



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ÜÇ BOYUTLU ORTAMDA EN KISA YOL PLANLAMA UYGULAMALARI

Elaf Jirjees DHULKEFL DHULKEFL

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Aralık-2019
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK TEZİ

ÜÇ BOYUTLU ORTAMDA EN KISA YOL PLANLAMA UYGULAMALARI

Elaf Jirjees DHULKEFL DHULKEFL

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr.Öğr.Üyesi Hakan TERZİOĞLU

2019, 108 sayfa

Jüri
Dr.Öğr.Üyesi Hakan TERZİOĞLU
Prof. Dr. Cemil SUNGUR
Prof. Dr. Şakir TAŞDEMİR

Robotik sistemlerde, engellerden kaçınmak ve çarpışma olmadan yolu planlamak mobil robotlar için önemli ve zor bir problemdir. Yol planlaması, hedefe en kısa yoldan en kısa sürede ulaşmak için gerçekleştirilmektedir. Böylece araçların hedefine hem kısa sürede ulaşarak hemde bataryadaki enerjinin optimum kullanılması sağlanacaktır. Yol planlaması için literatürde birçok yol planlama tekniği vardır. Özellikle son yıllarda, klasik yol planlama tekniklerindeki eksiklikleri ve sorunları ortadan kaldırmak için sezgisel yol planlama teknikleri kullanılmıştır.

Bu tez çalışmasında da insansız hava araçları (İHA) için önemli bir konu olan yol planlaması gerçekleştirilmiştir. Yol planlaması için Dijkstra, A* ve RRT algoritmaları kullanılmıştır. Bu algortimalar çevre şartları 2 boyutlu (2B) ve 3 boyutlu (3B) olarak oluşturulabilen simülasyon ile belirlenen alanlarda yol planlaması gerçekleştirilmiştir. 2B ve 3B simülasyonda A* algoritmasının, Dijkstra ve RRT algoritmalarına göre daha iyi sonuç verdiği görülmüştür. Gerçek ortamda gerçekleştirilen çalışma sonucunda A* algoritması ile daha hızlı sürede hedefe ulaştığı görülürken RRT algoritması hedefe daha kısa mesade fakat daha uzun sürede ulaştığı belirlenmiştir.

Anahtar Kelimeler: A* algoritması, Dijkstra algoritması, İnsansız hava araçları, Hızla keşfedilen rastgele ağaç algoritması (RRT).

ABSTRACT

MS THESIS

PATH PLANNING APPLICATIONS IN THREE DIMENSIONAL ENVIRONMENT

Elaf Jirjees DHULKEFL DHULKEFL

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE IN ELECTRICAL ELECTRONICS ENGINEERING

Advisor: Ass. Prof. Dr. Hakan TERZİOĞLU

2019, 108 Pages

Jury
Ass. Prof. Dr. Hakan TERZİOĞLU
Prof. Dr. Cemil SUNGUR
Prof. Dr. Şakir TAŞDEMİR

In robotic systems, avoiding obstacles and planning the path without collision is an important and difficult problem for mobile robots. Road planning is carried out to reach the destination as soon as possible via the shortest route. Thus, the target of the vehicles will be reached in a short time as well as the optimum use of the energy in the battery will be provided. There are many road planning techniques in the literature for road planning. Especially in recent years, heuristic road planning techniques have been used to eliminate deficiencies and problems in classical road planning techniques.

In this thesis, road planning, an important topic for unmanned aerial vehicles (UAVs), was carried out. Dijkstra, A* and RRT algorithms have been used for road planning. These algortimas environmental conditions 2-dimensional (2D) and 3-dimensional (3D) path planning was carried out in the areas determined by the simulation can be created. In 2D and 3D simulation, A* algorithm was found to be better than Dijkstra and RRT algorithms. As a result of the study performed in real environment, it was observed that the RRT algorithm reached the target in a faster time while the RRT algorithm reached the target in a shorter distance but longer time.

Keywords: A* algorithm, Dijkstra algorithm, unmanned aerial vehicles, rapidly discovered random tree algorithm (RRT).

ÖNSÖZ

Çalışmam sırasında değerli fikirlerini ve zamanını benden esirgemeyen tez danışmanım Sayın Dr. Öğr. Üyesi Hakan TERZİOĞLU'na ve ikinci danışmanım olan Sayın Dr. Öğr. Üyesi Akif DURDU' ya yardımlarından dolayı teşekkür ederim.

Elektrik Elektronik Mühendisliği Bölüm Başkanı Sayın Prof. Dr. Cemil SUNGUR'a ve bugüne kadar üzerimde fazlaca emekleri olan vermiş oldukları bilgilerle alanımda gelişmemi sağlayan bütün Elektrik Elektronik Mühendisliği Bölümü hocalarıma teşekkür ederim. Tez çalışmasında bana yardımcı olan arkadaşım Eyüp EKİNCİ'ye teşekür ederim.

Hayatımın her anında olduğu gibi eğitim-öğretim hayatımda da hep yanımda olan, yardımlarını benden esirgemeyen anneme, babama ve kardeşime Ayad Jirjees DHULKEFL'a çok teşekkür ederim.

Elaf Jirjees DHULKEFL DHULKEFL
Konya-2019

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

1.1. Mobil Robotlar İçin Yol Planlama 3

2. KAYNAK ARAŞTIRMASI **7**

2.1. İHA'lar İçin Literatür Taraması 7
2.2. Mobil Robotlar İçin Literatür Taraması 13

3. İNSANSIZ HAVA ARAÇLARI (İHA) **18**

3.1. İHA uygulama alanı 18
3.2. İHA'ların ayırt edici özellikleri 20
3.3. İHA teknolojileri 20
3.4. İHA çeşitleri 21
3.5. Gelecekteki İHA gelişimi 25
3.6. İHA'ların avantajları 26
3.7. Dezavantajları 26
3.8. Yol planlama teknikleri 26
3.9. Yol Bulma Algoritmaları 28

4. MATERYAL VE YÖNTEM **30**

4.1. A* Algoritması 30
4.2. Dijkstra Algoritması 35
4.3. Hızlı Rastgele Ağaç Keşfi algoritması (RRT_ Rapidly-exploring Random Tree) Algoritması 40

5. UYGULAMA VE SONUÇLARI **46**

5.1. Simülasyon Ortamı 47
5.1.1. Birinci uygulama 47
5.1.2. İkinci uygulama 59
5.1.3. Üçüncü uygulama 71
5.2. Gerçek Zamanlı Test Ortamı 83
5.2.1. Birinci test ortamı 84
5.2.2. İkinci test ortamı 88

5.2.3. Üçüncü Test Ortamı 92
5.2.4. Dördüncü test ortamı 96

6. SONUÇLAR VE ÖNERİLER 100

6.1 Sonuçlar 100
6.2 Öneriler 103

KAYNAKLAR 104

ÖZGEÇMİŞ 108

SİMGELER VE KISALTMALAR

Simgeler

g(n) : Başlangıç düğümünden mevcut düğüme kadar gelmenin maliyeti.
h(n) : Mevcut düğümden hedef düğüme varmak için tahmin edilen mesafe.
f(n) : Hesaplama yapan sezgisel fonksiyon.

Kısaltmalar

RRT : Hızla keşfedilen ağaçların planlaması **(**Rapidly exploring random tree)
İHA : İnsansız hava araçları
PSO : Parçacık sürüsü optimizasyonu
NNS : En yakın komşu araştırması
MDA : Modifiye Dijkstra algoritması
EDA : Yükseklik temelli dijkstra algoritması
FVF : Bulanık sanal kuvvet
VL : Görünürlük hattı (Visibility line)
A* : A yıldız
GA : Genetik algoritması
PRM : Olasılıksal yol haritası yöntemi (Probabilistic roadmap method)
ISAA : Geliştirilmiş seyrek A* algoritması (Improved sparse A* algorithm)
IFDS : Karışımlı sıvı dinamik sistemi (İnterfered fluid dynamic system)
AO : Karınca optimizasyonu (Ant optimization)
MOGA : Çok amaçlı genetik algoritma (multi-objective genetic algorithm)
APF : Yapay potansiyel alanı (Artificial potential field)
NN : Sinir ağı (Neural network)
IACÖ : Karınca kolisi (İmproved ant colony algorithm)
2B : İki boyutlu
3B : Üç boyutlu
BLP : İki seviyeli programlama

1. GİRİŞ

Günümüzde insansız hava araçlarının (İHA) kullanımı artmaktadır. Sabit kanatlı İHA'larda karşılaşılan en büyük sorunlardan birisi batarya sorunudur. Batarya sorunundan dolayı üç boyutlu (3B) yol planlaması, otonom mobil ve uçan robotlarda önemli bir konudur. Türkiye'de yeni bir konu olan otonom dronların kullanımı, otonom dronlar üzerinde maksimum otonomi bataryasıyla görev yapmak için yüksek performanslı yol planlama yöntemleriyle mümkündür. Tez çalışmasında 3B yol planlama yöntemlerinin geliştirilmesi ve mobil otonom robotlar için geliştirilen 2B yol planlama algoritmalarına dayalı olarak uygulamanın hızlandırılması hedeflenmektedir.

İHA'lar günümüzde hava durumu tahminleri, çevresel değerlendirmeler ve trafik kontrolü gibi çoğu sivil görevi yerine getirmek için kullanılmaktadır. İHA'lar ayrıca sınır kontrolü, gözetim, mayın tespiti, çevrenin korunması, orman yangını kontrolü, kargo yönetimi ve kayıt gibi askeri uygulamalarda yaygın olarak kullanılmaktadır. İHA'lar yaygın kullanımlarının nedenleri:

- Üretim ve satın alma maliyetleri düşüktür.
- Görev sırasında can kaybı riski yoktur.
- Görüntüleme teknikleri sayesinde takip yapabilme ve tehlikeli durumları haber verebilmektedir.
- Hava şartlarından az etkilenirler ve havada kalabilme süreleri bataryaya bağlı olarak arttırılabilmektedir.
- Gece ve gündüz görev yapabilmektedirler.
- Sensor ağları tabanlı takım halinde ve koordineli uçarak daha fazla alanı kısa sürede tarayabilmektedir.

Gelişmiş teknolojik gelişmelere paralel olarak, İHA sistemlerinin kullanımı gün geçtikçe artmaktadır. Uzun bir tarihsel gelişimi ve bireysel uygulamaları olmasına rağmen, İHA'lara olan talep ikibin yılı sonuna kadar çoğunlukla askeri alanda olmuştur. Ancak, hızlı teknolojinin gelişmesiyle birlikte drone'ların kullanımı sadece askeri alanda değil sivil alanda da günden güne artmaktadır. Sınır kontrolü, kaçakçılıkla mücadele, fotoğrafçılık ve meteoroloji gibi faaliyetlerle kullanımı yaygınlaşmıştır. Ayrıca, İHA'ları iş faaliyetlerine kullanmak için sayısız girişimler vardır.

İnsansız hava araçlarının (İHA) geliştirilmesi, 1916'da Archibald Low ile başlandı. Drone'lar daha çok insansız hava araçları (İHA) veya insansız hava aracı sistemleri (İHAS) olarak bilinir. İnsansız hava araçları (İHA), üzerinde insan olmadan, uzaktan ve otonom kontrol edilebilen bir hava aracıdır. İHA'lar iki bölüme ayrılır: birincisi uzaktan kumanda ile uçan, diğeri ise belirli bir uçuş planında otomatik olarak hareket edebilen araçlardır**.** İnsansız Hava Aracı uzaktan kumandalı veya önceden tanımlanmış uçuş planı, insan güvenliğinden ödün vermeden birçok uygulamada yaygın olarak kullanılabilir. Gerçek yaşam problemlerine uyumun sağlanması için çalışma 3B yüzeylere dayanılarak yapılmıştır. İHA'ların hareket planlamasında maliyet ve zaman optimizasyonu gerçek problemler için sağlanmaktadır. İHA Sistemleri terimi, son zamanlarda insansız araçları için kullanılmıştır. Bunun nedeni, İHA ifadesinin yalnızca düzlemin platformu anlamına gelmesi ve onu geçen tüm sistemi kapsamamasıdır. Ancak, hem devletlerin hem de kurumların bu uçaklar için farklı terimler kullandıkları düşünülmektedir.

1990'lı yıllarda askeri teknoloji devrimi ile uzun süre havada kalan İHA'lar her türlü hava koşulunda uzaktan kontrol edilip istihbarat, keşif ve gözetim yapma konusunda vazgeçilmez bir nokta haline geldi. En son yüksek teknolojiye sahip uçaklar artık çarpışma önleme sistemleri ile donatılmıştır. Bunlar, çevreyi taramak için engel algılama sensörlerini kullanırken, yazılım algoritmaları ve SLAM teknolojisi görüntüleri uçağın algılamasını ve önlenmesini sağlayan 3D haritalara dönüştürür (Yalçıner, 2014). Bu sistemler algılamak ve kaçınmak için aşağıdaki sensörlerden birini veya birkaçını birleştirir;

- Görme Sensörü
- Ultrasonik
- Kızılötesi
- Lidar
- Uçuş Süresi
- Monoküler Görme

Otomatik Pilotlar (OP), uzun uçuşlardan yorulmak, uykuya dalmak, dikkat dağınıklığı, uçuş verilerinin yanlış yorumlanması vb. insan doğasından kaynaklanan temel başarısızlık sebeplerine karşı üstünlük kazanır. Örneğin, bir OP arızası çok detaylı bir şekilde izlenebilir fakat bir pilotun kalp krizi sonucu tam olarak tespit edilemeyebilir. "İnorganik cihaz" alanındaki benzer problemler ortadan kalktığında

OP'ler bu açıdan daha güvenilir ve deterministik olarak adlandırılabilir. Bir elektrik kablosu gevşek bağlantı da ölümcül sonuçlara neden olur ancak bu noktada iki yeni parametre denklemi gelir; fazlalık ve fiyat. Artıklık ölümcül başarısızlık olasılığını büyük ölçüde azaltabilir. Uçaklarda, fazladan bir pilot daha vardır. Aynı uçakta buluna iki pilotunun da aynı anda ölüm şansının çok küçüktür. Bu çözüm, ana ünitelerin yedek bir eşdeğerine sahip olduğu otomatik sistem elektroniklerin de uygulanır. Ancak gereksiz pilot, yedek OP'lerden daha pahalıdır(Yalçıner, 2014).

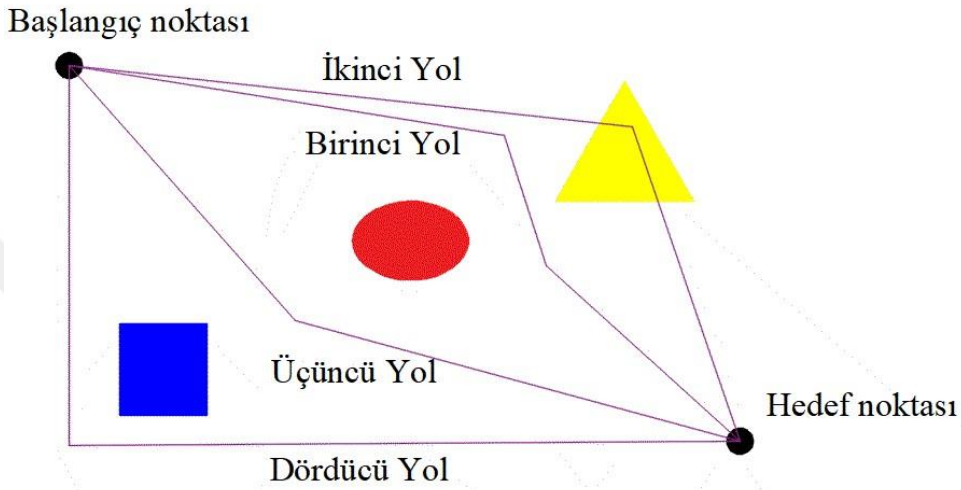
Teknolojik gelişmeler bileşenlere ve elektronik birimlere uygun fiyatlar ve yüksek yetenekler getirdi. Küçük ve orta ölçekli hava araçları, sıradan teknoloji meraklıları tarafından kolayca oluşturulabilir ve kullanılabilmektedir. En çok tercih edilen İHA tipleri sabit kanatlı uçaklar, helikopterler ve çok kanatlılardır. Bu hava taşıtları arasında, akademik çalışmalar da en popüler olan quadrotordır. Bir quadrotorun genel yapısı bir iskelet, güven rotorları, bıçaklar, radyo denetleyicisi ve entegre iç ölçüm birimleri bulunan bir uçuş denetleyicisinden oluşur. Görsel veri aktarıcıları da sisteme eklenebilmektedir (Yalçıner, 2014).

1.1. Mobil Robotlar İçin Yol Planlama

Yol planlama çalışmaları mobil robot faaliyetlerinde önemli bir rol oynamaktadır. Yol planlama, iki uzay noktası arasındaki mesafeyi hesaplayarak aralarındaki ideal yolu keşfetme tekniğidir. Şekil 1.1'de bir yol planlama örneğine ait kriterler görülmeltedir. 1960'lı yılların ortalarında başlamış ve 1980'lerde ivme kazanmış ve o zamandan bu yana önemli bir çalışma konusu haline gelmiştir. Taşınabilir robotlar, ortaya çıktıkları günden itibaren önemli ölçüde gelişmiştir. Geçmişte mobil robotların kullanımı sınırlı olmasına rağmen, günümüzde özellikle montajda, ambalajlamada, depoda, ulaştırmada, sağlık hizmetlerinde, askeri uygulamalarda, güvenlikte, enerji kaynaklarında, temizlik endüstrisinde ve tarım hizmetlerinde kullanılmaktadır.Bu büyük alanlarda kullanımıyla birlikte mobil robotların karşılaştığı sorunlar için birçok çözüm üretildi ve bu alternatiflerin geliştirilmesi günümüzde devam etmektedir (Suvaydan, 2011).

Şekil 1.1'e bakıldığında; İlk yol engelli bir çarpışma değildir ve yol uzunluğu açısından uygun bir yol olduğu görünmektedir. İkinci yol, engellerden biriyle doğrudan çarpışma halinde olduğu için uygunsuz görünmektedir. Üçüncü Yol, Engellere

çarpmadan, Birinci Yol'a kıyasla engellere olan mesafenin daha güvenli olduğu görünmektedir. Burada dikkat edilmesi gereken nokta, çarpışma olmamasına rağmen, yol planlama konusunun tam çözümü için ideal rota kriterinin de sağlandığını unutmamak gerekir. Dördüncü yol, engellere çarpmadığı için uygun bir yol olarak görünmesine rağmen birinci ve üçüncü yollarda en uygun yol olmaya uygun değildir. Sonuç olarak, birinci ve üçüncü yollar, yol planlama konusunun bu ideal çözümlerine çok yakındır (Suvaydan, 2011).



Şekil 1.1. Bir yol planlama örneğine ait kriterler

Yukarıdaki adımlara göre, yol çizelgeleme sorununu çözmek için yalnızca bir kriter yeterli değildir. Yol planlama konusunu aynı anda düzeltmek için tüm şartlar yerine getirilmelidir. Bazı yol planlama konularında bu temel gereksinimler dışında, yol planlama konularına farklı kriterler (zaman, enerji, güvenlik vb.) eklenebilmektedir. Tüm bu kriterler karşılandığında, yol planlama probleminin en doğru sonucu vereceğine inanmaktadır (Suvaydan, 2011).

Mobil robotlar, günümüzde robotik üzerine yapılan çalışmaların odak noktasını oluşturmaktadır. Birçok üniversitede, özerk araştırmacılar, firma mobil robotlar üzerinde çalışmış ve düzenli olarak mobil robotlar üzerine birçok bildiri yayınlamiştir. Günümüzde askeri, tıp, eğitim, araştırma, kurtarma gibi birçok alanda kullanılan mobil robotlar, özellikle elektronik, bilgisayar bilimi, iletişim ve internetteki yeniliklerle beraber kullanımı artmıştır. Mobil bir robot, herhangi bir ortamda hareket edebilen bir elektromekanik makinedir. Mobil robot, sensör bilgilerini toplama ve işleme, aktüatör iletişimini ve taban seviyesini kontrol etme, sensör bilgisini yorumlama ve davranışı

belirleme, yüksek seviye gibi yüksek seviyeli prosedürleri yürüten bir bilgisayar gibi düşük seviyeli aktiviteleri gerçekleştiren gömülü sistemlerden oluşmaktadır. İnternet kontrolü ve tüm teknolojiler, mekanik sistemler ve mobil robotun belirtilen ayardaki faaliyetleri yürütmesini ve yürütmesini sağlayan aktüatörler gibi iletişimdir (Suvaydan, 2011).

Robot yolunu, noktadan noktaya planlamak için robot sistemlerinde farklı yöntemler kullanılır. Yol planlama konusunun temel amacı, başlangıç noktasından robotun hedef pozisyonuna kadar herhangi bir engele çarpmadan en kısa yolu bulmasını sağlamaktadır. Kaynaktan hedefe giden yollar arasındaki en kısa ve güvenli yolu belirleyerek en iyi duruma getirir. Çalışma alanındaki bir öğenin hareketini belirlemeye yardımcı olduğu için buna hareket planlama da denir. Yol planlaması aynı zamanda belirli varlıkları optimize etmek, farklı hareketler oluşturmak ve rota hareketini birkaç tekrar eden adıma bölme yöntemi olarak da tanımlanabilir. Engel önleme sorununu gidermek için farklı teknikler kullanılır (Suvaydan, 2011).

Bir yol planlama işi tipik olarak birden fazla parametreye veya giriş değerine ihtiyaç duymaktadır. Bunlar;

1. Başlangıç pozisyonu: Robotun başlangıç pozisyonudur,
2. Hedef pozisyon: Robotun gitmesi istenen konumdur,
3. Engeller: Robotla etkileşime girmeyen nesnelerdir, başlangıç noktasından varış noktasına kadar yolda konumlandırılmışlardır.

Yol planlaması bir çevre haritası ve haritaya göre robotun konumunu gerektirir. Rutin yol planlamasındaki sorunlar, çevre bilindiğinde sıklıkla incelenir. Buna dayanarak, alanın etrafındaki rotayı ve robotun seyahat edeceği çevresel engelleri planlarken 4 ana bileşenle sınıflandırabiliriz (Suvaydan, 2011).

1. Bilinen devrelerde sabit engeller için tam yol planlaması,
2. Bilinen veya bilinmeyen ortamlardaki sabit engeller için yol planlaması,
3. İyi bilinen ortamlarda dinamik engeller için yol planlaması,
4. Bilinen veya bilinmeyen ortamlardaki dinamik engeller için yol planlaması.

Bu tez çalışmasında, üç boyutlu (3B) yolu bulmak için algoritmalarda birkaç adım kullanılmıştır. Başlangıç noktasından hedef noktaya kadar olan engellerin yükseklikleri hakkında bilgi edinilmiştir. Engel yüksekliklerine göre İHA'nın uçacağı

en uygun yükseklik belirlenmiştir. Engel yüksekliklerine göre 2B yol planlama algoritmalarına dayalı birkaç alternatif yol planlaması yapılmıştır. Yapılan yol planlamalarından en uygun olanı çeşitli yöntemler ile belirlenerek, en uygun yol planlaması için çeşitli yöntemler önerilmiştir. Önerilen bu yöntemler ise A*, Dijkstra ve RRT algoritmasıdır. Bu algoritmaların simülasyon sonucuna göre, üçüncü test sonucunu incelendiğinde, A* algoritmasında 22m'lik yolu 0.1256 saniyede, 19m'lik yolu ise 0.0555 saniyede gitmiştir. Dijkstra algoritmasında 22m'lik yolu 1.0446 saniyede, 19m'lik ise yolu 1.4563 saniyede gitmiştir. RRT algoritmasında 37m'lik yolu 4.1713 saniyede, 33m'lik yolu ise 6.4095 saniyede gittiği görülmektedir. A* algoritması diğer iki algoritmadan daha iyi performans göstermiştir. A*, RRT ve Dijkstra algoritmalarının sonuçlarını karşılaştırdığımızda Dijkstra ve RRT algoritmalarının fazla zaman harcadiği ve A* algoritmasının daha az zamanda sonuca vardiği görülmüştür. Bu durum A* algoritması alanı sadece hedefe doğru taradığı, RRT algoritmasının her simülasyonda hedefe doğru farklı noktalar belirleyerek yol çizdiği, Dijkstra algoritmasının ise çok daha geniş alanı tarayarak hedefe vardığı görülmüştür. Gerçek ortamda yapılan çalışma sonucunda RRT algoritması A* algoritmasına göre hedef için daha kısa yolu belirlemesine rağmen A* algoritması daha hızlı sürede hedefe vardığı görülmüştür.

2. KAYNAK ARAŞTIRMASI

2.1. İHA'lar İçin Literatür Taraması

Mobil robotlar için yol veya yol planlama problemleri ile ilgili çalışmalar 1960'lı yılların ortasında başlamış olup 1980'li ivme kazanarak, o yıllardan itibaren önemli bir araştırma konusu haline gelmiştir. Bu bölümde de, daha önceki yıllardan şimdiye kadar yapılan yol planlaması çalışmalarının bir literatür incelemesi yapılarak, özeti sunulmuştur.

Literatürde en çok kullanılan yöntemlerden biri araştırmacılar tarafından, A* algoritmasıdır. (Zhang Qi ve ark., 2010) 3B yol planlama problemini çözmek için gelişmiş bir sezgisel algoritma sunmaktadırlar. Yol planlama modeli ilk önce dijital haritaya dayalı olarak oluşturulmuştur ve sonra üç boyuttan iki boyuta kadar önemli miktarda arama alanını ortadan kaldırmak için sanal arazi tanıtılmıştır. Geliştirilmiş sezgisel A* algoritması, İHA yörüngesini oluşturmak için uygulandığını göstermektedirler. Diğer bir araştırmaya göre, A* algoritmasının daha geliştirilmiş şeklinde uygulandiğini anlatmaktadır, insansız hava araçları (İHA) üç boyutlu (3B) yol tasarımı ve gerçek sanal arazide düzeltme yöntemi sunmaktadırlar. Gerçek dijital alanı planlama alanına sokan enterpolasyon ve eğrilikten sonra bir Gelişmiş Seyrek Improved Sparse A* Algoritması (ISAA) önermişlerdir. Önerilen yöntemi A* algoritmasıyla karşılaştırılma sonuca göre optimal yol A* için 147.93 km iken optimal yol ISAA için 134.78 km göstermektedir. Karşılaştırmaya göre ISAA algoritması, arama alanın azalttığını ve yakınsama hızını arttırdığını göstermektedir (Miao Yong-Fei ve ark., 2013). Araştırdıkları bu makalede İHA'ların yüksek hayatta kalma oranları ve düşük yakıt tüketimi gerektirdiği sorunu çözmek için bir 3B geniş çaplı savaş alanı ortamında İnsansız Hava Araçlarının gerçek zamanlı yol planlaması için geliştirilmiş bir introduc algoritması sunar. Algoritma, hedef uzaydaki iki ara nokta arasındaki en uygun yolu bulabilir ve irtifa, algılama olasılığı ve yol uzunluğu gibi faktörleri kapsamlı bir şekilde dikkate alır. Sonuça göre, geliştirilmiş A* algoritmasının kararlı, yakınsak ve verimli olduğunu göstermektedirler (Weiwei Zhan ve ark., 2014). Diğer araştırmacılar anlatıkları bu makalede, uzun ve orta irtifa uçuşu olan insansız hava araçları için geliştirilen çok kriterli bir yol planlama modelini anlatmaktadırlar. Çevrenin dinamiklerini dikkate alarak insansız hava araçları için mesafe, zaman ve yakıt tüketimi

açısından A* algoritmasını önermişlerdir (Nafiz Arıca ve ark., 2012). Önerilen diğer algoritma ise Dijkstra algoritmasıdır. Bu araştırmada saha yüksekliğine göre sabit kanatlı İHA için bir Dijkstra yolu planlama algoritması önermişlerdir. Arazi yüksekliklerini temsil etmek için dijital yükseklik modelleri kullanmışlardır. MDA (Modifiye Dijkstra Algoritması) yöntemi, EDA'nın (Yükseklik Temelli Dijkstra Algoritmasının) hesaplama süresini önemli ölçüde azalttığını göstermişlerdir (Felipe Leonardo Lôbo Medeiros ve José Demisio Simões Da Silva, 2010). Sundukları bu yazıda, bir USV için gerçek zamanlı bir haritadaki kaynak ve bitiş noktaları arasındaki hesaplama süresi açısından en uygun yolu bulmak için pratik bir deniz ortamını, yani Portsmouth Harbour'u dikkate almaktadır. Mevcut çalışma orijinalden oluşturulan bir grid haritasını kullanmaktadır ve tek bir USV için en kısa yolu bulmak için Dijkstra algoritmasını kullanmaktadır. Önerilen Dijkstra algoritması, en iyi hesaplama yolunu bulmak açısından daha etkili olduğu bulunmuştur (Y Singh ve ark., 2018).

İHA'lar konusunda yapılan bu çalışmada takviye öğrenmesi yoluyla İHA'lar için bağımsız bir uçuş planı stratejisi önermişlerdir. Engellerden kaçınmak için İHA hedef noktasına ulaşana kadar bir çeşit takviye öğrenme algoritması olan Q-learning'i uygulamaktadırlar. Sonuca göre önerdikleri yöntem daha kısa varış zamanını göstermektedir (I Kim ve ark., 2017). Diğer araştırmacılar bu yazıda sunulan yöntemler ve algoritmalar, İnsansız Havacılık projelerinde, yüksek seviye karar algoritmalarında, uçuş optimizasyon prosedürlerinde ve düşük seviyeli uygulamalarda İHA hareket stratejilerinin uygulanması için kapsamlı bir şekilde gözden geçirilmiş, simüle edilmiş ve önerilmiştir (Victor Vladareanu ve ark., 2016). Diğer araştırmacılar yaptıkları bu çalışmada uçan baz istasyonları, optimum hava aracı dağıtım çerçevesi önermektedirler. Önerilen dağıtım yaklaşımının, sabit İHA siteleri ile standart Voronoi hücre ekleme tekniğine kıyasla sistemin güç verimliliğini 20 kat artırabileceğini göstermektedir. Sonuca göre toplam gerekli iletim gücünün İHA'lar için en uygun kapsama alanlarını belirleyerek önemli ölçüde azaldığını göstermektedirler (Mohammad Mozaffari ve ark., 2016). Farklı bir yöntem önerilen araştırmacılar, İHA yolu planlama parametrelerini ayarlamak için bir diferansiyel evrim algoritması optimizasyon yöntemi sunmaktadırlar. Geliştirilmiş algoritma, kullanıcının ihtiyacına göre gereken minimum üretim ile yakınlaşmak için gerekli olan yol ve hesaplama maliyeti arasında istenen ağırlığı tanımlamasını sağlar. İHA yolu planlaması için diferansiyel evrim algoritmasında ayar parametrelerinin önerdikleri optimizasyon, hesaplama maliyetini hızlandırdığni ve

iyileştirdiğini göstermektedir (Kai Yit Kok ve Parvathy Rajendran, 2016). Yapılan bu çalışmada Radar, hedef konum hakkında gerçek zamanlı geribildirim sağlamak, hedef konuma dayalı bir sonraki hareket durumunu öngörmek ve daha sonra dinamik yol planlaması yapmak için geri bildirim verilerini durum tahminleri ile birleştirmek için kullanılmaktadır. Kalman filtreleme durum pozisyonları ve öngörülen pozisyonları elde etmek için kullanmışlardır. Hedefleri ve çoklu hareketli hedefler yolları planlamak için kullanılan karınca kolonisi algoritmasıdır. Bu önerilen plan, hedefe ulaşmak için gereken zamanın kısaltıldığını ve yolun kısa olduğunu göstermektedir. (Bo Wang ve ark., 2018). Üç boyutlu yol bulma farklı algoritma kullanarak araştırdıkları bu makalede, İHA'ların özerk karar vermesi, kontrolü için çeşitli yol planlama ve navigasyon algoritmaları önermişlerdir. Sezgisel, sezgisel olmayan veya kesin teknikler, yol planlama algoritmalarını kategorize eden iki çözüm yöntemidir. Farklı senaryolar ve engel düzenleri için yol planlama algoritmalarının karşılaştırmalı bir çalışmasını sunmaktadılar. Potential Field, Flody-warshall, Genetik algoritma, MSLAP, A*, Dijkstra, Approximate RL, MILP yöntemleri birbiriyle karşılaştırdıklarında, GA zamana daha az duyarlılık göstermektedirler, MSLAP en hızlı çözümdür ancak çoğu zaman optimal değildir. Potansiyel alan algoritması makul bir çözüm süresi gösterir. A*, Dijkstra algoritması ve Floyd-Warshall algoritması senaryoları en iyi şekilde çözme yeteneğini göstermektedirler (Mohammadreza Radmanesh ve ark., 2018). Yapılan bu çalışmada, gerçek zamanlı bir yol arama prosedürünü İHA'ların ilgili performanslarındaki duyusal kabiliyet, manevra kabiliyeti, uçuş hızı limiti açısından çeşitlilik ve dalgalanmalarına uyarlamak için insansız hava araçları (İHA) için uyarlamalı bir yol planlayıcısı sunmaktadır. Gerekli yol planlama bileşenlerini modellemek için iki seviyeli programlama ve değişken planlama aşaması teknikleri tanıtılır ve daha sonra uyarlama ve optimizasyon amacıyla bir uyarlamalı yol planlayıcısı geliştirilirip, önerdikleri planlayıcının etkinliğini ve verimliliğini göstermişlerdir (Wei Liu ve ark., 2013).

İHA'larda yapay zeka algoritmalarıda kullanmaktadır. Yapılan bu çalışmada bulanık mantık kullanarak İHA'nın iki boyutlu hareket planlaması için bir metodoloji sunmaktadır. Bulanık çıkarım sistemi, engeller ve hedef yeri hakkında gerçek zamanlı bilgi alır, aynı zamanda yön açısı ve hızında bir değişiklik oluşturmakdadır. Sonuca göre, önerilen yöntem FLC yönteminin avantajlarından biri; düşük kontrol çabasını korurken, karmaşık senaryolara uyarlanabilirliğini göstermektedir (Chelsea Sabo ve

Kelly Cohen, 2012). Başka bir çalışmada FVF'ye dayanan bir yol planlama yaklaşımı incelenmiştir. FVF (Bulanık Sanal Kuvvet) yöntem, İHA yol planlaması için uygun ve hızlıdır. FVF yaklaşımı A* arama algoritmasıyla karşılaştırılır, FVF yöntemi karmaşık bir ortamda çevrimiçi İHA yol planlamasından üstündür (Dong Zhuoning ve ark., 2010). Yapılan bu çalışmada, 2B çerçeveye dayanan 3B yol planlama algoritmaları önermişlerdir. Poligonal engelleri olan bir ortamda, başlangıç noktasından hedef noktaya en kısa yolu oluşturabilecek olan Visibility Line (VL) yöntemini önerilmişlerdir. Engel sayısı arttıkça çalışma süresi katlanarak artar. Bir 3B yolu bulmak için algoritmalarda birkaç adım kullanılır. İlk önce, yerel bir başlangıç noktasından bir hedef noktaya yerel bir uçak üretilir. Daha sonra önceden tanımlanmış birkaç açıda döndürülür. Her dönüşte, en kısa yol 2B algoritmaları kullanılarak hesaplanır. Hesaplama açısından verimli olduğunu ve gerçek zamanlı uygulama için uygun olduğunu gösterdiler (Rosli Omar ve Dawei Gu, 2010). Anlatılan bu makalede, insansız hava taşıtlarının (İHA) rota optimizasyonu için bir dizi istihbarat temelli yöntem önermiştir. Harita kapsamı senaryosunda, Ant Sistem algoritmasının İHA sisteminin optimizasyonuna uygulanabileceğini gösterdiler. Başka bir yöntemin karşılaştırması olarak NNS (en yakın komşu araştırması), algoritmanın bulunmasında daha etkili olduğunu göstermişlerdir ve en yakın komşu arama daha uygun bir rota (Aleksandar Jevtic ve ark., 2010). Yapılan bu çalışmada, Alçaktan uçan insansız hava araçları (İHA) için karmaşık bir arazi ortamında üç boyutlu 3D yol planlaması sunmaktadır. Yol uzunluğu ve uçuş yüksekliğini en uygun 3B yolu elde etmek için (IFDS) interfered fluid dynamic system önerilmiştir. Önerilen yöntem AO (ant optimization) yötemle karşılaştırılarak sonuca göre önerilen yöntem çevresel modellemenin basit olduğunu ve yolun İHA için düzgün ve uygun olduğunu göstermektedir (Honglun Wang ve ark., 2015) . Bu makalede, GNRON problemini çözmek için göreceli bir mesafe fonksiyonu tanıtarak dinamik ortamda geliştirilmiş bir APF (Artificial Potential Field) önermişlerdir. Minimum karakterizasyon değerini bularak yerel bir sanal hedef oluşturulur ve APF (Artificial Potential Field) yardımı ile yerel yol planladığını göstermektedir. Sonuçlar, geliştirilmiş algoritmanın İHA gereksinimlerini karşılamak için yumuşak ve güvenli bir yol bulabileceğini göstermektedir (WANG Lei ve ark., 2017).

Diğer en çok kullanılan yöntem’de genetik algoritma, dinamik ortamlardaki genetik algoritmalara dayalı bir İHA yolu hazırlama yöntemi önermektedir. Önerilen

algoritmanın dinamik olarak değişen bir ortamda optimal olmayan engelsiz bir yol bulmakta etkili olduğunu göstermektedirler. Dinamik olarak değişen bir ortamda, yol ve algoritma, tüm aday çözümlerin uygun ve kullanılabilir bir yol alanı içinde garanti edilebileceğini göstermektedirler (Xiao-Guang Gao ve ark., 2005). Yaptıkları bu çalışmada, genetik Algoritmayı (GA) kullanarak, quadrotorun bir engele çarpmadan enerji ve zamandan tasarruf etmek için kat etmesi gereken en kısa yolu belirlenmektedir. Belirli bir zamanda enerji israfını önlemek için önemli olan aracın güç tüketimini en aza indirecektir. Algoritmanın, engellerin sınırın herhangi bir noktasında belirtildiğini varsayarak engellerden kaçabileceğini de gösterir. Sonuca göre GA en iyi rotaları aramayı gerçekleştirebileceğini göstermektedirler (Reagan L Galvez ve ark., 2014). Diğer araştırmacılar anlatıkları bu yazıda, bu durumda radar olan engelden kaçınmak için İHA'nın yol planlaması için Genetik Algoritmalar ve Yapay Sinir ağına dayanan yeni ve etkili bir yöntem sunulmaktadır. Genetik Algoritmalardan elde edilen çıktılar Yapay Sinir Ağlarını eğitmek için başarıyla kullanılmış ve tüm durumlarda performansı 3B simülasyonlar kullanılarak gerçekleştirilmiştir. Sonuca göre önerilen yöntem daha hızlı ve daha iyi performans göstermektedirler (S Aditya Gautam ve Nilmani Verma, 2014). Yapılan bu çalışmada, İHA'ların rota planlama problemi için bir optimizasyon çözümü incelenmiştir. Gerçek yaşam problemlerine uyumun sağlanması için çalışma 3B yüzeylere dayanılarak yapılmıştır. Böylece, İHA'ların hareket planlamasında maliyet ve zaman optimizasyonu gerçek problemler için sağlanmaktadır. Bu amaca ulaşmak için yaygın olarak tercih edilen optimizasyon yöntemlerinden biri olan Genetik Algoritma kullanılır (Hicran Yılmaz ve Şengül Doğan).

Yapılan bu çalışmada, sivil İHA'lara veri aktarımı sağlamak için mevcut 3G iletişim ağlarını kullanmaktadırlar. Önerilen yöntem MOGA'nın (Multi-Objective Genetic Algorithm) path-greedy ve signal-greedy algoritmalara kıyaslayarak 1.32 ve 3.22 kat daha iyi sinyal kalitesi verdiğini göstermektedirler (Fan-Hsun Tseng ve ark., 2017). PSO (Particle Swarm Optimization), araştırmacılar tarafından önerilen bu yöntemde benzersiz bir 3D yol planlama problemi formülasyonu ve çözüm yaklaşımı sunmaktadırlar. Düşman tehditleri nedeniyle riski en aza indirmek ve yakıt tüketimini en aza indirmek için tasarlanmıştır. İlk tasarım noktası, geçerli konum ile istenen hedef arasındaki düz yoldur. PSO'yu kullanarak, optimize edilmiş bir yol B-spline eğrileri aracılığıyla oluşturulur. Ortaya çıkan yollar, maksimum güvenlik, minimum yakıt tüketimi tercihiyle optimize edilebileceğini göstermektedir. (Jung Leng Foo ve ark.,

2006). Yapılan bu çalışmada, voroni diyagramı ve DPSO'ya dayanan ortak bir kontrol stratejisi açıklanmıştır. GA'nın zamanlama kısıtlamaları olan karmaşık grup görevi için kullanılabileceğini göstermektedir (Han Tong, 2012).

Diğer önerilen yöntem ise RRT, İHA yolunun planlanmasında hızlı ve etkili bir pratik yaklaşım olarak veri yapılarının ve algoritmaların yeni bir kombinasyonu sunulmuştur. RRT algoritması tarafından üretilen yollar, yoldaki noktalar arasında herhangi bir engel olmadan en kısa yolu ekleyen Dijkstra algoritması boyunca yolu göstermiştir. RRT yaklaşımının, yinelemelerin sayısı arttıkça keşif yaklaşma olasılığı olan karmaşık engelli alanlardan uygun bir yol bulması garanti edilirken, ortaya çıkan yollar, RRT algoritması tarafından oluşturulan yollar, yoldaki noktalar arasında en kısa yolu bulan Dijkstra algoritmasından geçtiğini gösterdiler (Jayesh Amin ve ark., 2006). Yapılan bu çalışmada, özerk bir İHA yol planlama çerçevesi önermektedir. Yürütme işlemi üst sınırların analiz edilmesi ve kaldırılması veya planlamacıyı PRM (probabilistic roadmap method) veya RRT (rapidly random exploring tree) için çağırarak yeni planlar oluşturmak için harcanabilecek zamanda eski planların onarılmasıyla gerçekleştirildi. Sonuçlar, bu tekniklerin insansız hava araçları (İHA) alanında kullanılmasının uygulanabilirliğini göstermişlerdir (Mariusz Wzorek ve Patrick Doherty, 2006). Yapılan diğer çalışmada, karışık doğal ortamlarda çalışan insansız hava aracı (İHA) için 3Bbir yol planlama algoritması sunmaktadır. Hızla keşfedilen Rastgele Ağaçlar (RRT) algoritması, çarpışma içermeyen ara noktaların üretilmesi için kullanılır. Sonuca göre önerilen yöntem daha iyi performans göstermektedir (Kwangjin Yang ve Salah Sukkarieh, 2008). Yapılan bu çalışmada, karmaşık 3B ortamlarda İHA navigasyonu için geliştirilmiş PRM (probabilistic roadmap method) algoritmasına dayanan gerçek zamanlı bir yol planlama yöntemi önermektedir. Yol haritası düğümlerinin 3B alanda daha dengeli dağılımını sağlamak için sınırlayıcı kutu dizisinde rastgele örnekleme yoluyla değiştirilmiş bir Olasılıklı Yol Haritası Yöntemi (PRM) tanıtılmıştır. Çalışma alanını bölerek 3B nokta bulutunu işlemek ve çevreyi temsil etmek için serbest vokselleri çıkarmak için octree algoritmasını kullanmaktadır, yol haritasında yol aramak için A* algoritması ile gerçekleştirilmiştir. Algoritmanın gerçek zamanlı olarak çarpışma dışı bir yol oluşturabildiğini göstermektedir (Fei Yan ve ark., 2012). Yapılan bu çalışmanın amacı, RRT'nin karmaşık, dengesiz dinamikleri ve kayda değer sapmalara karşı güvenliğini korurken, karmaşık, dengesiz dinamikleri ve kayda değer sapmaları olan robotik taşıtlarda çevrimiçi olarak kullanılmasını sağlayan standart

RRT algoritmasına sayısız uzantı sağlamaktır. Önerilen yöntemin daha iyi sonuçlar verdiği göstermektedir (Yoshiaki Kuwata ve ark., 2008). Sundukları bu yazıda, özerk mobil kara taşıtlarının evrensel kontrol ve yörünge planlaması için kullanılacak bir test ortamı tasarımıyla ilgilidir. Yörünge planlama algoritması çevrimdışı çalışır ve kullanılan yöntem, robot sistemi için örnekleme tabanlı artan bir arama yöntemi olan RRT algoritmasının uyarlanabilir bir versiyonudur (Mert Turanlı ve Emre Koyuncu).

2.2. Mobil Robotlar İçin Literatür Taraması

Mobile robotlarda farklı yol bulma algoritmaları kullanılmaktadır. İşlem süresinin kısaltılması için A* algoritmasındaki yapılacak değişiklikler bu yazıda belirtillmiştir. Önerilen A* algoritması, sezgisel işlevin değerini başlangıçta değil çarpışma aşamasından hemen önce belirler ve işlem süresinde daha yüksek hızda iyi bir düşüş sergilediğini göstermektedirler. Sonuca göre A* algoritması kullanarak işlem süresinde maksimum 95% iyileşme göstermektedirler. Diğer araştırmacılar, Hojjat Salehinejad ve ark. istenen iki nokta arasındaki en iyi duruma getirilmiş çok parametreli yolu bulmak için çok parametreli A* (A-yıldız) -antlara dayalı bir algoritma önermiştiler. Elektronik haritaları kullanarak kullanıcının istediği parametrelere göre yolları tanımaktadır. Önerdikleri yöntem ve karınca algoritması arasındaki karşılaştırma sonuçları, önerilen yöntemin karınca algoritmasına karşı etkinliği ve düşük maliyetli işlev sonuçlarını göstermektedir (Akshay Kumar Guruji ve ark., 2016). Yapılan bu çalışmada, bir labirentte mobil bir robotun yolunu planlamak için bir algoritma sunmaktadırlar. Kamera tarafından elde edilen bir görüntü kullanmaktadırlar. Algoritma, labirent'te son hedefe giden en kısa yolu bulduğununu göstermektedirler. Yörünge planlamasının amacı, bir seyyar robotun planlanan yol bouyuna harektini planlamaktır. Sonuça göre, algoritmanın bir labirentteki bir nesneyi büyük boyutta (1600 x 1013 piksel) 500 ms'den daha kısa bir sürede hareket ettirebildiğini göstermektedir (MILENA Karova ve ark., 2015). Anlatıkları bu makalede, bir mobil haritanın ızgara haritasını temel alarak planlamasını anlatmaktadırlar. A* algoritmasının birkaç modifikasyonunu (Temel Theta*, Phi*) ve iyileştirmeleri tanıtmaktadırlar. Sonuca göre, A* algoritmasının hesaplama süresi diğer algoritmalara göre daha az olduğunu göstermektedir (František Duchoň ve ark., 2014). Anlatıkları diğer makalede, statik küresel ortamda tekli ve çoklu mobil robotlar için en yakın güvenli yolu elde

etmesini sağladığını anlatmaktadırlar. Sonuç olarak güvenlik parametresi kullanılmış, güvenlik parametresinin optimizasyon ve yol uzunluğu üzerindeki etkisini göstermektedir (Daneshwar Sahu ve Anil Kumar Mishra, 2017). Yapay zeka algoritmalarıda mobile robotlarda kulanılmaktadır. Bilinmeyen bir statik ve dinamik ortamda otonom mobil robot navigasyon ve çarpışmayı önleme için bir bulanık mantık sistemi kontrolörü ve Fuzzy-WDO hibrid tanıtılmaktadırlar. Sonuclara göre önerilen yöntem fuzzy-WDO daha az yol uzunluğu göstermektedir (Anish Pandey ve Dayal R Parhi, 2017). Yapılan bu çalışmada, küresel ortamda bulunan engellerden kaçınmak ve istenen hedef pozisyona güvenli bir şekilde ulaşmak için bulanık bir mantık kullanarak akıllı bir algoritma olarak bulanık bir mantık kullanılarak bir insansı için bir gezinme kontrolörü geliştirilmiştir. Engel mesafeleri ve hedefe yönelik taşıma açısı kontrolöre giriş ve engellerden kaçınmak için gerekli hızlar çıkış olarak kabul edildiğni göstermektedir. Simüle edilmiş ve deneysel yol uzunluğunun karşılaştırılarak sonuca göre Simülasyon yol uzunluğuna göre (2.94m) Deneysel yol uzunluğuna göre (3.01m) olduğunu göstermektedir (Asita Kumar Rath ve ark., 2018). Bu makalede, Bulanık Mantık ve Genetik Algoritma kullanan bir mobil robot için bir hedefe yönelik global yol planlama için yeni bir algoritma sunulmaktadır. Kapalı ortamlarda mobil yönünü hedefe doğru yönlendirmek amaçlanmaktadır. Sonuca göre önerilen yöntem daha iyi sonuç elde edildiğini göstermektedir (Rame Likaj ve ark., 2017). Sundukları bu yazıda, iki farklı yöntem Bulanık Mantık Denetleyicisi (FLC-Sugeno) ve Yapay Nöronal Bulanık Mantığı anlatmaktadırlar. Engeller arasındaki yolların çok sıkı olması durumunda, nöronal algoritma kullanıldığında belirli durumlarda bulanık mantık arızalandığında robotun problemsiz bir şekilde hareket edebildiğini göstermektedir. Sonuca göre önerilen yöntem daha kısa yol uzunluğunu göstermektedir (X Bajrami ve ark., 2015). Bu makalede, hedef önyargı arama stratejisini ve rasgele ağaç algoritmasını temel hızlıca araştıran temel hızlı araştırmayı geliştirmek için hem mesafeyi hem de açısını dikkate alan yeni bir metrik işlev önermektedirler ve sinir ağları eğri işleme sonrası için kullanılmaktadırlar. Karmaşık bir ortamda simüle edilmesi ve basit, hızla keşfedilen rastgele ağaç algoritması ile karşılaştırılması sayesinde, iyileştirilmiş algoritmanın yol planlama problemini ele almada iyi bir performansa sahip olduğunu kanıtlayarak iyi gerçek zamanlı performans ve nispeten daha kısa ve daha yumuşak planlanmış bir yol gösterir (Pengchao Zhang ve ark., 2018). Sunduklar bu yazıda, Izgara ayarlarındaki mobil robotun başlangıç noktası ile bitiş noktası arasında en uygun yolu bulmasına yardımcı olmak için genetik bir algoritma kullanılmıştır. Önerilen kontrol algoritması,

dört bitişik hareketi gerçekleştirir, böylece yol planlaması karmaşık düşük karmaşık arama alanlarına uyum sağlayabilmektedir (AT Ismail ve ark., 2008).

Kullanılan diğer yöntem ise, modifiye karınca kolonisi algoritması, robotun büyüklüğü ve dönüş durumu dikkate alınarak optimal yolu bulmak için kullanılmıştır. Modifiye edilmiş karınca koloni algoritması, seçilen yolun sadece yolun uzunluğunu değil aynı zamanda mobil robotun dönüş durumunu da göz önünde bulundurarak yeniden değerlendirmesini sağlar ve robotun çalışması için optimum bir yol sağlayabildiğini göstermektedir (Cheng Chen ve ark., 2017). Yapılan bu çalışmada, Karmaşık ortamlarda mobil robotun yol planlama problemini çözmek için Ant Koloni Optimizasyonu algoritması önermişlerdir. Her durumda, engellerden veya duvar kenarlıklarıyla çarpışmadan, başlangıçtan son istenen pozisyonlara kadar optimal veya optimal seviyeye yakın minimum uzunluktaki yollar bulmak mümkün olduğunu göstermektedirler (Ronald Uriol ve Antonio Moran, 2017). Sundukları bu yazıda, Özellikle Dijkstra'nın algoritmasını en kısa yol algoritmasını kullanan ve kablosuz özel ağ kullanarak bu önerilen konsepti geliştirmek için kablosuz iletişim sağlayabilen akıllı bir yol planlama önerilmektedir. Elde edilen sonuçlar, akıllı yol planlama algoritmamızın üç farklı yaklaşımda planlandığı gibi çalıştığını ve bir rotada engellenen bölümlerden kaçınabilen bu algoritma kavramının başarıyla uygulandığını ve en kısa yol olan alternatif bir yolun tanımlandığını göstermektedir (Lyle Parungao ve ark., 2018). Yapılan bu çalışmada, Mobil robot tasarımı açık kaynaklı simülasyon yazılımı V-REP kullanılarak yaratılmış ve alt optimal ve çarpışmadan uzak yolun belirlenmesi için Dijkstra algoritması kullanılmıştır. Simüle edilen sonuçlar, oluşturulan robot yolu planlama ortamlarındaki azalma yaklaşımının zaman ve hız açısından etkili olduğunu göstermiştir (S Julius Fusic ve ark., 2018). Sundukları bu yazıda, dışbükey şekilli engellerin köşelerini ağ düğümlerine dönüştürmek, bir maliyet işleviyle en kısa yolu belirlemek ve başlangıç ve bitiş noktalarını birbirine bağlayan bir engelden kaçınma yolu bulmak için değiştirilmiş bir Dijkstra algoritması kullanılmıştır. Sonuçlar, sadece yol planlamada önerilen yöntemin değil, aynı zamanda iyi sonuçların alınabileceğini göstermektedir (Ter-Feng Wu ve ark., 2017).

Anlatılan bu makalede gelişmiş bir karınca kolonisi algoritması önerilmekte ve robot yolu planlamasına uygulamaktadır. Geliştirilmiş algoritmada, feromon miktarı her döngünün bazı kısa yollarında güçlendirilir ve feromon buharlaşma hızı dinamik olarak ayarlanır ve böylece geçiş olasılığı artar. Sonuçları, bu önlemlerin etkili olduğunu ve

global optimizasyon yeteneğini, arama hızını artırabildiğini ve erken olmasını önleyebileceğini göstermektedir (Jingang Cao, 2016). Bu makalede ise (Tyagi, 2017) önerdikleri bu yöntemde bir düğümden başka bir düğüme en kısa yolu grafikte bulmak için kullanıldığını anlatmaktadır. Sonuca göre önerdikleri yöntem daha iyi sonuçlar göstermektedir. Bu makalede Normal Olasılık Dağılımı üzerinde çalışılan değiştirilmiş Firefly Algoritmasını (FA) kullanarak etkin navigasyon kontrolörü geliştirmiştir. Sadece söz konusu zorlukları değil, aynı zamanda ateş böceklerinin hesaplanan hesaplamasını ve rastgele hareketini ek bir avantaj olarak en aza indirir. Önerilen denetleyicinin daha iyi performans gösterdiği kanıtlanmaktadır (BK Patle ve ark., 2017). Yapılan bu çalışmada, robot kolunun çeşitli uygulamalarda kullanılmasına çok dikkat edilmektedir. Bu nedenle, optimal yol bulmanın kol hareketini yükseltmek ve yönlendirmek için önemli bir rolü vardır. Yol planlamasının temel işlevi, engellerin çarpışmasını önleme, zaman aralığını azaltma, yol seyahat maliyetini düşürme ve kinematik kısıtlamalarını yerine getirme dahil hareketin amaçlarına uygun bir yol oluşturmaktır. Sonuçlara göre, değiştirilmiş PSO algoritması zorlu ortamlarda bile en uygun yolu bulmak için verimli bir şekilde çalıştığını göstermektedir (Ahmed T Sadiq ve ark., 2017). Anlatılan bu yazıda, bilinmeyen bir ortamda mobil bir robotun yapay potansiyel alan (APF) yöntemini kullanarak navigasyonunu ele almaktadır. Robotun yolundaki engellerden kaçınarak mobil robotun başlangıç noktasından, hedef noktaya yol planlaması için bir yöntem geliştirilmektedir(Alaa A Ahmed ve ark., 2015).

Literatür taramanın amacı yapılan çalışmaları öğrenmek ve kullanılan farklı yöntemleri tanımaktır. Bu tez konusunda araştırılan makaleler arasından çok sayıda yöntem arasından üç tane farklı algoritma kullanılmaya karar verilmiştir. Yöntemler sırasıyle, A*, Dijkstra ve RRT kullanılmıştır.

Simülasyon sonucuna göre üçüncü test sonucunu incelendiğinde, A* algoritmasına göre 19m'lik yolu 0.0555 saniyede gittiği görülmektedir. Dijkstra algoritmasına göre 19m'lik yolu 1.4563 saniyede gittiği görülmektedir. RRT algoritmasına 33m'lik yolu 6.4095 saniyede hedef noktasına gittiği görülmektedir. Önerilen algoritmalar literatürdeki algoritmalarla kıyasladığımızda daha hızlı çalıştıkları ve mobil otonom robotlarda en çok kullanılanlardır, bu nedenle bu algoritmalar tercih edilmiştir. A* algoritması alanı sadece hedefe doğru taradığı, RRT algoritmasının her simülasyonda hedefe doğru farklı noktalar belirleyerek yol çizdiği, Dijkstra algoritmasının ise çok daha geniş alanı tarayarak hedefe vardığı görülmüştür. A*

algoritmanın avantaji ise hızlı olmasıdır. Dijkstra algoritması ağırlıklı ve yönlü graflar için geliştirilmiştir. RRT algoritmasının avantajı ise programlanmasının oldukça basit olmasıdır; bu yüzden kalite bir yol bulmak için hızlı bir analiz yapılmasını sağlar. RRT algoritmanın amacı, dinamik bilinmeyen ortamda hızlı engel tespiti ve gereksiz manevra noktaların azaltılmasıdır. Dördüncü gerçek ortamın test sonucuna göre, A* algoritmasına göre 103m'lik yolu 0.3864 saniyede varmıştır. Dijkstra algoritmasına göre 103m'lik yolu 1.2077 saniyede varmıştır. RRT algoritmasına 35m'lik yolu 11.7482 saniyede hedef noktasına varmıştır. Gerçek ortamda yapılan çalışma sonucunda RRT algoritması A* algoritmasına göre hedef için daha kısa yolu belirlemesine rağmen A* algoritması daha hızlı sürede hedefe vardığı görülmüştür.

3. İNSANSIZ HAVA ARAÇLARI (İHA)

Günümüzde insansız hava araçlarının (İHA) kullanımı artmaktadır. Batarya sorunu, özellikle sabit kanatlı İHA'lardaki en büyük sorunlardan birisidir. Batarya sorunu nedeniyle, üç boyutlu (3B) yol planlaması, otonom mobil ve uçan robotlarda önemli bir konudur. 3B yol planlama yöntemlerinin geliştirilmesi ve mobil otonom robotlar için geliştirilen 2B yol planlama algoritmalarına dayalı olarak uygulamanın hızlandırılması istenmektedir. Bağımsız İHA'ların kullanımı yeni kurulan bir konu olduğundan, yüksek performanslı yol planlama teknikleriyle bağımsız İHA'lar, bataryaları ile maksimum bağımsız görevler üstlenebilmektedir. Bu yüzden bu tez çalışması oldukça önemlidir. İHA, 1916'da ilk İHA'dan sonra askeri amaçlar için kullanılmaktayken İnsansız hava araçları, 1950'lerden sonra sivil amaçlar için kullanılmıştır. Uzaktan algılama ve fotoğrafçılık için uçağın kullanımını belirlemek veya azaltmak mümkün değildir. Drone'lar günümüzde birçok alanda kullanılabilmektedir (Mustafa Dikmen, 2015).

3.1 İHA uygulama alanı

Birçok teknolojik yenilikte olduğu gibi, askeri alanlarda yaygın kullanımı, insansız uçakların geliştirilmesini başlattı. Askeri harekat, insansız uçakların bütçelerinin veya yatırımlarının çoğunu karşılamaktadır. Bu askeri operasyonlar esas olarak denetim, gözetim ve istihbarat faaliyetleridir. Ayrıca, tehlikeli bir spektrumda askeri, kimyasal, biyolojik, nükleer ve radyolojik işlemler kullanılmaktadır (Emrah Yiğit ve ark.).

İHA’ların uygulama alanları aşağıdaki gibidir:

- İzleme,
- Tarımsal ilaçlama,
- İmha gibi askeri amaçlar,
- Keşif,
- Ulaşım,
- Kamera fotoğrafçılığı,
- Yangın söndürücü,

- Tarama,
- Anlık / genel hedef belirlemek,
- Barış zamanında genel kontrolu,
- Savaş sırasında düşmanın faaliyetlerini denetlemek,
- Anlık hava durumu tahmini,
- Kapalı havalarda yol planlaması,
- Yön bulma,
- Düşman takibi,
- Kablosuz sinyal iletimi.

Bilinenlerin aksine, dronlar yalnızca askeri uygulamalarda kullanılmamaktadır. İHA sektörü, özellikle ABD hükümetinin İHA'ları askeri alanların dışında sivil havacılıkta kullanmasına izin verildiğinden hızlı bir şekilde gelişmektedir (Emrah Yiğit ve ark.).

- Ticari amaçlar için,
- Bilimsel çalışmalarda,
- Üniversite laboratuvar çalışmaları,
- Arama kurtarma faaliyetlerinde,
- Güvenlik / kamu düzeni ile ilgili uygulamalarda,
- Meteorolojik çalışmalar,
- Film endüstrisinde,
- Çevre ve Doğa Çalışmaları,
- Hava fotoğrafçılığında,
- Tarımsal alanlarda vb. Kullanım alanlarda.

İHA kullanımı tehlikeli uygulamalarda özellikle istenir ve uzun vadeli zaman alan alanlarda (örneğin, 40-50 saat) kullanımı kolaydır. Ayrıca, insan kullanımının tehlikede olduğu nükleer operasyonlar gibi işlemlerde de kullanılabilmektedir. Yaşamımızın her alanında kullandığımız web servisleri (özellikle Google) ve sosyal medya uygulamaları da İHA teknolojisinden yararlanmaktadır.

3.2 İHA’ların ayırt edici özellikleri

İHA'ların özellikleri aşağıdaki gibidir:

- Dizayn esnekliği bulunması,
- Tehlikeli ortamlarda uçuş yapılabilmesi,
- Görev esnekliği ve çeşitliliğinin olması,
- Havada kalış süresinin uzatılabilmesi,
- İnsanlı uçaklara göre maliyetli daha düşük olması,
- Operasyon alanında istihbarat toplama ve hedefe ulaşma kabiliyetinin olması,
- Kuvvet koruma görevini gerçekleştirebilmesi,
- Hedef tespiti ve hedef izleme özelliğinin olması şeklinde sıralanmaktadır (Mustafa Dikmen, 2015).

3.3 İHA teknolojileri

İnsansız hava araçları, herhangi bir bölgeden iniş ve biniş, uzaktan kumanda, yarı otomatik veya tam otomatik uçuş kapasitesi gibi benzersiz şekillerde tasarlanmıştır. Şekil 3.1'da insansız hava araçları örneği görülmektedir. Günümüzde İHA'ları birçok otopilot sistemine ve gelişmiş GPS / IMU sahiptir. Bu sistemler sayesinde otomatik uçuş planına yapılabilir ve toplanan görüntülerin yaklaşık koordinatları belirlenebilir. Ek olarak, GPS / IMU, imge kazanımı sırasında insansız hava aracına geçici resimlerin tanınmasını sağlamaktadır. UAV'lerin uzaktan algılama ve fotogrametri için en büyük yararı, bilgilerin mümkün olduğu kadar çabuk ve küçük alanlarda tekrarlanan kesin okumalar için elde edilebilmesidir. Konumsal hassasiyeti 10 cm'den az olan görüntüler, uygun ve uygun maliyetli uçaklar tarafından tanımlanan bir bölgede çok düşük irtifalarda (40 metre ila 1000 metre) uçarak elde edilebilmektedir. Günümüzde sivil amaçlarla uydu platformlarından ve insanlı uçaklardan elde edilen görüntülerin kesinliği 10 cm'den az değildir. Bu nedenle insansız hava araçlarının yüksek hassasiyet gerektiren uygulamalarda kullanılması bu nedenle oldukça yararlıdır (Mustafa Dikmen, 2015).



Şekil 3.1. İnsansız hava araçları örnekleri (Mustafa Dikmen, 2015)

3.4 İHA çeşitleri

Drone'lar çok çeşitli boyutlarda üretilir ve en büyüğü Predator drone gibi askeri alanlarda kullanılır. Kısa pist gerektiren sabit kanatlı insansız uçaklar vardır. Genellikle büyük bölgeleri kapsamak, coğrafi araştırmalar gibi alanlarda faaliyet göstermek veya yabani hayatı ile mücadele etmek için kullanılırlar.

Temel olarak iki gurup bulunmaktadır:

- Sabit kanatlı İHA lar
- Döner kanatlı İHA lar

3.4.1 Sabit kanatlı İHA lar

Sabit kanatlı İHA'nın birincil faydası, döner kanattan çok daha basit olmasıdır. Basit yapısı ile daha az karmaşık bir bakım ve onarım yöntemi sunar. Bu nedenle kullanıcının daha düşük bir fiyata, daha fazla çalışma süresine sahip olmasını sağlar. Daha da önemlisi, basit yapı daha yüksek hızlarda daha uzun uçuş sürelerinin avantajını sağlayan daha verimli aerodinamik sağlar ve böylece belirli uçuş başına daha büyük araştırma alanları sağlamaktadır (Dilek Funda Kurtuluş ve OzanTekin alp, 2010).

Sabit kanatlı bir yapının tek dezavantajı kalkış ve iniş için bir pist veya başlatıcının gerekli olmasıdır, ancak DKİL (dikey kalkış / iniş) ve KKİ (kısa kalkış / iniş) çözümleri bu sorunun ortadan kaldırılmasına yardımcı olmak için çok popülerdir.

Ayrıca, sabit kanatlı hava taşıtları, sürekli ileri hareket halinde kalması gereken, sürekli dönen bir kanat İHA'sının olabildiğince sabit kalmasını sağlamak için kanatları üzerinde hareket eden havaya ihtiyaç duyarlar. Sabit kanatlı çözümlerin inceleme araştırması gibi sabit uygulamalar için ideal olmadığını göstermektedir (Dilek Funda Kurtuluş ve OzanTekin alp, 2010). Şekil 3.2'da Sabit kanatlı İHA örneği görülmektedir.



Şekil 3.2. Sabit kanatlı İHA örneği (Dilek Funda Kurtuluş ve OzanTekin alp, 2010)

3.4.2 Döner kanatlı İHA lar

Döner kanat İHA'ları, rotor olarak bilinen sabit bir direk etrafında dönen 2 veya 3 rotordan oluşmaktadır.

- **3 rotor (tricopter):**

Her biri bir motora bağlı olan, üç kolu bulunan bir İHA'dır. Önü, iki kol arasında (Y3) olma eğilimindedir. Kollar arasındaki açı 120 dereceye kadar değişebilir. Bu İHA'nın dezavantajı, İHA tasarımı aynı olmadığı için arka motoru döndürmek için normal bir RC servo kullanmasıdır. Bu nedenle tasarım diğer motorlu motorlardan daha basittir (Hüseyin Şahin ve Tuğrul Oktay, 2019).

- **4 rotor (quadcopter):**

Dört motorlu / pervaneli ve en sık kullanılan çok rotorlu tasarıma sahip dört destek koluna sahip bir İHA'dır. Esnek ve kolay bir yapıya sahiptir. İHA'nın dezavantajı, bir motor veya pervane arızası meydana geldiğinde düşme ihtimalleri vardır. Quadcopter'e iki ekstra rotor eklendiğinde bir hexacopter elde edilmiştir 6 rotor (hexacopter) (Nanyaporn Intaratep ve ark., 2016).

- **8 rotor (octocopter) :**

Çok çeşitli kurulumlara sahip, her biri bir motora bağlı altı kol vardır. Ekstra motorlar ve parçalar helikoptere ağırlık katar bu nedenle akünün aynı quadcopter uçuş süresini gerçekleştirmesi için daha büyük bir enerji gerektirir. Dönen bir uçağın temel yararı düşey kalkış ve düşme kapasitesidir. Bu sayede kullanıcının büyük bir iniş / başlangıç bölgesine ihtiyaç duymadan daha dar bir bölgede çalışmasını sağlamaktadır. Uçma ve manevra yapma, kanatlı insansız uçakların döndürülmesi, hassas manevra kontrolleri ve uzun bir süre boyunca tek bir hedefe görsel görünürlük sağlama kapasitesi gibi uygulamalar için idealdir. Dönen kanatlı hava taşıtları daha fazla mekanik ve elektronik karmaşıklık içerir, genellikle daha karmaşık bakım ve onarım faaliyetlerine katkı sağlayarak kullanıcının çalışma süresini kısaltır ve daha yüksek işletme maliyetleri sağlamaktadir (Hüseyin Şahin ve Tuğrul Oktay, 2019). Şekil 3.3'da Döner kanatlı İHA'lar örneği görülmektedir.



Şekil 3.3. Döner kanatlı İHA örneği (Hüseyin Şahin ve Tuğrul Oktay, 2019)

- **Çok Rotor**

Teknoloji sürekli olarak gelişirken, birçok rotor temelde çok verimsizdir ve yerçekimi ile savaşmak, İHA'yı havada tutmak için çok fazla enerji gerektirir. Mevcut pil teknolojisi ile hafif bir kamera yükü taşırlar ve 20-30 dakika ile sınırlandırılırlar. Ağır kaldırma, çok rotordan daha fazla ağırlık taşıyabilir, ancak uçuş süreleri daha kısa olmaktadır. Stabilize edilmesi için hızlı ve yüksek hassasiyetli gaz modifikasyonlarına ihtiyaç duyulduğundan, çeşitli rotorlara güç sağlamak için bir gaz motoru kullanmak pratik olmadığından elektrik motorlarıyla sınırlıdır. Dolayısıyla uçuş zamanlarında yenilenebilir bir enerji kaynağı ile çok küçük artışlar sağlanabilinir. Çok rotorların

dezavantajı, sınırlı dayanıklılık ve hızlarıdır. Ayrıca geniş ölçekli hava haritalaması, uzun süreli izleme ve boru hatları, otoyollar ve enerji hatları gibi uzun mesafeli araştırmalar için uygun değildir (Nanyaporn Intaratep ve ark., 2016). Şekil 3.4'da Çok Rotor İHA örneği görülmektedir.



Şekil 3.4. Çok Rotor İHA örnekleri (Nanyaporn Intaratep ve ark., 2016)

- **Tek Rotorlu Helikopter**

Tek rotorlu bir helikopter, çok rotorlu motorlardan çok daha büyük bir verimlilik avantajına sahiptir ve ayrıca bir gaz motoruyla daha uzun ömürler için de çalıştırılabilir. Tek rotorlu helikopter İHA örneği Şekil 3.5'da görülmektedir. Aerodinamiğin genel prensibi; rotor kanadı ne kadar büyük ve yavaş döndükçe, o kadar etkili olmaktadır. Bu nedenle, bir quadcopter, sekiz helikopterden daha etkilidir. Tek rotorlu bir helikopter, bir pervaneye göre dönen kanat gibi daha uzun bıçaklara izin verir ve yüksek etkinlik sunar. Dezavantajları; karmaşıklıkları, titreşimleri ve büyük eğirme bıçakları riskidir (Hüseyin Şahin ve Tuğrul Oktay, 2019). İHA larının çeşit özellikleri çizelge 3.1'de görüldüğü gibidir.



Şekil 3.5. Tek Rotorlu Helikopter İHA örneği (Hüseyin Şahin ve Tuğrul Oktay, 2019)

Çizelge 3.1. İHA larının çeşit özellikleri

İHA ÇEŞITLERI	ÖZELLİKLERİ
Sabit Kanatlı	• Uzun ömürlü • Geniş alan kapsamı • Hızlı uçuş hızı
Çok Rotor	• Ulaşılabilirlik • Kullanım kolaylığı • İyi kamera kontrolü • Kapalı bir alanda çalışabilme
Tek Rotorlu Helikopter	• Uzun ömür (gaz gücüyle birlikte) • Daha ağır taşıma kapasitesi

3.5 Gelecekteki İHA gelişimi

Gelecekteki İHA gelişimi için önem kazanacak olan özerklik teknolojisi aşağıdaki kategorilere girer:

- **Sensör füzyonu:** Araçta kullanılmak üzere farklı sensörlerden gelen bilgilerin birleştirilmesi.
- **İletişim:** Eksik ve hatalı verilerin varlığında çeşitli görevliler arasında iletişim ve koordinasyonun sağlanması.
- **Yol planlama:** Engeller gibi belirli amaç ve kısıtlamaları yerine getirirken aracın gitmesi için en uygun yolu belirlemek.
- **Yörünge Üretimi:** Belirli bir rotayı takip etmek veya bir yerden diğerine gitmek için en uygun kontrol manevrasını belirlemek.
- **Görev Tahsisi ve Çizelgeleme:** Zaman ve ekipman sınırlamaları olan bir grup görevli arasında ideal görev dağılımını belirlemek.
- **İşbirliği Taktikleri:** Herhangi bir görev senaryosunda başarı şansını en üst düzeye çıkarmak için ajanlar arasında en uygun diziyi ve faaliyetlerin konum olarak dağılımını oluşturmak.

3.6 İHA’ların avantajları

İHA'lar hava savunma bileşenlerinin dikkatini dağıtmak ve birincil saldırı bileşenlerinin ateş hattını geçmesini sağlamak için yem olarak kullanılabilmektedir. Artık, özellikle kameraları ve görüntü aktarma özellikleriyle, gözetim amacıyla sıklıkla kullanılmaktadır.Ayrıca tarımda ve reklamcılıkta aktif olarak kullanılmaktadır(Osman İyibilgin ve ark., 2016).

- Üretim ve tedarik masrafları pilot uçağa göre çok küçüktür, havada 48 saate kadar kalabilirler,
- Takımlar halinde koordineli bir şekilde uçabilir ve kısa bir sürede daha fazla bölge tarayabilirler,
- Zamandan tasarruf, maliyetten tasarruf, daha uzun çalışma saatleri.

3.7 Dezavantajları

Hava saldırılarına ve savunma sistemlerine karşı hassastırlar ve İHA'lar yer kontrolünün bağlantısının kesilmesinde toprak bileşenlerine tehlike oluşturabilmektedir (Osman İyibilgin ve ark., 2016).

3.8 Yol planlama teknikleri

Mobil robotlar için yol planlaması konusu, uzun yıllardır devam eden ve gittikçe daha fazla araştırılan bir konudur. Çok sayıda bilim insanı alternatifleri için etkili ve verimli yöntemler oluşturmak için çok çalışmışlardır. Bilim adamları bu teknikleri geliştirmek için iki genel yöntemle ilgilenmişlerdir. Birincisi, önceden bilinen küresel ortamı veya engel bilgileri ve robot özelliklerini kullanarak yol planlama çözümüne erişmek ve ikincisi, robot özelliklerini kullanarak yerel bilgi ve hareket planlaması ile ilgilidir. Bu iki yaklaşımı karşılaştırdığımızda, küresel yol planlamasının daha yavaş bir süreçle sonuçlandığını ve yerel yolların planlamasından daha fazla olduğunu görünmektedir. Bunun nedeni, yerel yolların düzeninin etkileşimli ve dinamik bir süreç olmasıdır. Küresel yol planlama yaklaşımı, yerel engellerden daha yüksek bir düzeyde çalışıyor gibi görünmektedir. Bununla birlikte, çevre dünya çapında yol planlamasında

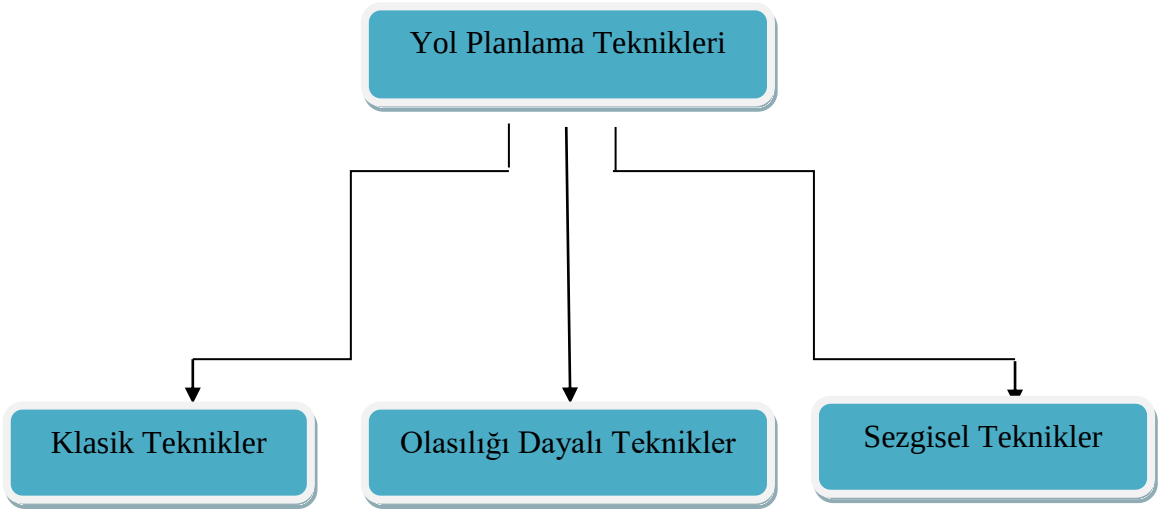
iyi bilindiğinden, dünya çapında planın yol kalitesini yüksek bir standarda yükseltmek için çalıştığı görülmektedir (Suvaydan, 2011).

Global ve lokal yol planlama yaklaşımları arasındaki temel farklar çizelge 3.2'de temel unsurları ile gösterilmiştir.

Çizelge 3.2. Global ve lokal yol planlamasının karşılaştırılması

GLOBAL YOL PLANLAMA	LOKAL YOL PLANLAMA
• Birikmiş ve önceden yola dayalı bir bilgi kullanılır. • Hedefe verimli ve uygun şekilde ulaşmayı planlar. • Dikkatli ve yavaş bir süreç vardır. • Uzun mesafeler ve uzun süreler için planlama yapmaktadır. • Basit model araçların planlaması için kullanılır. • Daha büyük engellerin bulunduğu alanlarda kullanılır.	• Sensör tabanlı acil durum bilgilerini kullanılır. • Daha hızlı ve daha olası seyahat planlar. • Reaktif ve hızlı işlemi vardır. • En yakın çevre ve en kısa süreler için planlama yapar. • Karmaşık model araç düzeni için kullanılır. • üçük engellerin bulunduğu alanlarda kullanılır.

Yol planlama ile ilgili sorunları çözmek için birçok yöntem geliştirilmiştir. Şekil 3.6'de yol planlaması için performans ve çözüm yöntemleri gösterilmektedir.



Şekil 3.6. Yol planlama teknikleri (Suvaydan, 2011)

3.8.1 Klasik yol planlama teknikleri

Yol planlama sorununu çözmek için mobil robotlar için en eski yöntem olarak tanımlanabilir. Yol planlama konularının çoğunluğu geleneksel yol planlama yöntemleriyle çözülebildiği için birçok çalışma klasik yöntemler kullanılmaktadır (Suvaydan, 2011).

3.8.2 Olasılığa dayalı yol planlama teknikleri

Olasılık tabanlı yol planlama yöntemleri, tüm arama alanını temsil eden vakaları seçerek geliştirildiğinden arama alanında oluşturulan yollar büyük olasılıkla ilk noktada seçilir. Daha sonra bu bölgede örneklenen noktalar birbirine bağlanır. Bu konunun arasındaki bağlantılardaki engellere bakarak yol sorununu gidermek için bir harita geliştirilmiştir (Suvaydan, 2011).

3.8.3 Sezgisel yol planlama teknikleri

Araştırma yöntemleri, genel kullanım nedenlerinden dolayı standart tekniklerle çözülemeyen sorunlara ve NP-Sert artışları gibi sonuçlar elde etmek için kullanılan hızlı ve etkili bir çözümdür. Deneysel yöntemler hemen hemen her sektörde, özellikle geometri sorunlarını çözmek için kullanılır. Robotlarda, özellikle son yıllarda sezgisel yöntemler üzerine çok fazla araştırma yapılmaktadır. Bu araştırmalardan biri mobil robotlar için yol planlaması veya hareket planlaması ile ilgili konularıdır. Mobil robotlar için rota planlama ile ilgili sorunları ele almak için deneysel yöntemleri kullanmanın nedenleri, bu tekniklerde, özellikle yukarıda belirtilen klasik tekniklerde ortaya çıkan sorunlardır (Suvaydan, 2011).

3.9 Yol Bulma Algoritmaları

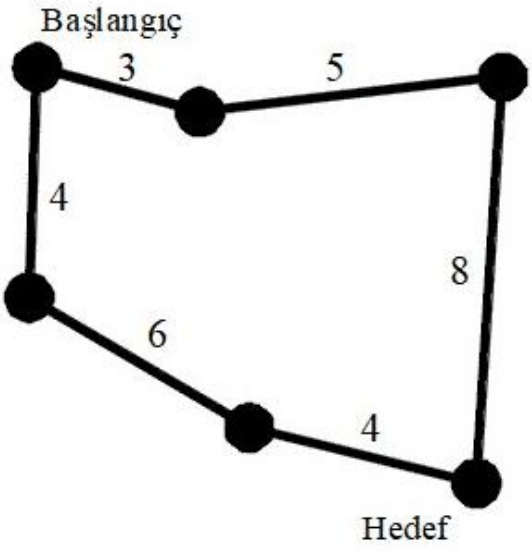
İnsansız hava araçları (İHA) için, yol planlamasını bir optimizasyon problemi olarak formüle eden bariyerler ile dolu üç boyutlu bir alanda gezinmek için pek çok yaygın yol planlama tekniği vardır. İki grup, bu optimizasyon konularını ele almak için

kullanılan algoritmaları netleştirir: sezgisel ve sezgisel olmayan (veya kesin) yöntemler. Sezgisel teknikler, hesaplamalı zaman etkinliği için çözümün optimizasyonunu (veya kesinliğini) değiştirir. Sezgisel olmayan teknikler, hesaplamalı olarak pahalı alternatifler elde etmek için genellikle matematiksel prensipleri kullanır. Yol planlama algoritmaları için yaygın yöntem, alan mozaiği, matris ayrımı veya ikisinin karışımı gibi yöntemler kullanılarak tanımlanan bölge veya haritayı hesaplama alanlarına ayırarak başlar. Bu veri yapıları, olası İHA yörüngelerini oluşturmaya yardımcı olur. Yörüngeden sonra, üretilen yörüngeler, İHA'nın hesaplama bölgesi analizinde alabileceği yolu düzleştirme yöntemini içermektedir (Mohammadreza Radmanesh ve ark., 2018). Literatürde İHA yol planlama ve engellerden kaçınma konusu oldukça son yıllarda artmaktadır. Yol planlama ve yörünge planlama olarak da adlandırılan hareket planlaması, başlangıç durumu adı verilen ve mevcut durumdan hedef durumu adı verilen son duruma giden bir yolun belirlenmesini içermektedir. Yerel hedef, İHA'nın başlangıçtan hedef duruma kadar başarmaya çalıştığı merkezi hedefi belirtir. Genel amaç, İHA'yı küresel hedefe yönlendirmenin yolunu veya yörüngesini keşfetmektir (Mohammadreza Radmanesh ve ark., 2018).

4. MATERYAL VE YÖNTEM

4.1. A* Algoritması

A* algoritması, sezgisel bir yaklaşım algoritmasıdır. Bu algoritma, yol bulma ve grafik geçişlerinde kullanılan en iyi ve popüler yöntemlerden birisidir. Yolun fiyatına sezgisel bir değer uygulanarak, toplam maliyet belirlenir. Algoritma, grafik üzerinde çoklu düğümler veya noktalar arasında bağlantılı bir yolu verimli bir şekilde çizer. En düşük maliyetli yolunun kullanılması tercih edilir. Yol bulma ve grafik geçişinde yaygın olarak kullanılan, düğümler adı verilen noktalar arasında etkin bir şekilde hareket ettirilebilir bir yol çizme işlemi olan bir bilgisayar algoritmasıdır (František Duchoň ve ark., 2014). Böylece tüm düğümlerin gezinme gereksinimini ortadan kaldırmaktadır. Şekil 4.1 A* Algoritması için örnek düğümler görülmektedir.



Şekil 4.1. A* Algoritması için örnek düğümler

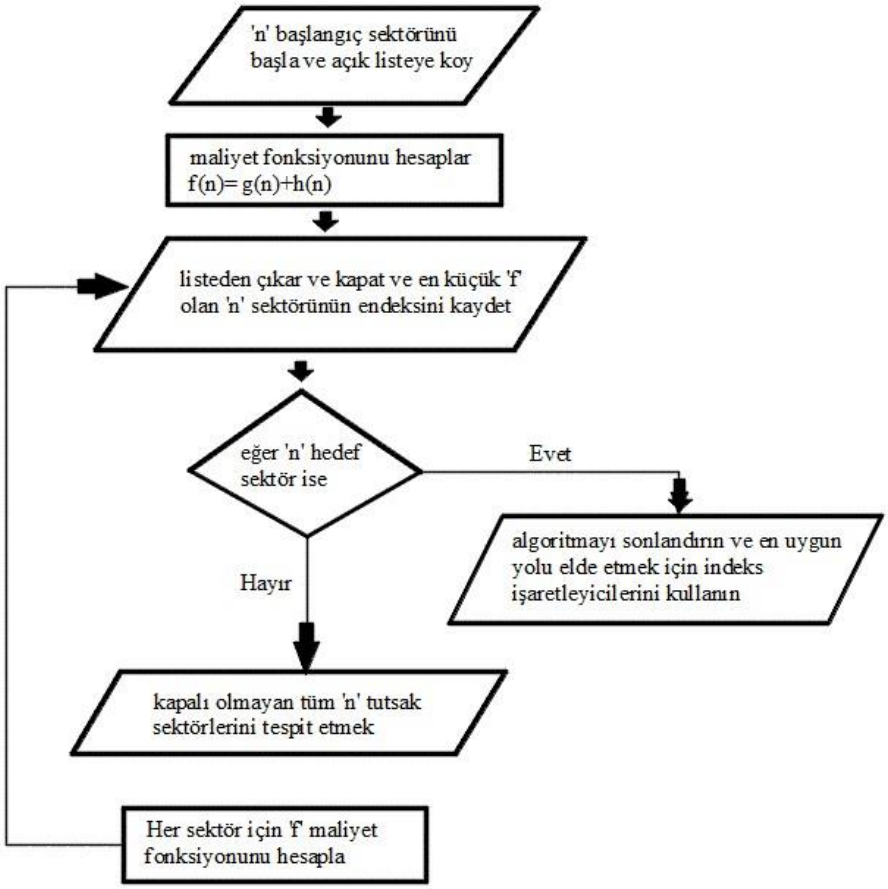
A* ilk kez 1963 yılında tanıtıldı ve o zamandan beri robot yolu planlama ve grafik geçişi için popüler bir yöntem haline geldi. A* algoritması uygun yol planlaması için sezgisel bir fonksiyon tabanlı algoritmadır. Bir ilk düğümden hedef düğüme en düşük maliyetli yolu bulmak için en iyi ilk aramayı kullanır. Bu yol planlamasının 2B uygulaması algoritması oldukça olgun ve kapsamlı bir şekilde geliştirilmiştir. Bununla birlikte, 3B uygulama hala hesaplamalı zorluklar sunmaktadır. Bu algoritmanın insansız hava araçllar için birçok rahatsızlık göz önünde bulundurulmadan yapılan deneysel

uygulaması, daha sonra herhangi bir fizibilite, verimlilik veya yakınsama kanıtı olmadan incelenmiştir. Bazı araştırmacılar, bağlı ağ grafiğini oluşturmak için hedef alanı birkaç bölüme ayırmak için iki boyutlu Voronoi haritasını kullanmaktadır. İnsansız hava araçların yüksek irtifa kalma oranları ve düşük yakıt tüketimi elde etmek için bir 3B büyük ölçekli savaş alanında insansız hava araçların gerçek zamanlı yol planlaması için bir A* algoritması sunulmaktadır (František Duchoň ve ark., 2014). Şekil 4.2 A* algoritma için akiş şeması görülmektedir.

Maliyet fonksiyonu aşağıda görüldüğü gibidir:

$$f(n)=g(n)+h(n) \tag{4.1}$$

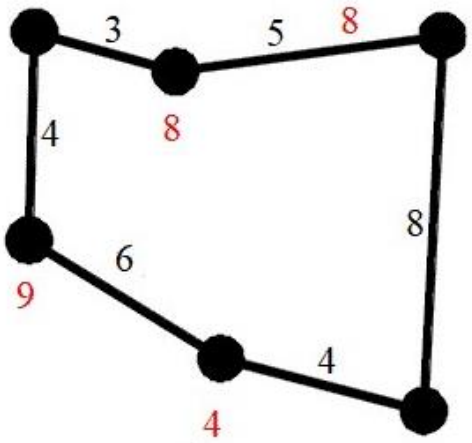
denklemde göründüğü gibi, f(n): hesaplama yapan sezgisel fonksiyon, g(n): Mevcut düğüme başlangıç düğümünden ulaşmanın maliyeti, h(n): Geçerli düğümden hedef düğüme olan tahmini mesafe.



Şekil 4.2. A* için akış şeması

Algoritmanın Çalışması:

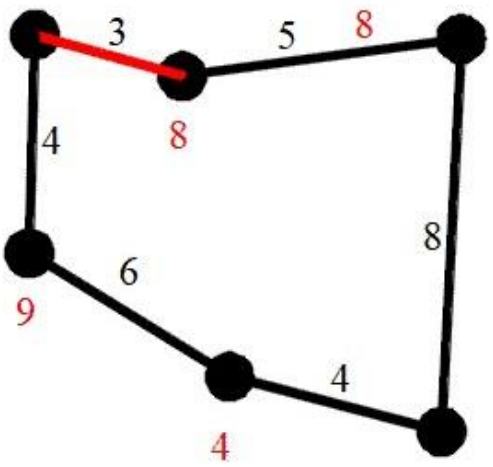
Algoritma toplama (4.1) işlemi oldukça basit bir yapıya sahiptir. Veri yapısı olarak bir öncelik sırası (priority queue) kullanan algoritmada en öncelikli olan düğüm f(n) değeri en düşük olan düğümdür. A*, en iyi ilk arama özelliğini kullanır ve belirli bir ilk düğümden bir hedef düğüme) en düşük maliyetli yolu bulur. A* grafiği geçerken, yol boyunca alternatif yol bölümlerinin sıralı bir öncelik sırasını koruyarak bilinen en düşük sezgisel maliyet yolunu izler. Sezgisel Fonksiyon olarak bir düğümün hedefe olan kuş uçuşu mesafesini almaktadır (Bknz. Şekil 4.3).



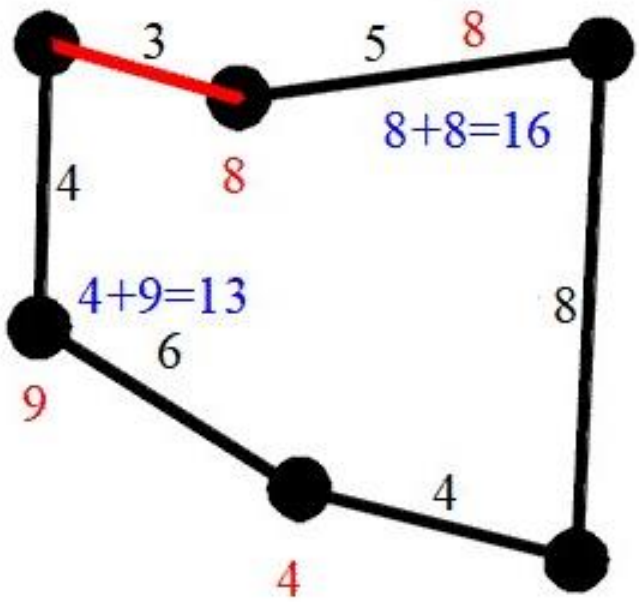
Şekil 4.3. A* Algoritması için kuş uçuşu mesafesi

Algoritmanın çalışması aşağıda sıraladığı gibidir:

- Algoritma, her adımda bir düğüm için en düşük değeri alır ve düğümü kuyruktan çıkarır (Bknz. Şekil 4.4 ve Şekil 4.5).

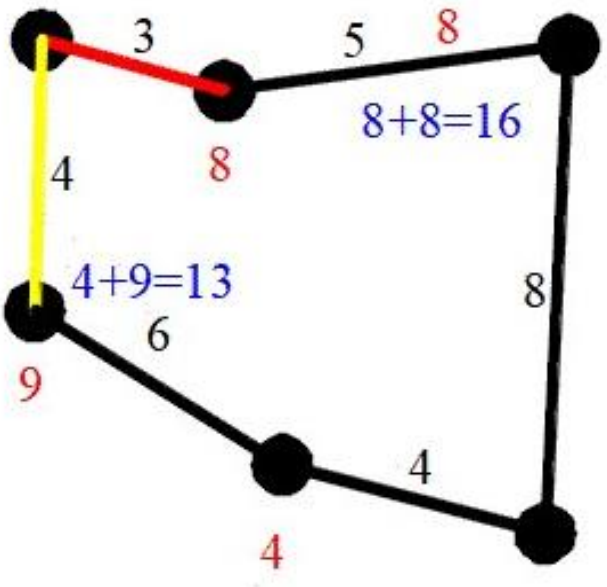


Şekil 4.4. Algoritmanın her düğümü güncelemesi

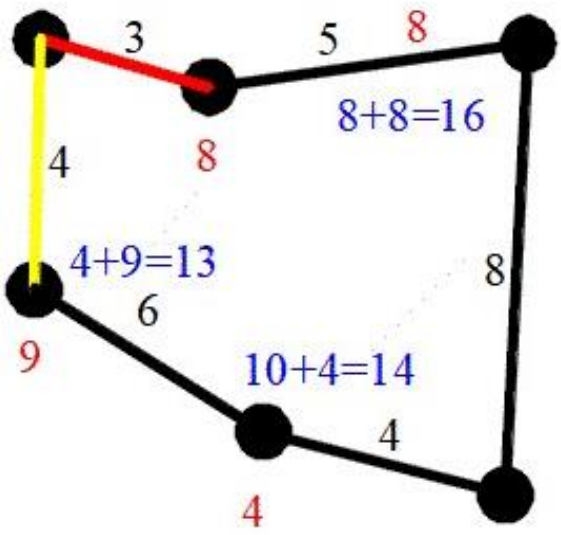


Şekil 4.5. Algoritma en düşük düğümü alması

- Bu düğüme göre, tüm komşu düğümlerin değerleri güncellenir (düğüme erişmenin bir maliyeti vardır ve bu değer f (n) işlevine dahil edilir (Bknz. Şekil 4.6 ve Şekil 4.7).

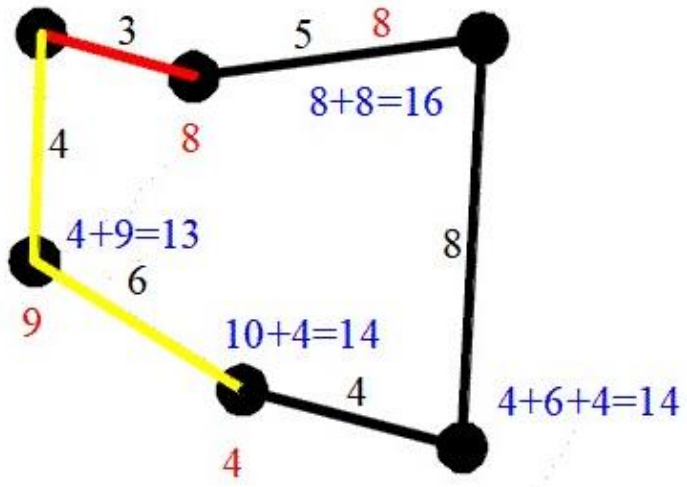


Şekil 4.6. Algoritma iki düğüm arasında en düşük olanı seçmesi



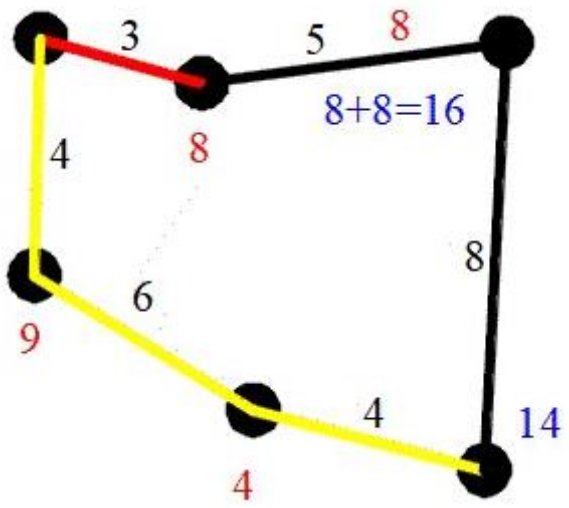
Şekil 4.7. Başlanğıç noktası ve hedef arasındaki en kısa yolu bulması

- Algoritma, yukarıdaki adımları, hedef hedef düğüme ulaşana kadar ya da hiçbir kuyruk olmayana kadar tekrarlar.



Şekil 4.8. İki yol arasında karar vermek

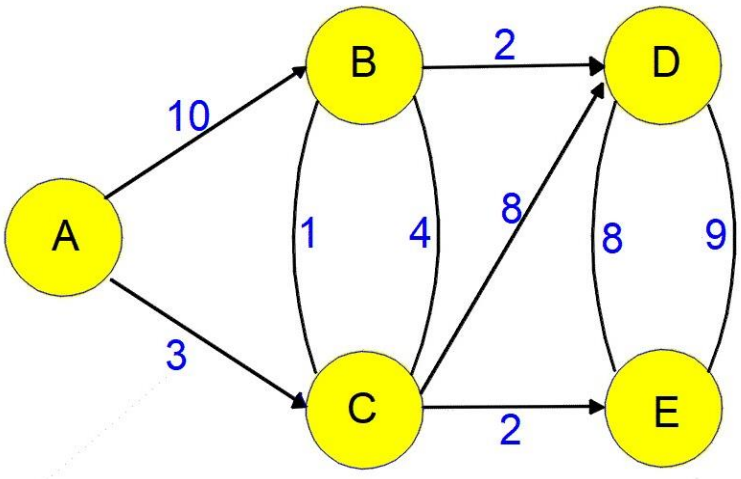
Şekil 4.8'e bakıldığında, son durumda hedefe ulaşılmış ve aşağıdaki yol kullanılmıştır. Sonuç olarak ulaşım için harcanan mesafe toplandığında 4+6+4 =14 mesafesi diğer alternatif olan 3+5+8 =16 değerine göre daha kısadır. En kısa yol Şekil 4.9' da görülmektedir.



Şekil 4.9. En kısa yolun belirlenmesi

4.2. Dijkstra Algoritması

Ağırlıklı bir graf üzerinde yani kenarları (edge) belli bir metrik değere göre herhangi iki düğüm arası en kısa mesafeyi bulmamızı sağlayan bir algoritmadır. Bu algoritma Hollandalı matematikçi ve bilgisayar bilimcisi Edsger Wybe Dijkstra tarafından bulunmuştur. Pek çok alanda, özellikle yönlendirmede algoritma kullanılmaktadır. Dijkstra algoritması aç gözlü yani Greedy bir algoritmadır. Dijkstra algoritması bir düğümden diğerine geçerken mevcut durumun "en iyi" çözümünü seçmektedir (Mohammadreza Radmanesh ve ark., 2018). Şekil 4.10'da Dijkstra algoritması için örnek düğümler görülmektedir.



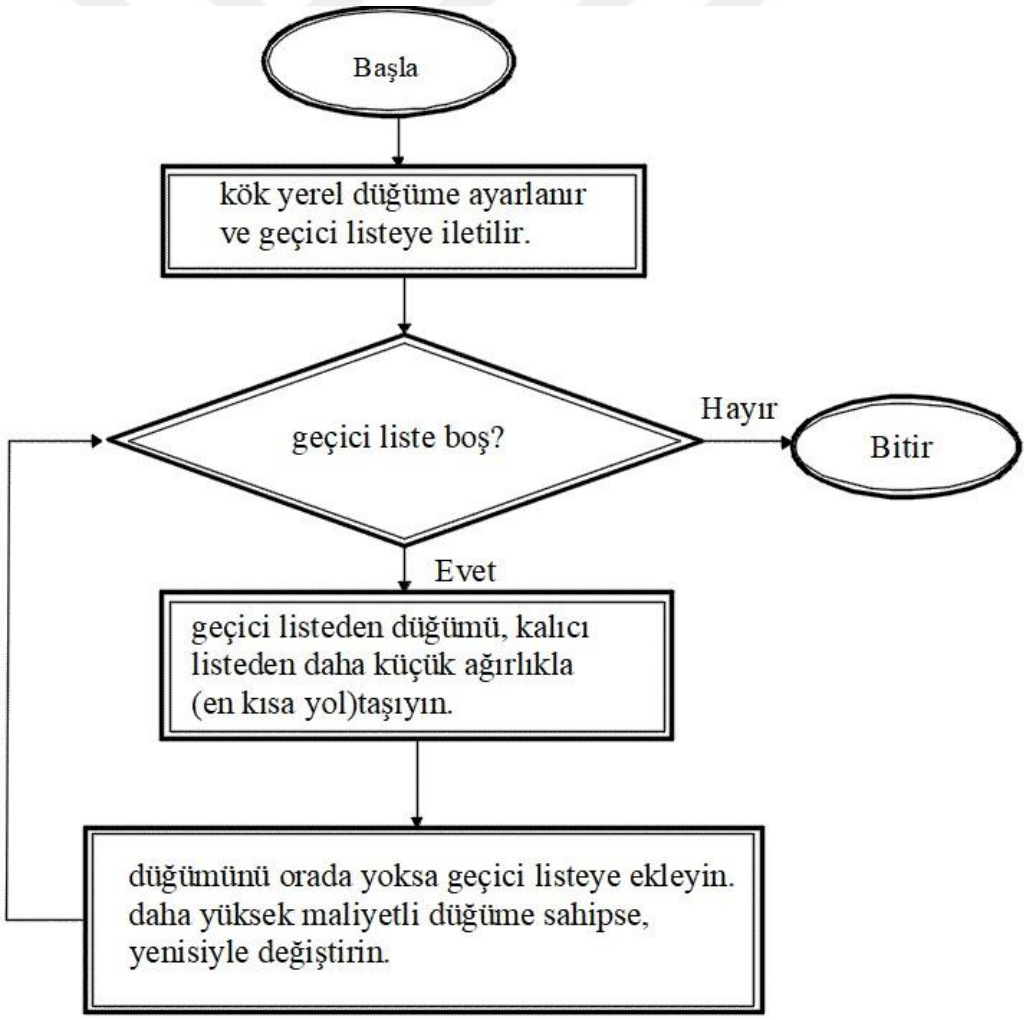
Şekil 4.10. Dijkstra algoritması için örnek düğümler

Bu algoritma 1956'da Edsger W. Dijkstra tarafından geliştirilmiştir. Bu algoritmanın asıl amacı, en kısa yol planlaması gibi uygulama için açgözlü bir yöntem geliştirmektir. Ağır modifikasyon, algoritmanın karmaşıklığını azaltmıştır. Bu algoritmanın temeli gevşeme ilkesidir. Daha doğru değerler, en sonunda en uygun çözüme ulaşana kadar doğru mesafe yaklaşımını kademeli olarak değiştirmektedir (Mohammadreza Radmanesh ve ark., 2018). Şekil 4.11'da Dijkstra algoritma için akiş şeması görülmektedir.

Bu algoritmada kullanılan hesap fonksiyonu:

$$f(n) = g(n) \quad (4.2)$$

Denklemde görüldüğü gibi, f(n): hesaplama yapan sezgisel fonksiyon, g(n): Geçerli düğüme başlangıç düğümünden girmenin değeri.



Şekil 4.11. Dijkstra için akiş şeması

Dijkstra algoritmasını kullanmamız için:

- Ağırlıklı ve yönlü graflar için geliştirilmiştir,
- Kenarların ağırlık değeri sıfır ya da sıfırdan büyük bir değer olmalıdır, eğer kenar değerleri sıfırdan küçükse Bellamn-Ford algoritması kullanılabilir,
- Negatif ağırlıklar için çalışmaz.

Algoritmanın Çalışması

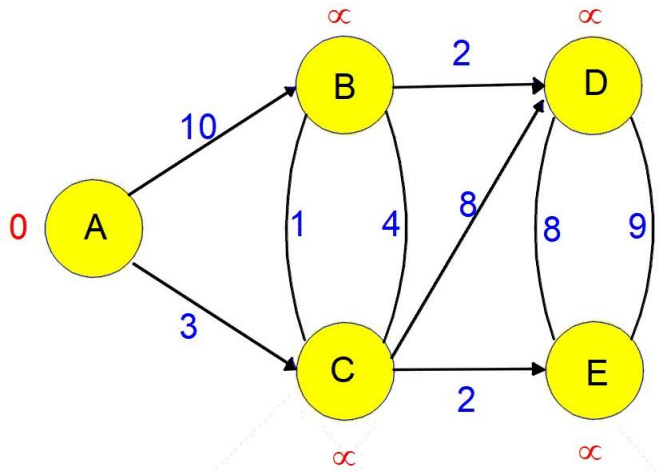
Dijkstra'nın Algoritması, nesnenin başlangıç noktasından başlayarak grafikteki köşeleri ziyaret ederek çalışmaktadır. Daha sonra, henüz incelenmemiş en yakın köşeyi tekrar tekrar inceleyerek, incelenecek köşeler kümesine köşelerini ekler. Başlangıç noktasından hedefe ulaşana kadar dışa doğru genişlemektedir. Dijkstra'nın Algoritmasının başlangıç noktasından hedefe giden en kısa yolu bulmasını garanti etmektedir.

Dijkstra algoritması bir düğümden diğer tüm düğümlere kadar olan en kısa yolları hesaplamaktadır. Kenarların ağırlık değeri, ağırlıklı ve yön grafikleri için oluşturulan sıfır veya sıfırdan büyük olmalıdır. Kenarların değerleri sıfırdan düşükse, Bellman-Ford daha genel algoritmalarda kullanılabilir. Çünkü negatif değerdeki kenar sürekli olarak mevcut durumdan daha iyi bir sonuç üretir ve algoritma hiçbir zaman için kararlı hale gelememektedir. Başlangıçta seçilen düğümden diğer tüm düğümlere giden en kısa yol Dijkstra algoritmasında hesaplanır. Dijkstra algoritmasında hiçbir hedef düğüm yoktur.

Bir hedef düğüm sorun olarak tanımlanırsa, Dijkstra algoritması tüm düğümlere en kısa yolu bulur ve bunlardan birini seçecektir. Eğer Dijkstra algoritmasını kullanılıyorsa, bir amaç olsun veya olmasın, tüm düğümlere en kısa mesafeyi belirleyemektedir. Şayet bir hedef varsa, bu hedefe giden en kısa yol da zaten hesaplanmış olunmaktadır. Dijkstra'da, her hareket bir düğümün komşularında kabul edilir ve daha kısa bir yol tanımlandığında mesafeler değiştirilmektedir.
Algoritmanın çalışması aşağıda sıraladığı gibidir:

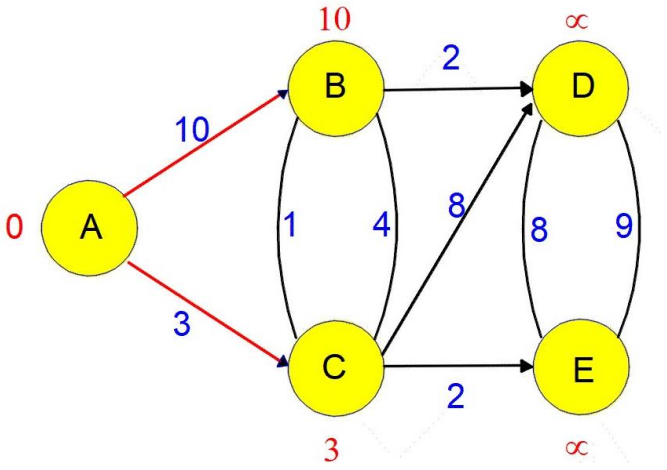
- Şekil 4.10'da görüldüğü gibi, İlk önce A düğümünü başlangıc noktası olarak Dijkstra algoritma tarafından seçilmektedir.
- İlk adım A düğümüne başlangıc noktası belirlendikten sonra sıfır uzaklık

değerini verilmektedir diğer tüm düğümlere sonsuz uzaklık değeri verilmektedir. Aşağıdaki şekil 4.12'de A düğüm başlangıc noktası görülmektedir.



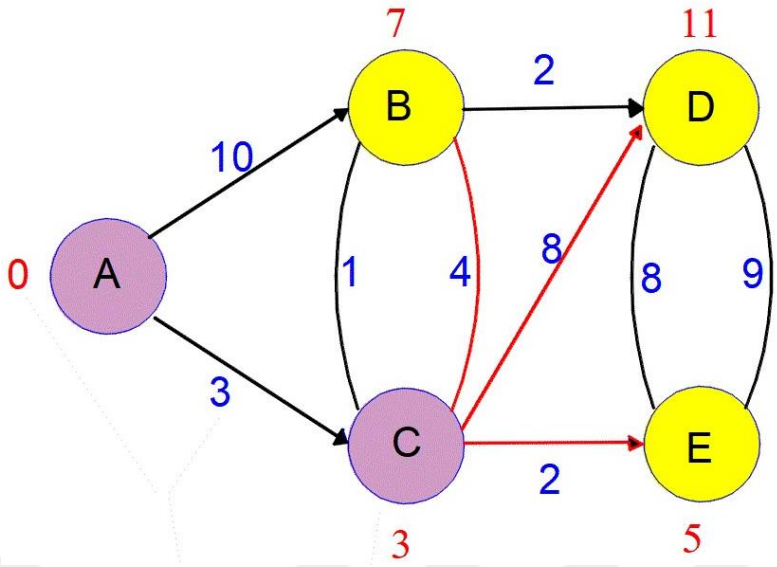
Şekil 4.12. A düğüm başlangıc noktası

- İkinci adım A düğümünden diğer yollara olan uzaklıkları hesaplanmaktadır . Sonra başlangıç düğümüne diğer tüm düğümleri geçerek düğümün erişimini günceller. Aşağıdaki şekil 4.13'de A düğümden diğer yollara olan mesafe görülmektedir.



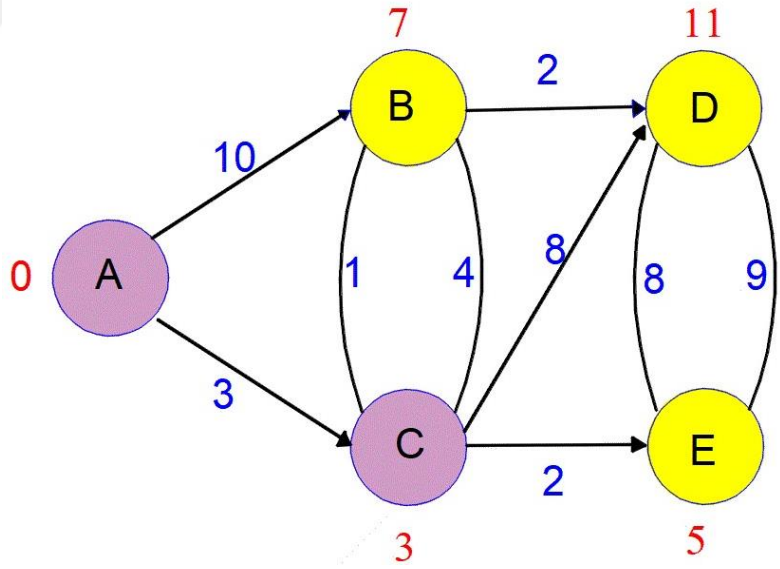
Şekil 4.13. A düğümden diğer yollara olan mesafe

- Üçuncu adım A düğümünden B ve C komşu düğüm olan en küçük mesafeyi seçmektedir. Bu güncellemeden sonra, güncellenen düğümleri günceller ve tüm düğümler güncellenene ve grafta yeni güncellemeler bulunmayana kadar devam etmektedir (Bknz. Şekil 4.14).



Şekil 4.14. A düğümden diğer yollara olan en düşük mesafe

- Dördüncu adım sadece E düğümünü ziyaret etmektedir. Şekil 4.15'de görülmektedir.



Şekil 4.15. E düğümünü ziyaret

Tüm adımların sonucunda özetilmiş hali çizelge 4.1'de görüldüğü gibidir.

Çizelge 4.1. Ziyaret edilen düğümler

A	B	C	D	E
0	∞	∞	∞	∞
	10	3	∞	∞
	7		11	5
	7		11	
			9	

Bu algoritmanın amacı:

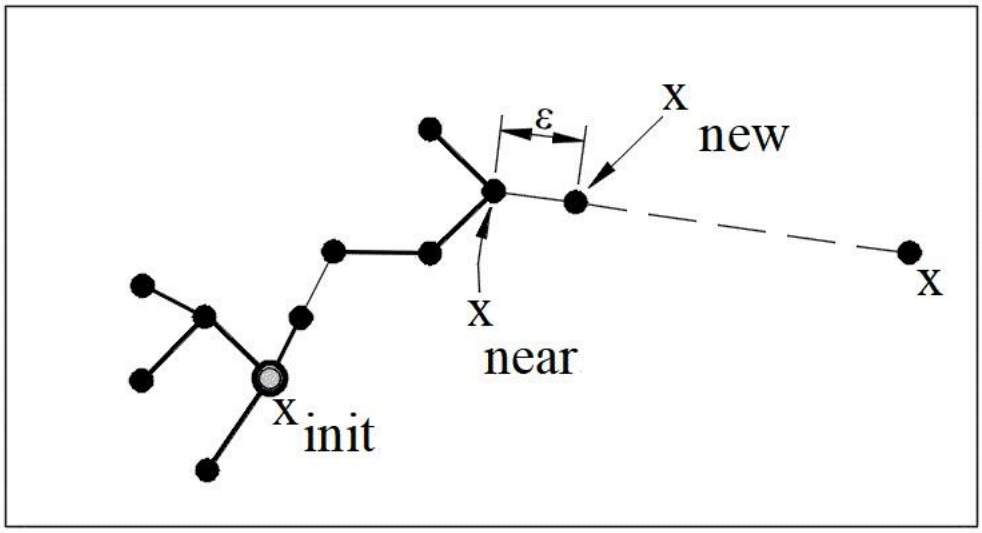
Bu algoritmanın amacı bulunduğu noktadan hedefe giden yollar arasındaki en az enerji harcanan, en kısa mesafeyi hesaplamaktır. Bir düğümden başlayarak, düğümün grafikteki tüm düğümlere en kısa mesafeden ulaşmasını garanti etmektedir. Ayrıca bundan daha kısa bir yol bulunmayacağını iddia etmektedir. Eşit mesafe olabilir. Googla mapte işlemlerde kullanılmaktadır. Bir grafikteki bir düğümden aynı grafik veri yapısı içindeki diğer tüm düğümlere giden en kısa yolu belirlemek için kullanılabilmektedir.

4.3. Hızlı Rastgele Ağaç Keşfi algoritması (RRT_ Rapidly-exploring Random Tree) Algoritması

Alan dolgulu bir ağacı rastgele oluşturarak, dışbükey olmayan, yüksek boyutlu alanları etkili bir şekilde aramak için tasarlanmış bir algoritmadır (Bknz. Şekil 4.16 RRT için örnek düğümler). Ağaç, arama alanından rasgele çekilmiş örneklerden artan şekilde inşa edilmiştir ve doğal olarak sorunun araştırılmamış büyük alanlarına doğru büyümek için önyargılıdır. RRT'ler Steven M. LaValle ve James J. Kuffner Jr. tarafından geliştirilmiştir. Engellerle ve diferansiyel kısıtlamalarla (holonomik olmayan ve kinodinamik) problemleri kolayca idare ederler ve özerk robotik hareket planlamasında yaygın olarak kullanılmaktadır(Iram Noreen ve ark., 2016).

Bu yöntemde amaç ilk andan itibaren rastgele noktalar seçerek dallanmalar yapmak ve bu dallanmalara göre sonuçta hedefe gidilebilecek en uygun rotayı oluşturmaktır. Bu amaçla birçok alanda rastgele noktalar seçer fakat yeni seçilecek noktayı seçilmiş olanların dışında ve hedefe yakınsayarak seçmesi işlem kolaylığı sağlamaktadır. RRT metodu klasik tek yönlü gerçekleştirimin dışında çift yönlü, çoklu ağaç yapılı, diferansiyel kısıtlı olarak karşımıza çıkmaktadırlar. Bu algoritmanın amacı

Dinamik bilinmeyen ortamda hızlı engel tespiti ve gereksiz manevra noktalarının azaltılmasıdır (Iram Noreen ve ark., 2016).



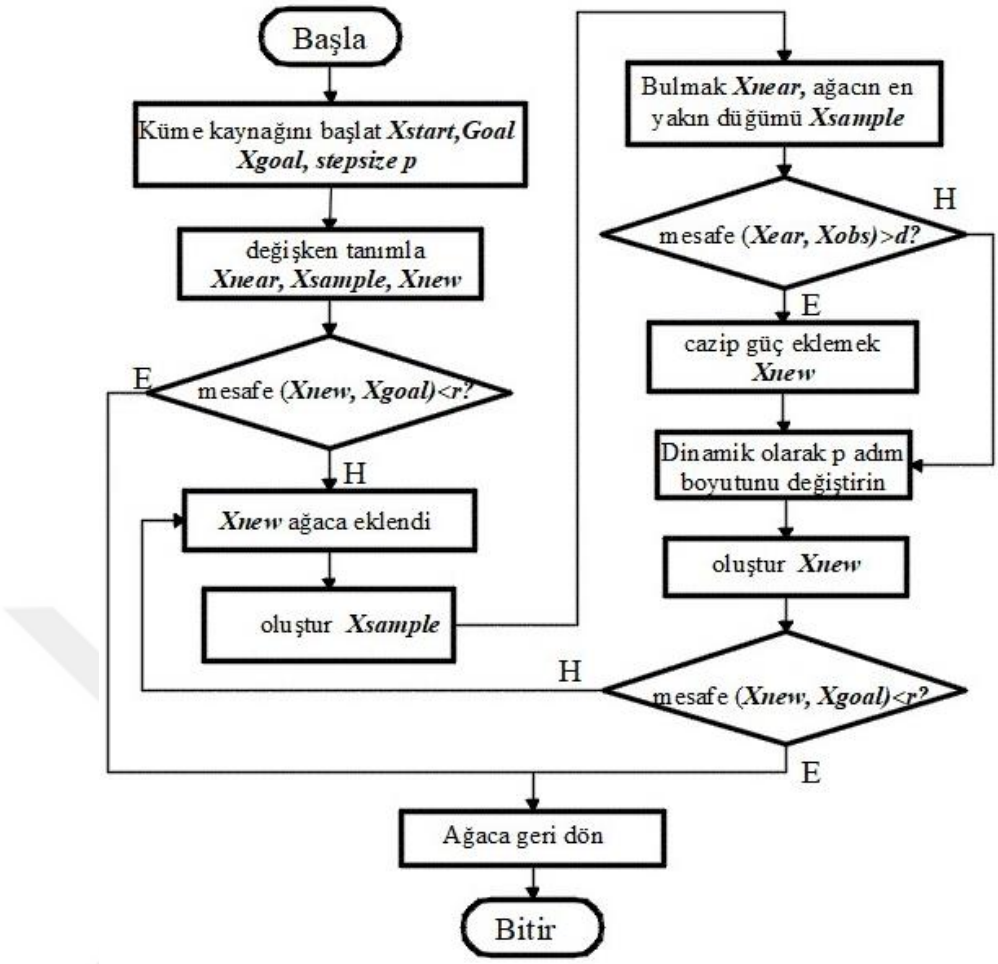
Şekil 4.16. RRT algoritması için örnek düğümler

Bir RRT, arama alanından rastgele örnekler kullanarak başlangıç konfigürasyonunda köklenmiş bir ağaç yetiştirmektedir. Her örnek çizilirken, ağaçla en yakın durum arasında bir bağlantı denenir. Bağlantı mümkün ise (tamamen boş alandan geçiyor ve herhangi bir kısıtlamaya uyuyorsa), bu yeni durumun ağaca eklenmesiyle sonuçlanır. Arama alanının düzgün bir şekilde örneklenmesi ile mevcut bir durumu genişletme olasılığı, Voronoi bölgesinin büyüklüğü ile orantılıdır. En büyük Voronoi bölgeleri aramanın sınırındaki devletlere ait olduğundan, bu, ağacın tercihen büyük, aranamayan alanlara doğru genişlediği anlamına gelmektedir (Iram Noreen ve ark., 2016). Şekil 4.17 RRT algoritma için akiş şeması görülmektedir.

Bu algoritmada kullanılan hesap fonksiyonu:

$$\left|AB\right| = \sqrt{((X_2-X_1)+(Y_2-Y_1))} \quad (4.3)$$

Burda, $\left|AB\right|$ iki düğüm arasında hesap yapan fonksiyon, X_2-X_1 yatay uzunluk Y_2-Y_1 ise dikey uzunluktur.



Şekil 4.17. RRT algoritması için akiş şeması

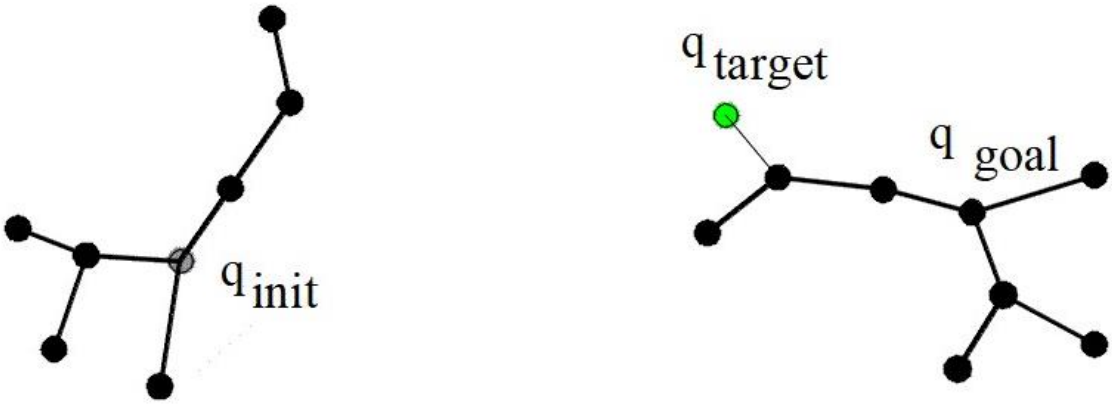
Eğer rastgele örnek, ağaçtaki en yakın durumundan bu sınırın izin verdiğinden daha fazla ise, çizgi boyunca rastgele numunenin yerine ağaçtan maksimum mesafedeki yeni bir durum kullanılır. Rastgele örnekler daha sonra büyüme faktörü oranını belirlerken ağaç büyümesinin yönünü kontrol ediyor olarak görülebilmektedir. Sonuç olarak artımlı büyümenin boyutunu sınırlandırırken, RRT'nin boşluk doldurma önyargısını korumaktadır (Iram Noreen ve ark., 2016).

RRT büyümesi, örnekleme durumlarının belirli bir bölgeden olasılığını artırarak önyargılı olabilir. RRT'lerin çoğu pratik uygulaması, planlama problemi hedeflerine yönelik araştırmaya rehberlik etmek için bunu kullanmaktadır. Bu, durumun örnekleme prosedürüne hedeflemenin küçük bir örnekleme olasılığı getirilerek gerçekleştirilir. Bu olasılık ne kadar yüksek olursa, ağaç açgözlülükle hedefe doğru büyümektedir (Iram Noreen ve ark., 2016).

RRT Algoritmasının çalışması

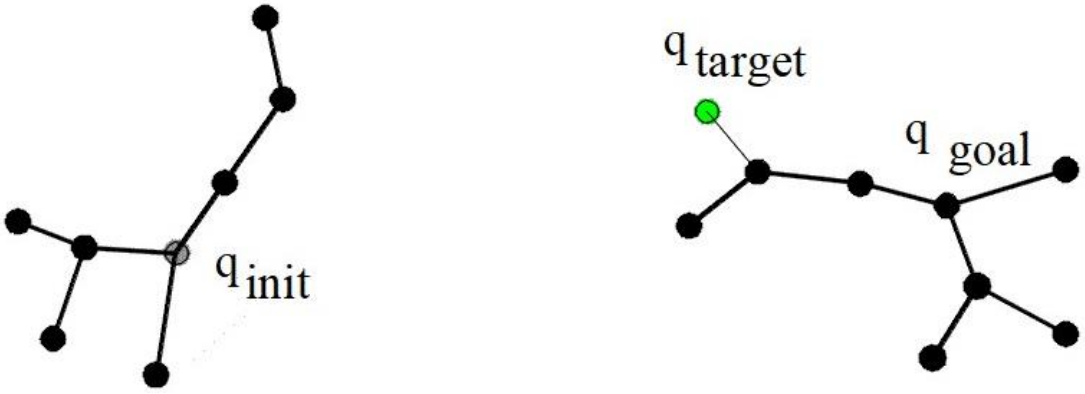
RRT algoritmasının adımları sıralandığı gibidir.

- Bir ağaç rastgele hedef kullanılarak büyütülür.



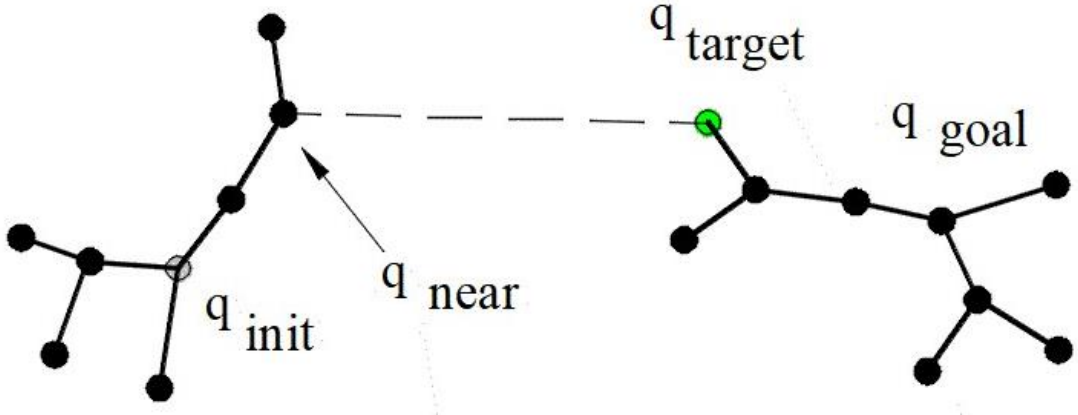
Şekil 4.18. RRT algoritması için başlangıc ve hedef noktası

- Yeni düğüm diğer ağaç için hedef haline gelir.



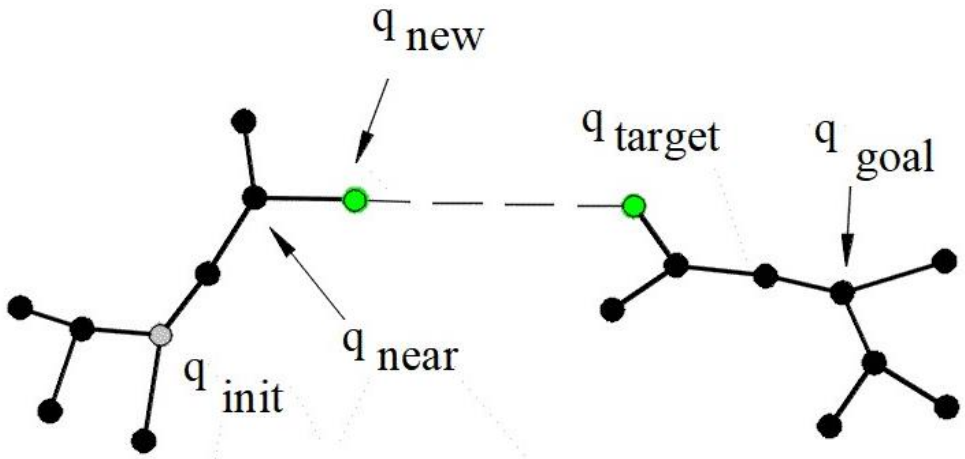
Şekil 4.19. RRT algoritması için ikinci adım

- Hedefe en yakın olan düğüm hesaplanır



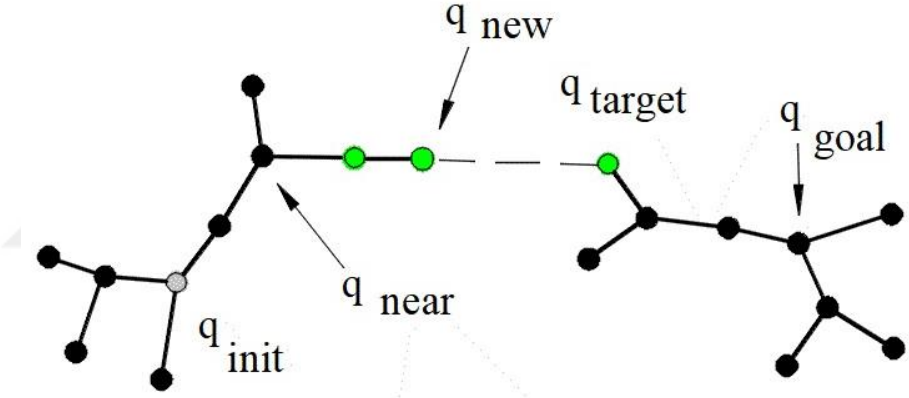
Şekil 4.20. RRT algoritması için iki düğüm arasında hesap yapması

- Çarpışmayan yeni yol eklemeye çalışılmaktadır.



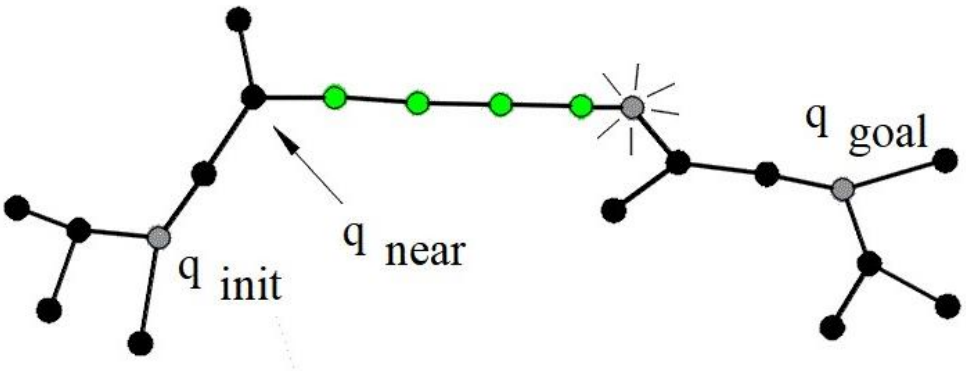
Şekil 4.21. RRT algoritması çarpışma olmadan yeni düğüm eklemektedir

- Başarılı olursa, dalı uzatmaya devam edilmektedir



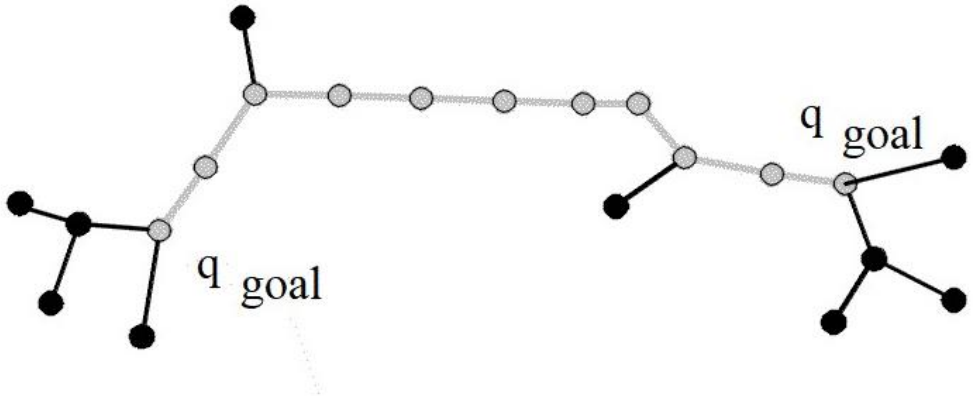
Şekil 4.22. RRT algoritması için hedef noktasına ulaşıncaya kadar devam etmektedir

- Şube hedefe ulaşırsa bulunan yol



Şekil 4.23. RRT algoritması için hedefe giden yol

- Başlangıç ve hedef bağlayan dönüş yolu



Şekil 4.24. RRT algoritması için bulunan yol

RRT Avantajları

- Bir ağaç, başlangıç düğümünün yakınında "kalmak" yerine, tüm alanı hızla araştırır.
- RRT algoritmasının programlanması oldukça basittir; bu yüzden bu kalite bir yol bulmak için hızlı bir analiz yapılmasına yol açar.
- Temel algoritmaya birçok ekstra seçenek eklenebilir.
- RRT algoritması birçok planlama sistemine dahil edilebilmektedir.

Temel bir RRT algoritması için dezavantajları da vardır:

- Algoritma deterministik değildir. Bulunan yol, bir sonraki seferde aynı başlangıç düğümünde bir yol bulunması gerektiğinde aynı değildir. Bu özellik nedeniyle, bulunan yollardan bazılarının bir insanın seçtiği yollar olmaması mümkündür.
- Devlet alanında çok fazla engel varsa, engelde bulunmayan ve ağaca bağlanabilen bir nokta bulma şansı çok azdır.
- Dar bir geçit de geçmek zordur.
- Bulunan yol keskindir ve kolay sürülemez.

5. UYGULAMA VE SONUÇLARI

Bu tezde, Mobil otonom robotlar için geliştirilen 2B yol planlama algoritmalarına dayalı 3B yol planlama yöntemlerinin geliştirilmesi ve uygulamada hızlı ilerleme kaydedilmesi amaçlanmaktadır. 2B ortamda yarattığımız engeller arasında insansız hava araçları, kaynak ile hedef arasındaki engelleri önleyerek ve önerilen yöntemleri kullanarak en kısa yolun elde edilmesini sağlamaktadır. Önerilen bu yöntemlerden Dijkstra ve A* algoritmaları özerk mobil robotlarda en yaygın kullanılanlardır. Dijkstra algoritması iki düğüm arasındaki en kısa mesafeyi bulurken, A* algoritması da en kısa mesafeyi bulmak için sezgisel yaklaşımı kullanmaktadır. Oluşturduğumuz harita üç unsurdan oluşmaktadır: engel, başlangıç noktası ve varış noktası. Bu haritayı oluşturmak için Matlab kullanılmıştır.

Algoritmalarda, 3B yolu bulmak için birkaç adım kullanılmaktadır. İlk olarak, yerel bir başlangıç noktasından engellerin yükseklikleri hakkında bir hedef noktaya bilgi verilmektedir. Daha sonra bu engel yüksekliklerine göre ve en kısa yol planlamasına göre İHA'nın uçacağı en uygun yükseklik belirlenir. Engel yüksekliklerine göre 2B yol planlama algoritmalarına dayalı birkaç alternatif yol planlaması yapılır ve yapılan yol planlamalarından en uygun olanı çeşitli yöntemler ile belirlenmiş olmaktadır.

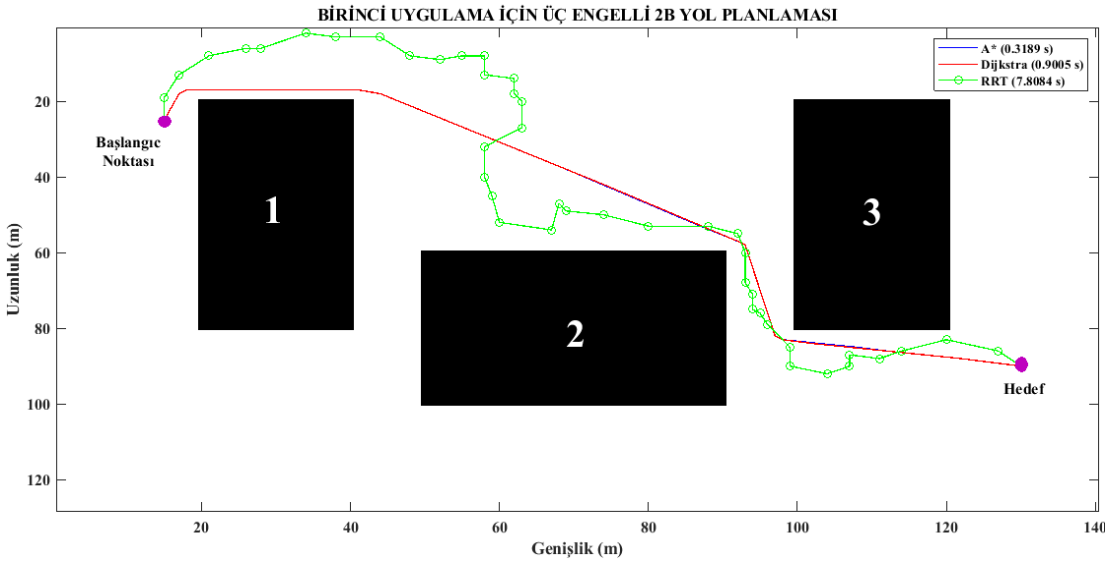
Mavi çizgi A* algoritması ile yol planlama sonucudur, kırmızı çizgi Dijkstra algoritması ile yol planlama sonucudur ve yeşil çizgi RRT algoritması ile yol planlama sonucudur.

5.1. Simülasyon Ortamı

5.1.1. Birinci uygulama

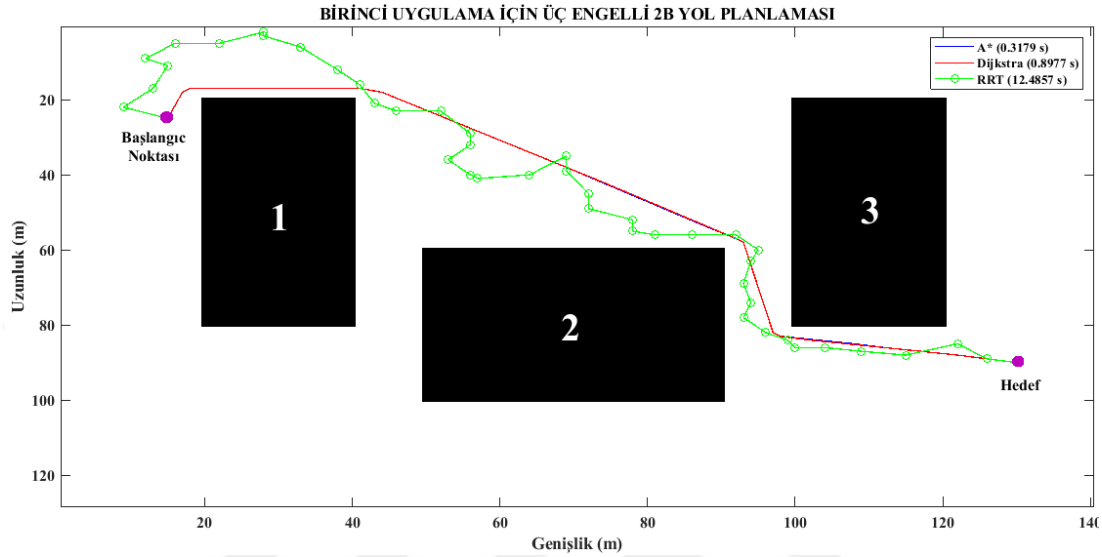
- ❖ Engel yüksekliğini ve drone yüksekliğini hesaba katmayan ortam 2B simülasyon ortamı olarak isimlendirilmektedir.

Şekil 5.1'de gösterildiği gibi, üç farklı yol planlama algoritması kullanılmaktadır. Algoritmalar sırasıyla A*, Dijkstra ve RRT'dir. RRT algoritmasında, algoritmanın 10 kez çalıştırılması sonucunda optimum yol seçilmektedir. Şekillerde gösterildiği gibi, her biri farklı bir uzunluk ve genişliğe sahip üç farklı engel vardır. Birinci ve üçüncü engeller 20 metre genişliğinde ve 60 metre uzunluğunda ve ikinci engel 40 metre genişliğinde ve 40 metre uzunluğundadır. Başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3189 saniyede, Dijkstra algoritmasına göre 48 metre yolu 1.9005 saniyede, RRT algoritmasına göre 45 metre yolu 7.8084 saniyede katettiği görülmektedir.



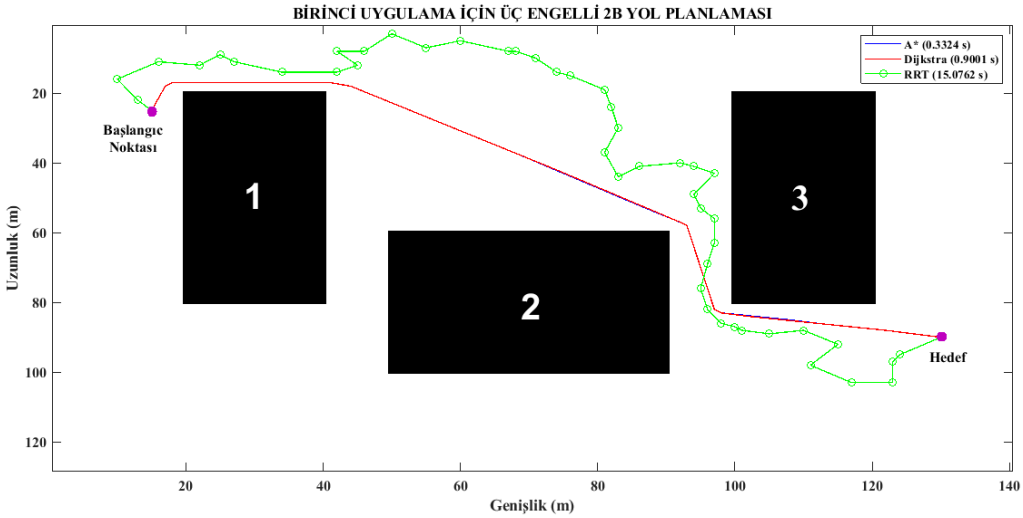
Şekil 5.1. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.2'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3179 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.8977 saniyede, RRT algoritmasına göre 44 metre yolu 12.4857 saniyede katettiği görülmektedir.



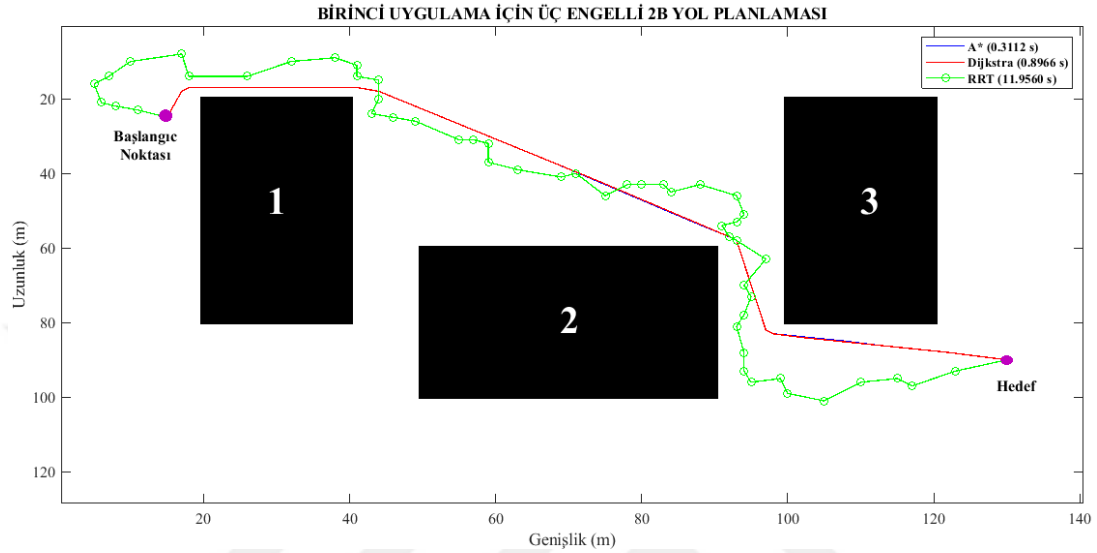
Şekil 5.2. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.3'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3324 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.9001 saniyede, RRT algoritmasına göre 48 metre yolu 15.0762 saniyede katettiği görülmektedir.



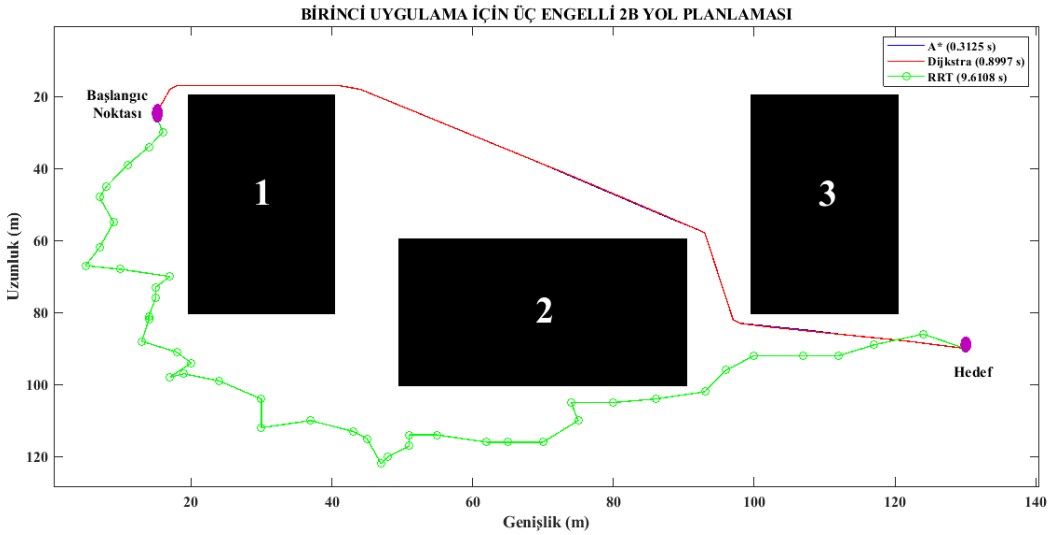
Şekil 5.3. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.4'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3112 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.8966 saniyede, RRT algoritmasına göre 54 metre yolu 11.9560 saniyede katettiği görülmektedir.



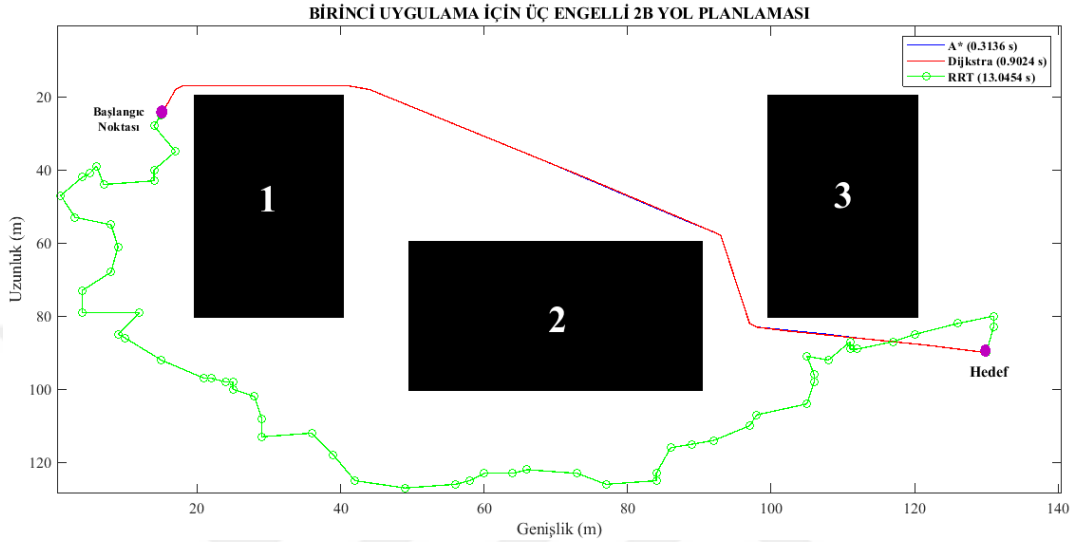
Şekil 5.4. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.5'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3125 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.8997 saniyede, RRT algoritmasına göre 46 metre yolu 9.6108 saniyede katettiği görülmektedir.



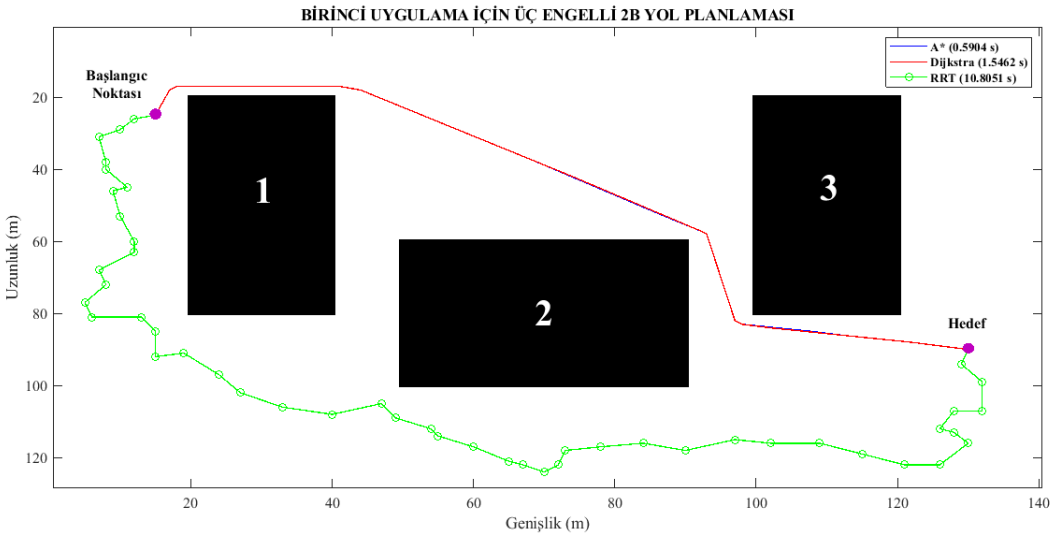
Şekil 5.5. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.6'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3136 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.9024 saniyede, RRT algoritmasına göre 44 metre yolu 13.0454 saniyede katettiği görülmektedir.



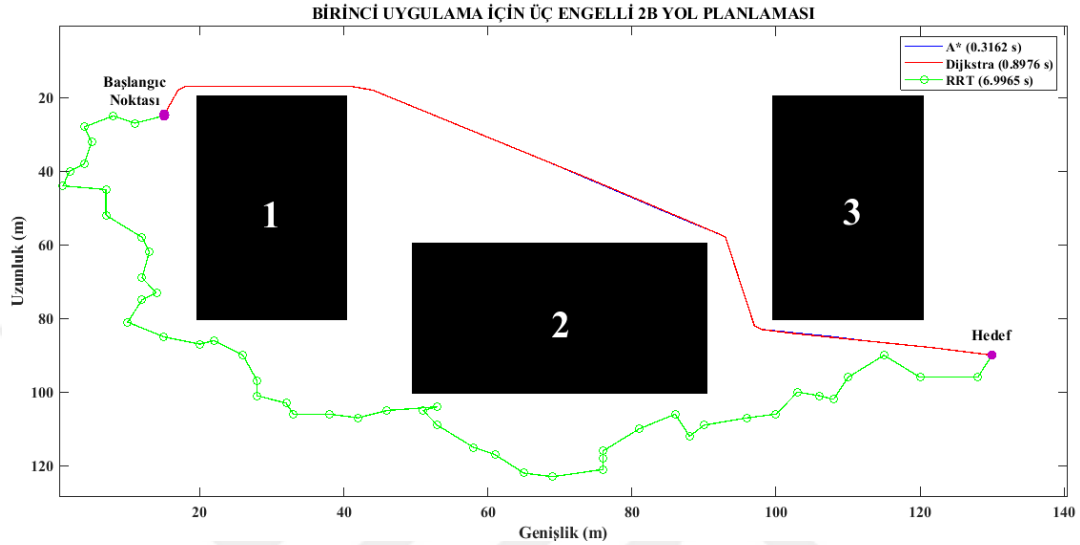
Şekil 5.6. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.7'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.5904 saniyede, Dijkstra algoritmasına göre 48 metre yolu 1.5462 saniyede, RRT algoritmasına göre 50 metre yolu 10.8051 saniyede katettiği görülmektedir.



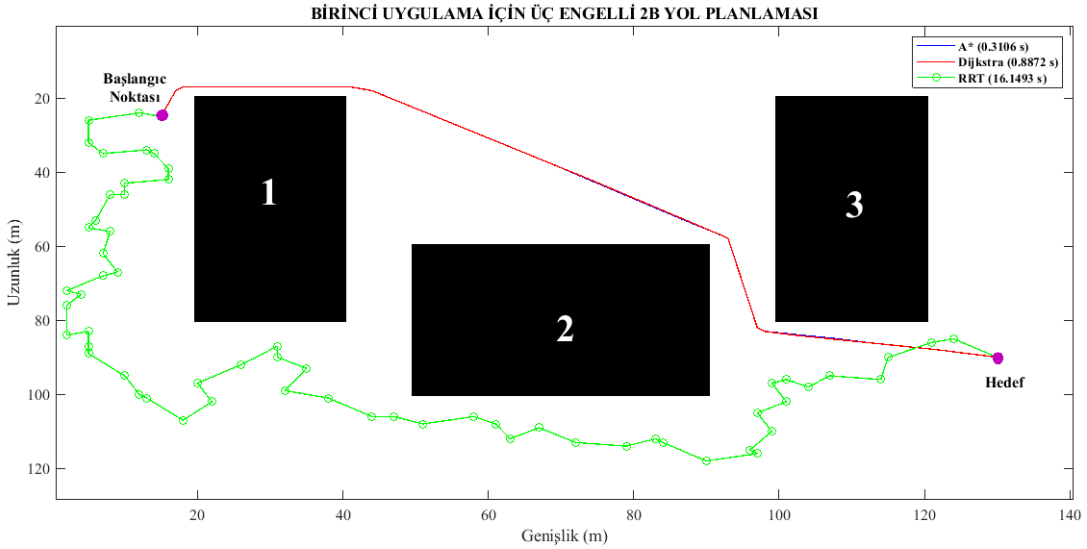
Şekil 5.7. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.8'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3162 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.8976 saniyede, RRT algoritmasına göre 51 metre yolu 6.9965 saniyede katettiği görülmektedir.



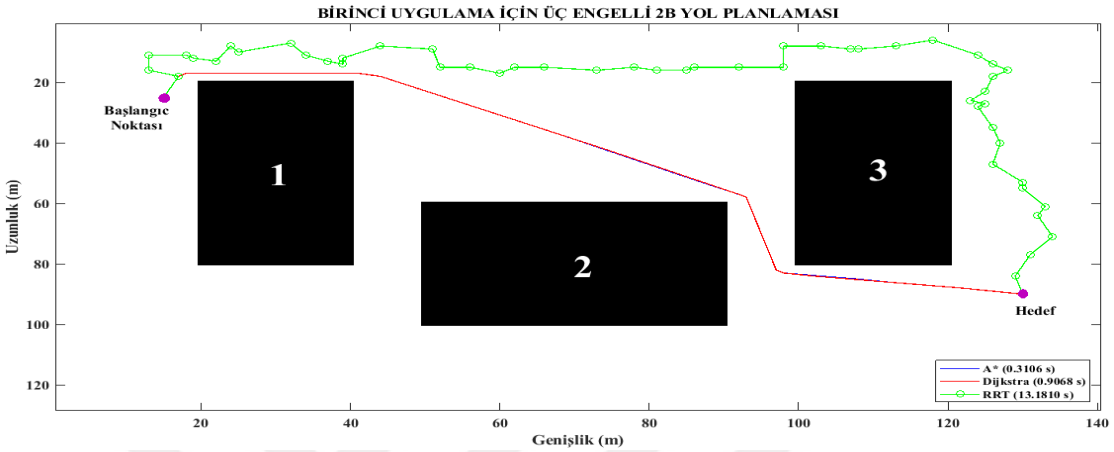
Şekil 5.8. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.9'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3106 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.8872 saniyede, RRT algoritmasına göre 63 metre yolu 16.1493 saniyede katettiği görülmektedir.



Şekil 5.9. Birinci uygulama için üç engelli 2B yol planlaması

Şekil 5.10'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 48 metre yolu 0.3106 saniyede, Dijkstra algoritmasına göre 48 metre yolu 0.9068 saniyede, RRT algoritmasına göre 53 metre yolu 13.1810 saniyede katettiği görülmektedir.



Şekil 5.10. Birinci uygulama için üç engelli 2B yol planlaması

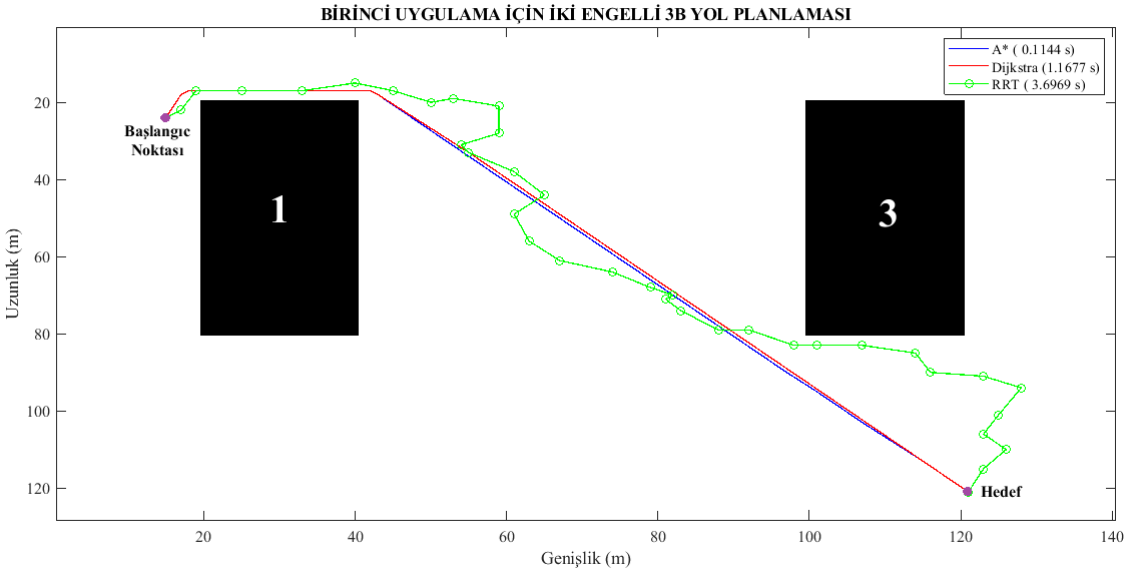
On farklı yol karşılaştırıldığında, Şekil 5.2'deki simülasyonun çizdiği yol uzunluk ve zamana göre daha uygun bir yol olduğu görülmüştür. On farklı yolun sonuçları Çizelge 5.l'de görülmektedir.

Çizelge 5.1. 2B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama Sayısı (1)	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)
5.1	0.3189	48	0.9005	48	7.8084	45
5.2	0.3179	48	0.8977	48	12.4857	44
5.3	0.3324	48	0.9001	48	15.0762	48
5.4	0.3112	48	0.8966	48	11.9560	54
5.5	0.3125	48	0.8997	48	9.6108	46
5.6	0.3136	48	0.9024	48	13.0454	60
5.7	0.5904	48	1.5462	48	10.8051	50
5.8	0.3162	48	0.8976	48	6.9965	51
5.9	0.3106	48	0.8872	48	16.1493	63
5.10	0.3106	48	0.9068	48	13.1810	53
Ortalama	0.3434	48	0.963	48	10.4628	51.4

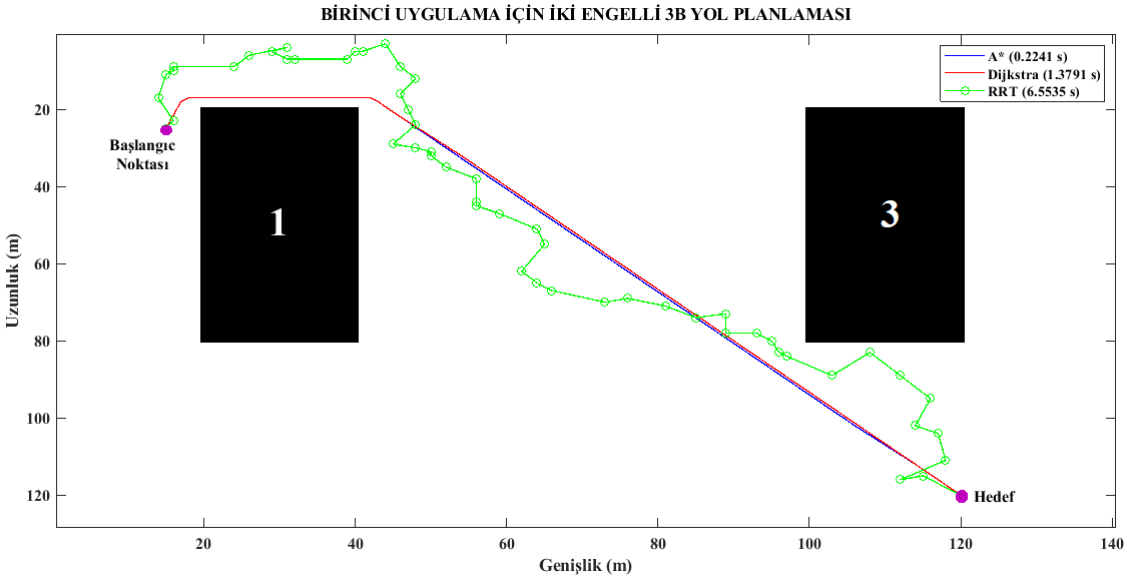
- Engel yüksekliğini ve drone yüksekliğni hesaba katan ortam 3B simülasyon ortamı olarak isimlendirilmektedir.

Şekil 5.11, uçağın yüksekliğini ve engellerin yüksekliğini belirledikten sonra, birinci ve üçüncü engellerin yüksekliği 60m ve ikinci engelin yüksekliği 40m'dır. Drone'un yüksekliği ikinci engelin yüksekliğinden daha yüksektir ve hedef noktaya daha kısa sürede ulaşmaktadır. Bundan dolayı Drone'un yüksekliği 45m olduğunda önerilen algoritmalar çalıştırıldığında ikinci engeli görmezden gelmektedir. Başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1144 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1677 saniyede, RRT algoritmasına göre 37 metre yolu 3.6969 saniyede katettiği görülmektedir.



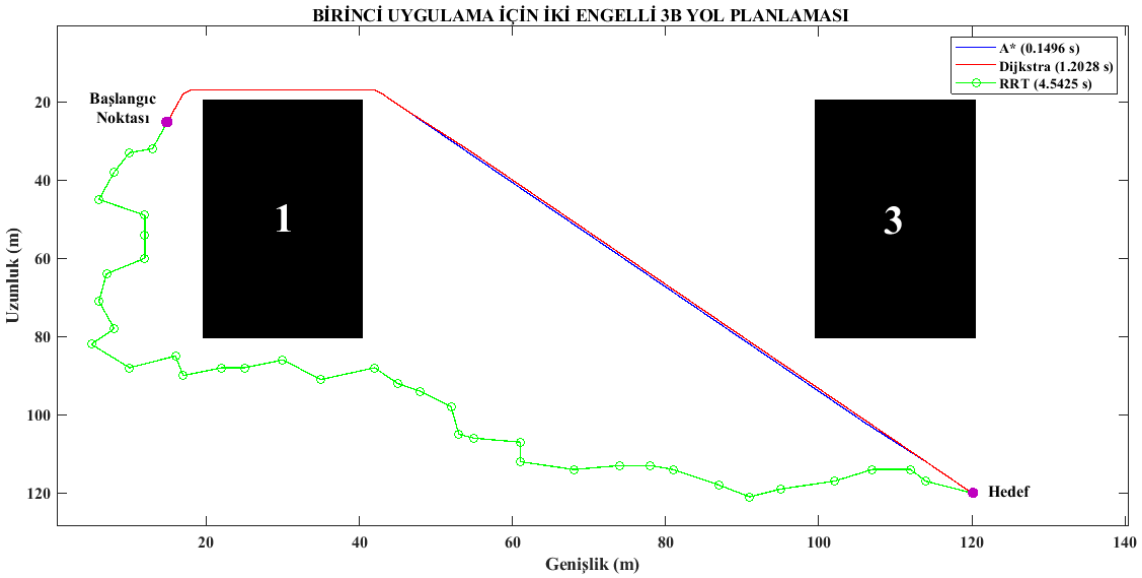
Şekil 5.11. Birinci uygulama için iki engelli 3B yol planlaması

Şekil 5.12'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.2241 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.3791 saniyede, RRT algoritmasına göre 55 metre yolu 6.5535 saniyede katettiği görülmektedir.



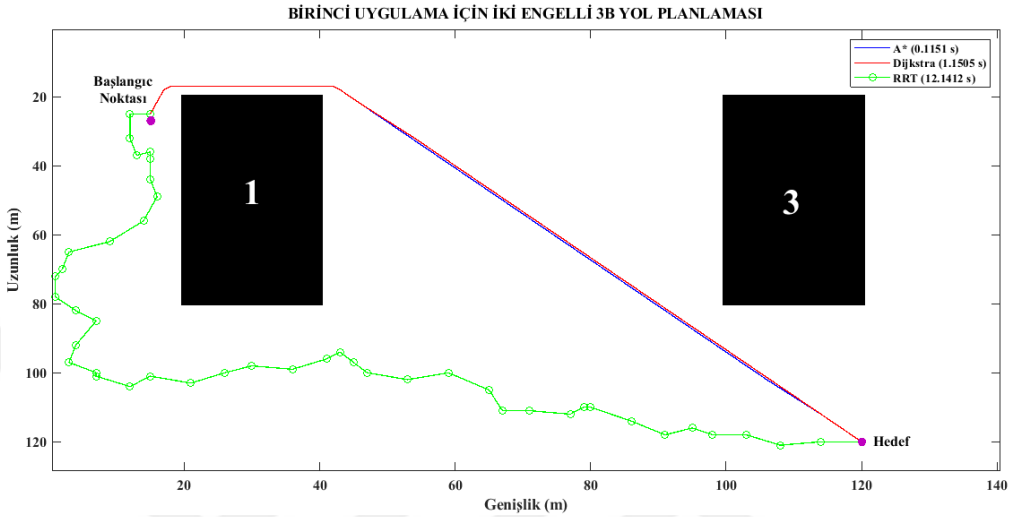
Şekil 5.12. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.13'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1496 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.2028 saniyede, RRT algoritmasına göre 39 metre yolu 4.5425 saniyede katettiği görülmektedir.



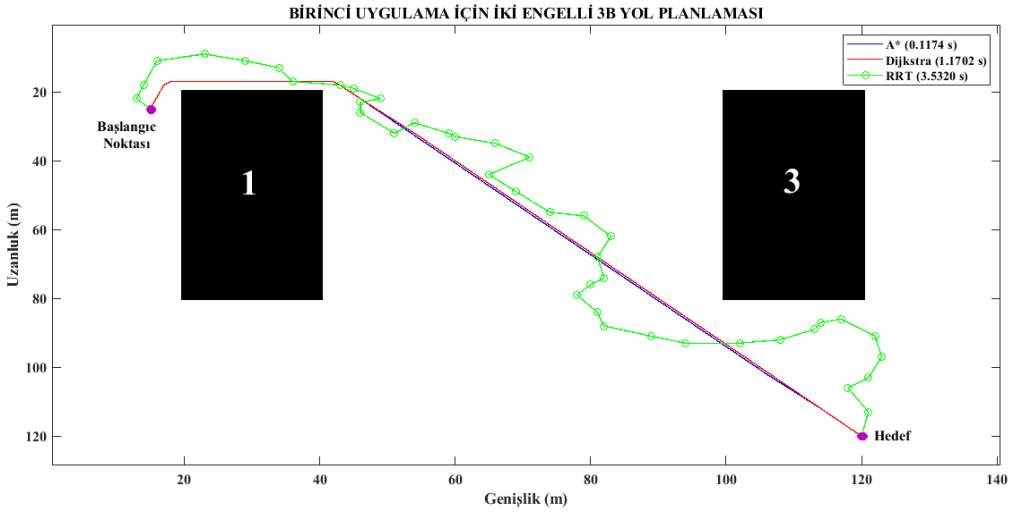
Şekil 5.13. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.14'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1151 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1505 saniyede, RRT algoritmasına göre 46 metre yolu 12.1412 saniyede katettiği görülmektedir.



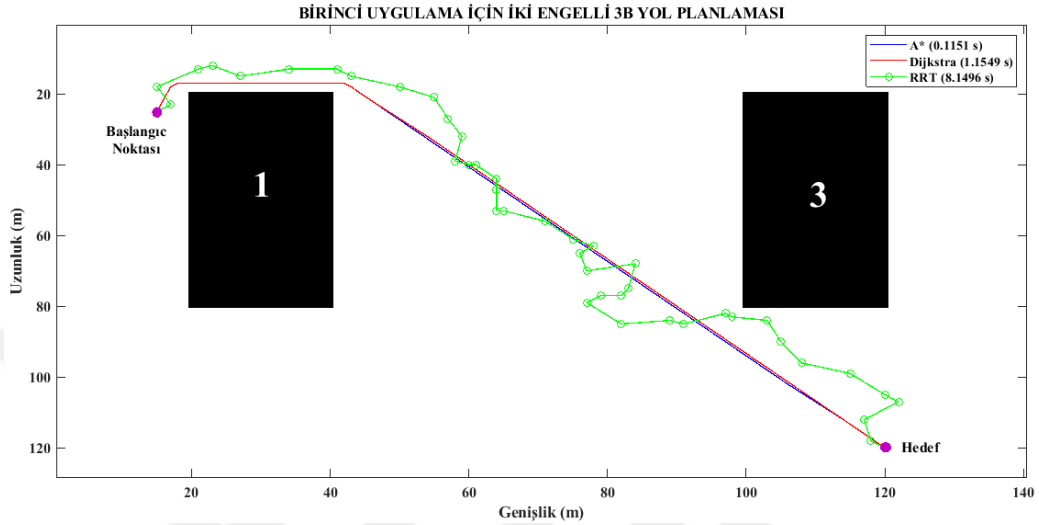
Şekil 5.14. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.15'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1174 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1702 saniyede, RRT algoritmasına göre 43 metre yolu 5.5320 saniyede katettiği görülmektedir.



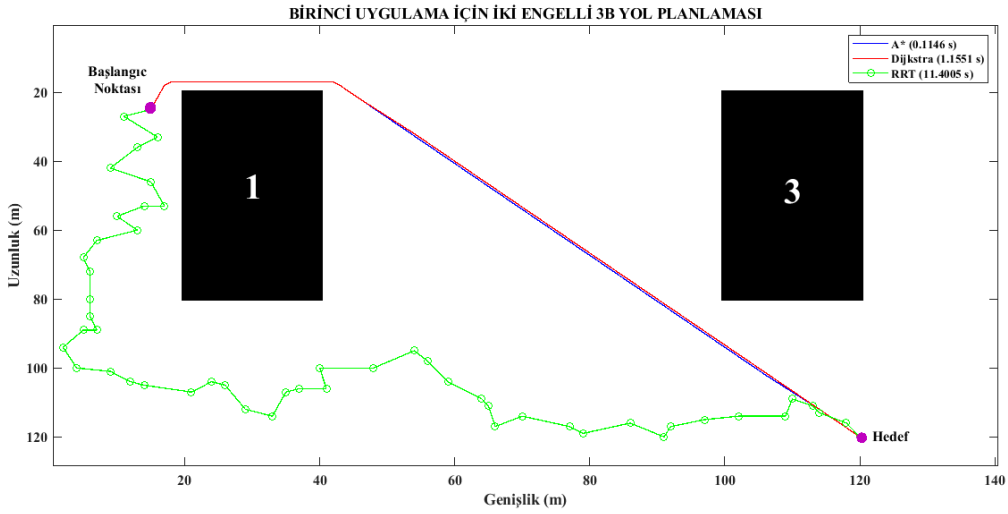
Şekil 5.15. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.16'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1151 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1549 saniyede, RRT algoritmasına göre 44 metre yolu 8.1496 saniyede katettiği görülmektedir.



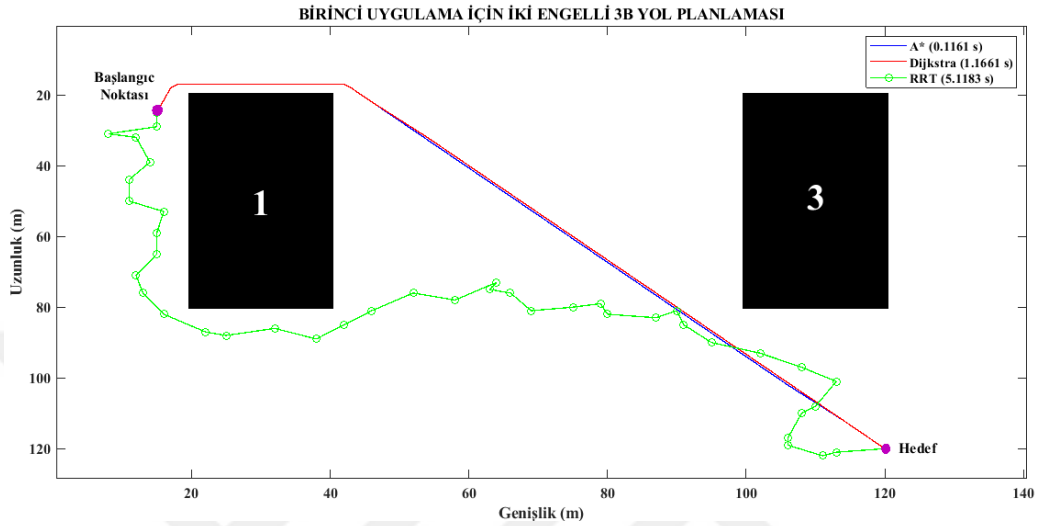
Şekil 5.16. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.17'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1146 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1551 saniyede, RRT algoritmasına göre 52 metre yolu 11.4005 saniyede katettiği görülmektedir.



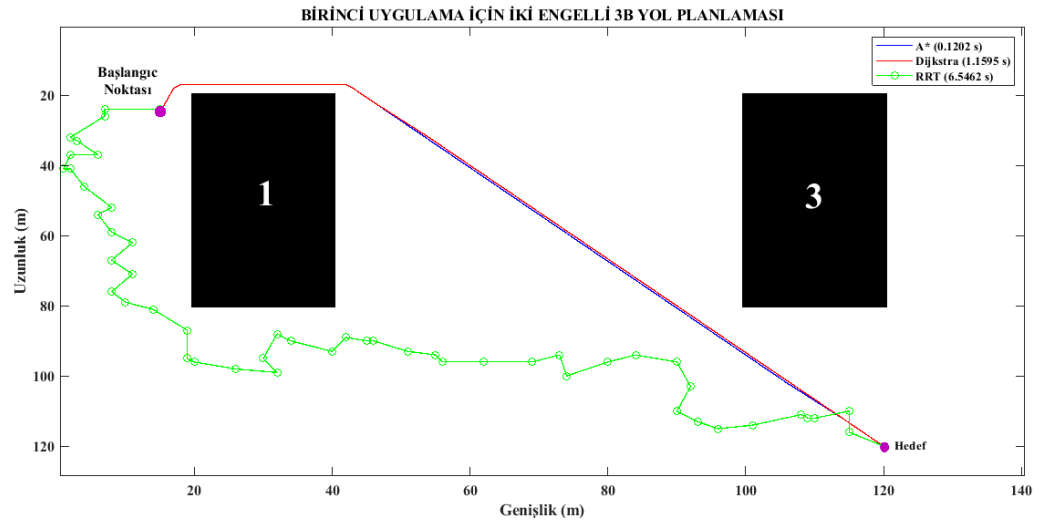
Şekil 5.17. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.18'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1161 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1661 saniyede, RRT algoritmasına göre 42 metre yolu 5.1183 saniyede katettiği görülmektedir.



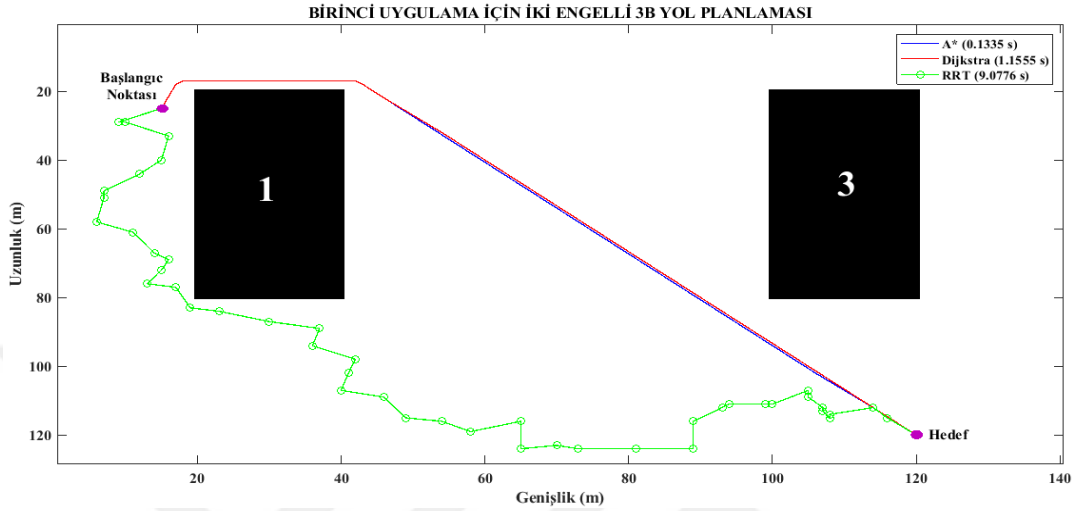
Şekil 5.18. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.19'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1202 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1595 saniyede, RRT algoritmasına göre 53 metre yolu 6.5462 saniyede katettiği görülmektedir.



Şekil 5.19. Birinci uygulama için üç engelli 3B yol planlaması

Şekil 5.20'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 42 metre yolu 0.1335 saniyede, Dijkstra algoritmasına göre 42 metre yolu 1.1555 saniyede, RRT algoritmasına göre 57 metre yolu 9.0776 saniyede katettiği görülmektedir.



Şekil 5.20. Birinci uygulama için üç engelli 3B yol planlaması

On farklı yol karşılaştırıldığında, Şekil 5.11'deki simülasyonun çizdiği yol uzunluk ve zamana göre daha uygun bir yol olduğu görülmüştür. On farklı yolun sonuçları Çizelge 5.2'de görülmektedir.

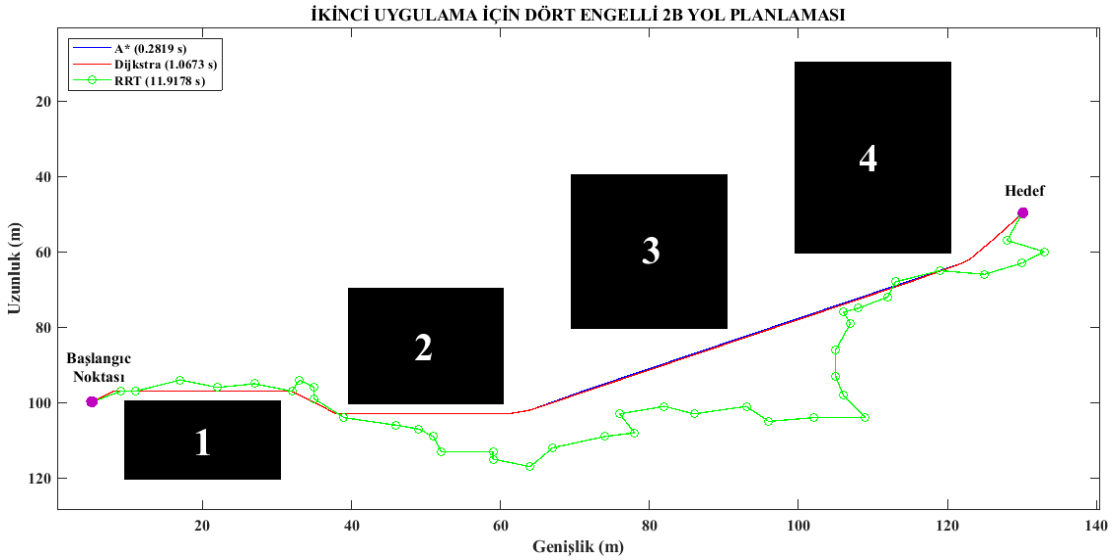
Çizelge 5.2. 2B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama Sayısı (1)	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)
5.11	0.1144	42	1.1677	42	3.6969	37
5.12	0.2241	42	1.3791	42	6.5535	55
5.13	0.1496	42	1.2028	42	4.5425	50
5.14	0.1151	42	1.1505	42	12.1412	46
5.15	0.1174	42	1.1702	42	3.5320	43
5.16	0.1151	42	1.1549	42	8.1496	44
5.17	0.1146	42	1.1551	42	11.4005	52
5.18	0.1161	42	1.1661	42	5.1183	42
5.19	0.1202	42	1.1595	42	6.5462	53
5.20	0.1335	42	1.1555	42	9.0776	47
Ortalama	0.1324	42	1.18577	42	7.22441	46.9

5.1.2. İkinci uygulama

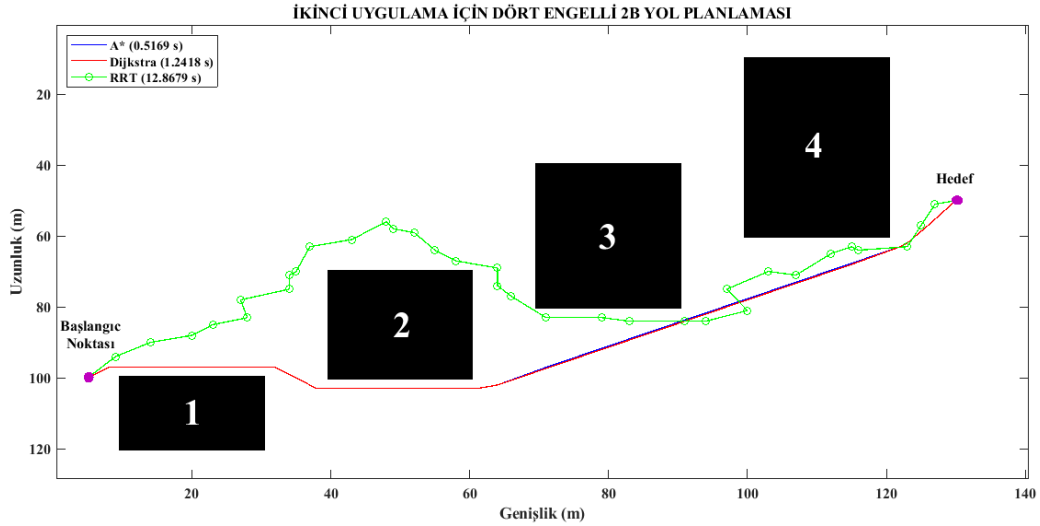
- ❖ Engel yüksekliğini ve drone yüksekliğni hesaba katmayan ortam 2B simülasyon ortamı olarak isimlendirilmektedir.

Şekil 5.21'de gösterildiği gibi, üç farklı yol planlama algoritması kullanılmaktadır. Algoritmalar sırasıyla A*, Dijkstra ve RRT'dir. RRT algoritmasında, algoritmanın 10 kez çalıştırılması sonucunda optimum yol seçilmektedir. Şekilde gösterildiği gibi, her biri farklı bir uzunluk ve genişliğe sahip dört farklı engel vardır. Birinci engel 20 metre genişliğinde ve 20 metre uzunluğunda ve ikinci engel 20 metre genişliğinde, 23 metre uzunluğunda, üçüncü engel 20 metre genişliğinde ve 40 metre uzunluğunda ve dördüncü engel 20 metre genişliğinde ve 50 metre uzunluğundadır. Drone, A* algoritmasına göre 71 metre yolu 0.2819 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0673 saniyede, RRT algoritmasına göre 42 metre yolu 11.9178 saniyede katettiği görülmektedir.



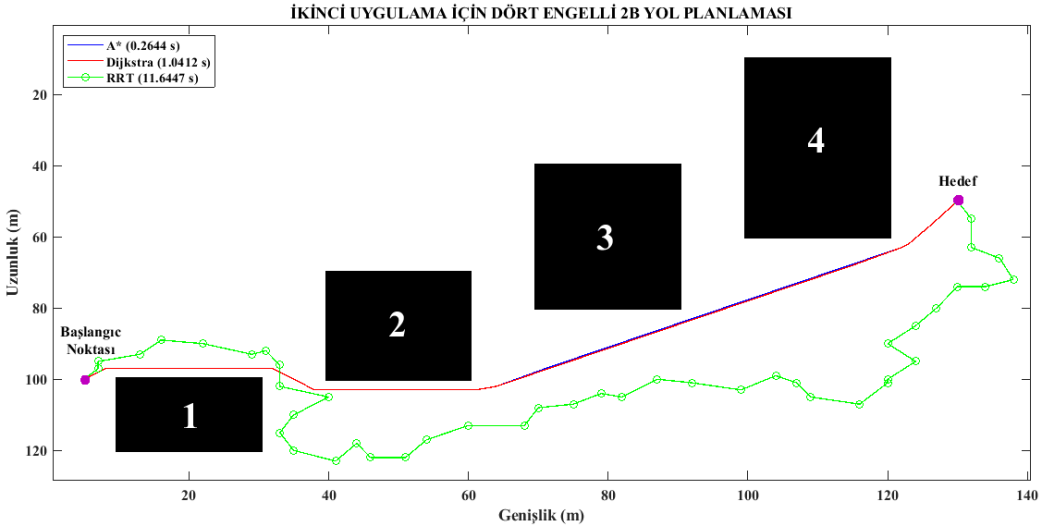
Şekil 5.21. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.22'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.5169 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.2418 saniyede, RRT algoritmasına göre 36 metre yolu 12.8679 saniyede katettiği görülmektedir.



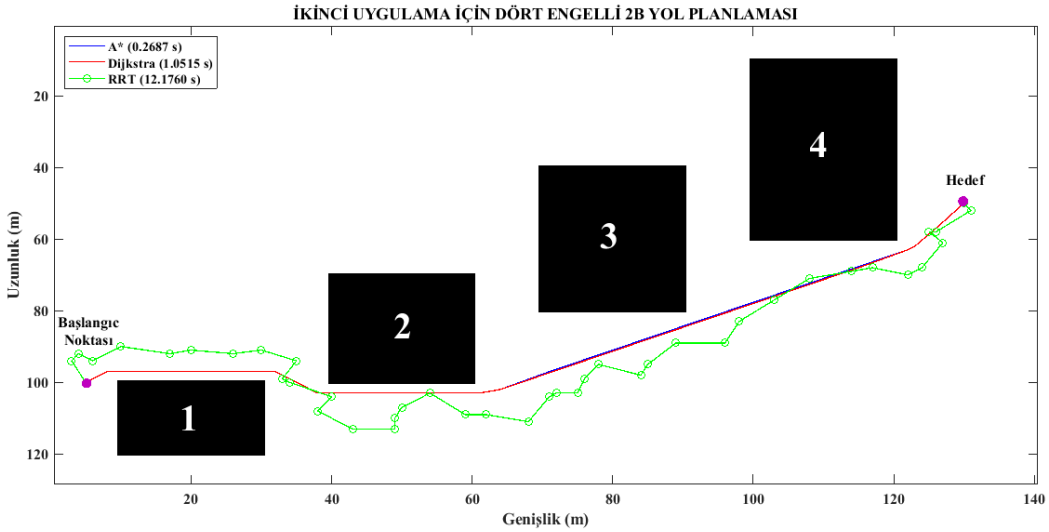
Şekil 5.22. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.23'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2644 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0412 saniyede, RRT algoritmasına göre 45 metre yolu 11.6447 saniyede katettiği görülmektedir.



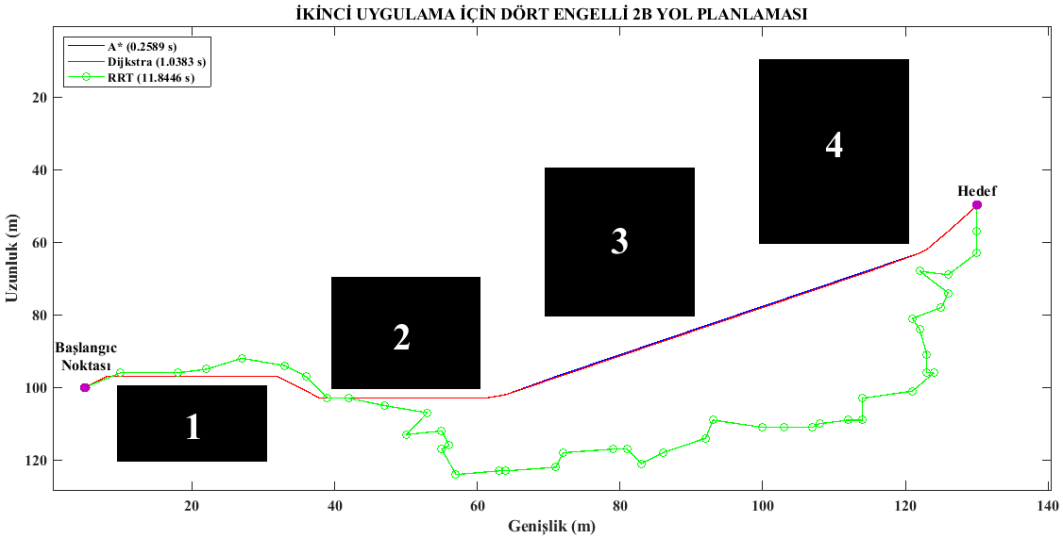
Şekil 5.23. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.24'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2687 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0515 saniyede, RRT algoritmasına göre 43 metre yolu 12.1760 saniyede katettiği görülmektedir.



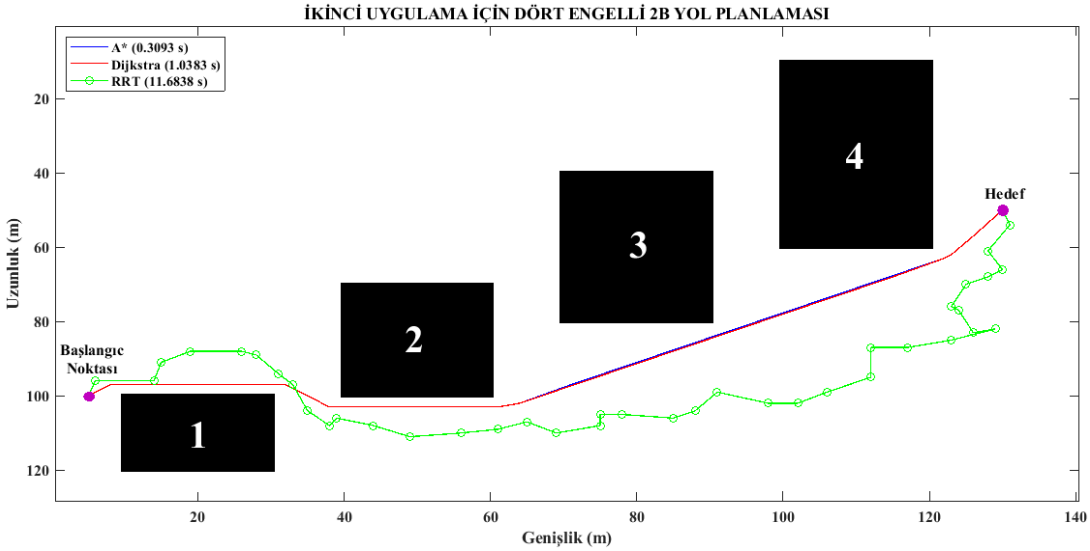
Şekil 5.24. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.25'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2589 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0383 saniyede, RRT algoritmasına göre 46 metre yolu 12.8446 saniyede katettiği görülmektedir.



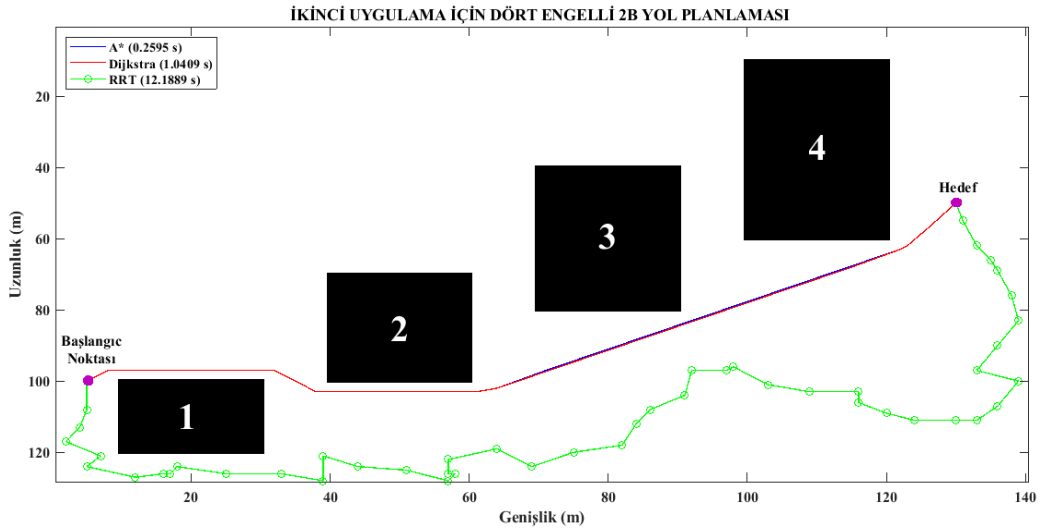
Şekil 5.25. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.26'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.3093 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0383 saniyede, RRT algoritmasına göre 41 metre yolu 11.6838 saniyede katettiği görülmektedir.



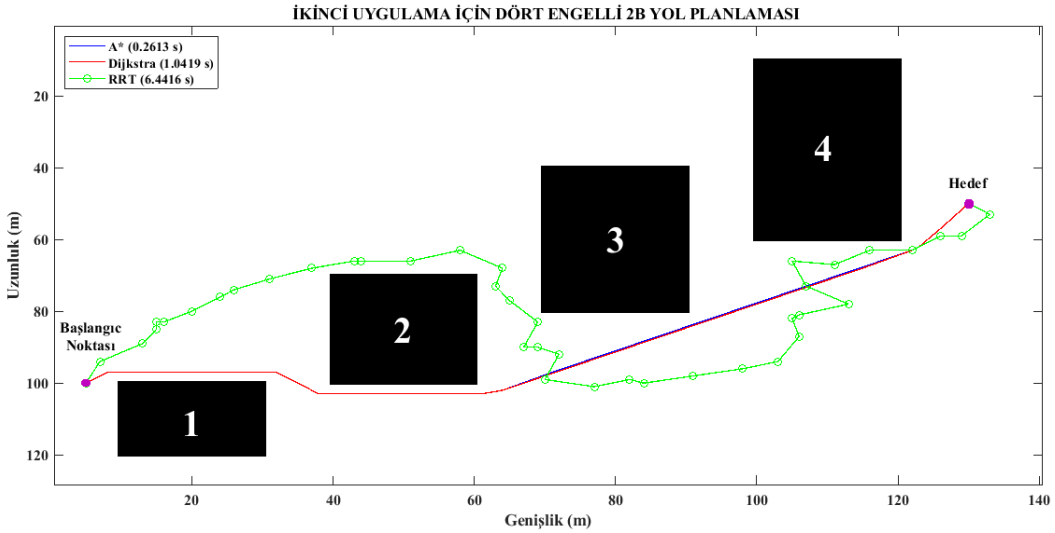
Şekil 5.26. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.27'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2595 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0409 saniyede, RRT algoritmasına göre 49 metre yolu 12.1889 saniyede katettiği görülmektedir.



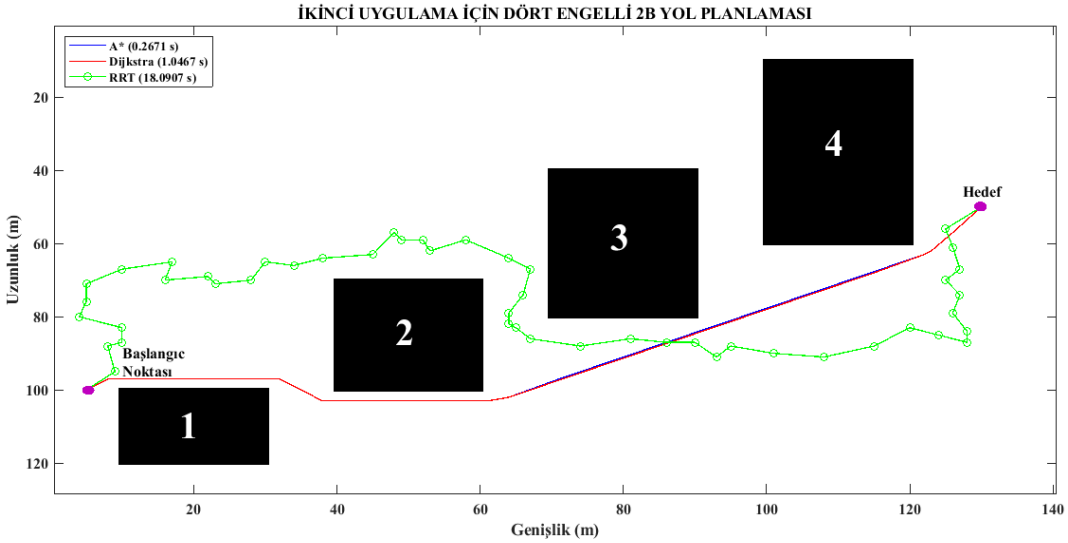
Şekil 5.27. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.28'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2613 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0419 saniyede, RRT algoritmasına göre 42 metre yolu 6.5516 saniyede katettiği görülmektedir.



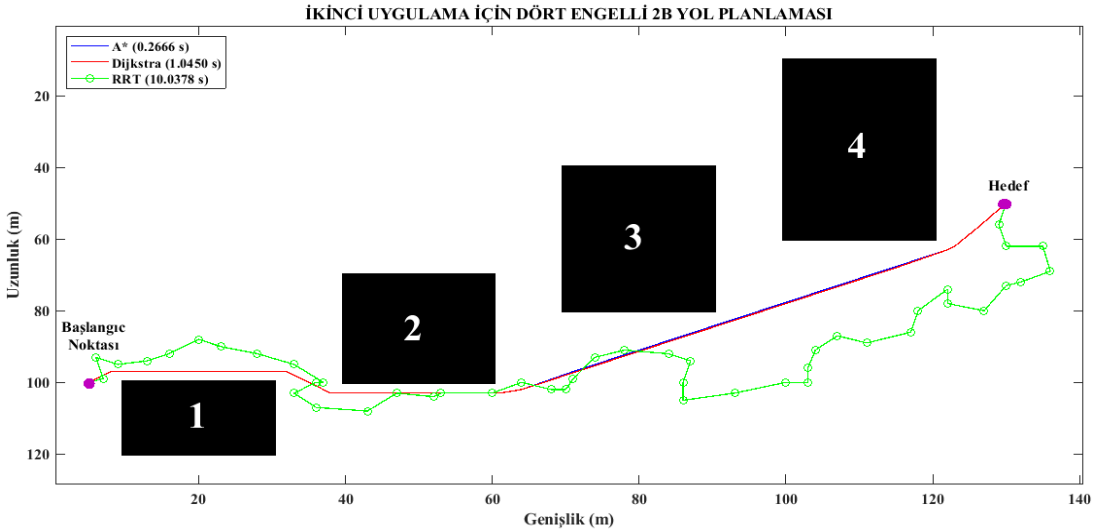
Şekil 5.28. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.29'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2671 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0467 saniyede, RRT algoritmasına göre 50 metre yolu 18.0907 saniyede katettiği görülmektedir.



Şekil 5.29. İkinci uygulama için dört engelli 2B yol planlaması

Şekil 5.30'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 71 metre yolu 0.2666 saniyede, Dijkstra algoritmasına göre 71 metre yolu 1.0450 saniyede, RRT algoritmasına göre 48 metre yolu 10.0378 saniyede katettiği görülmektedir.



Şekil 5.30. İkinci uygulama için dört engelli 2B yol planlaması

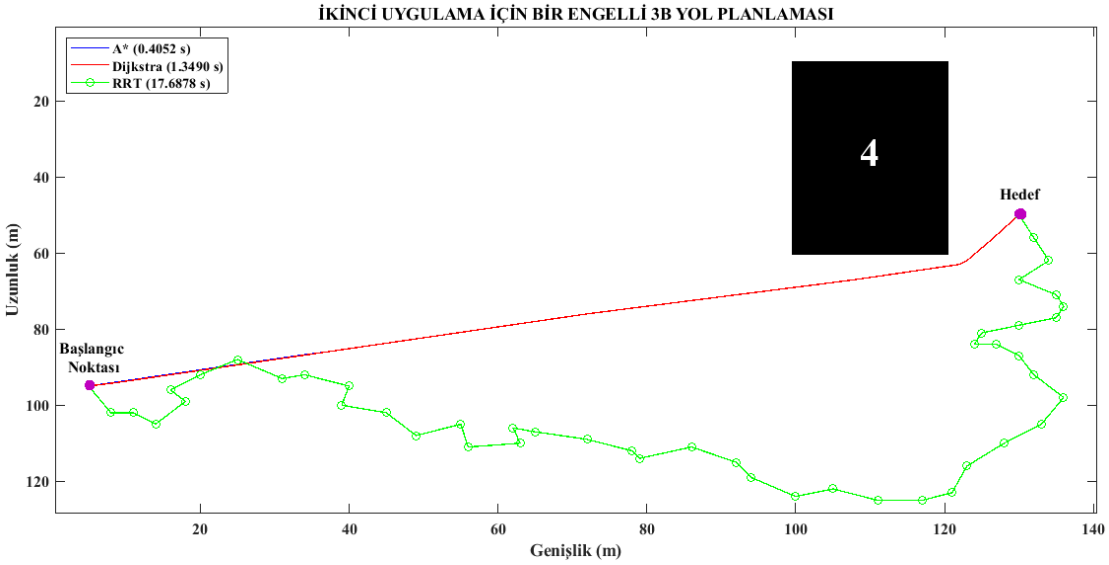
On farklı yol karşılaştırıldığında, Şekil 5.24'deki simülasyonun çizdiği yol uzunluk ve zamana göre daha uygun bir yol olduğu görülmüştür. On farklı yolun sonuçları Çizelge 5.3'de görülmektedir.

Çizelge 5.3. 2B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama Sayısı (2)	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)
5.21	0.2819	71	1.0673	71	11.9178	42
5.22	0.5169	71	1.2418	71	12.8679	36
5.23	0.2644	71	1.0412	71	11.6447	45
5.24	0.2687	71	1.0515	71	12.1760	43
5.25	0.2589	71	1.0383	71	11.8446	46
5.26	0.3093	71	1.0383	71	11.6838	41
5.27	0.2595	71	1.0409	71	12.1889	49
5.28	0.2613	71	1.0419	71	6.4416	42
5.29	0.2671	71	1.0467	71	18.0907	50
5.30	0.2666	71	1.0450	71	10.0378	48
Ortalama	0.29546	71	1.06529	71	11.85378	44.2

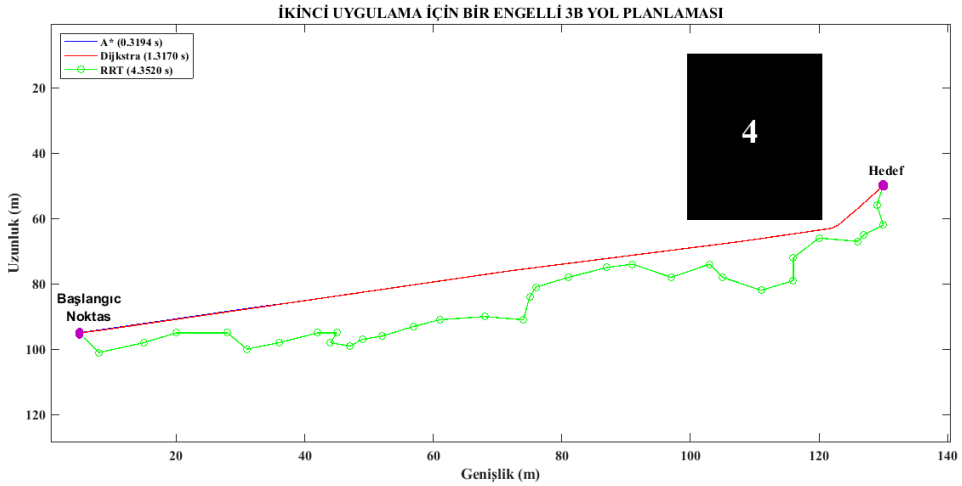
- Engel yüksekliğini ve drone yüksekliğni hesaba katan ortam 3B simülasyon ortamı olarak isimlendirilmektedir.

Şekil 5.31, uçağın yüksekliğini ve engellerin yüksekliğini belirledikten sonra, birinci engelin engellerin yüksekliği 20m, ikinci engelin yüksekliği 30m, üçüncü engelin yüksekliği 40m ve dördüncü engelin yüksekliği 50m 'dır. Dronun yüksekliği 45m olduğunda önerilen algoritmalar çalıştırıldığında birinci, ikinci ve üçüncü engelleri görmezden gelmektedir çünkü verilen dronun yüksekliği engellerin yüksekliklerinden daha yüksektir ve hedef noktaya daha kısa sürede ulaşmaktadır. Başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.4052 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3490 saniyede, RRT algoritmasına göre 47 metre yolu 17.6878 saniyede katettiği görülmektedir.



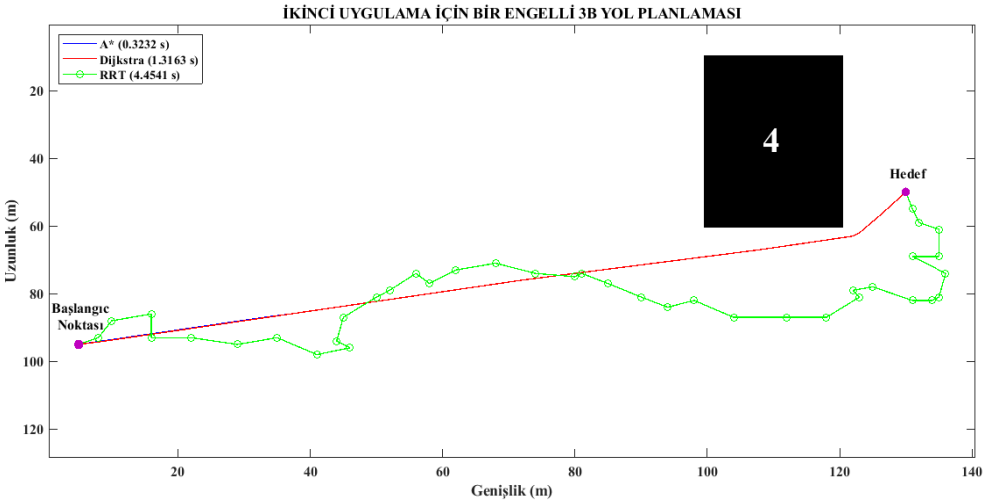
Şekil 5.31. İkinci uygulama için bir engelli 3B yol planlaması

Şekil 5.32'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3194 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3170 saniyede, RRT algoritmasına göre 34 metre yolu 4.3520 saniyede katettiği görülmektedir.



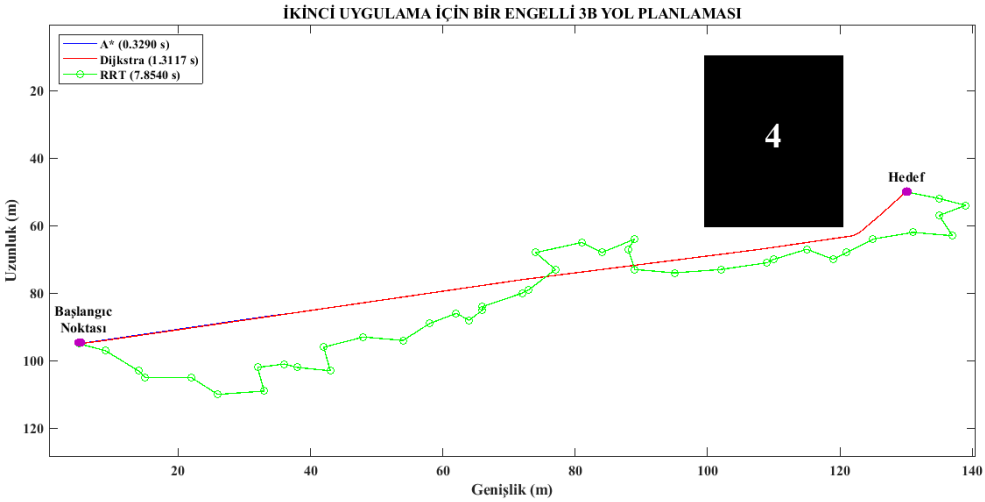
Şekil 5.32. İkinci uygulama için dört engelli 3B yol planlaması

Şekil 5.33'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3232 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3163 saniyede, RRT algoritmasına göre 41 metre yolu 4.4541 saniyede katettiği görülmektedir.



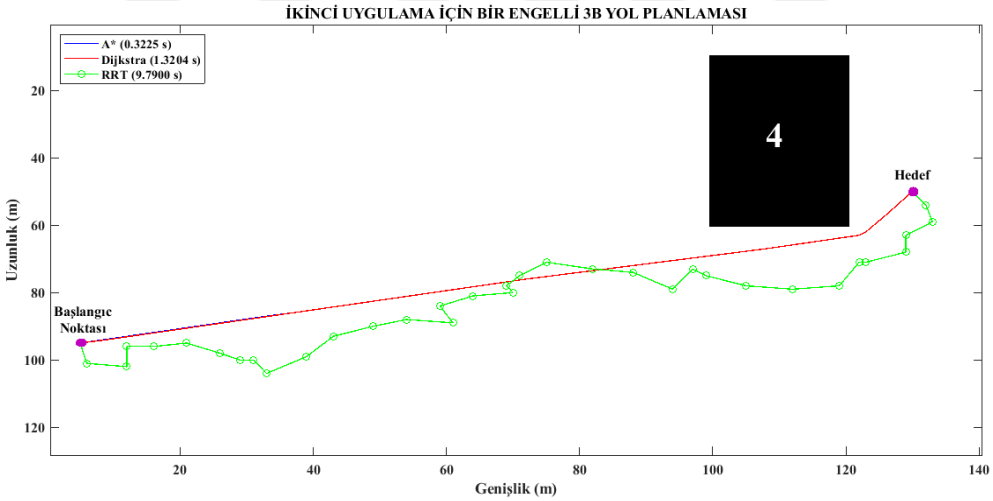
Şekil 5.33. İkinci uygulama için bir engelli 3B yol planlaması

Şekil 5.34'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3290 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3117 saniyede, RRT algoritmasına göre 42 metre yolu 7.8540 saniyede katettiği görülmektedir.



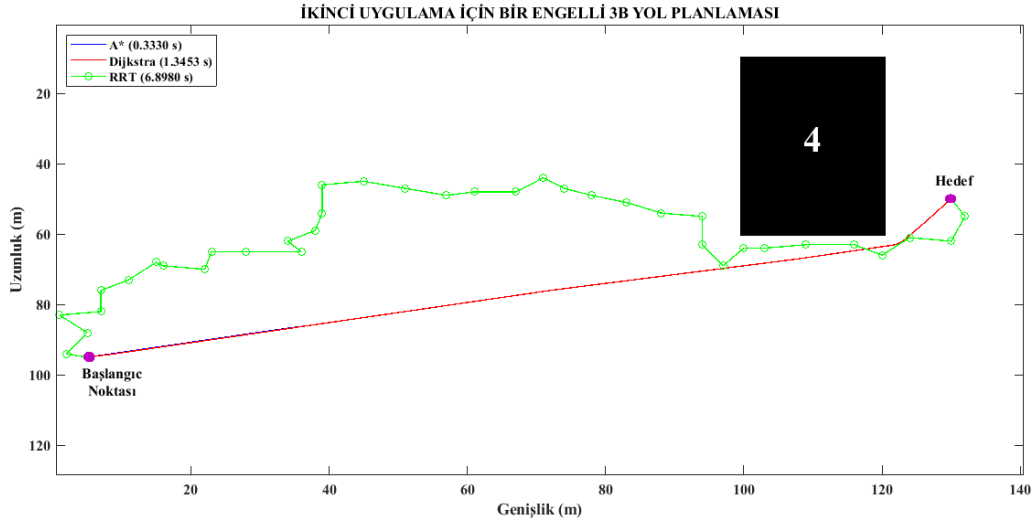
Şekil 5.34. İkinci uygulama için bir engelli 3B yol planlaması

Şekil 5.35'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3225 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3204 saniyede, RRT algoritmasına göre 36 metre yolu 9.7900 saniyede katettiği görülmektedir.



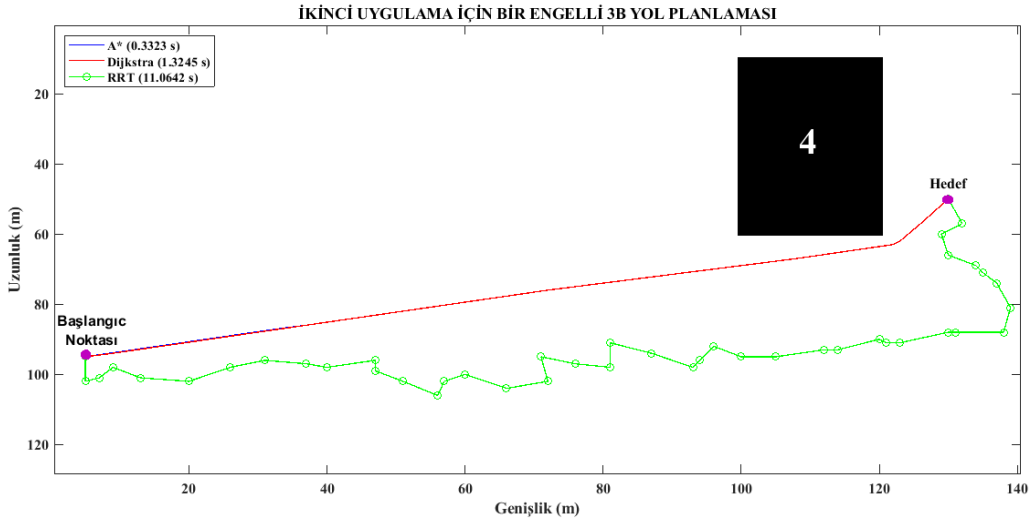
Şekil 5.35. İkinci uygulama için bir engelli 3B yol planlaması

Şekil 5.36'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3330 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3453 saniyede, RRT algoritmasına göre 39 metre yolu 6.8980 saniyede katettiği görülmektedir.



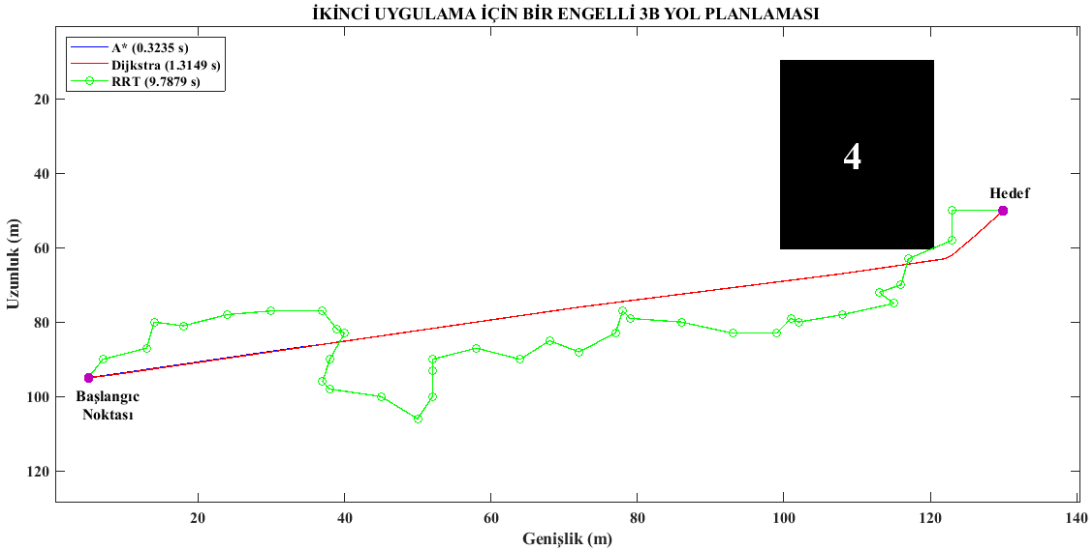
Şekil 5.36. İkinci uygulama için dört engelli 3B yol planlaması

Şekil 5.37'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3323 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3245 saniyede, RRT algoritmasına göre 44 metre yolu 11.0642 saniyede katettiği görülmektedir.



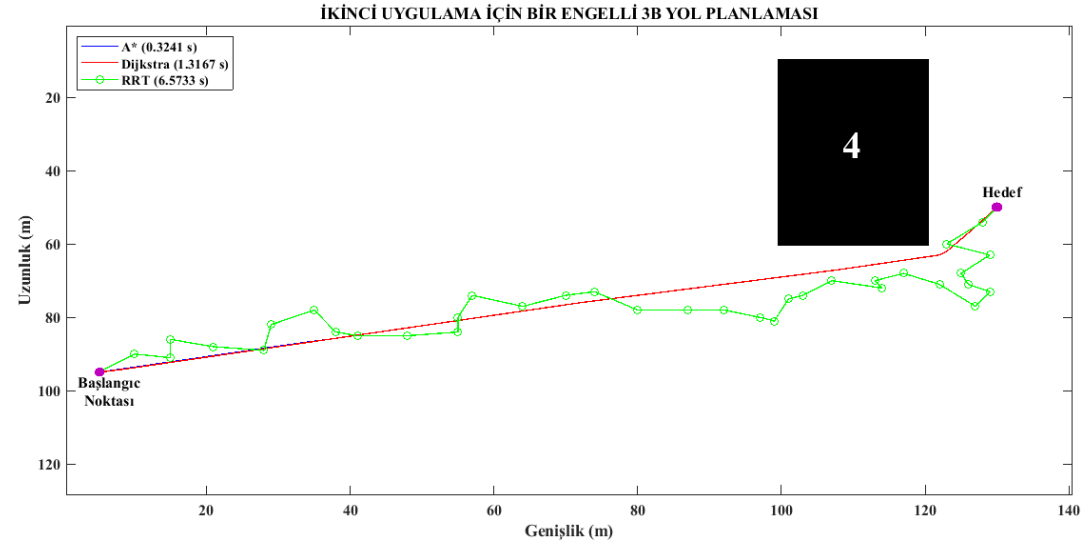
Şekil 5.37. İkinci uygulama için bir engelli 3B yol planlaması

Şekil 5.38'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3235 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3149 saniyede, RRT algoritmasına göre 38 metre yolu 9.7879 saniyede katettiği görülmektedir.



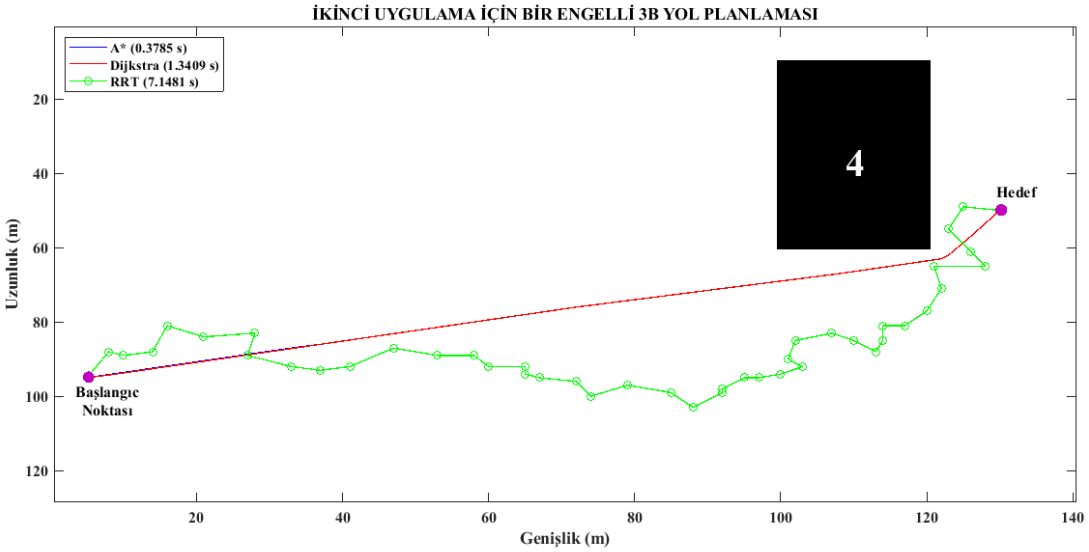
Şekil 5.38. İkinci uygulama için bir engelli 3B yol planlaması

Şekil 5.39'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3241 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3167 saniyede, RRT algoritmasına göre 37 metre yolu 6.5733 saniyede katettiği görülmektedir.



Şekil 5.39. İkinci uygulama için dört engelli 3B yol planlaması

Şekil 5.40'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 25 metre yolu 0.3785 saniyede, Dijkstra algoritmasına göre 25 metre yolu 1.3409 saniyede, RRT algoritmasına göre 45 metre yolu 7.1481 saniyede katettiği görülmektedir.



Şekil 5.40. İkinci uygulama için bir engelli 3B yol planlaması

On farklı yol karşılaştırıldığında, Şekil 5.39'deki simülasyonun çizdiği yol uzunluk ve zamana göre daha uygun bir yol olduğu görülmüştür. On farklı yolun sonuçları Çizelge 5.4'de görülmektedir.

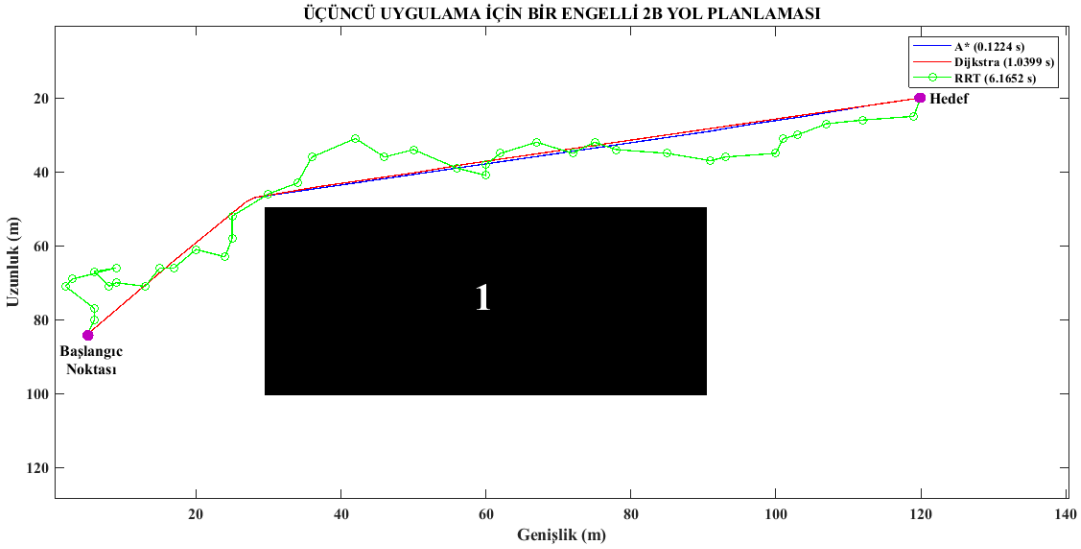
Çizelge 5.4. 2B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama Sayısı (2)	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu	Zaman (sn)	Yol uzunluğu	Zaman (sn)	Yol uzunluğu
5.31	0.4052	25	1.3490	25	17.6878	47
5.32	0.3194	25	1.3170	25	4.3520	34
5.33	0.3232	25	1.3163	25	4.4541	41
5.34	0.3290	25	1.3117	25	7.8540	42
5.35	0.3225	25	1.3204	25	9.7900	36
5.36	0.3330	25	1.3453	25	6.8980	39
5.37	0.3323	25	1.3245	25	11.0642	44
5.38	0.3235	25	1.3149	25	9.7879	38
5.39	0.3241	25	1.3167	25	6.5733	37
5.40	0.3785	25	1.3409	25	7.1481	45
Ortalama	0.33907	25	1.32567	25	8.56094	40.3

5.1.3. Üçüncü uygulama

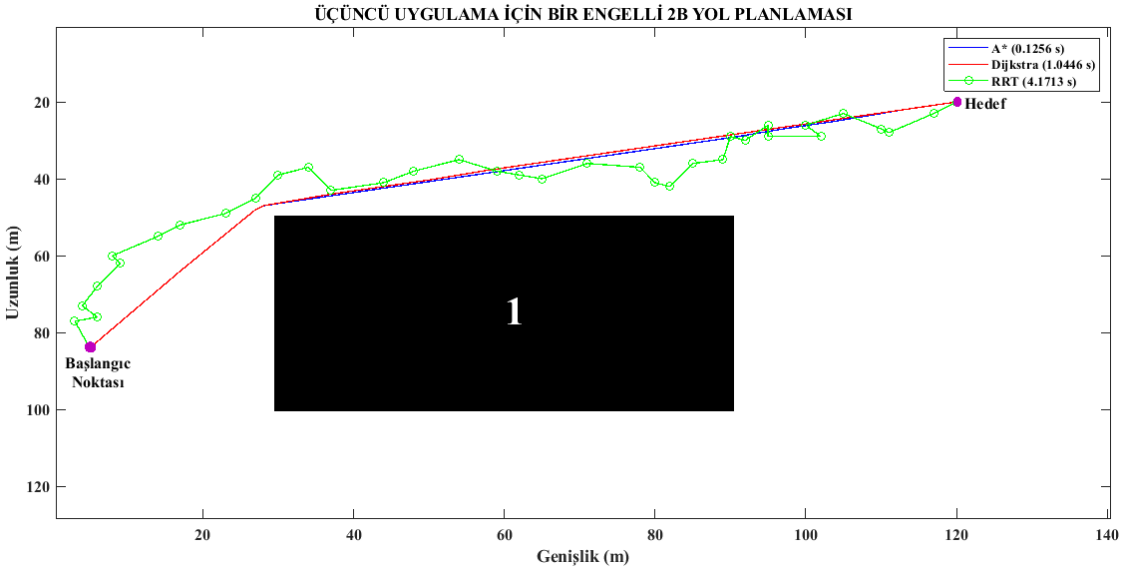
- Engel yüksekliğini ve drone yüksekliğni hesaba katmayan ortamı 2B simülasyon ortamı olarak isimlendirilmektedir.

Şekil 5.41'de gösterildiği gibi, üç farklı yol planlama algoritması kullanılmaktadır. Algoritmalar sırasıyla A*, Dijkstra ve RRT'dir. RRT algoritmasında, algoritmanın 10 kez çalıştırılması sonucunda optimum yol seçilmektedir. Şekilde tek bir engel vardır. Engel 50 metre genişliğinde ve 60 metre uzunluğundadır. Drone, A* algoritmasına göre 22 metre yolu 0.1224 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0399 saniyede, RRT algoritmasına göre 40 metre yolu 6.1652 saniyede katettiği görülmektedir.



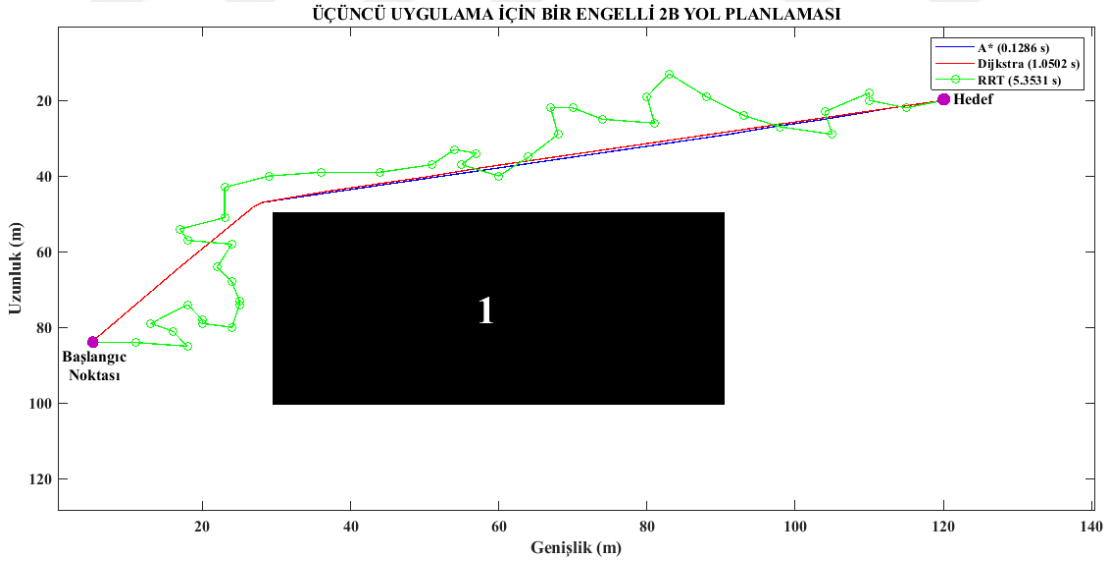
Şekil 5.41. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.42'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1256 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0446 saniyede, RRT algoritmasına göre 37 metre yolu 4.1723 saniyede katettiği görülmektedir.



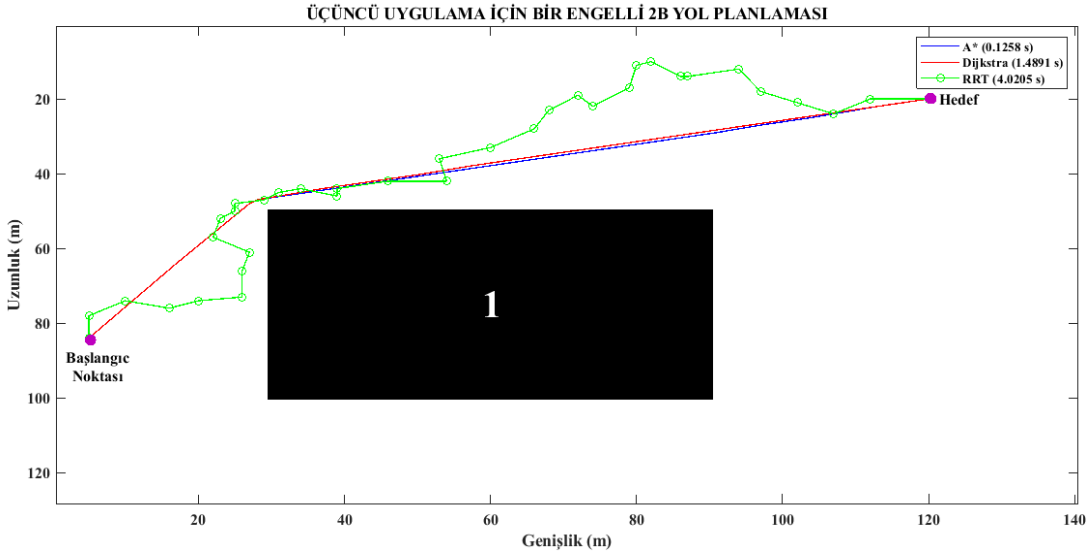
Şekil 5.42. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.43'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1286 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0502 saniyede, RRT algoritmasına göre 43 metre yolu 5.3531 saniyede katettiği görülmektedir.



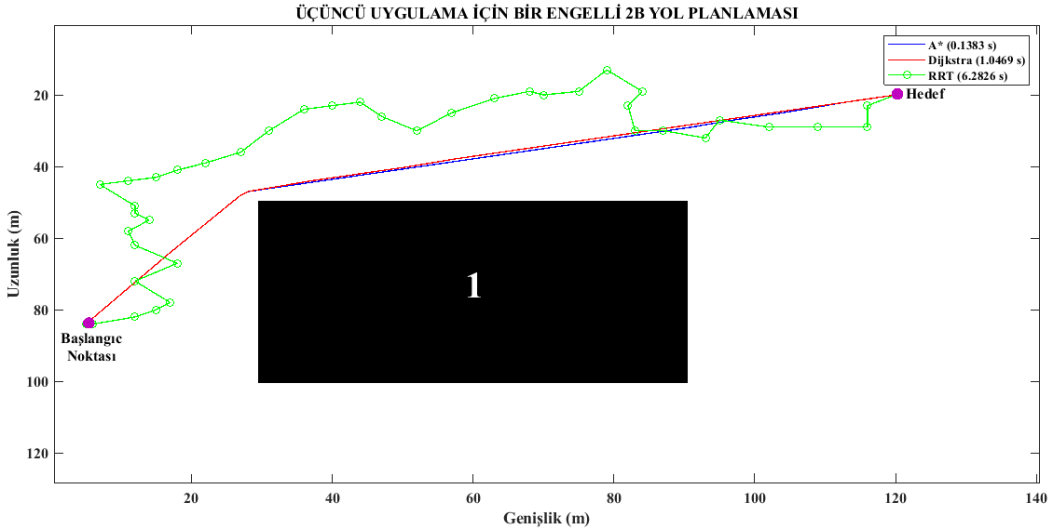
Şekil 5.43. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.44'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1258 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.4891 saniyede, RRT algoritmasına göre 36 metre yolu 4.0205 saniyede katettiği görülmektedir.



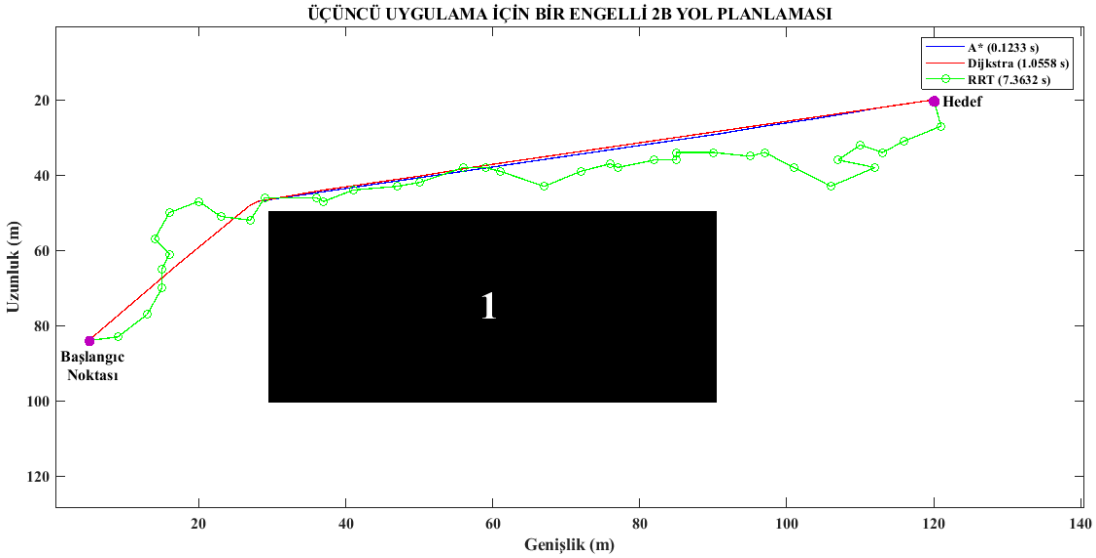
Şekil 5.44. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.45'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1383 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0469 saniyede, RRT algoritmasına göre 41 metre yolu 6.2826 saniyede katettiği görülmektedir.



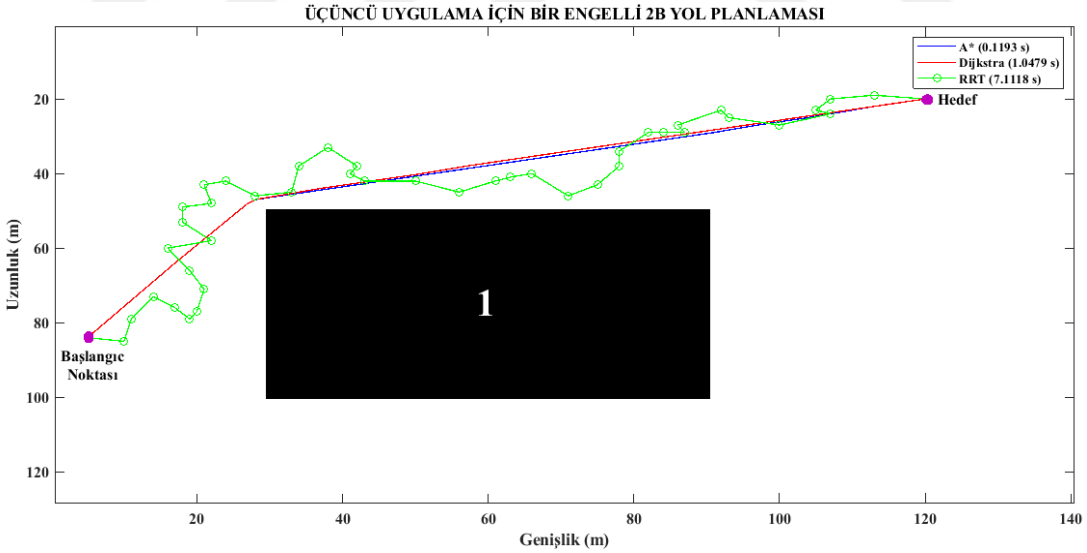
Şekil 5.45. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.46'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1233 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0558 saniyede, RRT algoritmasına göre 39 metre yolu 7.3632 saniyede katettiği görülmektedir.



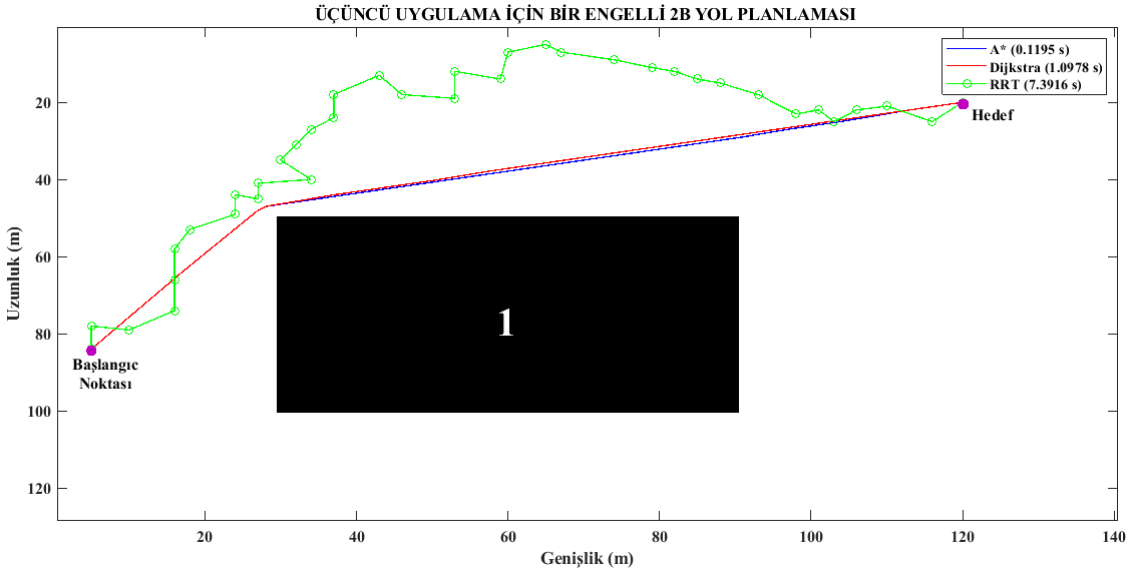
Şekil 5.46. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.47'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1193 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0479 saniyede, RRT algoritmasına göre 44 metre yolu 7.1118 saniyede katettiği görülmektedir.



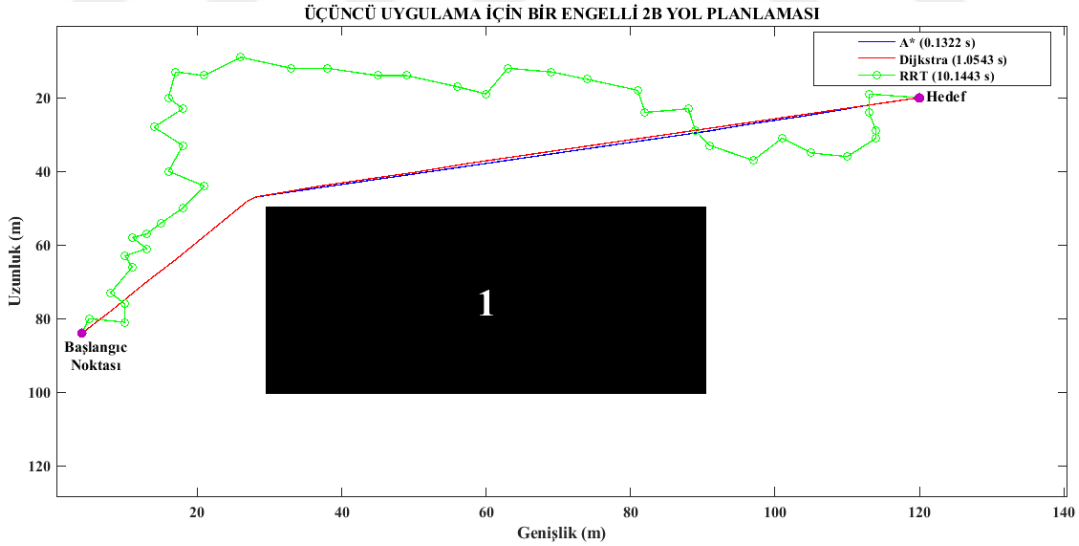
Şekil 5.47. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.48'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1195 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0978 saniyede, RRT algoritmasına göre 38 metre yolu 7.3916 saniyede katettiği görülmektedir.



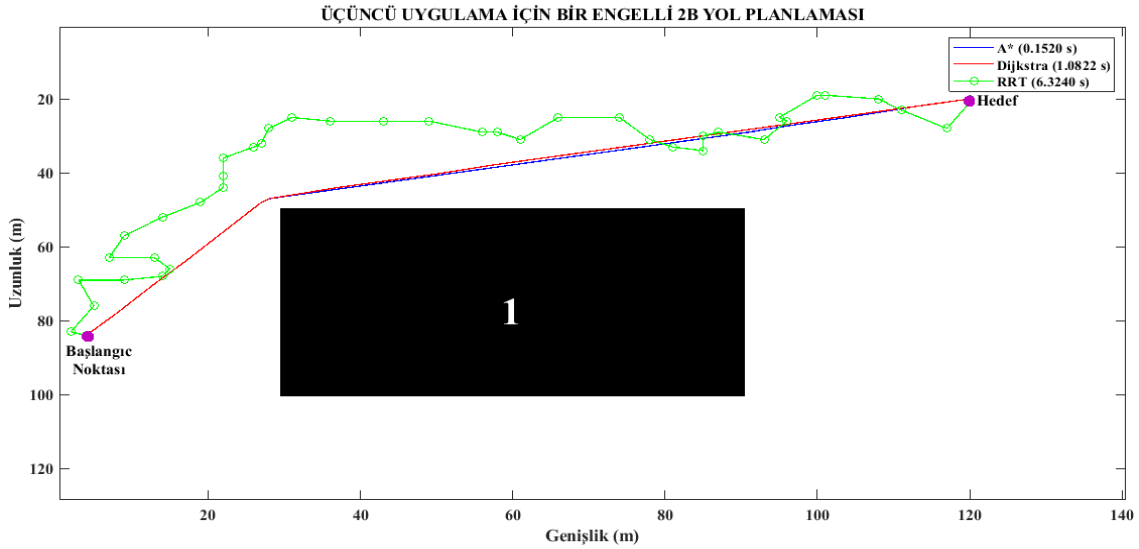
Şekil 5.48. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.49'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1322 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0543 saniyede, RRT algoritmasına göre 44 metre yolu 10.1443 saniyede katettiği görülmektedir.



Şekil 5.49. Üçüncü uygulama için bir engelli 2B yol planlaması

Şekil 5.50'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 22 metre yolu 0.1520 saniyede, Dijkstra algoritmasına göre 22 metre yolu 1.0822 saniyede, RRT algoritmasına göre 42 metre yolu 6.3240 saniyede katettiği görülmektedir.



Şekil 5.50. Üçüncü uygulama için bir engelli 2B yol planlaması

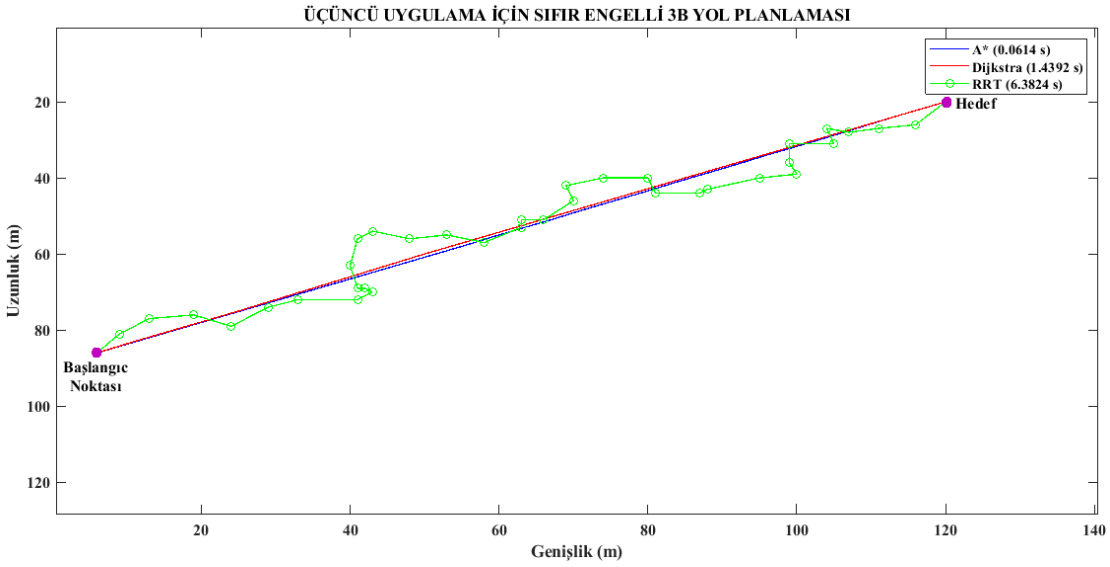
On farklı yol karşılaştırıldığında, Şekil 5.42'deki simülasyonun çizdiği yol uzunluk ve zamana göre daha uygun bir yol olduğu görülmüştür. On farklı yolun sonuçları Çizelge 5.5'de görülmektedir.

Çizelge 5.5. 2B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama Sayısı (3)	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu	Zaman (sn)	Yol uzunluğu	Zaman (sn)	Yol uzunluğu
5.41	0.1224	22	1.0399	22	6.1652	40
5.42	0.1256	22	1.0446	22	4.1713	37
5.43	0.1286	22	1.0502	22	5.3531	43
5.44	0.1258	22	1.4891	22	4.0205	36
5.45	0.1383	22	1.0469	22	6.2826	41
5.46	0.1233	22	1.0558	22	7.3632	39
5.47	0.1193	22	1.0479	22	7.1118	44
5.48	0.1195	22	1.0978	22	7.3916	38
5.49	0.1322	22	1.0543	22	10.1443	44
5.50	0.1520	22	1.0822	22	6.3240	42
Ortalama	0.12878	22	1.10087	22	16.324	40.4

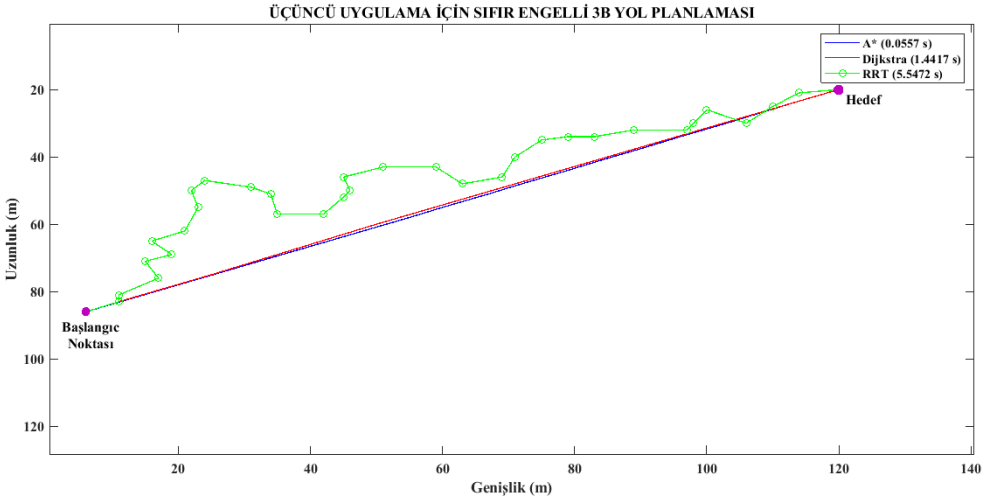
- Engel yüksekliğini ve drone yüksekliğini hesaba katan ortam 3B simülasyon ortamı olarak isimlendirilmektedir.

Şekil 5.51'de, engelin yüksekliği 50m dronun yüksekliği ise engelin daha yüksek olarak 55m olaraka belirlenmiştir. Böylece drone önerilen algoritmalar çalıştırıldığında engeli görmezden gelerek hedef noktaya daha kısa sürede ulaşmıştır. Başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0614 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4392 saniyede, RRT algoritmasına göre 37 metre yolu 6.3824 saniyede katettiği görülmektedir.



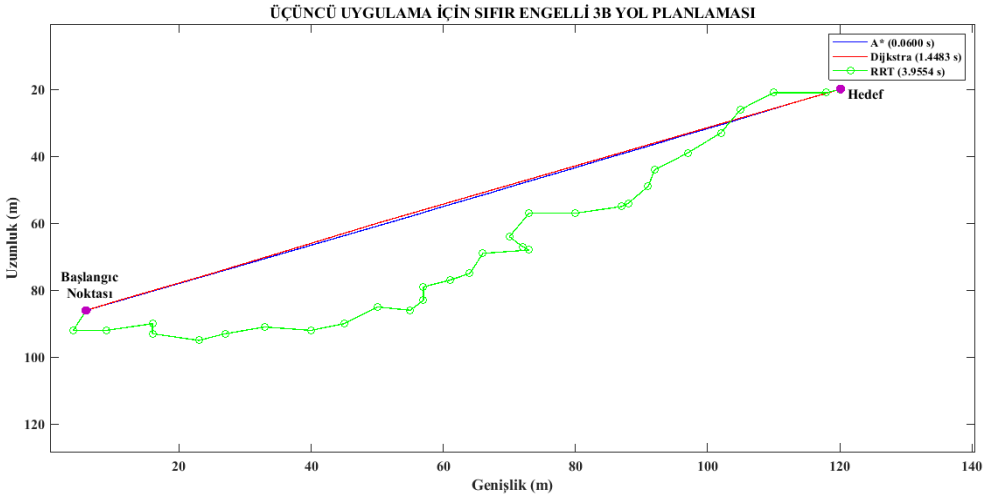
Şekil 5.51. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.52'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0557 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4417 saniyede, RRT algoritmasına göre 34 metre yolu 5.5472 saniyede katettiği görülmektedir.



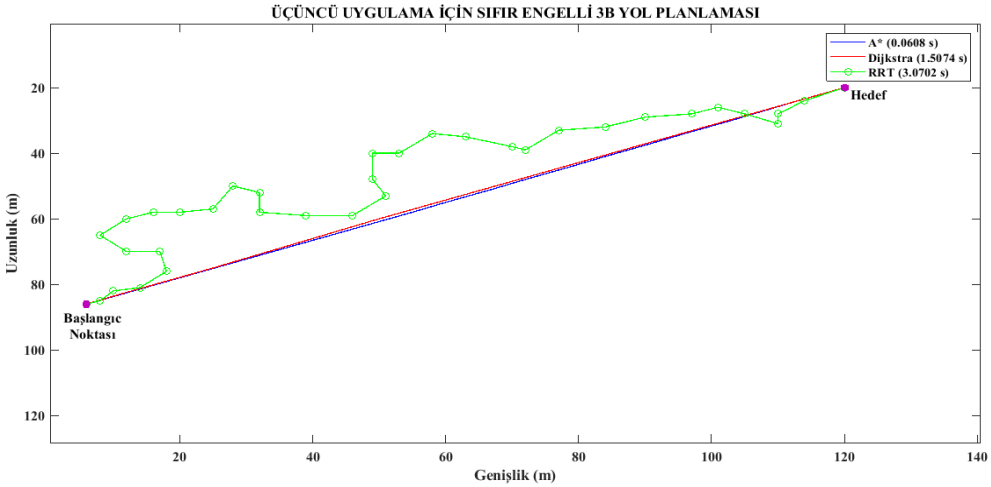
Şekil 5.52. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.53'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0600 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4483 saniyede, RRT algoritmasına göre 32 metre yolu 3.9554 saniyede katettiği görülmektedir.



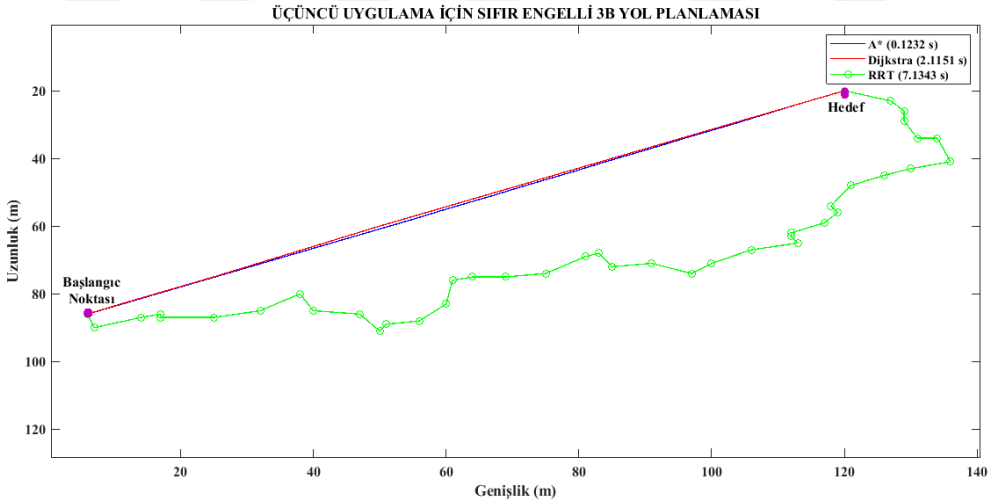
Şekil 5.53. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.54'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0608 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.5074 saniyede, RRT algoritmasına göre 35 metre yolu 3.0702 saniyede katettiği görülmektedir.



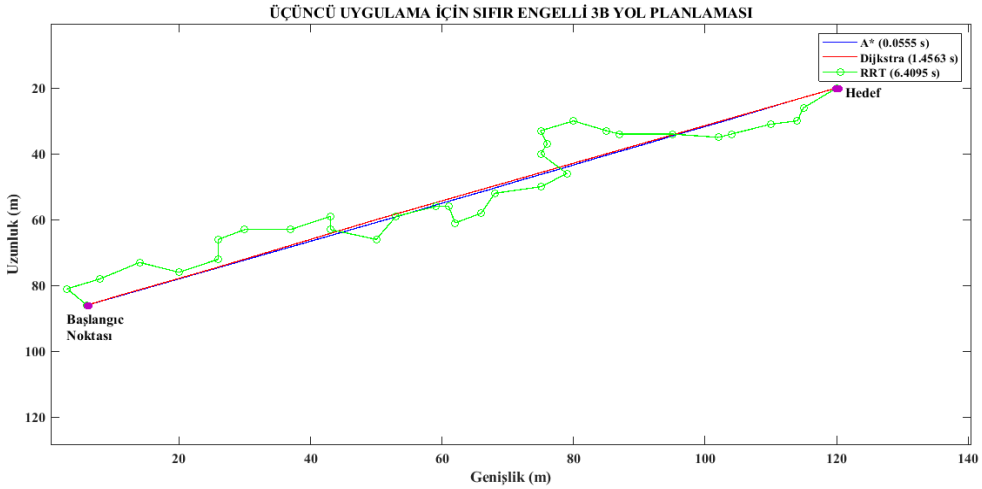
Şekil 5.54. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.55'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.1232 saniyede, Dijkstra algoritmasına göre 19 metre yolu 2.1151 saniyede, RRT algoritmasına göre 41 metre yolu 7.1343 saniyede katettiği görülmektedir.



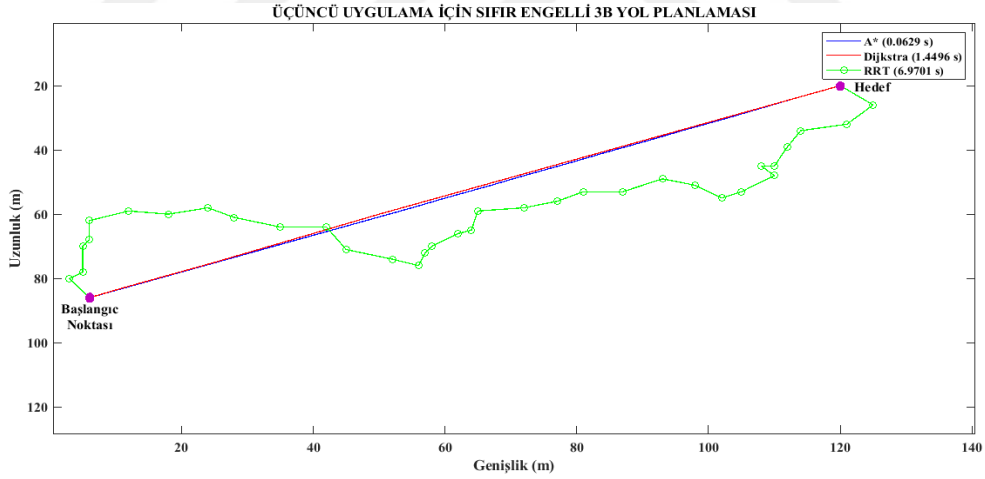
Şekil 5.55. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.56'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0555 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4563 saniyede, RRT algoritmasına göre 33 metre yolu 6.4095 saniyede katettiği görülmektedir.



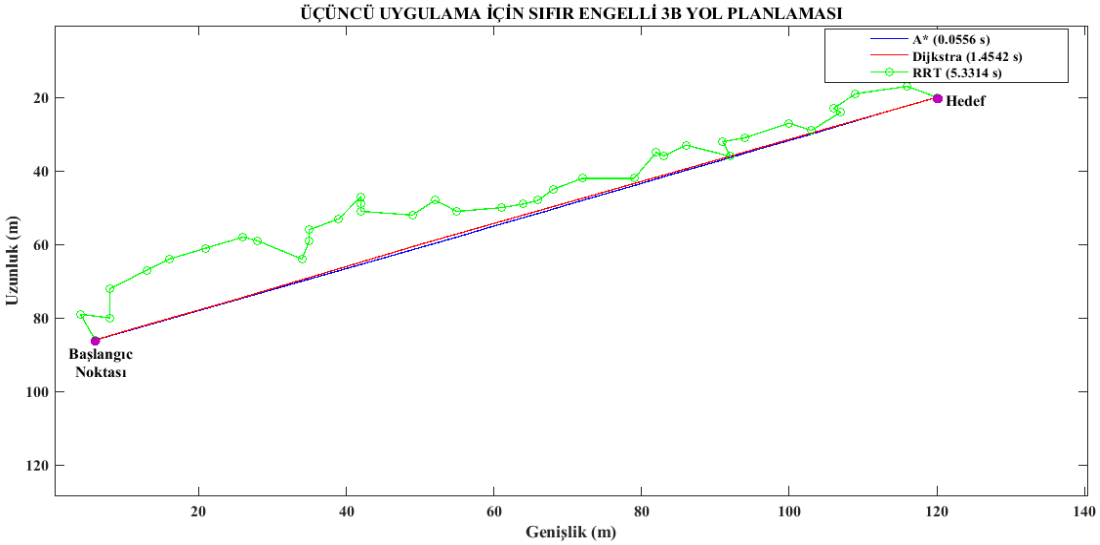
Şekil 5.56. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.57'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0629 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4496 saniyede, RRT algoritmasına göre 36 metre yolu 6.9701 saniyede katettiği görülmektedir.



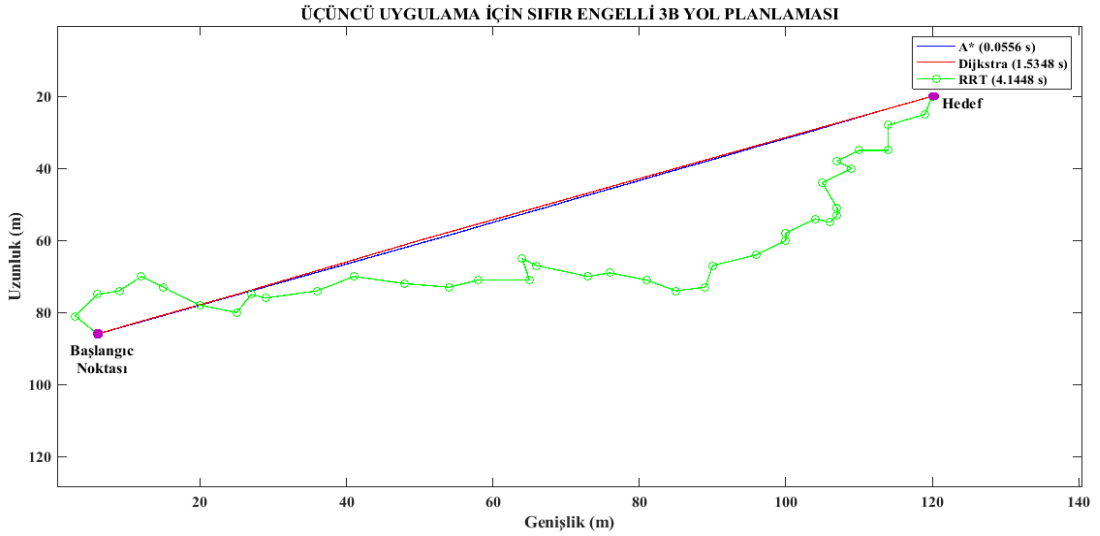
Şekil 5.57. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.58'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0556 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4542 saniyede, RRT algoritmasına göre 38 metre yolu 5.3314 saniyede katettiği görülmektedir.



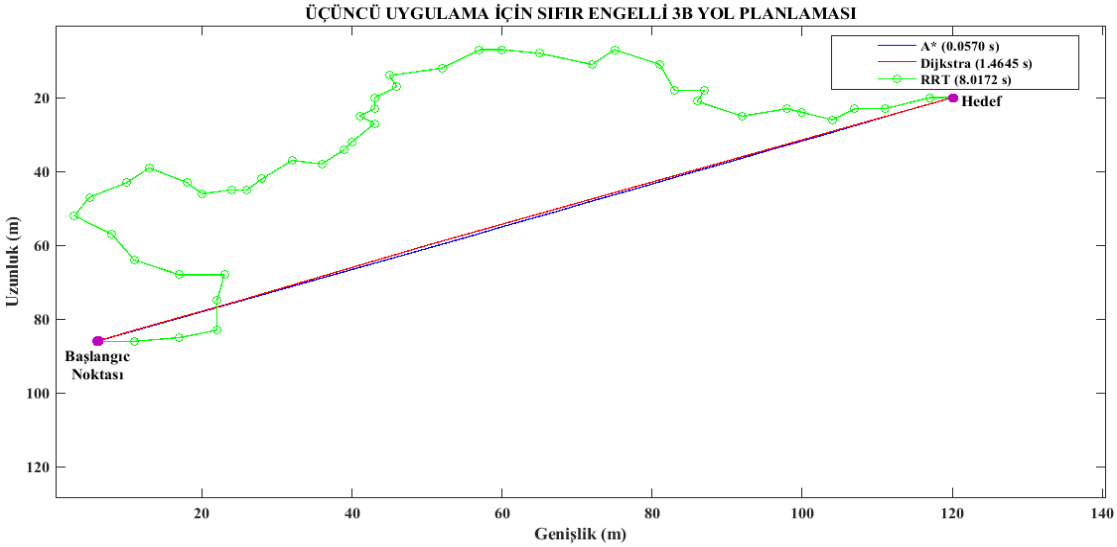
Şekil 5.58. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.59'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0556 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.5348 saniyede, RRT algoritmasına göre 39 metre yolu 4.1448 saniyede katettiği görülmektedir.



Şekil 5.59. Üçüncü uygulama için sıfır engelli 3B yol planlaması

Şekil 5.60'de başlangıç noktasından hedef noktaya giden yol 2B ortamında hesaplanmaktadır. Drone, A* algoritmasına göre 19 metre yolu 0.0570 saniyede, Dijkstra algoritmasına göre 19 metre yolu 1.4645 saniyede, RRT algoritmasına göre 46 metre yolu 8.0172 saniyede katettiği görülmektedir.



Şekil 5.60. Üçüncü uygulama için sıfır engelli 3B yol planlaması

On farklı yol karşılaştırıldığında, Şekil 5.56'deki simülasyonun çizdiği yol uzunluk ve zamana göre daha uygun bir yol olduğu görülmüştür. On farklı yolun sonuçları Çizelge 5.6'de görülmektedir.

Çizelge 5.6. 2B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama Sayısı (3)	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)
5.51	0.0614	19	1.4392	19	6.3824	37
5.52	0.0557	19	1.4417	19	5.5472	34
5.53	0.0600	19	1.4483	19	3.9554	32
5.54	0.0608	19	1.5074	19	3.0702	35
5.55	0.1232	19	2.1151	19	7.1343	41
5.56	0.0555	19	1.4563	19	6.4095	33
5.57	0.0622	19	1.4496	19	6.9701	36
5.58	0.0556	19	1.4542	19	5.3314	38
5.59	0.0556	19	1.5348	19	4.1448	39
5.60	0.0570	19	1.4645	19	8.0172	46
Ortalama	0.0647	19	1.53111	19	5.69625	37.1

5.2. Gerçek Zamanlı Test Ortamı

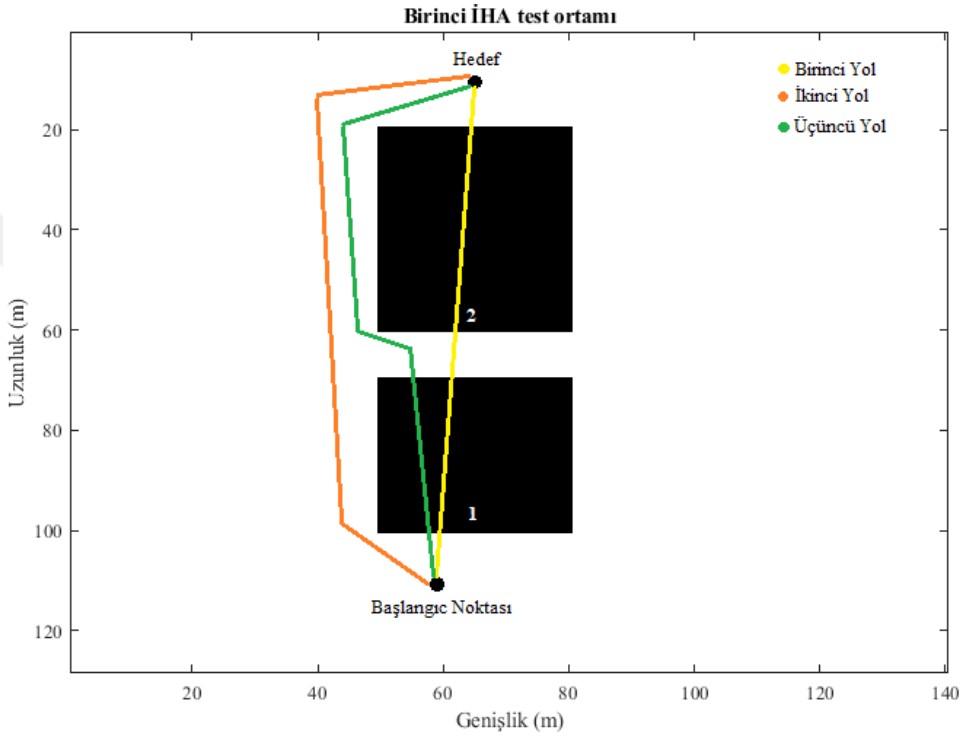
Gerçek zamanlı test ortamında 4 pervaneli (Quadcopter) kullanılmıştır. Quadcopter çeşitli pervanelere sahip, yüksek uçuş kapasitesine sahip bir hava aracıdır. Uçuş yapıldığı Quadcopter Şekil 4.61 de görüldüğü gibidir. Uçuş yolundaki görüntüler ise, Görev Planlayıcısı (Mission Planner) ile takip edilmektedir. Kullanılan bu program, Görev Planlayıcısı, ArduPilot için bir yer kontrol istasyonudur. Kayıtlı uçuşları gözden geçirmenin yanı sıra kurulum ve uçuş desteği sağlamaktadır. Pilin İHA'lara etkisi, Uçağı, gereken elektrik enerjisini sağlayan ve FPV sistemini ve tüm ekipmanı sağlayan ana güç kaynağıdır. Li-Po (Lityum Polimer) piller genellikle tercih edilir. Li-Po Bataryaya Lityum İyon polimer batarya denir. Lityum-iyon, bataryanın daha gelişmiş bir versiyonudur ve aynı özelliklere sahiptir. Bu pillerin hafıza etkisi yoktur ve kullanılmadıkları zaman enerji kayıpları yavaşlamaktadır. Doğru kullanılmazsa çok tehlikeli olabilmektedir. Özel şarj cihazlarıyla şarj edilmeleri gerekmektedir. Lipo piller hücrelerden oluşmaktadır. Tek bir hücrenin boş hali 3.7V dolu hali ise 4.2V olmalıdır. Lipo pillerin hücreleri seri olarak (S) veya paralel olarak (P) bağlı olabilmektedir.



Şekil 5.61. 4 pervaneli (Quadcopter)

5.2.1. Birinci test ortamı

Gerçek zamanlı birinci test ortamında insansız hava aracı uçuş örneği Şekil 5.62'de görüldüğü gibidir. İki tane engel var, her engel farklı yüksekliğe sahiptir. Birinci engel yüksekliği 30m ve ikinci engel yüksekliği ise 40m, başlangıç noktasından hedef noktasına yolları hesaplanmaktadır.



Şekil 5.62. Birinci İHA test ortamı

İHA test ortamının sonuçları, çizelge 5.7'de görülmektedir. İnsansız hava araçlarının uçuş yolundaki görüntüler Şekil 5.63, Şekil 5.65 ve Şekil 5.67'da görülmektedir.

Çizelge 5.7. Farklı Yüksekliklerde enerji tüketimi

Yollar	Drone Yüksekliği(m)	Drone Ağırlığı(kg)	Drone Hızı(m/s)	Batarya harcaması(v)	Pil tüketimi(%)
Birinci Yol	25	3.9	5	16.45-16.1	2.08
İkinci Yol	35	3.9	5	16.45-16.1	2.08
Üçüncü Yol	45	3.9	5	16.45-16.15	1.78

Pil Tüketimin yüzdesi aşağıda bu fonksiyonla hesaplanmaktadır:

$$\text{Pil yüzdesi} = \frac{V}{V_{max}} . 100 \quad (5.1)$$

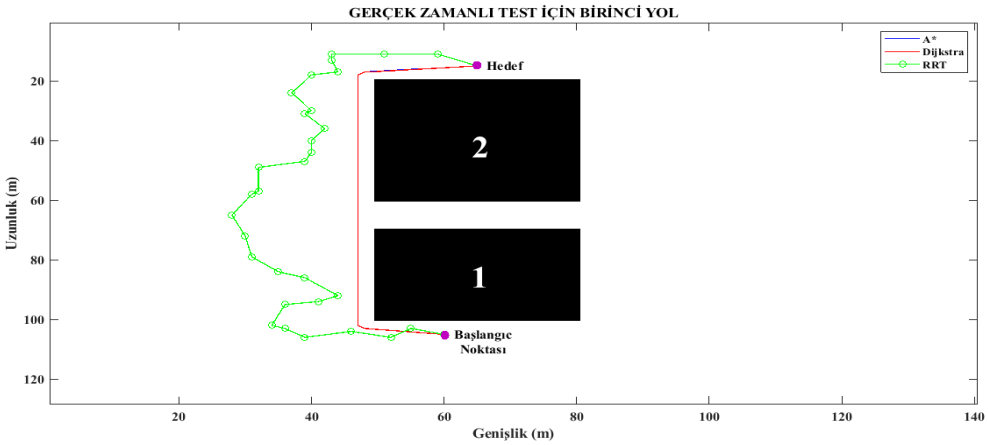
Drone yüksekliğ 45m olan üçmadan önce 16.45 v olan üçmadan sonra 16.1 v hesaplama yaparken 97.91% - 95.83= 2.08% pil harcaması olduğunu göstermektedir. Batarya dolu olduğunda 16.8 V'a eşit ve batarya boş olduğunda 13.2 V'a eşittir. Sonuçlara göre İHA ne kadar yüksek olursa, o kadar fazla batarya harcanmaktadır.

Simülasyon ortamında yapılan üç yol için deneme

- Birinci yol: Drone yüksekliği 25 m olduğunda önerilen algoritmalar çalıştırıldığında iki engelin etrafından geçmektedir. Drone yüksekliği iki engelin yüksekliğinden daha düşüktür. Şekil 5.64 görüldüğü gibi başlangıç noktasından hedefe yol çizdildiğini görülmektedir.



Şekil 5.63. Birinci Yol için Drone Yüksekliği 25m

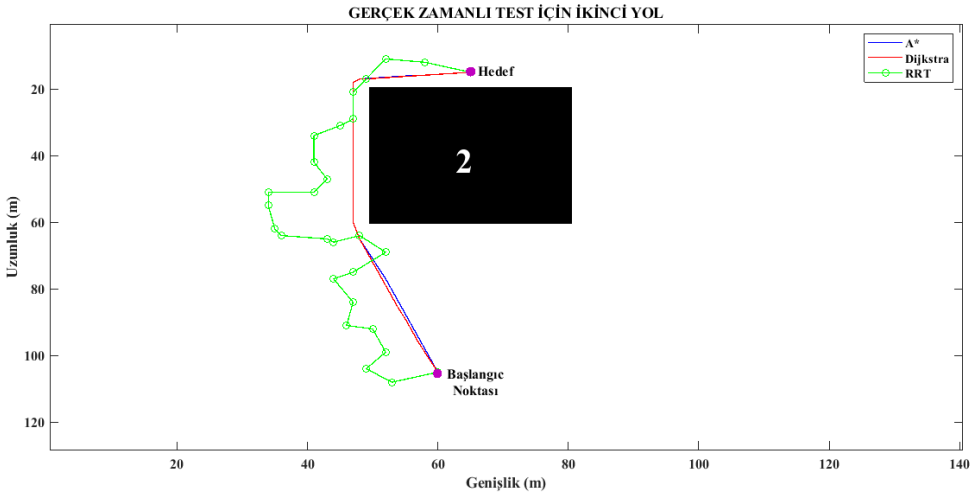


Şekil 5.64. Birinci test için birinci yol

- İkinci yol: Drone yüksekliği 35m olduğunda, önerilen algoritmalar çalıştırıldığında birinci engeli 30m yüksekliği olan görmezden gelmektedir ve ikinci engelin etrafından geçmektedir çünkü drone yükseliği birinci engelin yüksekliğinden daha yüksektir. Çizdiği yol şekil 5.66 de görülmektedir.

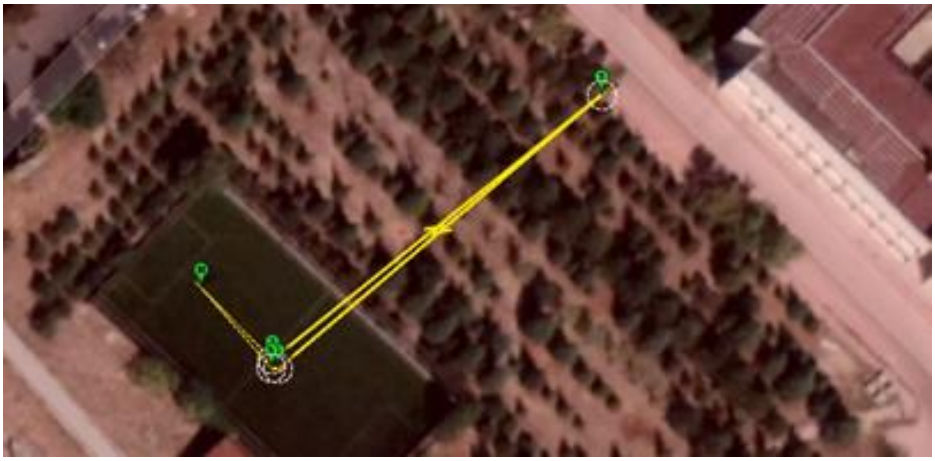


Şekil 5.65. İkinci Yol için Drone Yüksekliği 35m

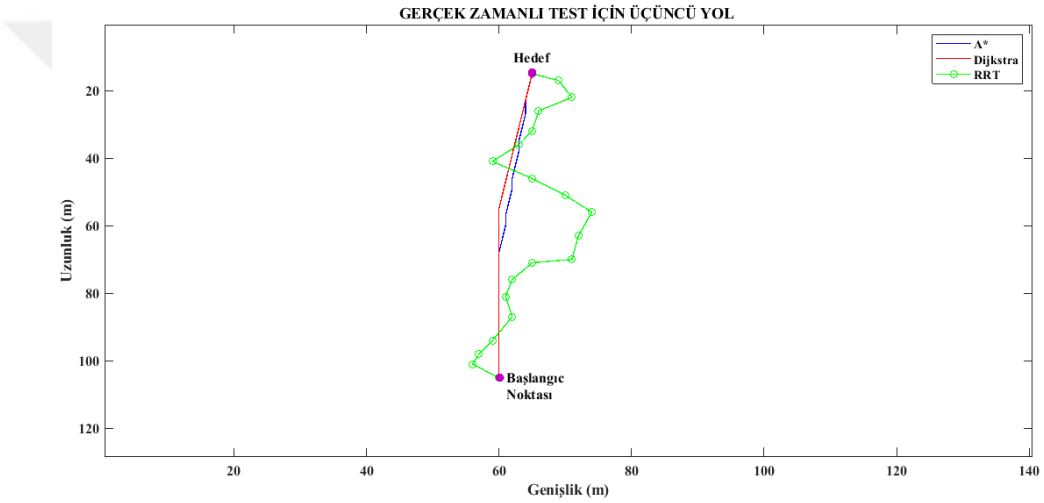


Şekil 5.66. Birinci test için ikinci yol

- Üçüncü Yol: Drone yüksekliği 45m olduğunda, iki engele verilen yükseklikten daha fazla bunun neticesinde önerilen algoritmalar çaliştirildiğinda iki engeli görmezden gelmektedir. Sonuc olarak Başlangıc noktasından hedef noktasına engeller olmadan hesaplamaktadır. Çizdiği yolu şekil 5.68 da görülmektedir.



Şekil 5.67. Üçüncü yol için Drone Yüksekliği 45m



Şekil 5.68. Birinci test için üçüncü yol

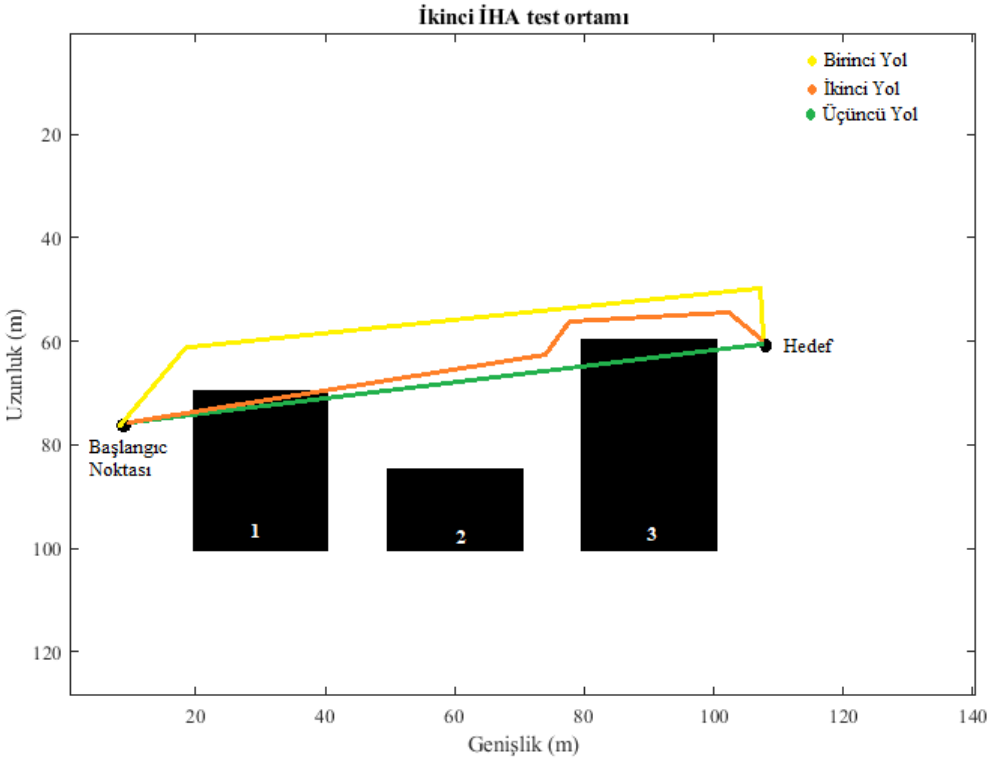
Simülasyon ortamın sonucları çizelge 5.8'de görülmektedir.

Çizelge 5.8. Birinci ortam için üç boyutlu simülasyon sonucu

Üç boyutlu simülasyon sonucu	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)
Birinci Yol	1.2760	92	1.2024	92	3.1303	33
İkinci Yol	0.3254	54	1.3230	55	4.1530	29
Üçüncü Yol	0.0787	56	1.2992	56	2.3733	23

5.2.2. İkinci test ortamı

Gerçek zamanlı ikinci test ortamında insansız hava aracı uçuş örneği şekil 5.69'de görülmektedir. Üç tane engel var, her engel farklı yüksekliğe sahiptir. Birinci engel yüksekliği 30m, ikinci engel 15m diğerise 40m, başlangıc Noktasından hedef noktasına yolları hesaplanmaktadır.



Şekil 5.69. İkinci İHA test ortamı

İHA test ortamının sonuçları, çizelge 5.9' da görülmektedir. İnsansız hava araçlarının uçuş yolundaki görüntüler Şekil 5.70, Şekil 5.72 ve Şekil 5.74'de görülmektedir.

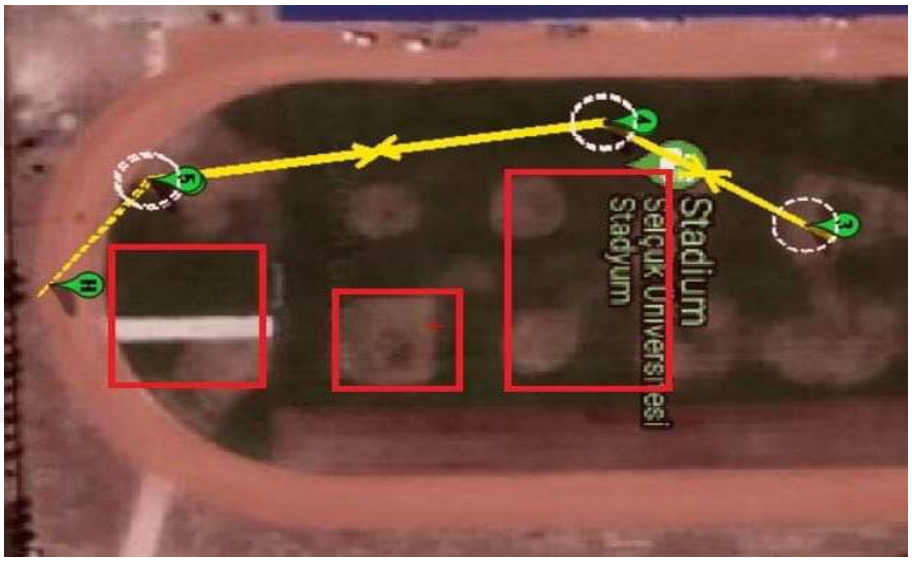
Çizelge 5.9. Farklı Yüksekliklerde enerji tüketimi

Yollar	Drone Yüksekliği(m)	Mesafe(m)	Drone Hızı(m/s)	Batarya harcaması(v)	Pil tüketimi(%)
Birinci Yol	20	264	5	3400	14.430184
İkinci Yol	35	249	5	3400	15.464194
Üçüncü Yol	45	264	5	3400	15.785886

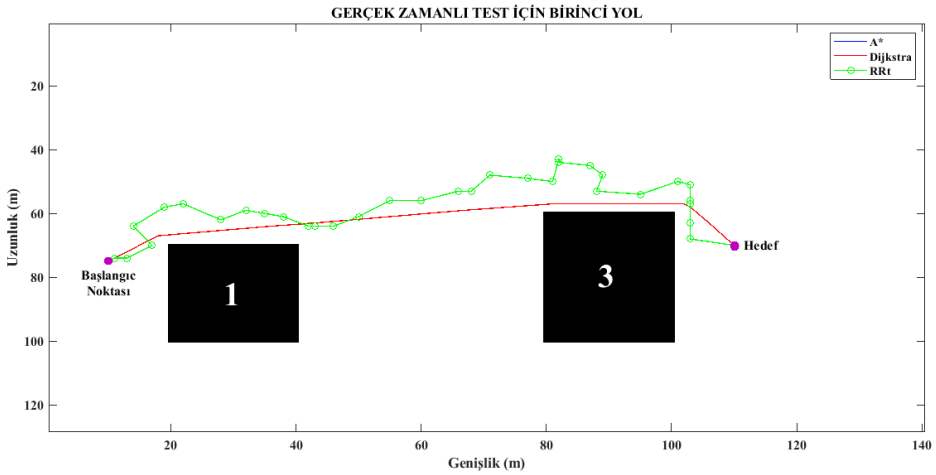
Batarya dolu olduğunda 16.8 V'a eşit ve batarya boş olduğunda 13.2 V'a eşittir. Sonuçlara göre İHA ne kadar yüksek olursa, o kadar fazla batarya harcanmaktadır.

Simülasyon ortamında yapılan üç yol için deneme

- Birinci yol: Drone yüksekliği 20m olduğunda önerilen algoritmalar çalıştırıldığında ikinci engeli görmezden gelmektedir. Drone yüksekliği birinci engelin yüksekliğinden daha yüksektir. Şekil 5.71 görüldüğü gibi başlangıç noktasından hedefe yol çizdildiğini görülmektedir.

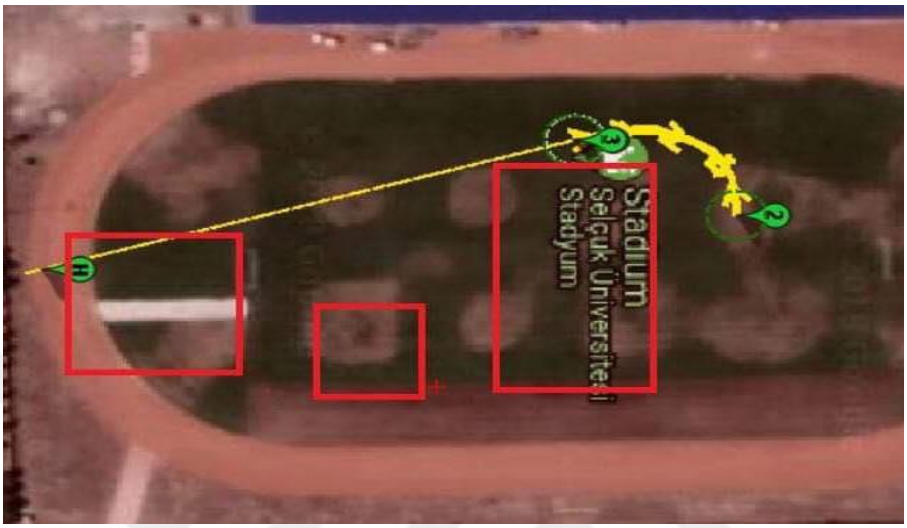


Şekil 5.70. Birinci yol için drone yüksekliği 20m

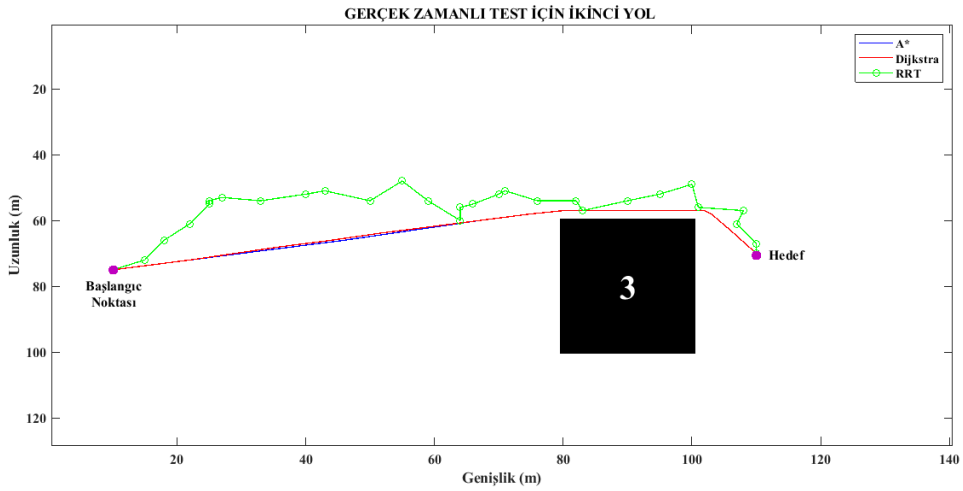


Şekil 5.71. İkinci test için birinci yol

- İkinci yol: Drone yüksekliği 35m olduğunda, önerilen algoritmalar çalıştırıldığında birinci engeli 30m yüksekliği ve ikinci engel 15m olan görmezden gelmektedir çünkü drone yükseliği birinci ve ikinci engelden daha yüksektir. Çizdiği yol Şekil 5.73 de görülmektedir.

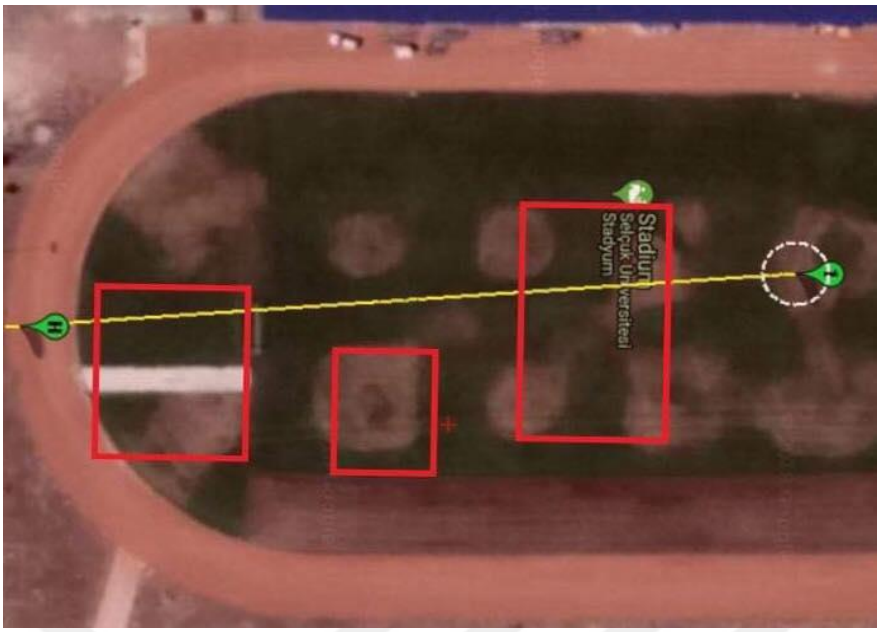


Şekil 5.72. İkinci yol için Drone Yüksekliği 35m

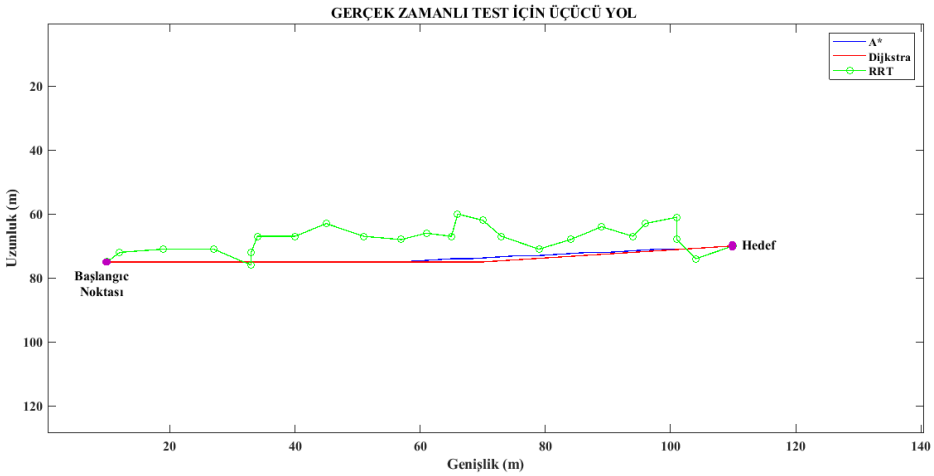


Şekil 5.73. İkinci test için ikinci yol

- Üçüncü Yol: Drone yüksekliği 45m olduğunda, üç engele verilen yüksekliğinden daha fazla bunun neticesinde önerilen algoritmalar çalıştırıldığında üç engeli görmezden gelmektedir. Sonuc olarak Başlangıc noktasından hedef noktasına engeller olmadan hesaplamaktadır. Çizdiği yolu şekil 5.75 de görülmektedir.



Şekil 5.74. üçüncü yol için drone yüksekliği 45m



Şekil 5.75. İkinci test için üçüncü yol

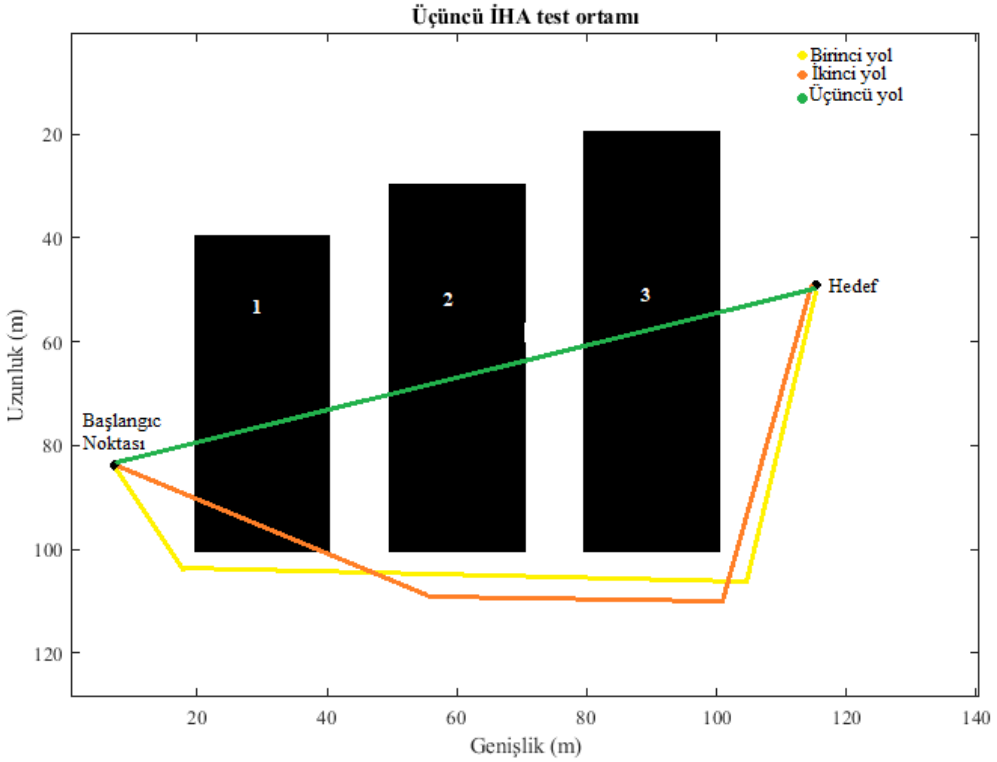
Simülasyon ortamın sonucları çizelge 5.10'da görülmektedir

Çizelge 5.10. İkinci ortam için üç boyutlu simülasyon sonucu

Üç boyutlu simülasyon sonucu	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)
Birinci Yol	0.2245	43	0.9637	43	5.3549	35
İkinci Yol	0.3000	36	1.0773	36	3.1755	29
Üçüncü Yol	0.0930	66	1.1265	66	2.0916	25

5.2.3. Üçüncü Test Ortamı

Gerçek zamanlı üçüncü test ortamında insansız hava aracı uçuş örneği şekil 5.76'da görülmektedir. Üç tane engel var, her engel farklı yüksekliğe sahiptir. Birinci engel yüksekliği 60m, ikinci engel 70m dve üçüncü engel ise 80m dır, başlangıc Noktasından hedef noktasına yolları hesaplanmaktadır.



Şekil 5.76. İHA test ortamı

İHA test ortamının sonuçları, çizelge 5.11'da görülmektedir. İnsansız hava taşıtlarının uçuş yolundaki görüntüler, Şekil 5.77, Şekil 5.79 ve Şekil 5.81'da görülmektedir.

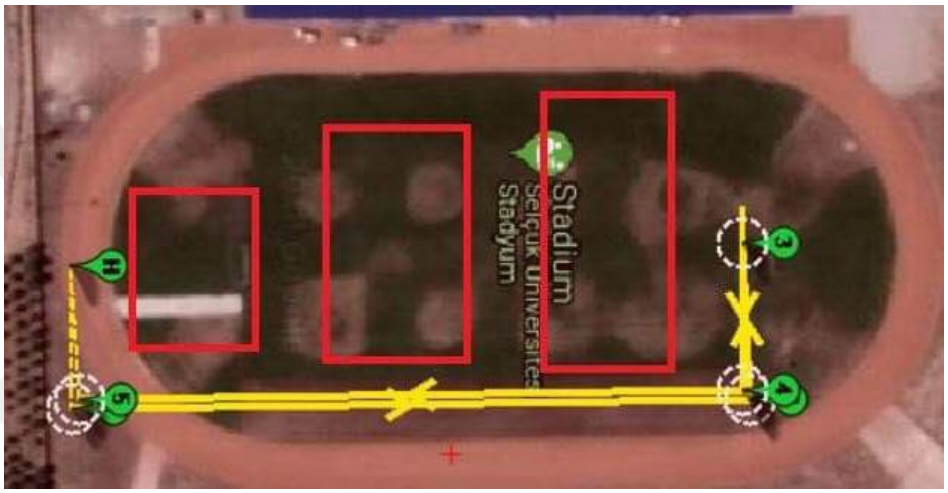
Çizelge 5.11. Farklı Yüksekliklerde enerji tüketimi

Yollar	Drone Yüksekliği(m)	Mesafe(m)	Drone Hızı(m/s)	Batarya harcaması(v)	Pil tüketimi(%)
Birinci Yol	50	385	5	3400	23.43756
İkinci Yol	65	387	5	3400	25.252822
Üçüncü Yol	85	272	5	3400	22.265682

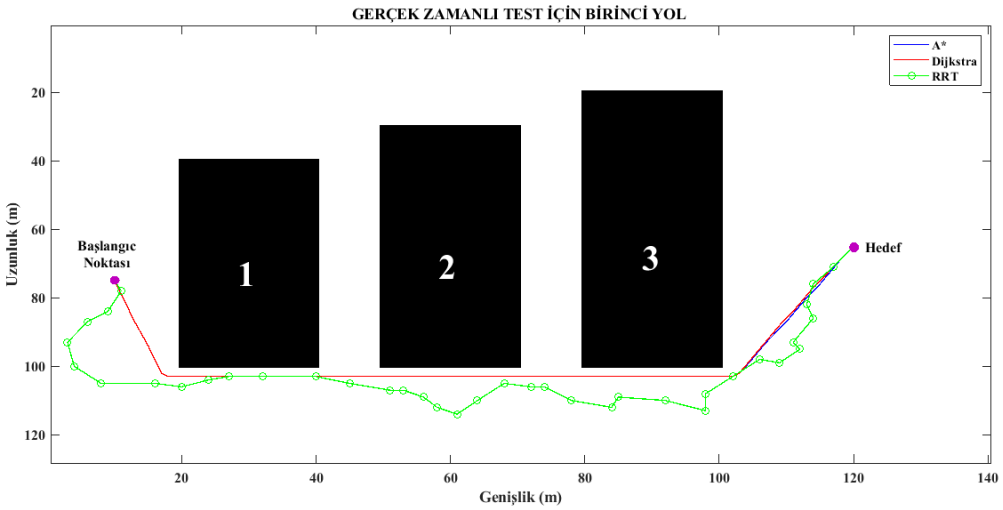
Batarya dolu olduğunda 16.8 V'a eşit ve batarya boş olduğunda 13.2 V'a eşittir. Sonuçlara göre İHA ne kadar yüksek olursa, o kadar fazla batarya harcanmaktadır.

Simülasyon ortamında yapılan üç yol için deneme

- Birinci yol: Drone verilen yükseklik üç engelin yüksekliğinden daha düşük olduğundan dolayı Drone yüksekliği 50m olduğunda önerilen algoritmalar çalıştırıldığında üç engelin etrafından geçmektedir. Şekil 5.78 de görüldüğü gibi başlangıç noktasından hedefe yol çizdildiğini görülmektedir.



Şekil 5.77. Üçüncü yol için drone yüksekliği 50m

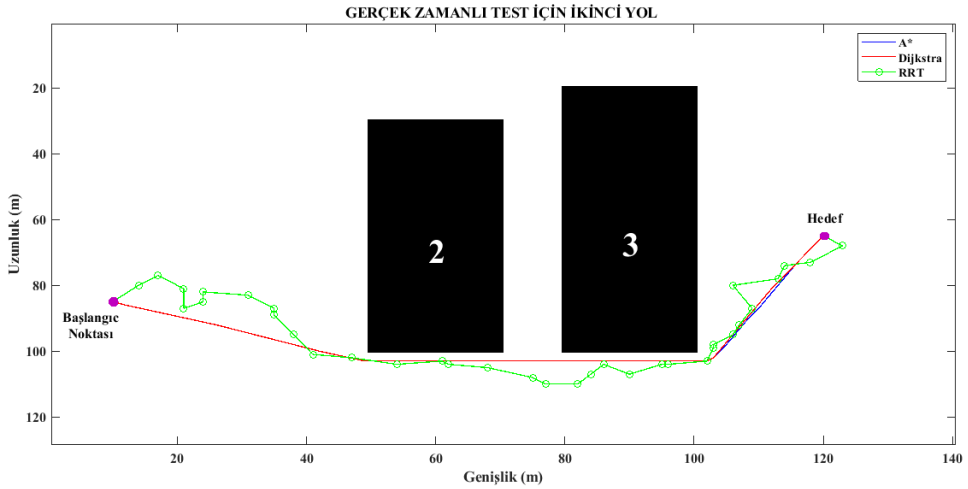


Şekil 5.78. Üçüncü test için birinci yol

- İkinci yol: Drone yüksekliği 60m olduğunda önerilen algoritmalar çalıştırıldığında birinci görmezden gelmektedir. Drone yüksekliği birinci engelin yüksekliğinden daha yüksektir. Şekil 5.80 de görüldüğü gibi başlangıç noktasından hedefe yol çizdildiğini görülmektedir

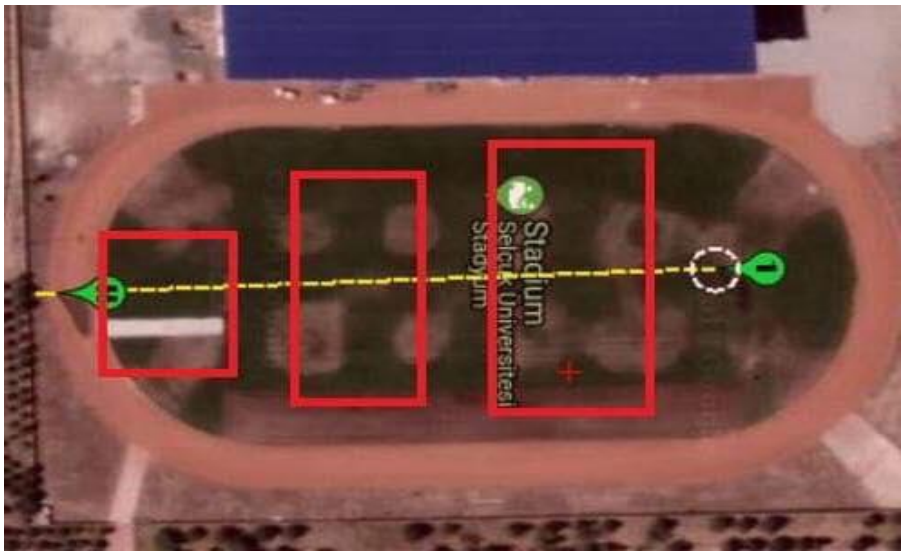


Şekil 5.79. İkinci yol için drone yüksekliği 65m

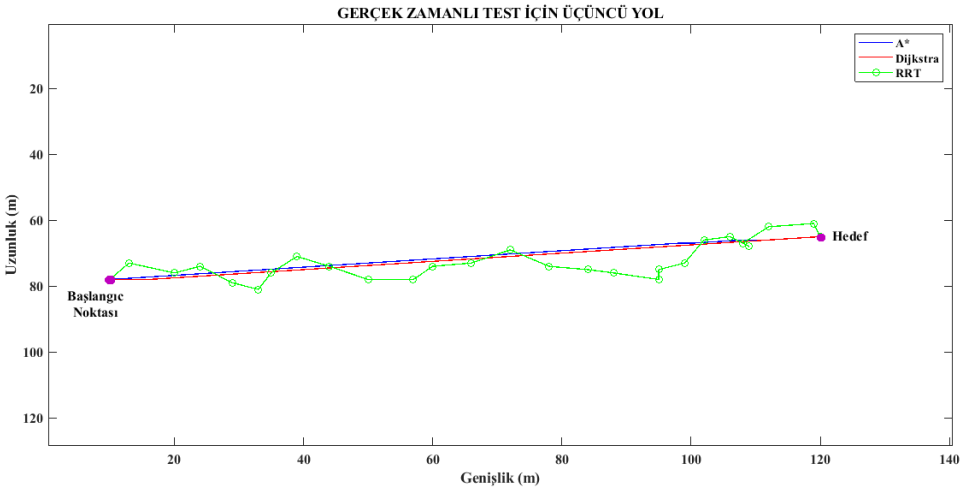


Şekil 5.80. Üçüncü test için ikinci yol

- Üçüncü Yol: Drone yüksekliği 85m olduğunda önerilen algoritmalar çalıştırıldığında üç engeli görmezden gelmektedir. Drone yüksekliği üç engelin yüksekliğinden daha yüksektir. Şekil 5.82 de görüldüğü gibi başlangıç noktasından hedefe yol çizdildiğini görülmektedir



Şekil 5.81. Üçüncü yol için yol için drone yüksekliği 85m



Şekil 4.82. Üçüncü test için üçüncü yol

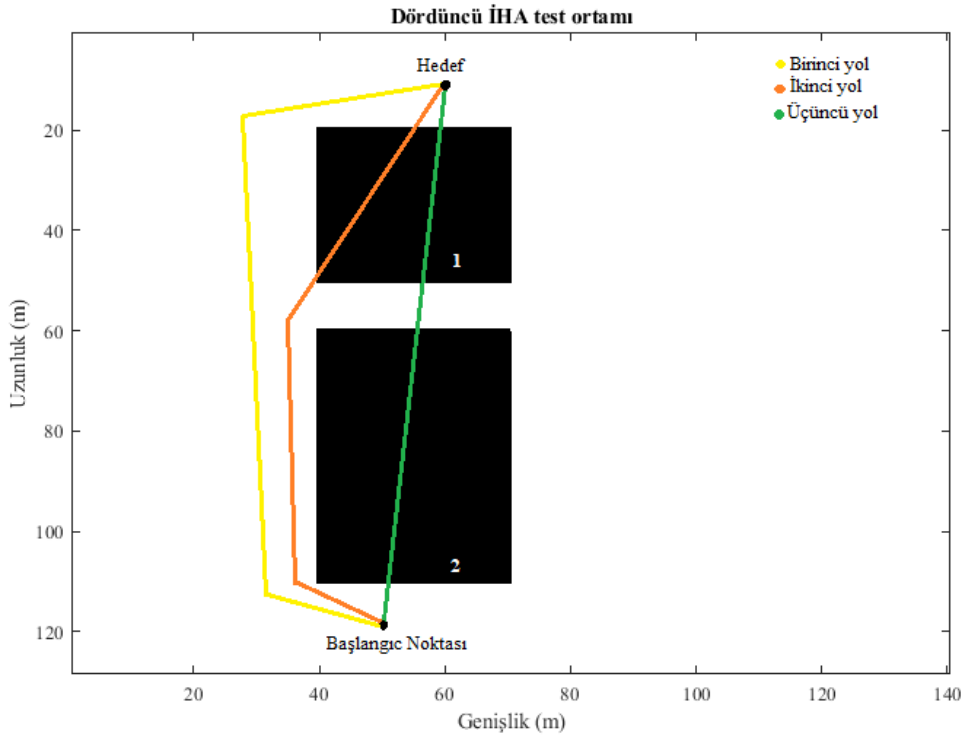
Simülasyon ortamın sonucları çizelge 4.12'de görülmektedir.

Çizelge 4.12. Üçüncü ortam için üç boyutlu simülasyon sonucu

Üç boyutlu simülasyon sonucu	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)
Birinci Yol	0.3115	98	0.8691	97	7.5799	39
İkinci Yol	0.3553	68	0.9681	66	11.3144	37
Üçüncü Yol	0.0698	20	1.2715	20	3.0597	27

5.2.4. Dördüncü test ortamı

Gerçek zamanlı dördüncü test ortamında insansız hava aracı uçuş örneği şekil 5.83'da görülmektedir. İki tane engel var, her engel farklı yüksekliğe sahiptir. İlk engel yüksekliği 90m, diğer engel ise 50m, başlangıc Noktasından hedef noktasına yolları hesaplanmaktadır.



Şekil 5.83. İHA test ortamı

İHA test ortamının sonuçları, Çizelge 5.13'de görülmektedir. İnsansız hava taşıtlarının uçuş yolundaki görüntüler, Şekil 5.84, Şekil 5.86 ve Şekil 5.88'da görülmektedir.

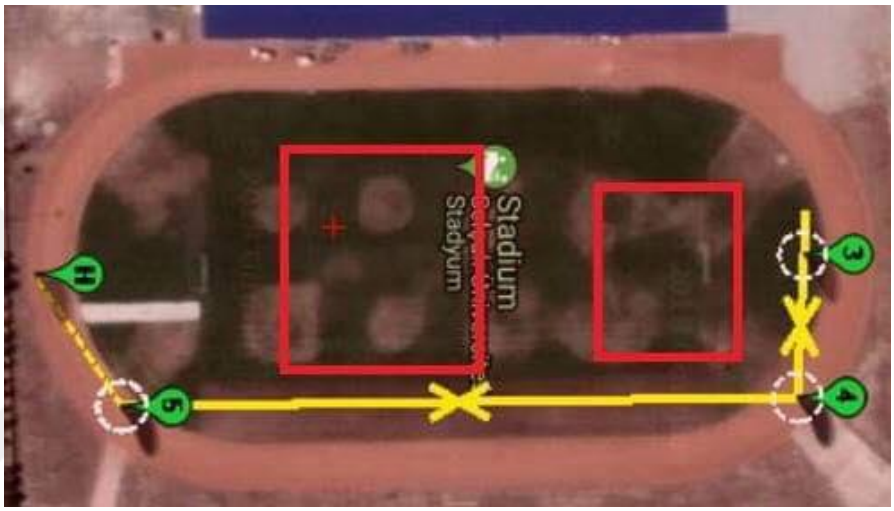
Çizelge 4.13. Farklı Yüksekliklerde enerji tüketimi

Yollar	Drone Yüksekliği(m)	Mesafe(m)	Drone Hızı(m/s)	Batarya harcaması(v)	Pil tüketimi(%)
Birinci Yol	50	410	5	3400	25.16091
İkinci Yol	70	372	5	3400	25.137932
Üçüncü Yol	100	321	5	3400	26.240876

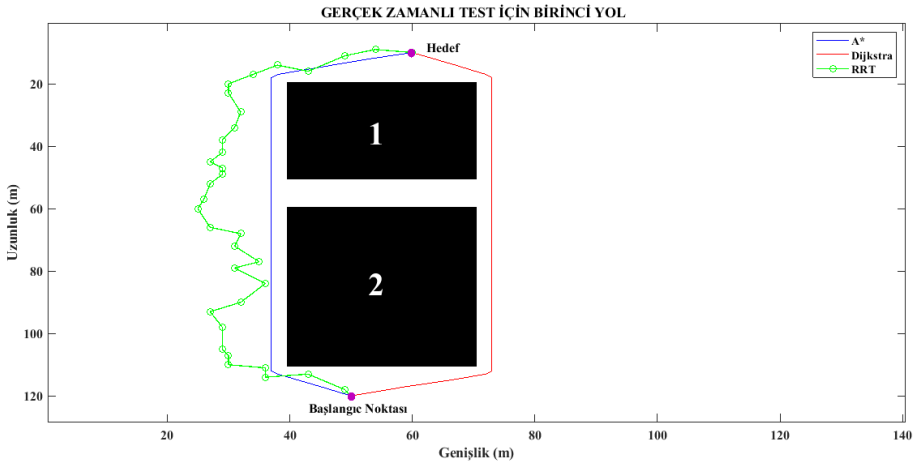
Batarya dolu olduğunda 16.8 V'a eşit ve batarya boş olduğunda 13.2 V'a eşittir. Sonuçlara göre İHA ne kadar yüksek olursa, o kadar fazla batarya harcanmaktadır.

Simülasyon ortamında yapılan üç yol için deneme

- Birinci yol: Drone yüksekliği 50m olduğunda, iki engelin etrafından geçmektedir çünkü drone'a verilen yükseklik iki engelin yüksekliğinden daha düşüktür. Çizdiği yol şekil 4.85'de görülmektedir.



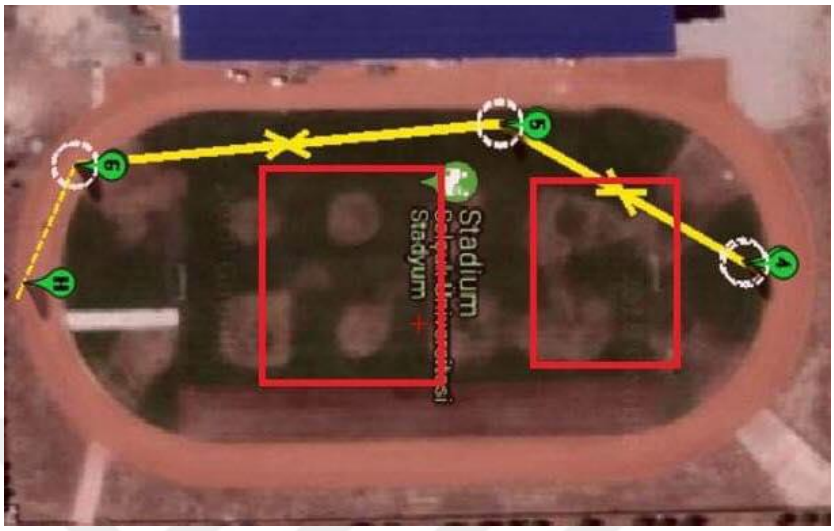
Şekil 5.84. Birinci yol için drone yüksekliği 50m



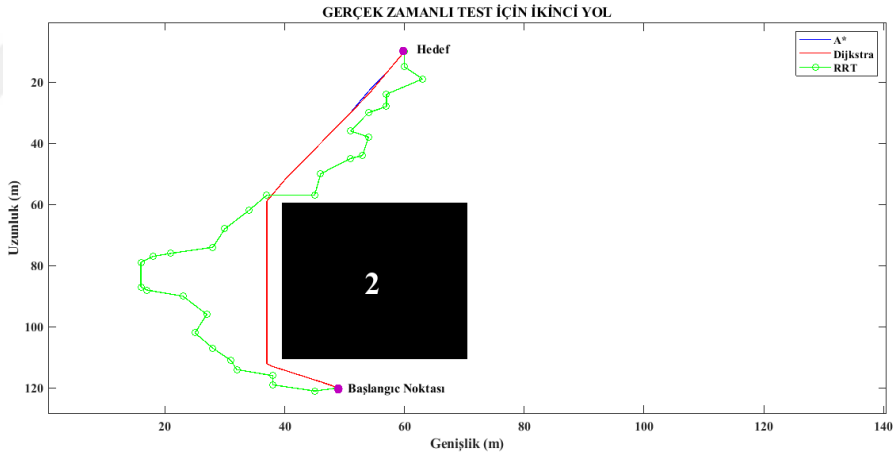
Şekil 4.85. Dördüncü test için birinci yol

- İkinci yol: Drone yüksekliği 70m, algoritma çalıştığında birinci engeli 50m yüksekliği olan görmezden gelmektedir çünkü drone yükseliği birinci engelin

yüksekliğinden daha yüksektir. Şekil 5.87 de görüldüğü gibi yol çizdiğini görülmektedir.



Şekil 5.86. İkinci yol için drone yüksekliği 70m

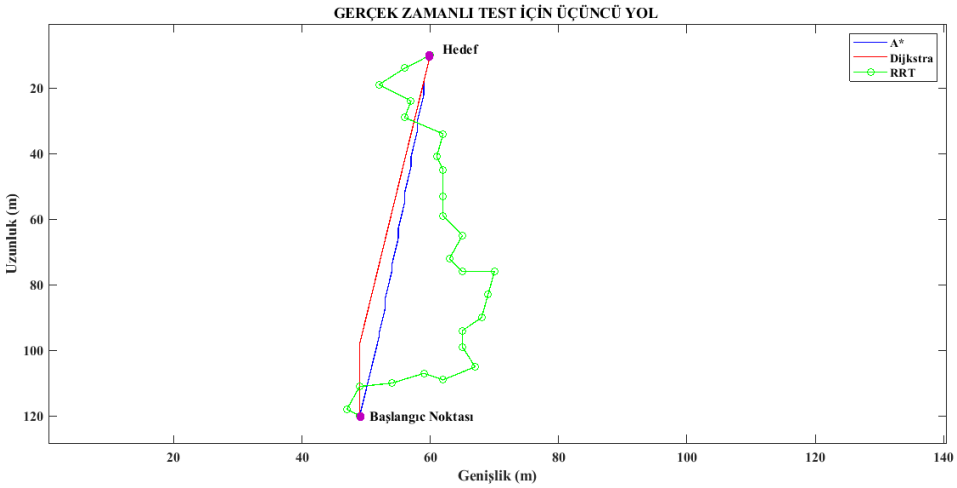


Şekil 4.87. Dördüncü test için ikinci yol

- Üçüncü Yol: Drone yüksekliği 100m olduğunda, iki engele verilen yükseklikten daha fazla bunun neticesinde önerilen algoritmalar çaliştiğinda iki engeli görmezden gelmektedir. Sonuc olarak Başlangıç noktasından hedef noktasına engeller olmadan hesaplamaktadır. Çizdiği yolu şekil 5.89 de görülmektedir.



Şekil 5.88. Üçüncü yol için drone yüksekliği 100m



Şekil 5.89. Dördüncü test için üçüncü yol

Simülasyon ortamın sonucları aşağıdaki çizelge 5.14'de görülmektedir.

Çizelge 5.14. Üçüncü ortam için üç boyutlu simülasyon sonucu

Üç boyutlu simülasyon sonucu	A* algoritması		Dijkstra algoritması		RRT algoritması	
	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)	Zaman (sn)	Yol uzunluğu (m)
Birinci Yol	0.3864	103	1.2077	104	11.7482	35
İkinci Yol	0.0892	68	1.2047	67	2.1202	31
Üçüncü Yol	0.1092	34	1.3891	34	8.0192	25

6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

Üç test ortamının sonuçları, çizelge 6.1'de görülmektedir.

Çizelge 6.1. 2B&3B yol planlama algoritmalarının zamana (sn) ve yol uzunluğuna (m) göre performansların karşılaştırılması

Uygulama sayısı	Şekiller	A* algoritması		Dijkstra algoritması		RRT algoritması	
		Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)	Zaman (sn)	Yol uzunluğu(m)
Ortam 1	5.2 (2B)	0.3179	48	0.8977	48	12.4857	44
	5.11 (3B)	0.1144	42	1.1677	42	3.6969	37
Ortam 2	5.24 (2B)	0.2687	71	1.0515	71	12.1760	43
	5.39 (3B)	0.3241	25	1.3167	25	6.5733	37
Ortam 3	5.42 (2B)	0.1256	22	1.0446	22	4.1713	37
	5.56 (3B)	0.0555	19	1.4563	19	6.4095	33

Üçüncü test simülasyon sonucunu incelendiğinde, A* algoritmasında 22m'lik yolu 0.1256 saniyede, 19m'lik yolu ise 0.0555 saniyede gitmiştir. Dijkstra algoritmasında 22m'lik yolu 1.0446 saniyede, 19m'lik ise yolu 1.4563 saniyede gitmiştir. RRT algoritmasında 37m'lik yolu 4.1713 saniyede, 33m'lik yolu ise 6.4095 saniyede gittiği görülmektedir. A* algoritması, Dijkstra algoritması ve RRT algoritmasının sonuçlarını karşılaştırdığımızda, her iki deneysel ortamda da farklı sonuçlar vermektedir. 2B ve 3B ortamları karşılaştırarak, önerdiğimiz üç algoritmadan her biri 3B ortamında daha iyi sonuçlar vermektedir. Sonuç olarak, A* algoritması diğer iki algoritmadan daha iyi performans göstermiştir. A* , RRT ve Dijkstra algoritmalarını karşılaştırdığımızda, Dijkstra ve RRT algoritmalarının fazla zaman harcadiği ve A* algoritmasının daha az zamanda sonuca vardiği görülmüştür. Bu durum A* algoritması alanı sadece hedefe doğru taradığı, RRT algoritmasının her simülasyonda hedefe doğru farklı noktalar belirleyerek yol çizdiği, Dijkstra algoritmasının ise çok daha geniş alanı tarayarak hedefe vardığı görülmüştür.

Avantajlar ve dezavantajlar

- A*, Dijkstra'nın algoritmasına göre daha hızlıdır.
- Dijkstra'lar A* ile kıyaslandığında basittir.
- Dijkstra'nın algoritmasının en büyük dezavantajı, gerekli kaynakların çok fazla zamanını boşa harcayarak orada kör bir araştırma yapmasıdır.
- Dezavantajlarından birisi de negatif kenarları kaldıramamasıdır

Dijkstra, esasen A* ile aynıdır, sezgisel değildir (H her zaman 0'dır). Sezgisel olmadığı için, her yöne eşit olarak genişleyerek arama yapar, ancak A* alanı yalnızca varış yönünde taramaktadır. Tahmin edebileceğimiz gibi, bu Dijkstra'nın nedeni genellikle hedef bulunmadan önce çok daha geniş bir alanı keşfetmekle sonuçlanmaktadır. Genellikle onu A* 'dan daha yavaş yapmaktadır. Ancak her ikisinin de kendi önemi vardır, örneğin A* çoğunlukla hem kaynağı hem de varış yerini bildiğimizde ve Dijkstra'lar hedef varış yerimizin nerede olduğunu bilmediğimizde kullanılmaktadır. Diyelim ki bir çeşit kaynak bulmaya ihtiyaç duyan bir kaynak toplama biriminiz var. Birkaç kaynak alanının nerede olduğunu biliyor olabilir, ancak en yakınına gitmek istiyor. Burada, Dijkstra'nın A* dan daha iyi olması nedeniyle hangisinin en yakın olduğunu bilmiyoruz. Tek alternatifimiz, her birine olan mesafeyi bulmak için A* tuşlarını tekrar tekrar kullanmak ve ardından bu yolu seçmektir. Aradığımız bir konum türünü bildiğimiz, en yakın olanı bulmak istediğimiz sayısız benzer durumlar vardır. Ancak nerede olduğunu veya hangisinin en yakın olabileceğini bilmiyorum. Dolayısıyla, hem başlangıç noktası hem de varış noktası bildiğimizde A* daha iyidir. Eğer kabul edilebilir bir sezgisel işlev kullanıyorsanız, A* hem tamamlanmış (varsa bir yol bulur) hem de en uygun (her zaman en kısa yolu bulmaktadır).

Yapılan SİMÜLASYON sonuclarını ve havada yapılan uçuş testini karşılaştırarak sonuçları çizelge 6.2 de görülmektedir

Çizelge 6.2. 2B ve 3B gerçek zamanlı test ve İHA gerçek zamanlı uçuş sonuclarının karşılaştırması

Uygulamalar	Şekiller	A* Algoritması		Dijkstra Algoritması		RRT Algoritması	
		Simülasyon zaman (s)	pil tüketimi (%)	Simülasyon zaman (s)	pil tüketimi (%)	Simülasyon zaman (s)	pil tüketimi (%)
Uygulama (1)	5.2 (2B)	0.3179	0.0165	0.8977	0.0466	12.4857	0.6492
	5.11 (3B)	0.1144	0.0059	1.1677	0.0607	3.6969	0.1922
Uygulama (2)	5.24 (2B)	0.2687	0.0279	1.0515	0.1093	12.1760	1.2663
	5.39 (3B)	0.3241	0.0168	1.3167	0.0684	6.5733	0.3418
Uygulama (3)	5.42 (2B)	0.1256	0.0130	1.0446	0.1086	4.1713	0.4338
	5.56 (3B)	0.0555	0.0028	1.4563	0.0757	6.4095	0.3332

Simülasyon sonucunu ve yapılan uçuş test sonucunu karşılaştırması aşağıdaki denklemden sonuçlanmakadır:

$$X = \frac{saniye * hız * pil\ yüzdesi}{dronun\ uçtuğu\ mesafe} \quad (6.1)$$

X: belli bir yükseklikte tüketiği pil.

Karşılaştırma sonucuna göre İHA test ortamı daha iyi performanslar göstermektedir. Üçüncü uygulama örneğini 3B ortamini incelediğimizde, A* algoritmasına göre, simülasyon ortamında belli bir yükseklikte başlangıc noktasından hedef noktaya 0.0555s yetiştiğini ve 0.0028% pil tüketiğini göstermektedir. Dijkstra algoritmasına göre, simülasyon ortamında belli bir yükseklikte başlangıc noktasından hedef noktaya 1.4563s yetiştiğini ve 0.0757% pil tüketiğini görülmektedir. RRT algoritmasına göre, simülasyon ortamında belli bir yükseklikte başlangıç noktasından hedef noktaya 6.4095s ve 0.3332% pil tüketiğini görülmektedir.

6.2 Öneriler

Bu tezin amacı İnsansız Hava Araçları (İHA) için en kısa yol ve engeller de dahil olmak üzere ortamdaki iki boyutlu (2B) yol planlama algoritmalarıyla hesaplanmaktır. Böylece robotlar en kısa sürede çevrede görevlerini yapabilmektir.

Bu tezin amacı engellerden kaçınmak ve hedef noktaya en kısa yolu bulmaktır. Engelli bir ortamda, iki ve üç boyutlu yol planlama algoritmaları, en kısa yolu ve çevre robotlarını görevlerini olabildiğince hızlı bir şekilde yerine getirmek için tasarlanmıştır. Farklı koşulların ve gereksinimlerin değerlendirildiği optimum yol planlaması için, çeşitli iki boyutlu (2B) yöntemler önerilmiştir. Önerdiğimiz Dijkstra, RRT ve A* algoritma yöntemleri, özerk mobil robotlarında en yaygın şekilde kullanılmaktadır. Dijkstra algoritması iki düğüm arasındaki en kısa mesafeyi aç gözlü yaklaşımla bulurken, A* algoritması ise en kısa mesafeyi sezgisel yaklaşım kullanarak bulmaktadır. RRT algoritması ise rastgele noktalar seçerek dallanmalar yapar, dallanmalara göre hedefe gidilebilecek en uygun rotayı oluşturmaktır. Bu tezde bilinen 2B yöntemlere dayalı 3B yol planlama algoritmalarının geliştirilmesi ve performans analizinin yapılması amaçlanmıştır.

Üç farklı yöntem kullanarak en uzun mesafeyi en kısa sürede alınması hedeflenmiştir. Simülasyon ortamı, İHA'ların önerilen algoritmalar kullanarak (A* , Dijkstra ve RRT algoritmaları) yol planlama ve hedefe varış zamanını değerlendirmek üzere oluşturulmuştur. Yapılan çalışma sonucunda RRT algoritması A* algoritmasına göre hedef için daha kısa yolu belirlemesine rağmen A* algoritması daha hızlı sürede hedefe vardığı görülmüştür. Belirlenen algoritmalardan farklı “ Genetik Algoritma, YSA, Bulanık mantık “ gibi yöntemlerde kullanılabilir.

KAYNAKLAR

Ahmed T Sadiq, Firas A Raheem ve Noor Alhuda F. Abbas, 2017, Robot Arm Path Planning Using Modified Particle Swarm Optimization based on D* algorithm, *Al-Khwarizmi Engineering Journal*, 13 (3), 27-37.

Akshay Kumar Guruji, Himansh Agarwal ve DK Parsediya, 2016, Time-efficient A* algorithm for robot path planning, *Procedia Technology*, 23, 144-149.

Alaa A Ahmed, Turki Y Abdalla ve Ali A Abed, 2015, Path planning of mobile robot by using modified optimized potential field method, *International Journal of Computer Applications*, 113 (4), 6-10.

Aleksandar Jevtic, Diego Andina, Aldo Jaimes, Jose Gomez ve Mo Jamshidi, 2010, Unmanned aerial vehicle route optimization using ant system algorithm, *2010 5th International Conference on System of Systems Engineering*, 1-6.

Anish Pandey ve Dayal R Parhi, 2017, Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm, *Defence Technology*, 13 (1), 47-58.

Asita Kumar Rath, D. R. P., Harish Chandra Das, Manoj Kumar Muni ve Priyadarshi Biplab Kumar, 2018, Analysis and use of fuzzy intelligent technique for navigation of humanoid robot in obstacle prone zone, *Defence Technology*, 14 (6), 677-682.

AT Ismail, Alaa Sheta ve Mohammed Al-Weshah, 2008, A mobile robot path planning using genetic algorithm in static environment, *Journal of Computer Science*, 4 (4), 341-344.

BK Patle, Dayal Parhi, Anne Jagadeesh ve OP Sahu, 2017, Real Time Navigation Approach for Mobile Robot, *JCP*, 12 (2), 135-142.

Bo Wang, Jianwei Bao, Li Zhang ve Qinghong Sheng, 2018, UAV autonomous path optimization simulation based on radar tracking prediction, *EURASIP Journal on Wireless Communications*, 2018 (1), 239.

Chelsea Sabo ve Kelly Cohen, 2012, Fuzzy logic unmanned air vehicle motion planning, *Advances in Fuzzy Systems*, 2012, 13.

Cheng Chen, Yuan-liang Zhang, Jun Gu ve Yong-qiang Liu, 2017, Path Planning for Robot Based on Modified Ant Colony Algorithm, *ELECTRICAL ENGINEERING AND AUTOMATION: Proceedings of the International Conference on Electrical Engineering and Automation (EEA2016)*, 813-819.

Daneshwar Sahu ve Anil Kumar Mishra, 2017, Mobile robot path planning by genetic algorithm with safety parameter, *International Journal of Engineering Science* 14723.

Dilek Funda Kurtuluş ve OzanTekin alp, 2010, İnsansız Hava Araçlarına bir Bakış, *%J SSM Gündemi Dergisi*, 2, 53-58.

Dong Zhuoning, Zhang Rulin, Chen Zongji ve Zhou Rui, 2010, Study on UAV path planning approach based on fuzzy virtual force, *Chinese Journal of Aeronautics*, 23 (3), 341-350.

Emrah Yiğit, Işıl Yazar ve T Hikmet Karakoç, İNSANSIZ HAVA ARAÇLARI (İHA)'NIN KAPSAMLI SINIFLANDIRMASI VE GELECEK PERSPEKTİFİ, *Sürdürülebilir Havacılık Araştırmaları Dergisi*, 3 (1), 10-19.

Fan-Hsun Tseng, Cho-Hsuan Lee, Li-Der Chou ve Han-Chieh Chao, 2017, Multi-Objective Genetic Algorithm for Civil UAV Path Planning Using 3G Communication Networks, 28 (6), 26-37.

Fei Yan, Yan Zhuang ve Jizhong Xiao, 2012, 3D PRM based real-time path planning for UAV in complex environment, *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1135-1140.

Felipe Leonardo Lôbo Medeiros ve José Demisio Simões Da Silva, 2010, A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation, *Brazilian Symposium on Artificial Intelligence*, 213-222.

František Duchoň, Andrej Babinec, Martin Kajan, Peter Beňo, Martin Florek, FicoTomáš ve Ladislav Jurišica, 2014, Path planning with modified a star algorithm for a mobile robot, *Procedia Engineering*, 96, 59-69.

Han Tong, 2012, Path planning of UAV based on voronoi diagram and DPSO, *Procedia Engineering*, 29, 4198-4203.

Hicran Yılmaz ve Şengül Doğan, İnsansız Hava Sistemleri için Güzergâh Optimizasyonu Route Optimization for Unmanned Aerial Systems.

Honglun Wang, Wentao Lyu, Peng Yao, Xiao Liang ve Chang Liu, 2015, Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system, *Chinese Journal of Aeronautics*, 28 (1), 229-239.

Hüseyin Şahin ve Tuğrul Oktay, 2019, Properties of Novel Tricopter And Comparision of Other Unmanned Aerial Vehicles, *%J Avrupa Bilim ve Teknoloji Dergisi* (16), 816-825.

I Kim, Saebyuk Shin, Juan Wu, Shin-Dug Kim ve Cheong-Ghil Kim, 2017, Obstacle avoidance path planning for uav using reinforcement learning under simulated environment, *IASER 3rd International Conference on Electronics, Electrical Engineering, Computer Science*, 34-36.

Iram Noreen, Amna Khan ve Zulfiqar Habib, 2016, A comparison of RRT, RRT* and RRT*-smart path planning algorithms, *International Journal of Computer Science Network Security*, 16 (10), 20.

Jayesh Amin, Jovan Boškovic ve Raman Mehra, 2006, A fast and efficient approach to path planning for unmanned vehicles, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 6103.

Jingang Cao, 2016, Robot global path planning based on an improved ant colony algorithm, *Journal of Computer Communications*, 4 (02), 11.

Jung Leng Foo, Jared Knutzon, James Oliver ve Eliot Winer, 2006, Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization, *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 6995.

Kai Yit Kok ve Parvathy Rajendran, 2016, Differential-evolution control parameter optimization for unmanned aerial vehicle path planning, *PloS one*, 11 (3), e0150558.

Kwangjin Yang ve Salah Sukkarieh, 2008, 3D smooth path planning for a UAV in cluttered natural environments, *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 794-800.

Lyle Parungao, Fabian Hein ve Wansu Lim, 2018, Dijkstra algorithm based intelligent path planning with topological map and wireless communication, *ARPN Journal of Engineering Applied Sciences*
13 (8), 2753-2763.

Mariusz Wzorek ve Patrick Doherty, 2006, Reconfigurable path planning for an autonomous unmanned aerial vehicle, *2006 International Conference on Hybrid Information Technology*, 242-249.

Mert Turanlı ve Emre Koyuncu, OTONOM MOBİL ROBOTLAR İÇİN ÇOK AMAÇLI BİR DENEME ORTAMININ TASARIMI: ITUKAL ROBOT TEST ORTAMI.

Miao Yong-Fei, Zhong Luo ve Xia Luo-Sheng, 2013, Application of improved sparse A* algorithm in UAV path planning, *Information Technology Journal*, 12 (17), 4058.

MILENA Karova, DANISLAV Zhelyazkov, MARIANA Todorova, IVAYLO Penev, VENTSISLAV Nikolov ve Vllıan Petkov, 2015, Path planning algorithm for mobile robot, *Proc. Int. Conf. Appl. Comput. Sci.*, 283-287.

Mohammad Mozaffari, Walid Saad, Mehdi Bennis ve Merouane Debbah, 2016, Optimal transport theory for power-efficient deployment of unmanned aerial vehicles, *2016 IEEE international conference on communications (ICC)*, 1-6.

Mohammadreza Radmanesh, Manish Kumar, Paul H Guentert ve Mohammad Sarim, 2018, Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study, *%J Unmanned systems*, 6 (02), 95-118.

Mustafa Dikmen, 2015, İnsansız Hava Aracı (İHA) Sistemlerinin Hava Hukuku Bakımından İncelenmesi, *Savunma Bilimleri Dergisi*, 14.

Nafiz Arıca, Halil Cicibaş ve Kadir Alpaslan Demir, 2012, İnsansız Hava Araçları için Çok Kriterli Güzergâh Planlama Modeli, *Journal of Defense Sciences/Savunma Bilmleri Dergisi*, 11 (1).

Nanyaporn Intaratep, William N Alexander, William J Devenport, Sheryl M Grace ve Amanda Dropkin, 2016, Experimental study of quadcopter acoustics and performance at static thrust conditions, *22nd AIAA/CEAS Aeroacoustics Conference*, 2873.

Osman İyibilgin, Yüksel Korkmaz ve Fehim Fındık, 2016, Geçmişten günümüze insansız hava araçlarının gelişimi, *Sakarya University Journal of Science*, 20 (2), 103-109.

Pengchao Zhang, Chao Xiong, Wenke Li, Xiaoxiong Du ve Chuan Zhao, 2018, Path planning for mobile robot based on modified rapidly exploring random tree method and neural network, *International Journal of Advanced Robotic Systems*, 15 (3), 1729881418784221.

Rame Likaj, Xhevahir Bajrami, Ahmet Shala ve Arbnor Pajaziti, 2017, Path finding for a Mobile Robot using Fuzzy and Genetic Algorithms, *International Journal of Mechanical Engineering*

Technology.

Reagan L Galvez, Elmer P Dadios ve Argel A Bandala, 2014, Path planning for quadrotor UAV using genetic algorithm, *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 1-6.

Ronald Uriol ve Antonio Moran, 2017, Mobile robot path planning in complex environments using ant colony optimization algorithm, *2017 3rd international conference on control, automation and robotics (ICCAR)*, 15-21.

Rosli Omar ve Dawei Gu, 2010, 3D Path Planning for Unmanned Aerial Vehicles using Visibility Line based Method, *ICINCO (1)*, 80-85.

S Aditya Gautam ve Nilmani Verma, 2014, Path planning for unmanned aerial vehicle based on genetic algorithm & artificial neural network in 3D, *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*, 1-5.

S Julius Fusic, P Ramkumar ve K Hariharan, 2018, Path planning of robot using modified dijkstra Algorithm, *2018 National Power Engineering Conference (NPEC)*, 1-5.

Suvaydan, F., 2011, PATH PLANNİNG PROBLEM FOR MOBİLE ROBOTS AND OPTİMAL SOLUTİON OF PATH PLANNİNG PROBLEMS WİTH ANT COLONY OPTİMİZATİON, (M.Sc. Thesis, *DÜZCE ÜNİVERSİTESİ* DÜZCE.

Ter-Feng Wu, Pu-Sheng Tsai, Nien-Tsu Hu ve Jen-Yang Chen, 2017, Combining turning point detection and Dijkstra's algorithm to search the shortest path, *Advances in Mechanical Engineering*
9(2), 1687814016683353.

Tyagi, M. K. a. M. A., 2017, Bellman Ford Shortest Path Algorithm using Global Positioning System,, *International Research Journal of Engineering and Technology (IRJET)*, 04 (04), 2503.

Victor Vladareanu, Elena-Corina Boscoıanu, Ovidiu-Ilie Sandrru ve Mircea Boscoıanu, 2016, Development of intelligent algorithms for UAV planning and control, *Scientific Research Education in the Air Force-AFASES*
1, 221-226.

WANG Lei, Bing-jie LI, Zhong-hai YIN, ZHOU Cheng, ZHAO Xin ve Ya-nan CHU, 2017, An Improved Artificial Potential Field for Unmanned Aerial Vehicles Path Planning, *DEStech Transactions on Computer Science Engineering*
(cst).

Wei Liu, Zheng Zheng ve Kaiyuan Cai, 2013, Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval, *Chinese Journal of Aeronautics*, 26 (3), 646-660.

Weiwei Zhan, Wei Wang, Nengcheng Chen ve Chao Wang, 2014, Efficient UAV path planning with multiconstraints in a 3D large battlefield environment, *Mathematical Problems in Engineering*, 2014.

X Bajrami, A Dërmaku ve N Demaku, 2015, Artificial neural fuzzy logic algorithm for robot path finding, *IFAC-PapersOnLine*, 48 (24), 123-127.

Xiao-Guang Gao, Xiao-Wei Fu ve Da-Qing Chen, 2005, A genetic-algorithm-based approach to UAV path planning problem, *Proceedings of the WSEAS International Conference on Simulation, Modeling, and Optimization*.

Y Singh, S Sharma, R Sutton ve D Hatton, 2018, Towards use of dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field, *TransNav: International Journal on Marine Navigation Safety of Sea Transportation*, 12.

Yalçıner, İ. F., 2014, DESIGN AND IMPLEMENTATION OF SMALL AND MIDDLE SCALE UNMANNED AERIAL VEHICLE (UAV) AUTO-PILOT, master, *FATİH UNIVERSITY*, 71.

Yoshiaki Kuwata, Gaston A Fiore, Justin Teo, Emilio Frazzoli ve Jonathan P How, 2008, Motion planning for urban driving using RRT, *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1681-1686.

Zhang Qi, Zhenhai Shao, Yeo Swee Ping, Lim Meng Hiot ve Yew Kong Leong, 2010, An improved heuristic algorithm for UAV path planning in 3D environment, *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*, 258-261.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Elaf Jirjees DHULKEFL DHULKEFL
Uyruğu	**:**	Irak
Doğum Yeri ve Tarihi	**:**	20.12.1994
Telefon	**:**	0534 855 02 25
Faks	**:**	
e-mail	**:**	elif.cercis94@gmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Al-Hariri Kız Lisesi, Kerkük, Irak	2012
Üniversite	:	Kerkük Teknik Üniversitesi	2016
Yüksek Lisans	:	Selçuk Üniversitesi-Elektrik_Elektronik Mühendisliği	2019

YABANCI DİLLER

Arabca, İnglizce, Türkçe

YAYINLAR

- Elaf Jirjees DHULKEFL and Akif DURDU (2019). Path Planning Algorithms for Unmanned Aerial Vehicles. International Journal of Trend in Scientific Research and Development (ijtsrd), 359-362.