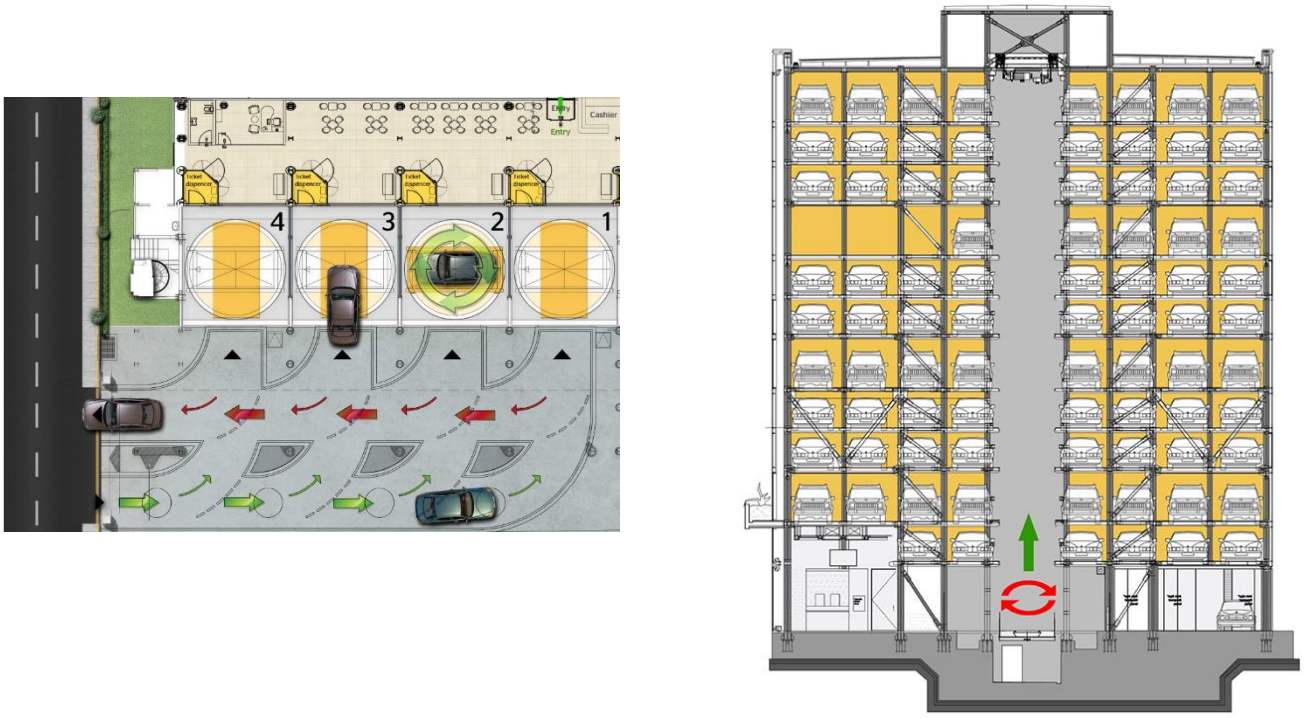
Nüfusun şehirlerde yoğunlaşması ve araç sahipliğinin artması beraberinde otopark sorununu getirmiştir. Otopark sorunlarını çözmek için Avrupa birliği ülkelerinin üyesi olduğu Avrupa Otopark Birliği (EPA 2020), ABD'de bulunan Uluslararası Otopark Enstitüsü (IPMI, 2020) ve farklı ülkelerdeki benzer organizasyonlarda otopark standartlarının geliştirilmesi konusunda çalışmalar yapılmaktadır. Ülkemizde 22.02.2018 tarih ve 30340 sayılı Resmi Gazetede yayınlanarak yürürlüğe yönetmelikle beraber otoparklarla ilgili düzenlemeler yenilenmiş, bunun yanı sıra mekanik otopark tanımı ve temel düzenlemeleri yapılmıştır. İlgili yönetmelikte mekanik otopark, "Taşıtları asansör görevi gören düzlemler ile düşey veya yatay olarak hareket ettirebilen, gerektiğinde insan eliyle de kontrol edilebilen, yer altı veya yer üstünde yapılan ve kesintisiz güç kaynağı ile beslenmesi zorunlu olan ilgili ulusal ve uluslararası standartlara uygun yapılan otopark sistemi" olarak tanımlanmıştır (Resmi Gazete, 2020).

Araç park yeri sorunundan sadece sürücüler değil yayalar, işleyen trafik, toplu taşıma, çevre kalitesi ve ekonomi doğrudan ya da dolaylı olarak etkilenmektedir. Otopark alanlarının yetersiz kaldığı durumlarda yeterli kontrol sağlanamadığında, yaya ve toplu taşıma için ayrılmış alanlar park amaçlı olarak işgal edilebilmektedir. Gerekli kontrollerin sağlanamadığı bu tür durumlarda özellikle merkezi bölgelerdeki ara sokakların tamamı park halindeki araçlarla ile dolabilmektedir. Ana caddelerde çift sıralı parkların artmasıyla yol kapasitesi azalmakta, park yeri arayan araçların arama süreleri artmakta ve bunların sonucunda hava kirliliği, trafik kazaları, gürültü, gereksiz yakıt harcamaları vs. artmaktadır. Sıralanan bu temel sorunların yanı sıra park yeri sorununa bağlı olarak ortaya çıkan çok sayıda dolaylı sorun bulunmaktadır.

İş merkezleri gibi şehir merkezlerindeki kısıtlı alanların kullanma ihtiyacından hareketle, çok katlı otoparklar inşa etmek belirli bir rahatlama sağlamaktadır. Bunun yanında çok katlı otoparkların kendi içinde de barındırdığı bazı problemler mevcuttur. Çok katlı otoparklarda, bilgilendirme işaretleri veya panoları sunulmadığı sürece, sürücü herhangi bir kattaki uygun olan park yerinin nerede olduğunu bilemeyebilmektedir. Uygun park yeri için bulunulan katta veya katlar arasında daire çizilmesi gerekebilmektedir. Bu, arama durumu önemli miktarda zaman kaybı, yakıt tüketimi, kirlilik ve araç yıpranmasına neden olmaktadır (Yardım ve Ağrıklı, 2005).

Tam otomatik otoparklar, otoparklarla ilgili yaşanan olumsuzlukların mevcut en verimli çözümü durumundadır. Mekanik otopark çeşitleri ve mekanik olmayan otopark sistemleri incelendiğinde, ülkemiz şartlarında minimum alanlarda maksimum kapasiteye ihtiyaç duyulan şehir merkezi gibi alanlarda otomatik mekanik sistemlerin kullanılması önerilmektedir (Yardım ve Ağrıklı, 2005).

Tam otomatik otoparklarda rampa, araç manevra alanı, yaya merdiveni, yaya asansörü gibi alanlar araç sayısı için değerlendirmiş olunur. Aydınlatma, havalandırma, hafriyat, betonarme, elektrik ve mekanik alt yapı maliyetleri azaltılarak daha fazla araç alanı oluşturulmaktadır. Otomatik otopark sistemlerinde sürücüler otopark içerisine girmeyip araçlarını kabul odasına bırakırlar ve araçlar akıllı sistemler tarafından otomatik taşıyıcılarla boş park hücrelerine taşınırlar. Bu sistemler dünyada yaygınlaşmakla birlikte ülkemizde de başarılı örnekleri mevcuttur. Ülkemizdeki örnekler incelenerek İzmir Alsancak otopark uygulama örneği üzerinden yapılan analizlerde sistemin katma değer ve sosyal fayda sağladığı görülmüştür (Can ve Ilıcalı, 2019). Alsancak tam otomatik otoparkının çizimi Şekil 1'de gösterilmektedir.



Şekil 1. İzmir Alsancak Tam Otomatik Otoparkı a) Araç Teslim b) Dikey Kesit (Parkolay, 2020).

2. Materyal ve Metot

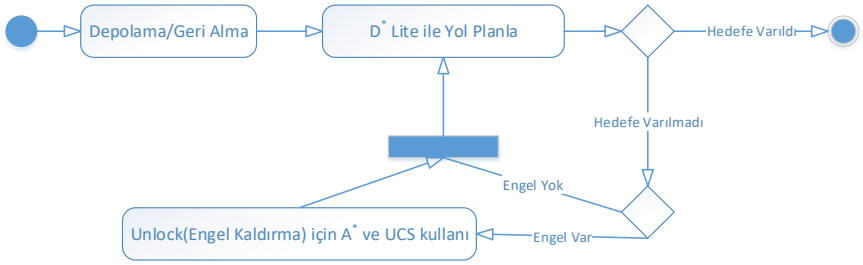
2.1. Yol Planlaması İçin Arama Algoritmaları

Çok katlı tam otomatik otoparklar, şehirlerde ciddi bir problem haline gelmiş olan otopark sorunu için verimli bir çözüm olmakla beraber kurulumu ve işletilmesi maliyetlidir. Otoparkın işletimi sırasında park işlemi için uygun hücresinin seçimi, aracın park hücresine götürülmesi ve geri verme işlemi sırasında izlenecek yol temel problemlerdendir. Aracın tam otomatik otopark içerisinde izleyeceği yolun optimize edilmesi zaman verimliliğini ve buna bağlı bağlı enerji sarfiyatını ayrıca bekleme sürelerinin düşürülerek buna bağlı müşteri memnuniyetinin arttırılmasını doğrudan etkilemektedir. Park etme ve geri verme işlemi sırasında izlenecek yol optimize edilmediği takdirde araç için bekleme süresi uzayacak ve otoparkın kullanılabilirliği azalacaktır. Kullanılacak olan yol bulma algoritması, sistemin enerji ve zaman verimliliğini belirleyecek ana faktör durumundadır.

Ele alınan sorun, dinamik bir ortamda her biri farklı başlangıç ve hedef konumlarına sahip birden çok yolun eşzamanlı olarak planlanmasını gerektirmektedir. Bir aracın konumu, atanan park yerine veya asansöre doğru değiştirilirken, robotik taşıyıcılar yeniden konumlandırılır ve bunun sonucu olarak da hareket halindeki diğer arabaların yolu tıkanabilir veya engellenebilir. Bu nedenle, yol planlama algoritması sürekli olarak artımlı bir şekilde yeniden planlayabilmeli ve bunu yaparken orijinal planın büyük bir kısmını mümkün olduğu ölçüde koruyabilmelidir.

Bu çalışmada depolama/geri alma işlemlerinin farklı aşamalarında bir dizi farklı arama algoritması kullanılmıştır:

- Yol planlaması için D* Lite algoritması,
- Yer değiştiren aracın yolundaki engellerin kaldırılması için UCS ve A* algoritmaları.



Şekil 2. Dinamik yol planlaması için UML diyagramı

2.1.1. D Lite Algoritması*

D* Lite, artımlı bir sezgisel arama algoritması olan LPA*'a dayanan Sven Koenig ve Maxim Likhachev tarafından geliştirilmiş artımlı bir sezgisel arama algoritmasıdır. D* Lite arama algoritması, D* arama algoritmasına dayanmaz ancak aynı davranışı uygular bunu yanında daha anlaşılabilir, kodlanması daha kolay ve ürettiği sonuçlar bakımından D* ile aynı etkinliğe sahip bir algoritmadır (Koenig ve Likhachev, 2002).

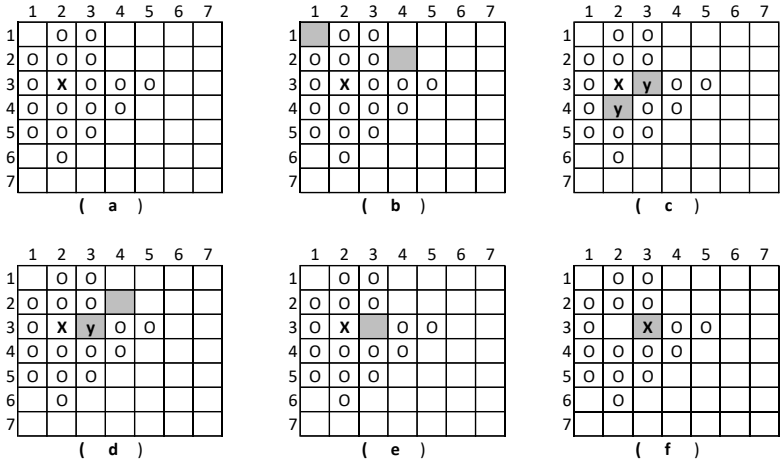
D* Lite algoritması, ortamdaki her park yerinin g(s) değerini varış noktasına olan uzaklık olarak belirleyerek varış noktasına aracın ulaşabilmesi için yol planlaması yapmaktır. Araç bulunan en uygun yol üzerinde ilerlerken ortamda değişiklik fark ettiğinde ortamdaki park yerlerinin g(s) değerlerini güncelleyerek yeni bir yol hesaplamasında bulunur.

2.1.2. UCS ve A Algoritması*

Sezgisel hesaplamaları kullanan ilk arama algoritması A* algoritmasıdır (Hart ve ark., 1968). A* arama algoritması değişmeyen ortamlarda kullanılmak üzere geliştirilmiş olup ortamda herhangi bir değişiklik olduğunda algoritmanın yeniden en baştan çalıştırılması gerekmektedir. UCS (Verwer ve ark., 1989) (Uniform Cost Search/ Sabit Maliyet Araması) algoritması, graflarda ağırlıklı değerler üzerinde çalışan bir arama algoritmasıdır. Bu çalışmada ağırlık değeri için Manhattan uzaklığı kullanılmaktadır.

2.2. Yoldaki Engelin Kaldırılması

Bir araç taşınırken, park etme veya geri alma işlemleri sırasında park edilmiş diğer araçlarla çevrelenmesi olası bir durumdur. Böyle bir durumda, en yakın boş hücreyi tanımlanarak taşınan aracın yanına getirmek gerekmektedir. Bu işlem, taşınan aracın hedef noktasına doğru ilerlemesini kolaylaştıracak şekilde yapılmaktadır. Depolanan veya geri alınan arabanın etrafının diğer park edilmiş araçlarla veya otopark sınırlarıyla çevrili olması durumunda bir bloke açma prosedürü uygulanmaktadır. Prosedürün ana fikri, şu anda arabanın bulunduğu bloke olmuş hücrenin yanına boş bir hücre taşımak ve bloke edilen aracın dışarı çıkmasına yardımcı olmak için boş hücreyi kullanmaktır. Örnek bir geri verme işlemi sırasında park edildiği hücreden alınarak belirlenmiş olan asansöre taşınmakta olan bir aracın, yolundaki blokeyi kaldırma işlemi uygulanır. Yapılan tüm bu işlemlerin adımları sırasıyla Şekil 3'de gösterilmektedir.



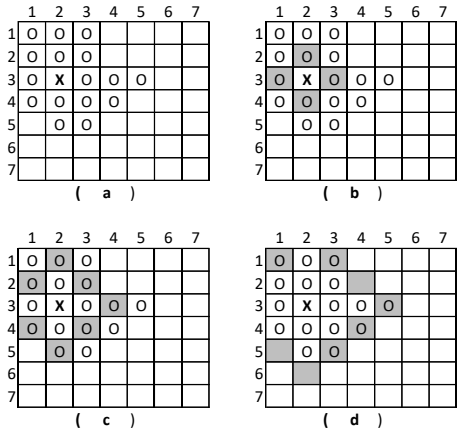
Şekil 3. Blokeyi kaldırma için işlem basamakları a) (3,2) konumunda hareket eden "x" hücresi, dolu durumdaki "o" hücreleri ile çevrilidir. b) hareketli "x" hücresine en yakın (1,1) ve (2,4) konumlarda iki boş hücre belirlenir. c) Hareketli "x" hücresinin yolundaki iki hedef hücre "y" belirlenir. d) (2,4) 'de en yakın boş hücre ve (3,3)' te hedef hücre tanımlanır. e) Boş hücre (3,3) 'e taşınır. f) Hareketli hücre (3,3)' e taşınır.

- **1. Adım:** UCS algoritması kullanılarak, taşınan aracın mevcut konumuna (3, 2) Manhattan uzaklığıyla ölçülen en yakın boş hücre bulunur. En yakındaki boş hücreler Şekil 3(b)'de gri renkle gösterilmektedir. Yer değiştirme için belirlenen boş hücreler hem taşınan araca hem de hedef asansöre en yakın konumda olmalıdır.
- **2. Adım:** Boş hücre için istenen yer değiştirme konumu belirlenir, bu konumlar Şekil 3(c)'de "y" ile işaretlenmiştir. İstenen yer değiştirme konumları sırasıyla; taşınan araca bitişik ve taşınan araçtan hedef asansöre doğru olan en kısa yol üzerinde olmalıdır. En kısa yol, boş hücrenin ya da taşınan aracın Manhattan uzaklığından birinden daha az olan bir yere taşınmasıyla oluşturulmaktadır.
- **3. Adım:** Hedeflenen yer hareketsiz/arızalı bir taşıyıcıya ait olması durumunda, hareketsiz hücrenin şu an bulunduğu yolla aynı Manhattan uzaklığıyla yeniden konumlandırılması için alternatif bir yol üzerinde yeni bir değiştirme hücresinin belirlenmesi gerekmektedir.
- **4. Adım:** Şekil 3(b)'de gösterildiği gibi bloke olan hücreye bitişik birden fazla yer değiştirme alanı olabilmektedir. Aynı şekilde Şekil 3(c)'de gösterildiği gibi bloke edilmiş hücreye aynı mesafede birden fazla boş hücre de mevcut olabilmektedir. Bu durumda, boş hücreden yeniden yerleştirme hedefine doğru, en küçük Manhattan uzaklığına sahip tüm kombinasyonlar arasından, birer tane boş hücre ve yer değiştirme hücresi seçilir. Şekil 3(d)'de görüldüğü gibi tanımlanan boş hücre konumu (2, 4) ve varış konumu (3, 3) olarak belirlenmiştir. Yer değiştirmenin varış konumu, araçtan hedef asansöre doğru Manhattan uzaklığıyla ölçülen en kısa yolda üzerinde olmalıdır.
- **5. Adım:** A* algoritması kullanılarak boş hücrenin yolu Şekil 3(e)'de gösterildiği gibi bloke olan hücreye bitişik istenen konuma yönlendirilmektedir.
- **6. Adım:** (3,2) konumundaki bloke olan taşıyıcı (3, 3) konumundaki boş hücreye Şekil 3(f)'de gösterildiği gibi hareket ettirilmektedir.

2.3. Uygun Boş Hücrenin Belirlenmesi

Boş hücre belirleme fonksiyonunda, maliyet fonksiyon değeri olarak Manhattan uzaklığını kullanan UCS algoritması uygulanmaktadır.

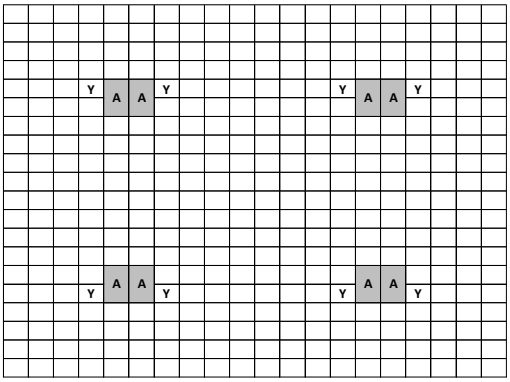
Şekil 4'te UCS algoritmasının nasıl uygulandığını gösterilmektedir. Şekil 4(a)'da, "x" ile işaretlenen hücre, taşınan aracın mevcut konumunu ve "o" ile işaretlenenler dolu hücreleri göstermektedir. UCS döngüsünün ilk yinelemesinde, vurgulanan hücreler ardıllar olarak üretilmekte ve bunlardan herhangi birinin boş olup olmadığı Şekil 4(b)'de gösterildiği gibi kontrol edilmektedir. İlk yinelemede boş hücre algılanmadığından döngü tekrarlanır ve arama işlemi yeniden yapılır. İkinci yinelemede Şekil 4(c)'de görüldüğü üzere boş hücre tespit edilemez. Döngünün üçüncü yinelemesinde 3(d)'de görülen (5, 1), (6, 2) ve (2, 4) konumlarında üç boş hücre tespit edilmektedir. Bu hücreler boş hücre listesine eklenir.



Şekil 4. UCS algoritmasının, engellenmiş bir taşıyıcının etrafındaki boş hücreleri aramak için nasıl kullanıldığını gösteren örnek. a) "x" hücresi, "o" engeli ile çevrilidir. b) UCS algoritması, işaretli hücreleri ilk yineleme sırasında Manhattan "1" uzaklığında araştırır. c) UCS algoritması, işaretli hücreleri ikinci yineleme sırasında Manhattan "2" uzaklığında araştırır. d) UCS algoritması, işaretli hücreleri üçüncü yineleme sırasında Manhattan "3" uzaklığında araştırır.

2.4. Otopark Düzeni

Park alanı, iki boyutlu ve ızgara tabanlı bir düzen olarak temsil edilmektedir. Bu ızgarada m sütun ve n satır bulunmaktadır. 20x20 düzenindeki bir otopark yapısı Şekil 5'te gösterilmektedir.



Şekil 5. 20 × 20 Otomatik otopark için park katının düzeni

Park noktaları üç farklı kategoride sınıflandırılır. Bunlar sırasıyla;

- **Asansörler(A):** Her bir asansör noktası iki bitişik hücreyi kaplamaktadır. Otoparktaki asansör sayısı alan sayısına bağlıdır. Her asansörün yaklaşık 50 park alanına hizmet verdiği varsayılmaktadır. Asansörler her zaman sabit engeller olarak kabul edilecektir. Bu durumda arabaları taşımak için kullanılan hareketli taşıyıcılar asansör hücresinden geçemeyecek veya asansör noktasında park edemeyecektir.
- **Yükleme/Boşaltma Alanları(Y):** Her asansörün yanında belirli bir yükleme alanı bulunur ve her yükleme alanı bir park hücresini kaplamaktadır. Yükleme alanları bir arabanın depolanması işleminin başlangıç noktası ve geri alınma işleminin bitiş noktasıdır. Depolama işleminde, araç önce yükleme alanlarından birinde görünecektir ki bu, aracın asansörden boşaltılmış olduğu anlamına gelir ve sonrasında hareket başlar. Alma işlemi sırasında, araç önce yükleme alanlarından birine ulaşır ve daha sonra ilgili asansöre aktarılır. Bu alanlar da yolun bir parçası durumundadır, ancak park yeri olarak kullanılamaz.
- **Park Yerleri:** Otopark alanının geri kalanı park yerleri olarak belirlenmiştir. Park yerleri boş veya dolu olabilirler. Boş noktalardan birine araç park edildiğinde, taşıyıcıdaki bir arıza nedeniyle hareketsiz hale gelmediği sürece hareketli bir engel olarak kabul edilir. Bununla birlikte, belirlenmiş tüm park yerlerine araç park edilerek alanlar doldurulamaz. Hareketli taşıyıcıların geçiş yolu olarak hizmet etmesi için minimum sayıda nokta her zaman boş tutulmalıdır.

2.5. Simülasyon Çalışması

Otomatik park sistemi için önerilen algoritmaların performansını göstermek için Java'da multithreading (çoklu iş parçacıklı) bir simülasyon yazılımı geliştirilmiştir. Simülasyon çalışması, aşağıdaki gibi donanım yapılandırmasına sahip Windows ™ OS tabanlı bir masaüstü bilgisayarda gerçekleştirilmiştir. Yazılımın geliştirildiği bilgisayarın ve geliştirici programın genel özelikleri sırasıyla aşağıda verilmiştir.

- **İşlemci:** Intel(R) Core (TM) i-5 6400 CPU @ 2.70GHz 2.71GHz

- **Bellek:** 8 GB RAM
- **İşletim sistemi:** Microsoft Windows10.
- **Java sürümü:** SE (Standart Sürüm) 1.8.0-91

3. Araştırma Sonuçları ve Tartışma

3.1. Simülasyon Sonuçları

Geliştirilen yazılımda otopark düzeni, 20x20 boyutunda, 4 asansör, 4 yükleme boşaltma alanı ve %13 boş hücre şeklinde modellenmiştir. Simülasyon, tasarlanan model üzerinde 10 defa çalıştırılarak elde edilen değerlerin ortalaması alınarak Tablo 1'de gösterilmiştir.

Tablo 1. Tasarlanan modelin simülasyon sonuçları.

Her Bir Araç İçin Adım Sayısı		Arama Süresi (ns)		Çalışan Maksimum Thread Sayısı	
Ortalama	**Optimum**	**Toplam**	**Ortalama**	**Teoride**	**Uygulamada**
6.87	4.54	7550396343	9856914.286	26	19
6.62	4.49	3357762774	4321445.012	26	13
6.50	4.51	3109387996	4017297.152	26	16
6.69	4.56	1795506189	2250007.756	26	15
6.97	4.56	2483361657	3216789.711	26	13
6.58	4.57	6094211134	7873657.796	26	15
6.87	4.60	4381226321	5653195.253	26	17
6.47	4.53	1777947047	2300060.863	26	15
6.54	4.53	2194900486	2846822.939	26	14
7.31	4.54	167251011	223001.348	26	17
6.74	**4.54**	**3291195096**	**4255919.212**	**26**	**15.4**

4. Sonuç

Bu makale, tam otomatik ve robotik otopark yapıları için entegre bir arama ve planlama algoritması paketinin tasarımını sunmaktadır. Önerilen tasarım, asansör aracılığıyla ilgili kata erişildiğinde, araçları park alanlarına taşımak için hareketli robotik taşıyıcıların kullanıldığı tam otomatik otoparklar için tasarlsanmıştır. Önerilen tasarımın en önemli yönü, birden fazla eşzamanlı park etme ve geri alma talebinin gerçek zamanlı olarak sunulmasıdır. Bir sonraki adım olarak, sunulan çalışmanın birden fazla hikayeye sahip bir otopark yapısına genişletilmesini öngörmekteyiz. Ayrıca kullanılan arama algoritması paketleri optimize edilerek performansı arttırılmaya çalışılacaktır.

Kaynakça

EPA, 2020, European Parking Association, https://www.europeanparking.eu

IPMI, 2020, International Parking Institute, http://www.parking.org

Resmi Gazete, 2020, https://www.resmigazete.gov.tr

Yardım, M. S., & Ağrikli, M. (2005). Otomatik Otoparklar ve Türkiye'deki Otopark Probleminin Çözümü İçin Uygulama Potansiyeli. Makina mühandisleri odası (s. 363-371). İstanbul: Makina mühandisleri odası.

Can, M., & Ilıcalı, M. (2019). Türkiye'de ileri otopark sistemleri İzmir Alsancak otopark uygulama örneği ve öneriler sunulması.

Parkolay, 2020, Otomatik Otopark, https://www.otomatik.com.tr

Koenig, S., & Likhachev, M. (2002). Incremental a. In Advances in neural information processing systems (pp. 1539-1546).

Koenig, S., & Likhachev, M. (2002). D* lite. Aaai/iaai, 15.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics, 4(2), 100-107.

Verwer, B. J. H., Verbeek, P. W., & Dekker, S. T. (1989). An efficient uniform cost algorithm applied to distance transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(4), 425-429.