



**UÇUŞ KISITLI BÖLGELERDE İNSANSIZ HAVA ARAÇLARI İÇİN ROTA  
OPTİMİZASYONU**

**Sabina AGHAZADA**

**YÜKSEK LİSANS TEZİ  
İLERİ TEKNOLOJİLER ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ARALIK 2020**

Sabina AGHAZADA tarafından hazırlanan “UÇUŞ KISITLI BÖLGELERDE İNSANSIZ HAVA ARAÇLARI İÇİN ROTA OPTİMİZASYONU” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi İleri Teknolojiler Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

**Danışman:** Doç. Dr. Bünyamin CİYLAN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

.....

**Başkan:** Prof. Dr. Şahin EMRAH

Bilgisayar Mühendisliği Ana Bilim Dalı, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

.....

**Üye:** Prof. Dr. Nurettin TOPALOĞLU

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

.....

Tez Savunma Tarihi: 06/12/2020

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Cevriye GENCER

Fen Bilimleri Enstitüsü Müdürü

## ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Sabina AGHAZADA

06/12/2020

# UÇUŞ KISITLI BÖLGELERDE İNSANSIZ HAVA ARAÇLARI İÇİN ROTA

## OPTİMİZASYONU

(Yüksek Lisans Tezi)

Sabina AGHAZADA

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Aralık 2020

### ÖZET

İnsansız hava araçlarının (İHA), özellikle de dronların gerek askeri gerekse sivil alanlarda her geçen gün daha da yaygın olarak kullanıldığı görülmektedir. İHA'ların görev planlaması söz konusu olduğunda ilk anlaşılan hareket-rota planlamasıdır. Bu çalışmada uçuş kısıtlı bölgelerde İHA'ların başlangıç noktasından bitiş noktasına uçuşu için en uygun rotanın belirlenmesi konusu araştırılmıştır. Uçuş kısıtlı bölgeler dikdörtgen veya daire ile değiştirilmiş ve problem, matematiksel olarak verilmiş doğru parçaları ile dikdörtgenin veya dairenin kesişimi ile tanımlanmıştır. Problemin çözümü için mevcut yaklaşımlardan farklı olarak dışbükey dörtgenin özelliklerine göre bir matematiksel model geliştirilmiş ve bu modeli kullanarak optimuma yakın mesafeyi hesaplayabilen bir yazılım geliştirilmiştir. Bunun için Karınca Kolonisi Algoritması uygulanmıştır. Klasik Karınca Kolonisi Algoritmasından farklı olarak her bir karıncanın hücreye girişinin serbest veya yasak olduğunu belirten özel bir “tabulist” matrisi dahil edilmiştir. “Tabulist” matrisinin dahil edilmesi karıncanın bir sonraki hücreyi seçme olasılığı da dikkate alındığı için engellere girip girmediğini kontrol etmeye gerek kalmaz. İterasyon sonunun sağlanması için karıncaların olduğu hücre ile başlangıç ve son noktayı birleştiren kırık çizgiye kadar olan en kısa mesafenin tersi – “karıncanın hedefe ulaşma” parametresi dahil edilmiştir. Her iki algoritma için yazılım geliştirilmiş ve hesaplama deneyleri yapılmıştır. Yapılan hesaplama deneylerine göre birinci algoritma ile ikinci algoritma arasındaki fark 4 birim olmuştur. Bu fark engellerin geometrisine, boyutuna, hizasına ve sayısına bağlı olarak artabilir veya azalabilir.

Bilim Kodu : 92402

Anahtar Kelimeler : İnsansız Hava Aracı, rota, dışbükey dörtgen, iterasyon, engel, algoritma

Sayfa Adedi : 74

Danışman : Doç. Dr. Bünyamin CİYLAN

# ROUTE OPTIMIZATION FOR UNMANNED AERIAL VEHICLES IN FLIGHT RESTRICTED AREAS

(M. Sc. Thesis)

Sabina AGHAZADA

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

December 2020

## ABSTRACT

It is seen that Unmanned Aerial Vehicles (UAVs), especially drones, are being used more and more every day in both military and civilian areas. Route planning is the first goal when it comes to the task planning of UAVs. In this study, it has been investigated determining the most suitable route for the flight of UAVs from the starting point to the endpoint in flight restricted areas. The restricted areas of flight are changed by rectangles or circles and the problem is defined by the intersection of the mathematically given line segments and rectangles or circles. For this problem, as distinct from the present condition, a mathematical model was developed according to the properties of the convex quadrilateral and by using this model a software has been developed that can calculate the optimum distance. For this, the Ant Colony Algorithm has been applied. Unlike the Classical Ant Colony Algorithm, a special “Tabulist” matrix has been included, indicating that each ant's entry into the cell is free or prohibited. The inclusion of the tabulist matrix also takes into account the possibility of the ant choosing the next cell, there is no need to check whether or not it is entering into obstacles to ensure the end of the iteration, “Ant’s reaching the target” parameter was included by connecting the ants’ cell to the start and the endpoint till the broken line to the inverse of the shortest distance. A software has been developed and calculation experiments have been done for both algorithms. According to the calculation experiments, the difference between the first algorithm and the second algorithm was 4 units. This difference may increase or decrease depending on the geometry, size, alignment and number of obstacles.

Science Code : 92402

Key Words : Unmanned Aerial Vehicle, route, convex quadrilateral, iteration, obstacle, algorithm

Page Number : 74

Supervisor : Assoc. Prof. Dr. Bunyamin Ciylan

## TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde, çalışmalarımı yönlendiren, araştırmalarımnda yardımlarını esirgemeyen değerli danışman hocam Doç. Dr. Bünyamin CİYLAN'a, yüksek lisans eğitimim boyunca yardım, bilgi ve tecrübeleri ile bana sürekli destek olan İleri Teknolojiler Ana Bilim Dalı bölüm başkanı Prof. Dr. Selim ACAR'a, İleri Teknolojiler Ana Bilim Dalı öğretim üyeleri değerli Prof. Dr. Sema Bilge OCAK'a, Prof. Dr. Şükrü ÇAVDAR'a teşekkür ederim.

Ayrıca çalışmalarım sırasında bana sürekli destek olan, bilgileri ile bana yol gösteren “Azerbaycan Teknik Üniversitesi” Mühendis Matematiği bölüm başkanı, canım babam Prof. Dr. Nadir AGHAYEV'e, dualarını üzerimden eksik etmeyen canım annem Humar AGHAYEVA'ya sonsuz sevgiler ve teşekkürlerimi sunmaktan onur duyarım.

## İÇİNDEKİLER

|                                                                  | Sayfa |
|------------------------------------------------------------------|-------|
| ÖZET .....                                                       | iv    |
| ABSTRACT.....                                                    | v     |
| TEŞEKKÜR.....                                                    | vi    |
| İÇİNDEKİLER .....                                                | vii   |
| RESİMLERİN LİSTESİ .....                                         | ix    |
| ŞEKİLLERİN LİSTESİ.....                                          | x     |
| SİMGELER VE KISALTMALAR.....                                     | xi    |
| 1. GİRİŞ.....                                                    | 1     |
| 2. İNSANSIZ HAVA ARAÇLARI.....                                   | 3     |
| 2.1. İHA'ların Avantajları ve Dezavantajları .....               | 7     |
| 2.2. İHA'ların Sınıflandırılması .....                           | 8     |
| 3. İNSANSIZ HAVA ARAÇLARININ ROTA OPTİMİZASYONU .....            | 13    |
| 3.1. Araç Rotalama Problemi (ARP).....                           | 16    |
| 3.1.1. Mesafe kısıtlı araç rotalama problemi.....                | 17    |
| 3.1.2. Kapasite kısıtlı araç rotalama problemi .....             | 17    |
| 3.1.3. Zaman pencereli araç rotalama problemi.....               | 17    |
| 3.1.4. Yolların durumuna göre araç rotalama problemleri.....     | 18    |
| 3.1.5. Dinamik araç rotalama problemi .....                      | 18    |
| 3.1.6. Statik araç rotalama problemi .....                       | 18    |
| 3.1.7. Gezgin satıcı problemi .....                              | 18    |
| 3.2. Rota Planlama Yöntemleri .....                              | 20    |
| 3.2.1. Grafik teorisini kullanarak rota planlama yöntemleri..... | 20    |
| 3.2.2. Dekompozisyon yöntemlerini kullanarak rota planlama ..... | 22    |
| 3.2.3. Potansiyel alan yöntemleri ile rota planlama .....        | 24    |

|                                                                                                              | <b>Sayfa</b> |
|--------------------------------------------------------------------------------------------------------------|--------------|
| 3.2.4. Optimizasyon yöntemleri ile rota planlama .....                                                       | 25           |
| 3.2.5. Entelektüel teknolojiler kullanarak rota planlama .....                                               | 26           |
| 3.3. İnsansız Hava Araçlarının Rota Optimizasyonunda Kullanılan Algoritmalar....                             | 28           |
| 3.3.1. Parçacık sürü optimizasyonu .....                                                                     | 28           |
| 3.3.2. Karınca kolonisi algoritması .....                                                                    | 30           |
| 3.3.3. Arı algoritması .....                                                                                 | 34           |
| 3.3.4. Süpürme algoritması .....                                                                             | 36           |
| 3.3.5. Tabu arama algoritması.....                                                                           | 36           |
| 3.3.6. En yakın komşu algoritması.....                                                                       | 37           |
| 3.3.7. Genetik algoritma.....                                                                                | 37           |
| 3.4. Literatürde Yer Alan İnsansız Hava Araçları'nın Rota Optimizasyonu ile İlgili<br>Diğer Çalışmalar ..... | 39           |
| 4. ÇALIŞMANIN AMACI ve PROBLEMİN TANIMLANMASI.....                                                           | 41           |
| 4.1. Matematiksel Modelleme.....                                                                             | 42           |
| 4.2. Hesaplama Deneyi .....                                                                                  | 48           |
| 4.3. Deneylerden Çıkan Sonuçlar.....                                                                         | 49           |
| 5. İNSANSIZ HAVA ARAÇLARININ KARINCA KOLONİSİ<br>ALGORİTMASI İLE OPTİMAL ROTASININ BULUNMASI.....            | 51           |
| 5.1. Hesaplama Deneyi .....                                                                                  | 55           |
| 6. SONUÇ VE ÖNERİLER .....                                                                                   | 59           |
| KAYNAKLAR .....                                                                                              | 73           |
| ÖZGEÇMİŞ .....                                                                                               | 73           |

## RESİMLERİN LİSTESİ

| Resim                                                                                                     | Sayfa |
|-----------------------------------------------------------------------------------------------------------|-------|
| Resim 2.2. Bombalarla donatılmış insansız savaş hava aracı .....                                          | 4     |
| Resim 2.3. Hewitt-Sperry otomatik uçak.....                                                               | 4     |
| Resim 2.4. GNAT 750 İHA .....                                                                             | 5     |
| Resim 2.6. ANKA-s İHA .....                                                                               | 6     |
| Resim 2.7. Bayraktar TB’2 İHA .....                                                                       | 6     |
| Resim 2.8. Karayel İHA .....                                                                              | 7     |
| Resim 3.1. İnsansız Hava araçlarının rota optimizasyonu .....                                             | 14    |
| Resim 3.2. Araç Rota planlama .....                                                                       | 17    |
| Resim 3.3. Clarke ve Wright tasarruf algoritması .....                                                    | 19    |
| Resim 3.4. Biyoloji nöron .....                                                                           | 27    |
| Resim 4.1. Engeller olduğunda İHA’ların en optimal uçuş yörüngesinin belirlenmesi .                       | 41    |
| Resim 4.2. St. Patrick’ten Service Ontario’ya IHS uçuşunun ilk talimatı.....                              | 48    |
| Resim 4.3. St. Patrick’ten Service Ontario’ya İHA’nın uçuş kısıtlı bölgelerde en kısa uçuş mesafesi ..... | 49    |
| Resim 5.1. St. Patrick’ten Service Ontario’ya İHA’nın engeller olduğunda en kısa uçuş mesafesi.....       | 55    |

## ŞEKİLLERİN LİSTESİ

| Şekil                                                                                                                                | Sayfa |
|--------------------------------------------------------------------------------------------------------------------------------------|-------|
| Şekil 3.1. İHA'ların uçuş yönetiminin yazılım sistemi yapısı.....                                                                    | 15    |
| Şekil 3.2. Rota Planlama Yöntemleri.....                                                                                             | 20    |
| Şekil 3.3. Grafik teorisini kullanarak rota planlama yöntemleri.....                                                                 | 21    |
| Şekil 3.4. Dekompozisyon yöntemleri.....                                                                                             | 23    |
| Şekil 3.5. Entelektüel teknolojiler kullanarak rota planlama sistemi.....                                                            | 26    |
| Şekil 3.6. Nöronun matematiksel modeli.....                                                                                          | 27    |
| Şekil 3.7. Parçacık sürü algoritması kullanılarak yol planlama probleminin akış şeması .....                                         | 30    |
| Şekil 3.8. En yakın komşu algoritmasıyla bulunan rota .....                                                                          | 33    |
| Şekil 3.9. Karınca sistem algoritmasıyla bulunan rota .....                                                                          | 33    |
| Şekil 4.1. Düzlem üzerinde $n$ sayıda doğru parçasıyla, verilmiş doğru parçasının kesişimini belirleyen algoritma .....              | 47    |
| Şekil 5.1. Karıncaların iterasyonlara bağlı olarak rotalarının uzunluk dinamiği .....                                                | 56    |
| Şekil 5.2. Karıncaların belirlediği rotanın uzunluklarının diğer çalışmada verilmiş yöntemle bulunmuş rotasındaki bağıl hatası ..... | 57    |

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

### Simgeler

### Açıklamalar

|                  |                                  |
|------------------|----------------------------------|
| $\%$             | Yüzde                            |
| $\mu_{i,j}$      | Karınca'nın hedefe ulaşma isteği |
| $\theta_{ik,jk}$ | “Tabulist” matrisi               |
| $\alpha$         | Feromonun buharlanma hızı        |

### Kısaltmalar

### Açıklamalar

|      |                                |
|------|--------------------------------|
| ACO  | Ant Colony Optimization        |
| ACS  | Ant Colony System              |
| ARP  | Araç Rotalama Problemi         |
| GA   | Genetik Algoritma              |
| İKA  | İnsansız Kara Aracı            |
| İHA  | İnsansız Hava Aracı            |
| KKA  | Karınca Kolonisi Algoritması   |
| KKE  | Karınca Kolonisi Eniyileme     |
| HALE | High Altitude Long Endurance   |
| MALE | Medium Altitude Long Endurance |
| MMAS | Max-Min Ant System             |
| RRP  | Rota Planlama Problemi         |
| TSK  | Türk Silahlı Kuvvetleri        |
| YSA  | Yapay Sinir Ağları             |

## 1. GİRİŞ

İnsansız hava araçları (İHA) kullanımı son yıllarda oldukça yaygınlaşmıştır. İHA'lar öncelikle askeri işlerde, daha çok savunma programlarında, askeri stratejilerin geliştirilmesinde ve uygulanmasında kullanılmaktadır [1]. Çeşitli askeri çatışmalarda İHA'lar askeri tesislerin yer tespiti ve farklı sorunların da çözümünde yaygın olarak kullanılabilir. İHA'lar hava kirliliğinin ölçülmesi, iletişim hatlarının kurulması, çeşitli amaçlarla kargo taşımacılığı, olağanüstü hallerde halkın ve tesislerin güvenliğinin sağlanması vb. pek çok amaçla kullanılabilmektedirler. Son yıllarda, İHA'ların büyük şehirlerin sorunlarının çözümünde kullanımı ile ilgili olarak teknik araçların hareketinin gözlemlenmesi, yardım, iletişim ve diğer konuların araştırılmasında pek çok avantajı bulunmaktadır [2]. Askeri amaçlı kullanımla beraber İHA'ların kişisel veya ticari amaçla kullanımı da oldukça yaygınlaşmaktadır [3]. İHA'ların kullanımı özellikle acil durumlarda çevrenin izlenmesinde, deprem, yangın ve diğer olağanüstü durumlarda hayat kurtarabilir. Kriz zamanında tehlikeli yerlerdeki insanlara yeterli miktarda su, gıda ve tıbbi malzeme taşıyabilir, ayrıca acil müdahale için irtibat tesisleri kurabilirler. Büyük şehirlerde günlük olarak bilimsel ve pratik çalışmalarında birçok problemlerin olduğu ortaya çıkmıştır. Bu problemler içinde en çok çevrenin karmaşıklığı ve değişkenliği nedeniyle İHA'ların yönetiminde problemlerle karşılaşmaktadır. Bu problemler; teknik, yönetim, çevresel etki vs. gibi gruplara ayrılabilir. Diğer alanlarda, örneğin askeri işlerde kullanımının aksine İHA'ların büyük şehirlerde kullanımında karşılaşılan bu sorunların kendine özgü özellikleri vardır [4].

Teknik problemlere İHA'ların fiziksel boyutlarının küçük olması, iniş kalkış sahalarının olmaması veya az sayıda olması, uçuş süresinin ve yük taşıma kapasitesinin kısıtlı olması, ses, titreşim vb. etkilerin araç üzerinde daha fazla olması da eklenmektedir.

Yönetim problemleri ise otonom yönetim ve merkezi kontrol sistemlerinin her ikisinin eşzamanlı varlığı, uçuş yüksekliğine ve çevreye göre yüksek manevra zorunluluğu gibi sorunları içerir [5].

Büyük şehirlerde İHA uygulamalarında karşılaşılan sorunlardan birisi de çevre kaynaklı sorunlardır. Bunun başlıca nedeni yüksek binaların etrafında hava akışlarının hızının ve yönünün keskin değişikliği sonucu ortaya çıkan sorunlar, çeşitli iletişim, elektrik ve diğer

elektromanyetik alan kaynaklarının yönetiminde engellerin oluşması gibi sorunları içermesidir. Bu problemlerin çözümüne yönelik çalışmalar, teorik ve uygulamalı araştırmalar gibi çeşitli bilimsel ve dijital bilgi kaynaklarında bulunabilir. İHA'ların kullanımı ile ilgili pratik sorunların en ilgi çekenini ise mini İHA'ların kullanımı problemi. Bu İHA'lar en fazla 5 fut (1,5 m) uzunluğuna sahip, sabit kanat boyutu olan, pille çalışan, iniş-kalkış pisti gerekmeyen İHA'lardır. Düşük taşıma kapasitesi, üzerinde yönetim ve iletişim araçlarının kurulumu olanaklarını kısıtlar. Bu nedenle, böyle İHA'ların kullanımı sırasında otomatik pilotlarla ilgili problemler daha belirgindir [6]. İHA'ların otomatik şekilde uçuşunun sağlanması probleminin çözümü öncelikle uçuş rotasının önceden belirlenmesi, bu rotanın haritaya aktarılması, gerçek zamanda rotada uçuşu için bir kontrol planı geliştirmesiyle gerçekleştirilebilir. Bu problemlerin teknik bir çözümün önce teorik bir konu olarak ele alınması gerekmektedir. Uçuş rotasının önceden belirlenmiş kısıtlar içinde belirlenmesi problemi teorik olarak matematiksel programlama problemi olarak sunulmakta ve çeşitli sayısal, sayısal-analitik, sezgisel ve diğer yöntemlerle çözülebilir. Uçuş rotasında olan kısıtlı alanlardan yan geçerek rotanın belirlenmesinin yanı sıra, İHA'nın fiziksel ve uçuş özelliklerinden kaynaklanan sınırlamalar da bu kapsamdadır. Sorunun matematiksel formülasyondaki çözümü, klasik matematiksel programlamada olduğu gibi sadece basit durumlarda mümkündür. Diğer durumlarda gerçek zamanlı dikkate alınması, problemi daha da zorlaştırır ve bu çözüm sırasında geleneksel olmayan çeşitli yöntemlerin kullanılmasını gerekli kılar. Bu tür çözümler arasında bulanık mantık teorisi, nöro modelleme ve akıllı algoritmalar kullanarak soruna sezgisel çözümler bulmak daha yaygındır [7].

Bu çalışmada iki farklı algoritma geliştirilmiştir. Birinci algoritma iki doğru parçasının kesişiminden alınan dışbükey dörtgenin özelliklerine dayanmaktadır. Bu çalışmanın mevcut çalışmalardan farkı sadece rota üzerinde olan engellerin çevresini hesapladığı için daha az hesaplama kaynağı gerektirir. Engellerin bir-birine yakın olduğu durumlarda ise bu algoritma optimal olmadığı için bu çalışmada Karınca Kolonisi Algoritması uygulanarak problemin çözümü verilmiştir. Bu algoritmanın mevcut algoritmalarından farkı özel bir "tabulist" matrisi ve "karıncanın hedefe ulaşma" parametreleri dahil edilmiştir.

## 2. İNSANSIZ HAVA ARAÇLARI

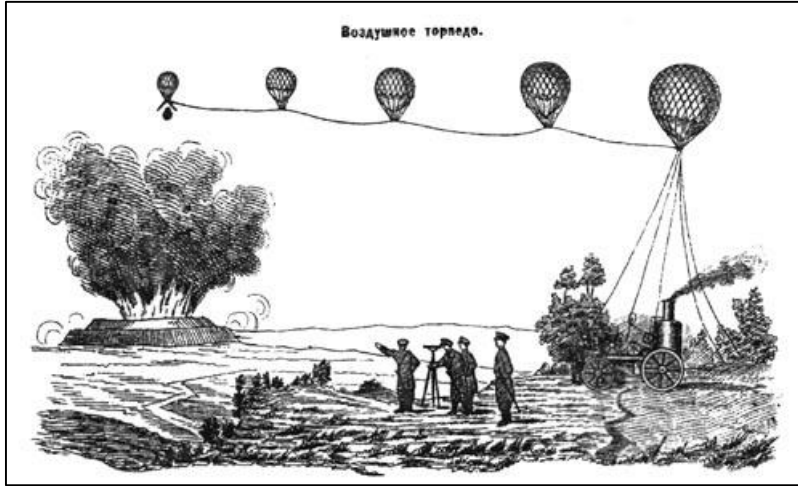
İnsansız sistem, kullanıcı olarak bir insan taşımayan, otonom ya da uzaktan komutla çalışabilen, faydalı yükleri taşıyabilen, kendi kendine hareket yeteneğine sahip olan bir araçtır [8]. İnsansız sistemler arasında hava, kara ve deniz araç sistemleri vardır. İnsansız Kara Araçları (İKA), kara için geliştirilen ve insan olmadan çalışan araçtır. İKA'lar bir bombanın imhasında, herhangi bir bölgenin güvenlik kontrolünde, düşmanı etkisiz hale getirilmesi için kullanılabilmektedir [9].

İnsansız Deniz Araçları, uzaktan çalışan operatör tarafından kontrol edilen, uzaktan kumandalı sualtı araçları ve doğrudan insandan bağımsız olarak çalışan Otonom sualtı araçları olarak iki kategoriye ayrılır [10]. İHA'lar ise önceden programlanarak veya yerden komuta ile harici bir pilot tarafından kontrol edilen, uçuş sonrası tahrip olan veya yeniden kullanılabilen, motor gücüne sahip hava araçlarıdır [11].

Uzaktan kumandalı uçan makinelerin tarihi çok eskidir. 1849 yılında Avusturyalılar ilk kez zamanlayıcı cihazlarla çalışan bombalarla donatılmış 200 insansız savaş amaçlı hava aracı ile İtalya'nın Venedik şehrine saldırmışlardır (Resim 2.1, 2.2) [12].

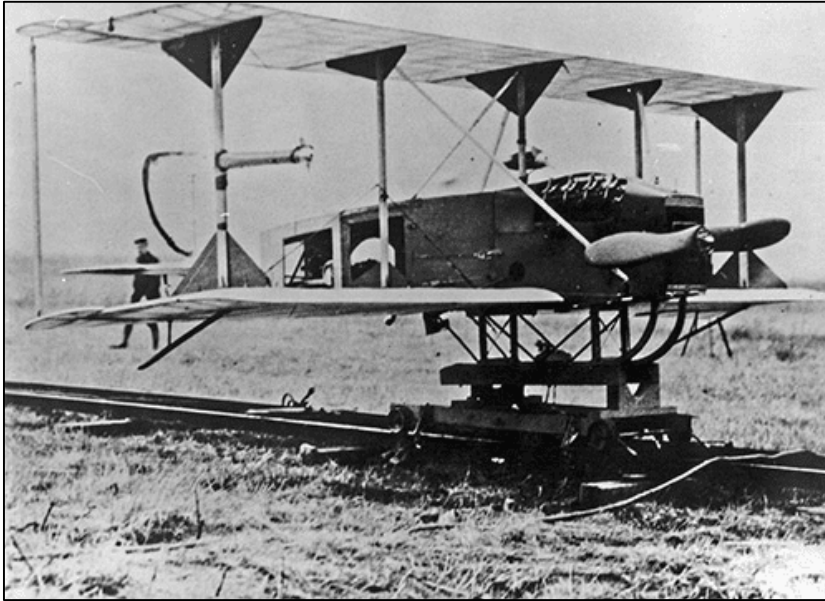


Resim 2.1. Bombalarla donatılmış insansız hava aracı [13]



Resim 2.2. Bombalarla donatılmış insansız savaş hava aracı [12]

İnsansız ilk uçak Birinci Dünya Savaşı'nın sonlarında (1916) tasarlanmıştır. İlk insansız kullanımı ise “Hewitt-Sperry” isimli, entegre jiroskop kontrollü “uçan bombalar” olarak bilinen otomatik uçak ile gerçekleştirilmiştir (Resim 2.3).



Resim 2.3. Hewitt-Sperry otomatik uçak [14]

İHA'larla ilgili çalışmalar İkinci Dünya Savaşı ve sonrasında da devam etmiş, günümüze kadar birçok askeri operasyonda kullanılmış ve halen de kullanılmaya devam etmektedir. Son yıllarda İHA'lar üzerinde birçok çalışma yapılmaktadır. Bu çalışmalar İHA'ların farklı alanlarda kullanılacağını ve bu alanda çalışan araştırmacı sayısının da gelecek yıllarda artmasına sebep olacağını göstermektedir [15].

Türkiye’de İHA sistemlerinin gelişmesi, dünyadaki gelişmelere paralel olarak 1980 yılının sonunda başlamıştır. Türk Silahlı Kuvvetleri (TSK) tarafından kullanılmaya başlanan ilk İHA Meggitt firması üretimi olan Banshee sistemi olmuştur. TSK’ya gerçek anlamda alınan ilk İHA sistemi, 1995 yılında ABD firması üretimi olan General Atomics’in GNAT-750 İHA’sıdır (Resim 2.4).



Resim 2.4.GNAT 750 İHA [16]

Yerli İHA üretim çalışmaları 1990’lı yılların başında başlamış, son yıllarda daha da yoğunlaşmıştır. Türkiye’de ilk yerli İHA üretimi 1990 yılında TUSAŞ tarafından İHA-X1 Şahit sistemi ile başlamıştır. Projenin uçuş testleri tamamlanmasına rağmen, bütçe kısıtlılığı nedeni ile daha fazla ilerletilememiştir [17].

2004 yılında ANKA Geliştirme Projesi başlatılmıştır. Proje, üç prototip aşamadan oluşmuştur. Birinci aşamasının ilk uçuşu 2010 yılında yapılmıştır. Bu prototipin uçuş kabul testleri ise 2013 yılında tamamlanmıştır (Resim 2.5) [18].



Resim 2.5. ANKA-s İHA [19]

Anka-s İHA'nın teknik özelliklerini sıralarsak: gövde uzunluğu: 8 m, hızı: 217 km/saat, havada kalış süresi: 24 - 32 saattir [20].

2005 yılında Kalekalıp-Baykar Makina Ortak Girişimi tarafından geliştirilmesi başlatılan Bayraktar mini İHA sistemi olmuştur. 29 Şubat 2012'de Katar Silahlı Kuvvetleri için 20 adet İHA üretilmiştir. Böylece Bayraktar mini İHA Türkiye tarihinde ihraç edilen ilk yerli İHA sistemi olmuştur [21]. Bayraktar TB2'nin genel özellikleri; toplam kalkış ağırlığı 650 kg, faydalı yük ağırlığı: 55 kg, havada kalma süresi 24 saatten fazladır (Resim 2.7) [22].



Resim 2.6. Bayraktar TB'2 İHA [22]

Aralık 2010 tarihinde Vestel Savunma Sanayi ile SSM arasında imzalanan anlaşma ile tasarım çalışmalarına başlanan 550 kg kalkış ağırlığına ve 150 km veri linki menziline sahip Karayel'in tasarımı tamamlanmış ve uçuş testleri yapılmıştır. Karayel insansız hava aracının yapısının büyük bir kısmı kompozitten oluşmaktadır. Bu İHA'da bir adet 70 beygir gücünde motor vardır. 6,5 metre uzunluğunda ve 10,5 metre kanat açıklığına sahiptir. Toplam kalkış ağırlığı 550 kg olan araç 70 kg faydalı yük taşıyabilir. Aralıksız 20 saat üstü uçuş yapabilecek olan aracın uçuş hızı 110-130 km/saattir. Araç faydalı yük olarak; renkli gündüz kamerası, gece görüş kamerası, lazer mesafe bulucu ve lazer işaretleyici taşımaktadır (Resim 2.8) [20].



Resim 2.7. Karayel İHA [23]

## 2.1. İHA'ların Avantajları ve Dezavantajları

Normal uçaklarda, uzun keşif ve gözetleme gerektiren işlerde, insan zaafından kaynaklı bazı sıkıntılar çıkma ihtimalinin yüksek olmasına karşın, İHA'lar taşıdıkları kamera, sensör ve radarlar yardımı ile daha verimli ve etkin olarak kullanılabilir. Araştırma ve geliştirmelerde daha hızlı ve daha ucuz çözüm alma adına İHA'lar kullanılabilir. İHA'lardan büyük şehirlerde çeşitli işler için, özellikle gün içinde trafiğin yoğun olduğu zamanlarda, ayrıca olağanüstü hallerde, örneğin, istihbarat, keşif ve gözetleme, hedef tespiti ve izleme, yangın tespiti ve izleme, arama ve kurtarma, sivil güvenlik, haberleşme ve elektronik harp, sınır güvenliği, atmosferik ve denizaltı gözlem, tarımsal gözlem ve ilaçlama gibi çalışmalar için kullanılabilir [24].

İHA'lar, teslimat için gerekli olan maliyet ve zamanı önemli ölçüde azaltma potansiyeline sahiptir [11]. Genel olarak İHA'ların avantajları sıralandığında daha az enerji sarfıyatı,

gürültü düşüklüğü ve dışarıya bıraktığı atığın azlığı göz önünde bulundurulduğunda daha çevre dostu olması gibi durumlar da ön plana çıkmaktadır. İHA sistemlerinin avantaj ve dezavantajları ile ilgili temel bulgular aşağıda verilmiştir.

İHA'ların avantajları:

- İHA'ların üretim, yakıt tüketimi satın alma ve uçuş maliyeti, pilotlu uçaklara göre çok daha azdır.
- Görev sırasında risk altında pilot veya operatör yoktur.
- Kimyasal, biyolojik ve radyoaktif olarak kirlenmiş riskli arazilerde görev yapabilmektedir.
- Kötü hava şartlarından pilotlu uçaklara göre daha az etkilenmektedir.
- Çeşidine göre 48 saate kadar havada kalabilmekte ve 65 bin ft. yüksekliğe kadar çıkabilmektedir.
- Küçük ve sessiz olduğu için havada tespit edilmesi zordur.
- Birçok İHA'ların bir aradayken daha fazla alanı kısa sürede tarayabilmektedirler.

İHA'ların dezavantajları:

- Bazı modelleri düşman uçaklarının saldırısına karşı savunmasızdır.
- İHA operatörünün risklere karşı duyarlılığı normal uçaklardan daha düşük olduğu için operatörün gelen tehditleri algılama yeteneği zayıftır.
- İHA'ların sistemleri siber saldırılarına karşı daha düşüktür.
- İHA'lar ile yer arasında eğer bağlantı koparsa büyük riskler oluşabilmektedir [18].

## 2.2. İHA'ların Sınıflandırılması

İHA'lar askeri ve sivil alanlarda kullanılmak üzere 2 kategoriye ayrılırlar. Askeri İHA'lar aşağıdaki amaçlarla kullanılmaktadır:

- Hedef ve yem – kendisini hedef haline getirerek düşmanın pozisyonunu ortaya çıkarmasını sağlayan araçlar,
- Keşif ve gözetleme – düşman tarafla ilgili bilgiler toplayan araçlar,
- Çatışma – riskli yerlerde görev yapan ve saldırı kapasitesine sahip araçlar.

Günümüzde en çok askeri alanda İHA MALE (Medium Attitude Long Endurance) tipli İHA'lar kullanılmaktadır. MALE tipi İHA'ların en büyük avantajı riskli bölgelerde insan kaybı yaşanmamasını sağlamasıdır. Dünyada MALE tipi İHA'ları üretilen satan beş ülke vardır, bu ülkeler Türkiye, İsrail, ABD, BAE, Çin'dir. İstihbarat ve saldırı için de en etkili İHA sistemi yine MALE tipli İHA'lardır. İçlerinde bulunan otonom sistemi sayesinde düşmanı anlık olarak izleyebilir ve komuta merkezine canlı görüntü gönderebilirler. Bu İHA sistemleri yüksek irtifalarda gerektiğinde motoru kullanmadan uzun süre havada kalabilmektedir [25].

Askeri ve ARGE çalışması için kullanılan diğer bir İHA sistemi HALE (High Altitude Long Endurance) tipi İHA'lardır. Bu İHA'ların ortalama operasyon süresi 17-22 saat, operasyon irtifası ise 45000 ile 60000 ft aralığındadır [26].

Günümüzde askeri İHA'lar ile birlikte sivil amaçlı İHA'lar da yaygın şekilde kullanılmaya başlanmıştır:

1. Geliştirme ve gelecekte kullanılmak amacıyla farklı İHA teknolojilerinin denendiği araçlar;
2. Ticari amaçlar için kullanılan İHA'lar;
3. Eğlence amaçlı kullanılan İHA'lar.

İHA'lar ile ilgili çalışmalarda birçok sınıflandırmalar bulunmaktadır. Mevcut İHA'lar ağırlık, dayanıklılık, menzil, motor tipi, yükleme gücü, maksimum yükseklik ve kanat yüküne göre sınıflandırılabilir.

İHA'ların ağırlıklarına göre sınıflandırılması:

- Çok ağır İHA'lar 2 ton ve daha fazla kalkış ağırlığına sahip İHA'lardır. Örneğin, Darkstar, Predator-B, Boeing X-45.
- Ağır İHA'lar 2 ton ile 200 kg arasında kalkış ağırlığına sahip İHA'lardır. Örneğin, Fire Scout.
- Orta ağırlıklı İHA'lar 50 ile 200 kg arasında kalkış ağırlığına sahiptir. Örneğin, Phoenix UAV

- Hafif ağırlıklı İHA'ların kalkış ağırlığı 5 ile 50 kg arasında değişmektedir. Örneğin, RPO Midget
- Mikro İHA'ların kalkış ağırlığı 5 kilogramın altındadır. Örneğin, Lockheed Martin Desert Hawk, Dragon Eye [27].

İHA'ların dayanıklılığı uçuş sırasında alınan toplam sürede tanımlanabilir [28]. İHA'lar havada kalma sürelerine bağlı olarak uzun, orta ve düşük dayanıklılığa göre sınıflandırılır.

Uzun ömürlü İHA'lar genellikle askeri amaçla kullanılır. Bu İHA'lar 24-48 saat aralığında havada kalabilir ve 1500-20 000 km mesafeye kadar gidebilir. Örneğin, Northrop

Grumman RQ-4 Global Hawk- havada kalma süresi 32 saatten fazladır. Predator için ise havada kalma süresi 24 saattir.

Orta ömürlü İHA'lar yakıt ikmali olmadan 5-24 saat aralığında uçuş gerçekleştirebilirler. Onlar da çok uzun mesafelerde gözetim yapabilirler. Örneğin, Silver Fox'un havada kalma süresi 8 saattir [15].

Kısa ömürlü İHA'lar 5 saatten daha az dayanıklılığa sahip İHA'lardır. Bunlar tarımsal amaçlı kullanılmaktadır. Bu nedenle, uçuş yolu önceden belirlenerek daha verimli bir şekilde kullanılabilir. Örneğin, Pointer 1 saat kadar havada kalabilmektedir [29].

İHA'lar menzile göre “çok yakın mesafeli”, “yakın mesafeli”, “kısa mesafeli” ve orta mesafeli” olarak sınıflandırılabilir. Maksimum menzil, İHA'ların kalkış ve iniş sırasında hızı ve hava koşullarını dikkate alındığında uçabileceği maksimum mesafedir. İHA'ların menzili, uçağın ürettiği hava akımının miktarı, dayanıklılık ve uçuş hızı değerlerine bağlıdır.

$$Range(miles) = \frac{kV \cdot V \cdot 60 \cdot Pitch}{12 \cdot 5260} \cdot Endurance(hrs) \quad (2.1)$$

Burada “Pitch” değeri İHA üzerindeki pervanenin eğimidir [28].

- Çok yakın mesafeli İHA'lardan genellikle havadan çekilen fotoğraf yardımıyla yapılan araştırmalar tarım bölgelerinde kullanılır. Bu İHA'lar 5 km menzile sahiptir.

- Yakın mesafeli İHA'lar 50 km menzile sahip İHA'lardır. Bu İHA'ların havada kalma süresi 1-6 saattir. Bu İHA'lar genellikle havada çekilen fotoğrafları inceleme, keşif tesislerin gözetimi için kullanılıyor.
- Kısa mesafeli İHA'lar 150 km menzile sahip İHA'lardır. Bu İHA'ların havada kalma süresi 8-12 saattir.
- Orta mesafeli İHA'lar ise genellikle 16.000 ft'e kadar çıkabilen yüksek-hızlı İHA'lardır. Bu İHA'lar hedef tanılama, askeri gözetim ve bir alanı bombalamak için kullanılır [29].

Aerodinamikte kanat yükü, toplam uçak ağırlığının kanat alanına bölünmesiyle elde edilen orandır [30]. İHA'lar kanat yüküne göre aşağıdaki gibi sınıflandırılabilir;

- Düşük kanat yüklü İHA'lar tarım için kullanılır ve kanat yükü değeri 5 kg'dır. (Örn. Seeker)
- Orta kanat yüklü İHA'lar kanat yükü 50-100 kg arasında olan İHA'lardır. (Örn. Fire Scout)
- Yüksek kanat yüklü İHA'ların kanat yükü 100 kg'dan büyüktür. (Örn. Global Hawk) [31].

İHA'lar çıkabileceği irtifaya göre alçak, normal ve yüksek irtifa olarak sınıflandırılabilir.

- Alçak irtifalı İHA: Çıkabileceği en fazla yükseklik 1000 m'dir. Bu tip İHA'lar "Mikro İHA'lar" olarak bilinir. Örneğin, FPASS, Pointer.
- Orta irtifalı İHA: Bu kategoride olan İHA'ların çıkabileceği maksimum yükseklik 1000 ile 100000 m arasındadır. Günümüzde en yaygın kullanılan İHA'lardır. Örneğin, Ion Tiger.
- Yüksek irtifalı İHA: 100000 m'den çok yüksekliğe uçabilen İHA'lardır. Yüksek irtifa en çok askeri alanlarda, hedef, keşif gözetlemelerinde kullanılır. Örneğin, X-45, Predator B [32].



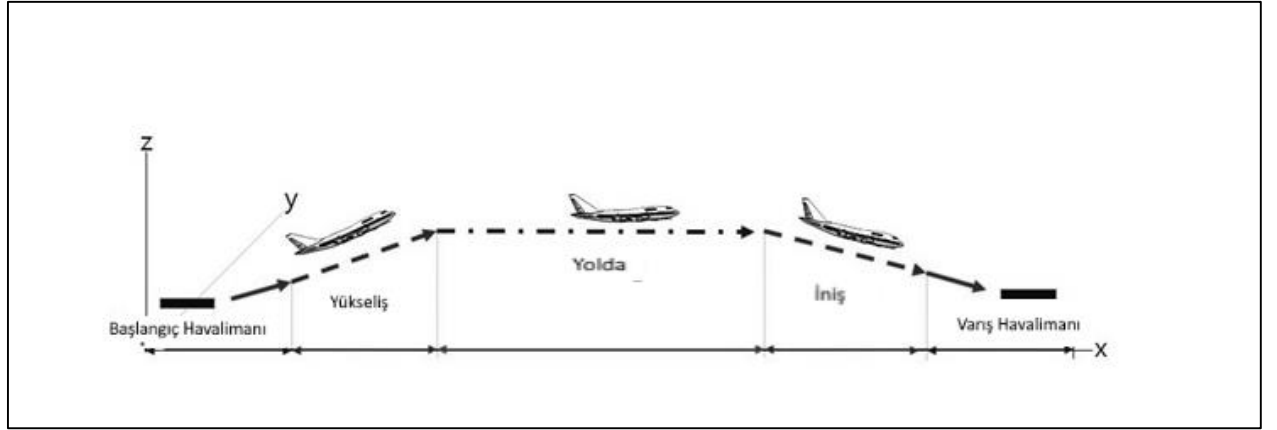
### 3. İNSANSIZ HAVA ARAÇLARININ ROTA OPTİMİZASYONU

İHA'nın askeri ve sivil alanda kullanım alanları ve yaygınlığı her geçen gün daha da artmaktadır. İHA'ların görev planlaması söz konusu olduğunda ilk anlaşılan hareket-rota planlamasıdır. Hava trafiğini düzenlemek ve artan hava yoğunluğunu önlemek için hava trafiği kontrolü sistemlerini geliştirme ihtiyacı ortaya çıkmaktadır. Aynı zamanda İHA'ların en büyük problemlerinden birisi de yakıt ve zaman problemidir. Bu nedenle yakıtı en verimli şekilde kullanarak hedefe en kısa şekilde erişilmesi gerekmektedir. Rota planlaması, herhangi bir aracın başlangıç noktasından başlayarak, engel olarak belirlenmiş bölgelere girmeden (bulunduğu irtifadan yüksek noktalar, düşman tarafından engellenmiş veya korunan alanlar) son noktaya ulaşmaya kadar geçeceği noktalardır. Böylece İHA'ların rota optimizasyonu denildiğinde;

- Planlanan yol belirli bir başlangıç noktasından belirli bir bitiş noktasına kadar ilerlemelidir.
- Bu yolda olan engelleri aşarak İHA'ların hareketini sağlamalıdır.
- Yol, ilk iki gereksinimi karşılayan tüm olası yollar arasında optimal olmalıdır.

Rota optimizasyonu, yol üzerinde kısıtlamalar olduğunda uygun maliyetli ve kısa zamana sahip olan rotalar arasında en iyi rotayı bulmaktır [33].

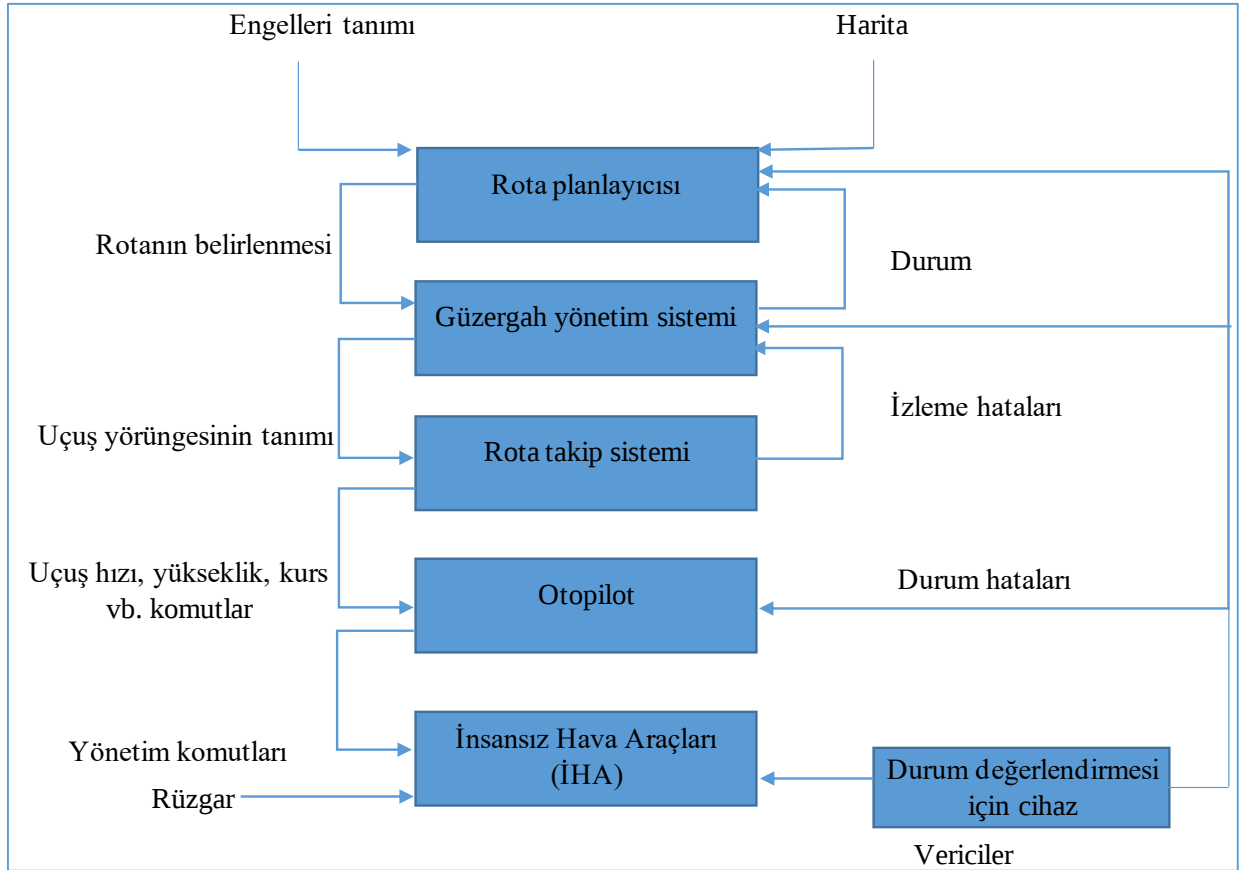
En basit rota hesaplama modeli olarak tabir edilebilecek model Resim 3.1'de görülmektedir. (Resim 3.1). Bunun için herhangi bir uçuşu seçilebilir. Uçuş başlangıç noktasından son noktasına kadar birkaç aşamadan oluşur. Basit bir ifadeyle, hava sahası Öklid uzayından oluşur, başlangıç ve bitiş noktalar sıfır yüksekliğinde bulunmaktadır. Tüm hava araçları aynı şekilde hareket eder ve aynı irtifaya inme ve çıkma aşamalarını takip eder.



Resim 3.1. İnsansız Hava araçlarının rota optimizasyonu

Engeller olduğunda İHA'ların uçuş yönetimi probleminin çözümü aşağıdaki alt problemlerin çözümünden ortaya çıkar:

1. Rota planlayıcı
2. Rota yönetiminin sistemi
3. Rota izleme sistemi
4. Oto-pilot
5. İnsansız Hava Aracı (İHA)
6. Durum değerlendirme cihazı (Şekil 3.1) [34].



Şekil 3.1. İHA'ların uçuş yönetiminin yazılım sistemi yapısı

Rota planlayıcı, harita verilerini kullanarak uçuşun başladığı ve bittiği koordinatlar arasında engel olmadan rotanın belirlenmesi amacı taşımaktadır. Bu amaçla harita verilerinin İHA uçuşunun matematiksel modeli ile bütünleştirmek için koordinat sistemleri seçilmeli ve bir koordinat sisteminden diğer koordinat sistemine geçiş formülleri göz önünde bulundurulmalıdır [35]. Alt sistem diğer alt sistemden, rota yönetim sisteminden, İHA'nın durumu ile ilgili verileri alır. Sistem gerçek zamanlı olarak çalıştığında durum değerlendirme cihazı, vericilerden aldığı bilgileri planlanan rotada ayarlamalar yapmak için kabul eder. Bu bilgiler diğer bilgilerle düzeltilerek tekrar rota yönetim sistemine gönderilir.

Rota yönetim sistemi, diğer alt sistemlerdeki verilere dayanarak İHA'nın uçuş rotasını tanımlar ve verileri rota izleme sistemine gönderir. Bu sistem rota planlayıcıdan rota üzerinde yörüngenin her noktasıyla ilgili bilgileri işler. İşlem sürecinde rota takip sisteminden, rota dışında bilgiler ve durum değerlendirme cihazının vericilerden aldığı veriler göz önünde bulundurulur. Verilerin eşzamanlı olarak işlenmesi rota boyunca doğru hareketi sağlamak içindir.

Rota takip sistemi, gerçek zamanlı olarak belirlenmiş yörünge boyunca uçuş yönünün izlenmesi ve elde edilen sonuçların İHA kontrol cihazına oto-pilot olarak aktarılması için kullanılması amaçlanmıştır. Yazılım sisteminde bu bölüm, durum değerlendirme cihazına gelen ve İHA'da bulunan çeşitli sensörlerden (konum, ivmeölçerler, yükseklik ölçerler, açı, hızı ve basınç sensörleri vb.) alınan bilgilerin sisteme iletimi için kullanılmaktadır. Bu bilgiler yazılım tarafından işlenir ve izleme hataları tespit edildikçe sonuçlar rota yönetim sistemine, rotayı düzenlemek için gönderilir.

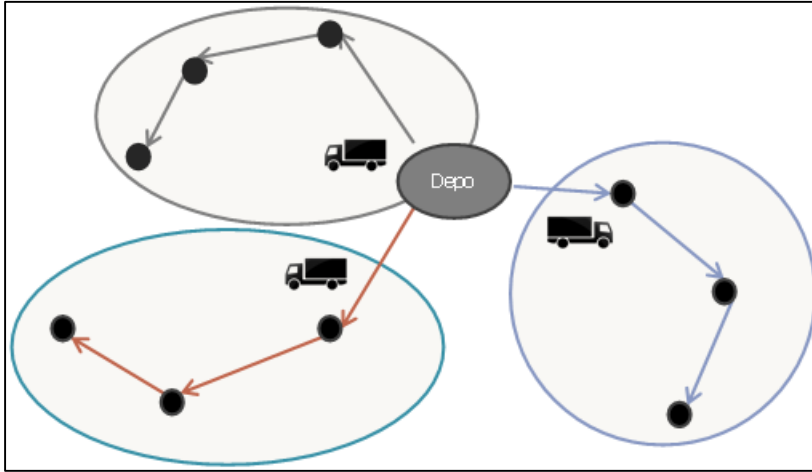
Oto-pilot, İHA uçuş kontrol cihazı ile doğrudan bağlantılı olup, önceki bloklardan gelen verileri ve İHA uçuşunun matematiksel modele dayalı geliştirilen özel algoritması ile uçuş hızı, irtifa, seyir gibi değerleri hesaplayıp yönetim komutlarını formüle ederek İHA'ya iletir. Yazılım sistemi İHA'dan aldığı verilere dayanarak gerçek durumu değerlendirir ve hataları tanımlar.

İnsansız Hava Aracı (İHA), fiziksel (gerçek) İHA'nın yazılım sisteminin modellenmiş halidir. Gerçek İHA'larda uçuş yönetiminde temel sorun rüzgar kontrol sorunudur. Üst hava katmanlarında, rüzgar hızı genellikle İHA'ların hızından daha güçlüdür. Bu noktalarındaki rüzgar hızının belirsizliği, uçuş sırasında İHA'nın yörüngeden ayrılmasına neden olur.

Rota Planlama Problemleri (RPP) ile ilgili literatüre ilk defa 1959 yılında Ramster ve Danztig, istasyonlara benzin aktarılması ile ilgili problemlerin çözümünde matematiksel algoritmalar geliştirdi. RPP ile ilgili en önemli algoritma ise 1981 yılında Christofides'in iki makale yayımlaması ile başladı. Bu makalelerde dinamik programlamaya dayalı bir algoritma geliştirmiştir. O zamandan bugüne kadar, rota planlama problemi araştırmacılar tarafından büyük ilgi görmektedir. [36, 37].

### **3.1. Araç Rotalama Problemi (ARP)**

Bir veya birkaç depodan K sayıda aracın, ürünleri müşterilere dağıtımı ve bu ürünlerin müşterilerden alınması olarak bilinmektedir. Araç rota problemi, ürünleri depoya gönderen üretici bir firma, siparişlerin müşterilere getirildiği depo ve ürünleri alan müşterilerden oluşmaktadır (Resim 3.2). Amaç en kısa sürede maliyeti de en aza indirerek bir noktadan diğer noktaya olan rotayı en aza indirmektir [38].



Resim 3.2. Araç Rota planlama [39]

Araç rotalama probleminin birkaç çeşidi vardır.

### 3.1.1. Mesafe kısıtlı araç rotalama problemi

Mesafe kısıtlı Araç Rotalama Probleminde, her bir aracın gidebileceği maksimum mesafe kısıtlanmıştır. Burada kısıt, araçların belirli bir süreden fazla yolculuk yapmaması veya herhangi bir durumdan dolayı gecikmemesi olarak anlaşılır. Bazı durumlarda araç sayısı bile kısıtlı olabilmektedir. Bu durumlarda maksimum verilen süre geçilebilir. Böylece, her bir geçen süreye ceza maliyeti verilerek, toplam ceza maliyeti minimum hale getirilebilir [40].

### 3.1.2. Kapasite kısıtlı araç rotalama problemi

Kapasite kısıtlı araç rotalama probleminde her araç eşit kapasiteye sahip olup, müşteriye ürünler bir defada teslim edilmektedir. Bazı durumlarda kapasite kısıtlı araç rotalama probleminde sabit maliyet bulunmaktadır. Bu problem diğer problemler içinde en yaygın olanıdır. Burada asıl amaç eşit kapasitede olan araçların başlangıç noktasından son noktaya gitmesi için en kısa rotayı bulmaktır [42].

### 3.1.3. Zaman pencereci araç rotalama problemi

Zaman pencereci araç rotalama probleminde her bir araç için müşteriye dağıtım süresi önceden verilmekte olup bu süreyi aşmaması gerekmektedir. Eğer araç müşteriye gelmesi

gereken zamandan daha erken gelmişse belirtilen süreye kadar beklemesi gerekmektedir. Burada amaç toplam mesafeyi ve toplam süreyi en aza indirmektir [43].

#### **3.1.4. Yolların durumuna göre araç rotalama problemleri**

- Simetrik Araç Rotalama Problemleri.
- Asimetrik Araç Rotalama Problemleri.

Simetrik araç rotalama probleminde bir noktadan diğer noktaya gidiş-dönüş yolu aynıdır. Burada her bir araç için belli bir mesafe sınırı vardır.

Aracın başlangıç noktasından müşteriye kadar olan mesafenin, aynı müşteriden başlangıç noktasına kadar olan mesafeye eşit olmadığı durumdur. Bu durum asimetrik araç rotalama problemi olarak tanımlanır [44].

#### **3.1.5. Dinamik araç rotalama problemi**

Tüm ARP'lerde mesafe ve zaman bilinmekte ve değişmemektedir. Fakat bazı durumlarda, bu girdiler değişebilmektedir. Bu tip problemler dinamik araç rotalama problemi olarak adlandırılmaktadır.

#### **3.1.6. Statik araç rotalama problemi**

Statik araç rotalama probleminde araç kapasitesi, maliyet, rota, zaman gibi veriler önceden belirlenmiş olup, problem boyunca değişmeyecek şekilde kalmaktadır [45].

#### **3.1.7. Gezgin satıcı problemi**

Gezgin satıcı probleminde amaç bir satıcının başlangıç noktasından başlayıp her bir yere sadece bir kez uğradıktan sonra başlangıç noktasına geri dönerek en kısa ve en az maliyetli rotayı bulmasıdır. GSP çizge teorisi, şehirlerin noktalarla, yolların kenarlarla temsil edildiği bir çizge üzerinde en kısa Hamilton turunun bulunmasıdır. Hamilton turu bir çizge üzerinde başladığı noktadan itibaren her geçtiği noktadan sadece bir kere geçen, son noktaya vardıktan sonra başlangıç noktasına geri dönen, 19. yüzyılda yaşamış matematikçi William Hamiltonun adıyla anılan bir turdur [33, 46].

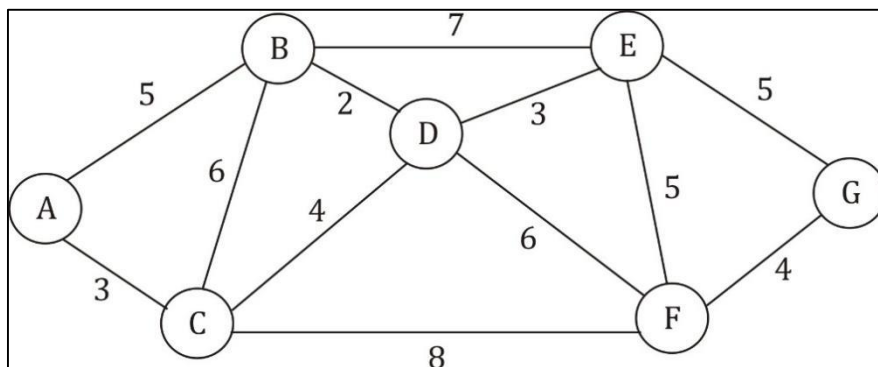
Literatürde rota planlama teknikleri geliştirmekle ilgili birçok algoritma mevcuttur.

Bunlardan en kısa yol algoritmasının temeli Dijkstra's algoritmasıdır. Dijkstra algoritması 1959'da Hollandalı bilgisayar bilimcisi Edsger Dijkstra tarafından geliştirilmiştir [47]. Dijkstra algoritması başlangıç düğümünden, diğer tüm düğümlere en kısa yolu belirler. Dijkstra algoritması şu alanlarda kullanılmaktadır;

- Grafik
- Mesafe
- Öncelik sırası
- Gevşeme ve mesafe başlatma [48]

Dijkstra algoritması, Greedy yaklaşımını kullanarak en kısa yolu belirler. Her seferinde bir düğümden diğer düğüme gitmesi Greedy yaklaşımı sayesinde gerçekleştirilir [49].

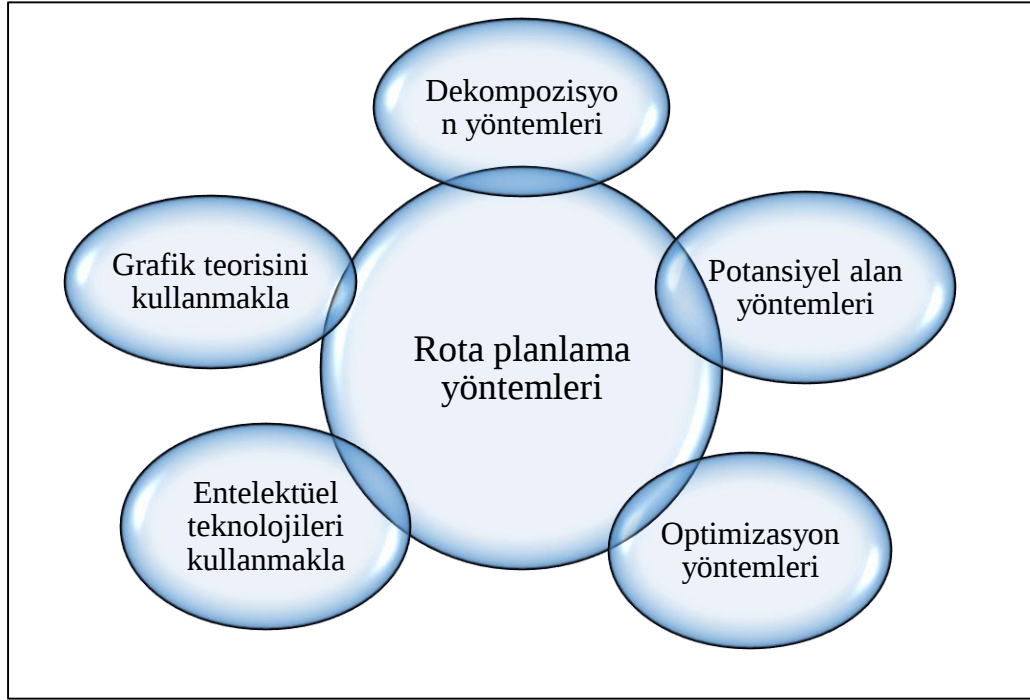
Clarke ve Wright 1964 yılında Dantzig ve Ramser algoritmasından esinlenerek araç rota problemi için tasarruf algoritmasını geliştirdiler. Algoritmada ilk önce gidilecek olan her bir nokta için farklı araçlar belirlenir. Daha sonraki adımlarda bu noktalar birleştirilir ve tasarruf oluşturulur. Özetle, başlangıçta gidilecek noktalar birbirine bağlanmadıysa araç bir noktaya gidip tekrar depoya döndükten sonra diğer noktaya gidecektir. Fakat noktalar birbirine bağlanmışsa araç sırayla her bir noktaya gidip, en sonunda depoya dönecektir. Böylece bu farka göre tasarruf oluşacaktır (Resim 3.3) [40].



Resim 3.3. Clarke ve Wright tasarruf algoritması [41]

### 3.2. Rota Planlama Yöntemleri

Rota planlama yöntemleri farklı kriterlere göre sınıflandırılabilir. Tarihsel bakış açısına göre, bu yöntemler geleneksel yöntemlere ve sezgisel yöntemlere ayrılabilir. Her ne kadar bu yöntemler ilk önce matematiksel kurallara uyarak oluşturulmuş olsa da yapay zeka teknolojilerinin kullanımı modern zamanlarda sezgisel yaklaşımlara tercih edilir. Bu durumda bilgilerin tam olması veya eksik olması ana koşulları oluşturur. Eğer rota ile ilgili tüm bilgiler tam ise o zaman problem, global planlama problemi ile çözülebilir, aksi takdirde, sorun belirli koşulları göz önünde bulundurarak yerel planlama problemi ile çözülür (Şekil 3.2).

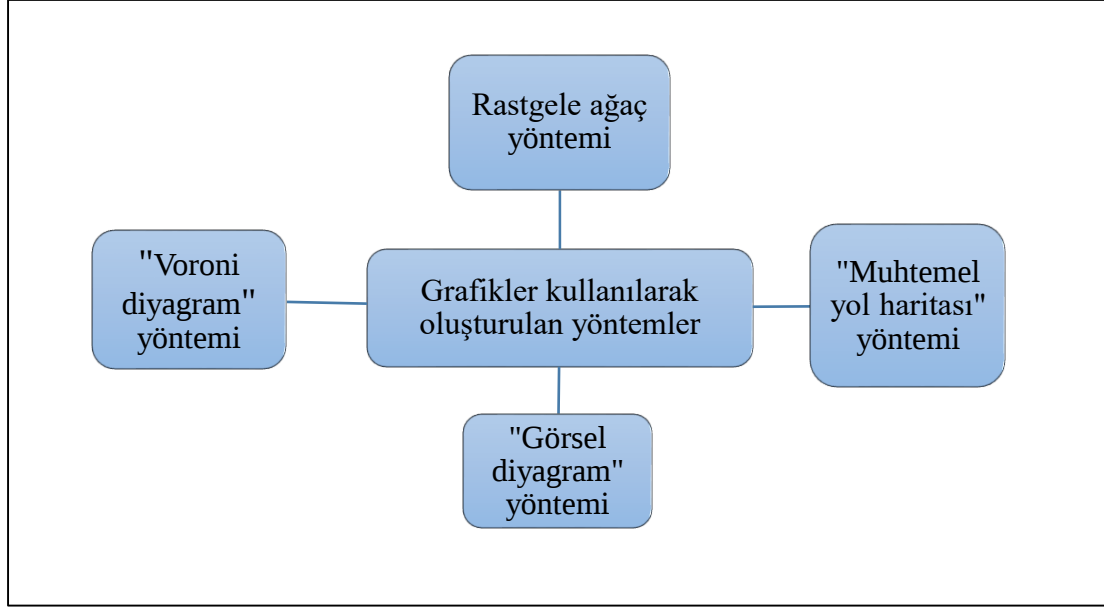


Şekil 3.2. Rota Planlama Yöntemleri

#### 3.2.1. Grafik teorisini kullanarak rota planlama yöntemleri

Bu yöntemlerin temel avantajı, anlaşılabilir olmasıdır. Bu özelliğinden dolayı bu yöntemler araştırmacılar tarafından daha sık kullanılmaktadır. Bu yöntemlerin her birinde İHA'nın durumu grafiksel olarak gösterilebilir. Bu durum İHA'nın koordinatı, uçuş yönü, hızı, ivmesi gibi değerler olabilir. Farklı durumlar arasındaki geçişler özel fonksiyonlar ile sağlanır. Çoğu durumlarda bu fonksiyon kayıp fonksiyonu olur. Bu durumda en az kayıpla

hedefe ulaşmak problemi çözülebilir [50]. Grafik teorisini kullanarak rota planlama yöntemlerinin birkaç çeşidi vardır (Şekil 3.3).



Şekil 3.3. Grafik teorisini kullanarak rota planlama yöntemleri

Huang ve Chung'un yaptığı bir makalede engellerin olduğu durumlarda grafikler kullanarak robotlar için yol planlaması problemine [51] odaklanmışlar. Bu makalede problemin çözümü için yol planlaması amacıyla dinamik görünürlük grafiği yöntemi önerilmiştir. Yöntem, engeller çokgen olduğu halde araştırılmıştır. Burada engelleri, planlanan yolun başlangıç ve son noktaları yönlendirilmemiş, grafik olarak verilmiştir. 1958 yılında yapılan mobil robotlar için hareket planlaması ile ilgili çalışmada [52] problemi çözmek için bir  $m$  vektörü kavramı önerilmiştir. Her iki çalışmada robotların engeller olduğu zaman yeryüzündeki gidişi modellenmiş olsa da, bu fikir İHA'ların rota optimizasyonunun belirlenmesinde de kullanılabilir. Bu çalışmalarda temel amaç, gidiş yolunda hangi engelin göz önünde bulundurulması, hangi engelin dikkate alınmaması olarak bakıldığında, bazen elde edilen sonuç en uygun rota olmayabilir. Habib ve Asaman'ın "Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach" çalışmasında ise yeni bir fikir "boş alan yapılandırma" yöntemi önerilmiştir. Burada gidiş yolunda "serbest şebeke, free links" terimi ile engellerin tepe noktalarını birbirine bağlayan, fakat onlarla kesişmeyen düz çizgi parçalarını isimlendirerek bu parçalardan çokgen oluşturulmuştur. Bu verilere dayanarak, grafik oluşturulmuş, aranan rota ise "grafiklerdeki yol arama algoritması" kullanılarak bulunmuştur [53]. Bu fikre dayanarak

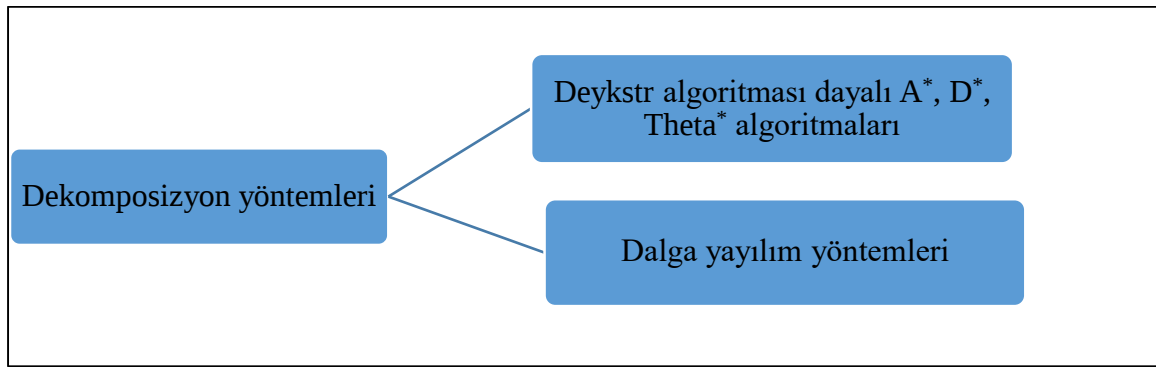
Wallgrun'un çalışmasında Varoni diyagramı kullanılarak aranan rota oluşturulmuştur. Önceki çalışmaların aksine burada sadece engellerden serbest bölgeler değil, engellerin karşılıklı durumunu göz önüne alarak onlardan maksimum mesafe geçerek, rotanın bulunması fikri öne sürülmüştür [54]. Engeller olmadığı halde rotanın bulunması için rastgele örnekleme yöntemi de kullanılabilir. Bu durumda muhtemel yaklaşım fikri kullanılır. Bu fikir ilk kez Ladd ve diğerleri tarafından "Muhtemel yol haritası algoritması" olarak ortaya çıkartılmıştır. Yöntemin ana fikri, verilmiş alana rastgele noktalar atılır. Eğer bu noktalar engelin üzerine düşerse, bu nokta reddedilir ve kümeye eklenmez. Böylece, engellerin olmadığı noktalardan oluşan "boş alan" oluşturulur. Fakat, bu yöntemle bulunan rota optimum olmayabilir [55].

Rastgele örneklemeye dayanan yöntemlerden biri de "Rastgele orman yöntemidir". Bu yöntemde rotanın oluşturulması yöntemi başlangıç noktasından son noktaya kadar ardışık genişleyen rota ağacından oluşur. Bir ağaç oluşturma işlemi, dalı olmayan (kenarları olmayan), yalnızca başlangıç tepe noktalarında oluşan ağaçtan başlar. Uygulama sürecinin sonraki adımlarında yeni bir rastgele yol seçilir. Eğer engeller bu yoldan uzak olursa bu yol seçilir, aksi takdirde rastgele örnekleme devam eder. Eğer bu yol bulunmazsa, rastgele noktalar kaldırılır. Seçimler arasında en uygun mesafe olarak en kısa olan mesafe kaydedilir ve ana ağaca dal olarak eklenir. İşlem son amaca veya dalların maksimum sayına ulaşana kadar devam eder. Rastgele orman yönteminin çeşitli farklılaştırılmış metotları önerilmiştir. Bunlardan birisi Amato ve Wu'nun yaptığı bir çalışmada [56] önerilen yöntemdir. Burada, bir sonraki dalı oluştururken bitiş noktasının konumunun dikkate alınması önerilmiştir. Yapılan başka bir çalışmada [53] bu yöntemi iki yönde, hem başlangıç noktasından hem de son noktasından uygulayarak oluşturulması önerilmiştir. Yukarıdakileri özetleyerek, grafikler teorisine dayanan yöntemlerin sadece çevrenin değişmeyen, statik olduğu durumlarda uygulanması daha iyidir. Bu yöntemin uygulanmasında engeller hakkında tam bilgi gerekmektedir. Eğer cisimler dinamik ise grafikleri kurma zorluğundan dolayı bu yöntemin uygulanması mümkün değildir.

### **3.2.2. Dekompozisyon yöntemlerini kullanarak rota planlama**

Engellerin olduğu durumlarda rota planlama konusunun çözüm yollarından birisi de çevrenin dekompozisyon (ayırıştırma) yöntemi ile bu problemin çözülmesidir [57]. Bu çözümler yaklaşık ayırıştırma yöntemi ve kesin ayırıştırma yöntemi olarak iki gruba ayrılabilir.

Yaklaşık ayrışma yöntemlerinden en yaygın olanı, alanı kapsayan bir ağ (ızgara haritası) oluşturmak için önerilen yöntemdir. Bu yöntem Elfes A. tarafından önerilmiştir [58]. Bu yöntemin temel fikri, tüm alanı aynı boyutta hücrelere ayırarak engel olan hücreyi 1, engel olmayan hücreyi ise 0 olarak isimlendirerek ağın oluşturulmasına dayanır. Bu yöntemin dezavantajı ağın boyutları küçüldükçe, hesaplama işlemlerinin artmasıdır. Bu eksiklikler [59, 60, 61] çalışmalarında iki boyutlu olduğunda kareler (quadtree), üçboyutlu olduğu durumda ise küp (octree) ağaçları kullanarak onları ortadan kaldırmaya çalışmaktadır (Şekil 3.4).



Şekil 3.4. Dekompozisyon yöntemleri

Kesin çözüm yöntemleri engelin sınırlarını göz önünde bulundurarak çözümün arandığı bölgenin gridlere bölünmesine dayanır. Bu durumda gridler (hücre) dışbükey çokgen şeklinde olur. Chazelle ve Palios yaptıkları çalışmada çokgenleri dikey ve ikizkenar yamuk şeklinde ayırmakla problemin çözümü için bir yaklaşım önermiştir [62].

Ayrıştırma yöntemlerinde problemin çözümü için grafik ağaçlarını oluşturduktan sonra grafiklerde en optimum yolun bulunması ile problem çözülür. Bu çözümler içinde klasik arama yöntemleri, geniş alan araması, derin arama, kısıtlama olduğunda derin arama, ikili arama gibi yöntemler gösterilebilir [63].

Klasik yöntemlerden farklı olarak, sezgisel araştırma yöntemleri son yıllarda daha yaygın hale gelmiştir. Bu yöntemler arasında 1968'de Peter Hart, Nilsom Nilson ve Bertram Raphael tarafından önerilen A\* algoritması yer almaktadır. Bu algoritma, 1959'da Dykrist tarafından önerilen algoritmanın genişletilmiş bir versiyonudur. Bu algoritmanın iki amaç fonksiyonu vardır:

- Son noktaya ulaşmanın değer fonksiyonu
- Başlangıç noktasıyla son nokta arasında mesafenin sezgisel fonksiyonu.

Sezgisel fonksiyon, mesafenin azalması yönünde oluşturulmalıdır. Bu yöntemlerin dezavantajı, onların ağ yapımına bağlı olarak yön seçiminde sınırlamalarının olmasıdır. Bu eksiklikleri D\* algoritması ile ortadan kaldırmaya çalışmışlardır [64]. D\* algoritması, A\* algoritmasının yeni bir versiyonudur. Burada hücreler dinamik olarak yeniden oluşturularak A\* algoritmasından daha az hesaplama yapılır. Bu yöntem, Koenig ve diğerlerinin [65, 66] çalışmasında önerilmiştir.

A\* arama algoritması grafik geçişi ve yol bulmada kullanılan bir algoritmadır. Bilgisayar bilimlerinde en kısa yolu bulmak için kullanılan diğer algoritmalar arasında en iyisidir. Başlangıç düğümünden hedef düğüme giden ve en kısa yolu bulan algoritmadır. Peter Hart, Nils Nilsson ve Bertram Raphael 1968’de A\* arama algoritmasını keşfettiler. Bu algoritma Dijkstra’nın algoritmasının bir uzantısıdır [67].

A\* ve Theta\* algoritmalarını kullanarak İHA’ların üç boyutlu uzayda De Philippis tarafından [68, 69, 70] incelenmiştir. Alvarez ve diğerleri uçakların dinamik özelliklerini dikkate alarak A\* algoritmasının kinematik A\* algoritmasını önermiştir.

Dekompozisyon yöntemleri arasında “Dalga yayılımı yöntemi” ve “Hızlı yürüyüş yöntemi” veri tabanında oluşturulmuştur. Bu yöntemler arasındaki en önemli fark, bir arama aşamasının olmamasıdır. Yöntem çevrede dalgaların yayılımına dayanır:

Eğer muhit homojen ise dalgalar daire boyunca yayılır, ancak muhit homojen olmadığında (engeller olduğunda) dalgalar daire boyunca yayılmayacaktır. Böylece bu yöntemler kısmi türevli diferansiyel denklemler, dalga denkleminin verilen başlangıç ve sınır koşulları dahilinde çeşitli yöntemlere dayanarak önerilmiştir [71].

### 3.2.3. Potansiyel alan yöntemleri ile rota planlama

Rota planlama yöntemleri en yaygın potansiyel vektör alanlarının özelliklerine dayanan yöntemlerdir. Potansiyel fonksiyonlar tipine göre sanal kuvvet alanı [72], Newton potansiyel alanı [73], harmonik vektör alanları [74] olarak belirtebiliriz. Yöntemlerin amacı şu şekildedir;

Söz konusu olan alan iki kısma ayrılmıştır. Hedef noktası çekici vektör alanı gibi engeller ise, itme vektör alanı olarak seçilir. Bu iki alanın birleşimi, hedef noktaya ulaşmayı ve engelleri aşmayı sağlayacaktır. Yöntemde muhit, son nokta ve engeller statiktir. Bu yöntemin dezavantajı ise çözümlerde yerel minimumunun olmamasıdır (yerel minimumda optimum olmayan fakat, minimum şartları sağlayan ara aşama kastedilmektedir).

Bazı çalışmalarda [75, 76], bu sorunları gidermek için çalışılmış olsa bile problem sadece özel durumlarda çözüldüğü için bu sorun hala açık olarak kalmıştır. Borenstein ve Koren “sanal güç alanı” yöntemini önermiştir [77]. Bu yöntemde [78] potansiyel alan yöntemlerini ağ yöntemiyle birleştirmişlerdir ancak engeller küçük olduğunda bu yöntem uygulanamaz. Bu eksikliği gidermek için Borenstein “vektör alan histogramı” (VFH) [79] yöntemini ve bu yöntemin değiştirilmiş olan VFH+ yöntemini [80] ve VFH\* yöntemini [81] önermiştir.

#### **3.2.4. Optimizasyon yöntemleri ile rota planlama**

Engellerin şekilleri ve konum yöntemleri birleşik olan muhitte rota problemleri, optimizasyon problemi olarak çözülebilir. Bu durumda İHA’lar dinamik nesnelerin hareketi olarak kabul edilir. Engeller kısıtlayıcı koşullar olarak verilir, olası rotalarsa belirli fonksiyonlarla ifade edilir. Sonuç olarak, rota planlama problemi optimal kontrol olarak çözülür ve onun çözümü sadece engelleri aşmak değil, aynı zamanda zamana, hıza veya enerji verimliliğine göre optimum çözümü bulmayı sağlar [82, 83].

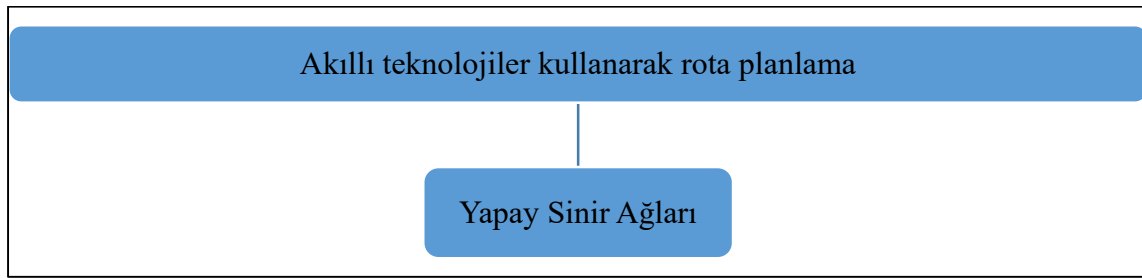
Optimizasyon probleminin çözümü sayısız hesaplamalar gerektirir ve sadece doğrusal olduğu durumlarda bu gerçekleştirilebilir. Bu nedenle, bu tür sorunlar genellikle sayısal yöntemlerle çözülür. Bu yöntemler düz çizgi yöntemleri ve dolaylı yöntemler olarak ikiye ayrılır. Bilindiği üzere optimum kontrol probleminin çözümü için gerekli olan koşul Pontryagin’in maksimum prensibini sağlamalıdır. Bu prensibin uygulanması diferansiyel bir denklemde ifade edilir. Diferansiyel denklemin çözümü herhangi bir sayısal yöntemle gerçekleştirilir. Bu durumda, diferansiyel denklemi çözmek için kullanılan yöntem dolaylı yöntem denir, çünkü bu durumda minimum bulma problemi çözülmez, önce diferansiyel denklem çözülür, sonra elde edilen fonksiyonun minimumu bulunur. Düz çizgi yöntemlerinde hedef fonksiyonun minimum miktarı doğrudan yöntemin içinde bulunur. Fakat, bu yöntemler sadece özel durumlarda ve amaç fonksiyonunun özel ayırık durumlarında uygulanabilir. Bazı durumlarda rota planlama problemi optimizasyonunu hem

kısıtlayıcı koşullarla, hem de amaç fonksiyonu sonlu parametrelerle doğrusal olmayan programlama problemine dönüştürülebilir [84]. Bu durumda problemi mevcut uygulama yazılım paketleriyle, örneğin Matlab’da “Toolbox Optimization” ve “SNOPT” [85], “NPSOL” [86] paketleriyle kullanılabilir. Çoğu durumlarda rota planlama problemi doğrusal programlama olarak belirlenir ve çözülür. Problemde bazı parametreler tam sayılı (İHA’ların sayısı, engellerin sayısı vb.), diğerleri ise seçmeli olduğu için bu problemler karma tip programlama olarak ele alınmaktadır [87, 88]. Pratik çözüm sürecinde herhangi bir zamanda İHA’nın durumu için ayrık zaman hesaplaması gerekmektedir. Sonuç olarak her bir zaman anı için doğrusal programlama problemi ele alınır [89].

Zamanın ayrıklaştırılması durumunda engeller dikkate alınmamaktadır. Eğer rota engelle karşılaşır, o zaman bu engel göz ardı edilir. Bu eksikliği gidermek için ayrışma sayısını çoğaltmak gerekir, bu ise ek hesaplamalara yol açar. Başka bir çalışmada ise [90] tahmin yöntemleri kullanılarak bu eksiklik giderilmeye çalışılmıştır. Bazı durumlarda, problem ikili tam sayıya sahip bir programlama problemi olarak çözülür [91].

### 3.2.5. Entelektüel teknolojiler kullanarak rota planlama

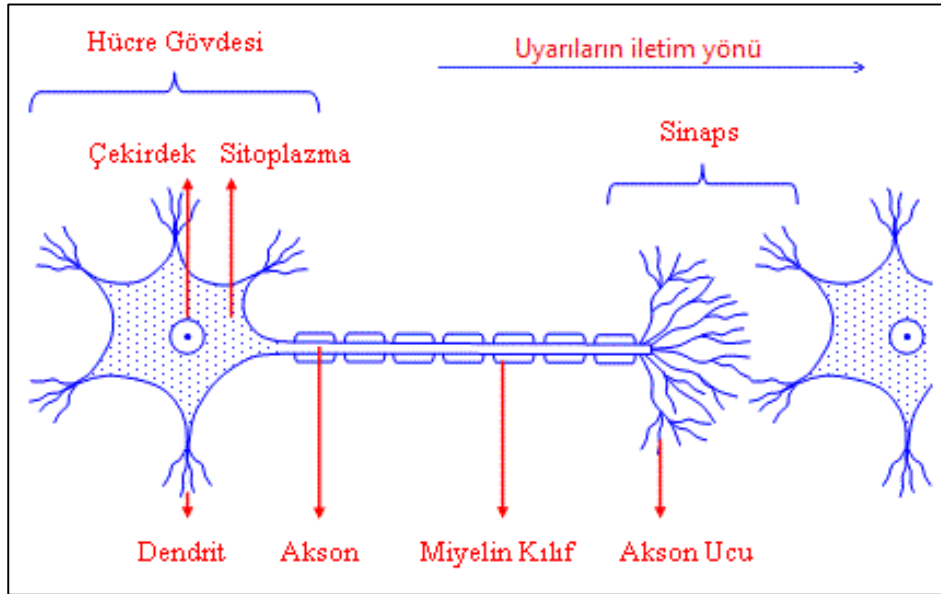
Rota planlama yöntemlerinden biri de Entelektüel teknolojiler kullanarak rota planlama yöntemidir.



Şekil 3.5. Entelektüel teknolojiler kullanarak rota planlama sistemi

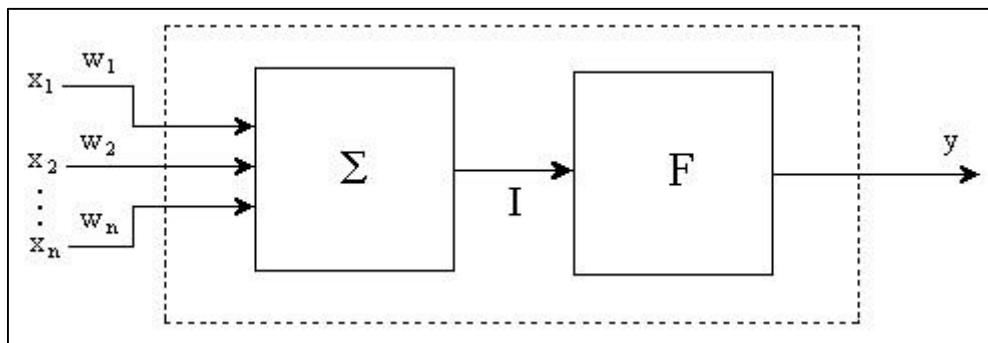
Yapay Sinir Ağları (YSA) beynimizin sinirsel yapısının bilgisayar yazılımları ile benzeştirilmesidir. Birçok sinir hücresi türü, genel olarak nöron terimi altında birleştirilir. Nöron temelde, organizmanın işlevsel yönetimi olan elektriksel aktiviteye sahip özel şekilli bir hücre olmasıdır. Beyin bir tür organik anahtar gibi nöronlar içermektedir. Bir insanın beynindeki sinir ağı, herhangi bir nöronun çıktısının, binlerce diğer nöronun girdisi olabileceği birbiriyle çok bağlantılı bir nöron ağıdır. Nöronlara ayrıca girdi (dendrit) ve

çıktı (akson ve uçları) ağacı dahildir. Aksonun son dalları başka hürelere sinirsel bağlantılarla (sinapslar) birleşir. Aksondan ayrıca diğer hücreleri birbirine bağlayan çıkıntılar, kollaterallar çıkar. Dendritik ağacının giriş sinyalleri (postsinaptik potansiyeller) aksonun başlangıç bölümüne doğru toplanır ve sonuç olarak çıkış impulsu elde edilir (Resim 3.4) [27, 92].



Resim 3.4. Biyoloji nöron [93]

Nöronları birbiriyle belirli bir şekilde birleştirerek yukarıda bahsedilen modellere dayanarak bir sinir ağı inşa etmek mümkündür (Şekil 3.6).



Şekil 3.6. Nöronun matematiksel modeli

Yapay sinir ağıları İHA'ların rota optimizasyonu ile ilgili [94] çalışmasında bakılmıştır. Burada İHA'ların rota planlamasının her prosedürü için bir sinir ağı geliştirilmiştir. Bunlar aşağıdakilerdir;

- Durumun genel stratejik sınıflandırması,
- Her bölgeye düşen nesnelerin sayısının sayılması,
- Öncelik bölgesinin belirlenmesi,
- Öncelikli bölgedeki bir nesnenin seçimi, öngörülmüştür.

Sinir ağlarının bu durumda görevleri şu şekildedir, sinir ağı önce x eksenini boyunca İHA'ya en yakın rotayı belirler. Daha sonra rotanın seçilen noktasının İHA'nın görüş alanına düşüp düşmediğini kontrol eder. Öge bu alanda ise, bu ögeyi dikkate almaz ve bir sonrakine geçer. Bu aşamaların her biri için farklı sinir ağları geliştirilerek İHA'lar için en az maliyetli uçuş rotası belirlenmiştir [95].

### 3.3. İnsansız Hava Araçlarının Rota Optimizasyonunda Kullanılan Algoritmalar

İHA'ların rota optimizasyonunda çeşitli algoritmalar kullanılmaktadır. Bunlar aşağıdaki gibi sıralanabilir.

#### 3.3.1. Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu (PSO), kuş sürülerinin davranışından ilham alan popülasyon bazlı olasılıksal bir optimizasyon yöntemidir. 1995 yılında Russell C. Eberhart ve James Kennedy tarafından sunulmuştur. Algoritma, kuş ve balık sürülerinin yaşam tarzının öğrenilmesi sonucunda oluşturulmuştur (Resim 3.5, 3.6). Sürü algoritmasının evrimsel ve genetik algoritmalar ile benzerlikleri vardır. Bu algoritmanın avantajları;

- Hızlı birikim
- Ayarlanabilir parametrelerin az olması
- Geniş alanlarda bile kolay arama imkanı olmasıdır.

Sürü algoritmalarının prensipleri ise:

- Her bir eleman (sürü bireyi) verilen ana kadar ulaştığı en iyi durum (pbest) özel koordinat olarak işaretlenir.
- Diğer iyi durumlar (gbest) aynı elemanın etrafına ulaşan diğer elemanlar olur.

Algoritmanın temel fikri, her bir elemanı belirli bir ağırlıkta bir pbest ve gbest durumuna getirmektir.

$$v_i \leftarrow \omega v_i + \varphi_p r_p (p_i - x_i) + \varphi_g r_g (g - x_i) \quad (3.1)$$

$x_i$  – verilen eleman

$p_i$  – elemanın belirlenen en iyi durumu

$g$  – sürünün en iyi konumu

$v_i$  – elemanın hızını düzenli dağılımı

$r_p$  ve  $r_g$  -  $rnd(0,1)$

$\omega$ ,  $\varphi_p$ , ve  $\varphi_g$  – kullanıcı tarafından seçilir ve yöntemin verimliliğini sağlar.

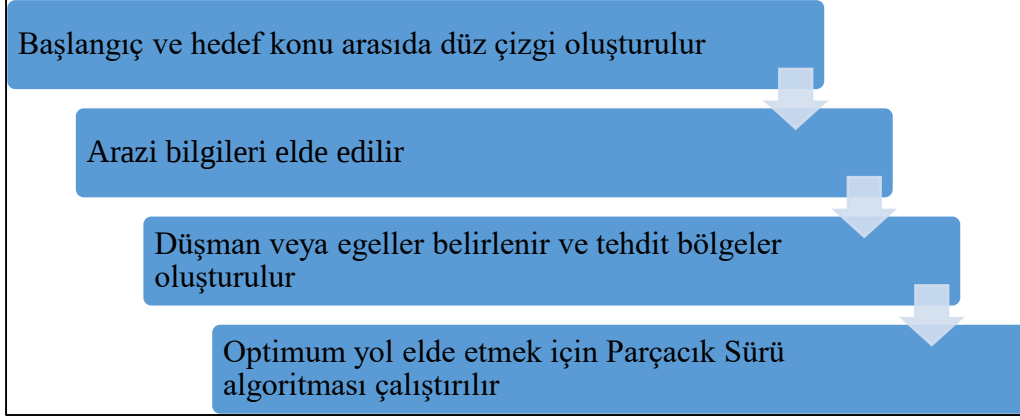
$$v_{it+1} = x[v_{i,t} + \varphi_p r_p (p_i - x_{i,t}) + \varphi_g r_g (g_i - x_{i,t})] \quad (3.2)$$

$\varphi = \varphi_p + \varphi_g$ ,

$\varphi > 4$  [96].

Parçacık sürüsü algoritması kullanılarak İHA'ların rota planlama problemi konusu Jung ve diğerleri tarafından araştırılan çalışmada ele alınmıştır. Bu çalışmada parçacık sürüsü optimizasyonu (PSO) kullanılarak üç boyutlu yol planlama problemi formülasyonu ve çözüm yaklaşımı sunulmuştur. Problem formülasyonu, aynı zamanda yakıt tüketimini en aza indirirken, düşman tehditleri nedeniyle oluşacak olan riski en aza indirmek için tasarlanmıştır. PSO'yu kullanarak, optimize edilmiş bir yol B-spline eğrileri yardımıyla

oluşturulur. Ortaya çıkan yollar, maksimum güvenlik, minimum yakıt tüketimi veya ikisinin bir kombinasyonunun tercihiyle optimize edilebilir. Burada sınırlamalar olarak engeller, amaç fonksiyonu olarak yakıt tüketimi ele alınmıştır (Şekil 3.7) [97].

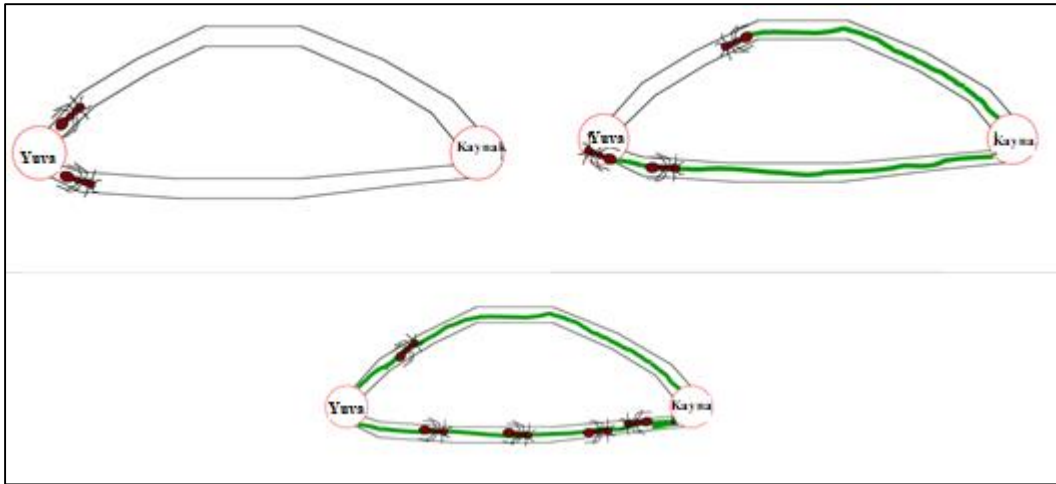


Şekil 3.7. Parçacık sürü algoritması kullanılarak yol planlama probleminin akış şeması

### 3.3.2. Karınca kolonisi algoritması

Karıncaların besine doğru en kısa yolu bulma özellikleri aşağıdaki gibidir.

İki karınca farklı yollarla besine doğru ilerler. Kısa yol ile ilerleyen karınca besine en erken varır. İlk karınca yuvaya vardığında, ikinci karınca daha yolun yarısında olur. Bir sonraki karınca besine doğru ilerler. Birkaç tekrardan sonra en kısa yol ile daha fazla karınca ilerlemiş olacaktır.



Resim 3.5. Karıncaların besine giden en kısa yolu bulma şekli

Belli bir süre sonra tüm karıncalar en kısa yolda ilerlerler. Karınca algoritmalarının temelinde karınca kolonisinin yaşam tarzının modellenmesi dayanır:

- Her bir karınca kolonide basit bir harekete sahiptir.
- Kolonide merkezi kontrol yoktur.
- Bilgi alışverişi özel bir sıvı olan feromonlarla yapılır.
- Zamanla feromonun azalması hareket adaptasyonunu sağlar.
- Sürü yuvadan besine kadar olan en yakın mesafeyi bulabilir.
- Eğer önceden bulunan kısa yol kapalıysa, sürü yeni bir kısa yol bulabilir [50].

Karınca Kolonisi ilk defa 1992’de Marko Dorigo tarafından en uygun yolun bulunması problemini çözmek için önerilmiştir [98]. Algoritma doğada karıncaların besin arayışı sürecinin modelleştirilmesi sonucunda oluşturulmuştur. Besin arayışında karıncalar önceden rastgele gider ve besini bularak kolonisine geri döner. Bu sırada geçtiği yolu özel kokuya sahip feromonla kaydeder. Belli bir zaman içinde feromon buharlaşmaya başlar ve böylece çekicilik gücü azalır. Besine varma ve geri dönme, harcanan zamana bağlı olarak feromonun buharlaşması farklıdır. Mesafe küçüldükçe kalan feromon daha çok ve kokusu daha keskin olacaktır. Bu koku diğer karıncaları da çekecektir ve sonuç olarak feromon karıncalar için en kısa rotayı belirleyecektir [89]. Son zamanlarda karınca kolonisi algoritmasının bazı üst sezgisel modelleri önerilmiştir. Onlar arasında en çok kullanılan ACS (Ant Colony System), MMAS (Max-Min Ant System) [99], ACO (Ant Colony Optimization) ve diğerleri gösterilebilir [88, 100, 101]. Böylece Karınca Kolonisi Algoritması en uygun rotanın belirlenmesi problemini ele alır. Karınca Kolonisi Algoritmasının bu özelliği onun birçok pratik taşıma probleminin konularını [102], şehir içi ulaşım optimizasyonlarını [103], imalat optimizasyon konularını [104] ve diğer problemlerin çözümünü sağlar [105].

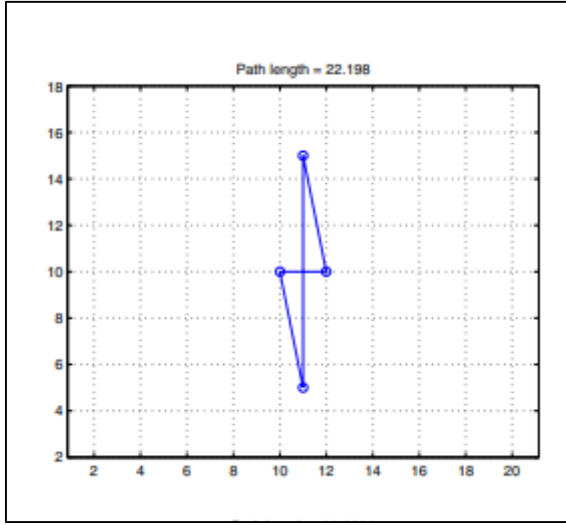
Karınca Kolonisi Algoritmasını kullanarak İHA’ların rota planlama problemi ile ilgili Murat Karakaya’nın “En Az Sayıda İnsansız Hava Aracı Kullanarak Sabit Hedeflerin Gözetlenmesinin Planlanması” çalışmasında bakılmıştır [106]. Bu çalışmada amaç, en az sayıda İHA kullanarak Karınca Kolonisi Eniyileme (KKE), meta-sezgiseli tabanlı bir çözüm yöntemi ile İHA’lar için uygun bir rotayı belirlemektir. Önerilen algoritma, KKE yöntemi ile konumun başlangıç ve bitiş noktaları, kısıtları girdi olarak alarak tek İHA ile en çok sayıda hedefe gidecek rota belirlenmiştir. Daha sonra ise hedefler listesinden bu İHA’nın gittiği yolları çıkararak kalan diğer hedefler için diğer İHA ile aynı adımları uygulanmıştır.

Hedef noktalar boşaldığında en az sayıda İHA için rota bulunmuştur. Sonuç olarak deneyde kullanılan İHA sayısının %20 azaldığını ortaya çıkmıştır [106].

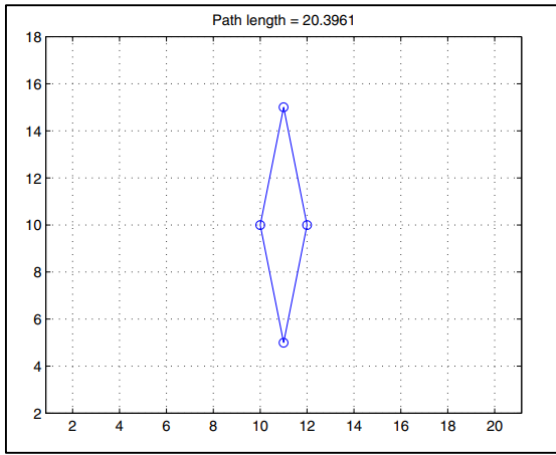
Jevti ve diğerleri tarafından geliştirilen makalede [107] İHA'ların en uygun rotayı bulması için Karınca Sistem algoritması uygulanmıştır daha sonra “En yakın komşu algoritması” ile karşılaştırılmıştır. Algoritmada her karınca hücre sayısına eşittir ve her karıncanın başlangıç noktası farklı hücrelerdedir. Her kenar feromon izi ile aynı değerde başlatılır. Böylece kenarların seçilmesi için ilk olasılıklar yalnızca kenarların uzunluklarına bağlıdır. Dğümler arasındaki kenarlar İHA'ların uçuş dinamiklerini hesaba katmadan bir konumdan diğerine hareket etmek için gittiği yollarıdır. Bu durum karıncaların ilk iterasyonda iyi çözümler bulmasına yardımcı olur.

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\mu_{ij})^\beta}{\sum_{h \notin tabu_k} (\tau_{ih})^\alpha (\mu_{ih})^\beta} & \text{if } j \notin tabu_k \\ 0, & \text{otherwise} \end{cases} \quad (3.3)$$

Karıncalar yukarıdaki denklemlerle düğümden düğüme hareket ederler. Son düğüm ile başlangıç düğümü arasındaki mesafe toplam tur uzunluğuna eklenir. Her iterasyonda karıncalar yeni çözümler üretir ve en iyi tur iterasyonu en iyisi olarak kaydedilir. Daha sonra yeni iterasyonla önceki iterasyonların en iyi çözümleri karşılaştırılır. Maksimum iterasyon sayısına ulaşıldığında algoritma durur. Daha sonra en yakın komşu algoritmasıyla bulunan rota ile karınca sistem algoritmasıyla bulunan rotayı karşılaştırılır (Şekil 3.8, 3.9). Deneylerden çıkan sonuçta karınca sistem algoritmasıyla bulunan rotanın en iyi rota olduğu ispat edilmiştir [107].



Şekil 3.8. En yakın komşu algoritmasıyla bulunan rota



Şekil 3.9. Karınca sistem algoritmasıyla bulunan rota [107]

Ma ve diğerleri “Improved Ant Colony Algorithm for Global Optimal Trajectory Planning of UAV under Complex Environment” çalışmasında [108] KKA algoritması kullanarak tehlikeli araziler olduğunda İHA’lar için hedefe gidilecek rotayı belirlemişlerdir. Bu algorithma rüzgar hızı önceden bilinmektedir ve İHA’nın uçuş yüksekliği ise sabittir. Engeller radar yardımıyla belirlenir. Algoritmanın başlangıcında tüm karıncalar başlangıç noktasındadır. Daha sonra karıncalar düğümü seçer ve hedefe ulaşır. Her karıncanın maliyet fonksiyon değeri hesaplanır ve iterasyon koşulunu karşılayıp karşılanmadığına karar verilir, eğer karşılanırsa, iterasyon sonlandırılır ve en iyi çözüm elde edilir. Aksi takdirde, iterasyon koşulu karşılanana kadar algoritma yeniden hesaplanır. Deneysel sonuçlar, önerilen iyileştirilmiş KKA’nın temel KKA algoritmasından daha verimli olduğunu doğrulamıştır. Gelecekte yapılacak olan çalışmalarında ise karmaşık ortamlarda İHA’nın dinamik küresel

en uygun rota planlamasını çözmek için KKA algoritmasını geliştireceklerini belirtmişlerdir [108].

### 3.3.3. Arı algoritması

Yapay Arı Kolonisi Algoritması ile ilgili ilk defa 2005 yılında Dr. Karaboğa, meta-sezgisel popülasyona dayalı bir sürü algoritmasını geliştirdi [110]. Yapay Arı Kolonisi Algoritması bal arılarının yiyecek arama davranışlarından ilham almıştır. Bal arısı sürüsünün, arıların bilgileri iletebilmesi, çevreyi ezberleyebilmesi, bilgileri saklayabilmesi, paylaşabilir ve buna dayanarak kararlar alabilmesi gibi birçok niteliği vardır. Arı, besin ararken, kendine özgü bir çiçek seçer. Bu besin kaynağından arı, besin kaynağının içerdiği nektarın miktarı, bu nektarın çiçekten ne kadar kolay çıkarılabileceği, yuvadan ne kadar uzakta olduğu ve hangi yönde olduğu ile ilgili bilgileri toplar. Arı, bu durumları kolay olması için hafızasına tek bir veri olarak kaydeder. Bu veri yiyecek kaynağının toplam karlılığı olarak adlandırılır. İşçi arılar topladıkları bilgileri diğer arılarla paylaşır. İşsiz yiyecek arılar ise işçi arılardan topladıkları bilgileri toplamaktan ve elde edecek gıda kaynağını seçmekten sorumludur. İşsiz arılar gözcü arılar ve keşif arıları olarak iki gruba ayrılır. Keşif arıları kovandaki arılardan bilgi toplayıp kendilerine yeni gıda kaynağı arayışındadırlar. Gözcü arılar ise çevresinde yeni gıda kaynağı bulmasından sorumludur. İşçi arılar kovandaki diğer arılara besin kaynağının ne kadar uzakta ve hangi yönde olduğunu, besin kaynağının ne kadar bol olduğunu “dans” adı verilen adımlarla gösterir [109]. Bu algoritmada bir kolonide işçi arı, gözcü arı ve keşif arısı vardır. Burada arıların yiyecek için gittiği yerler problemin çözümlerini, buradaki nektarlar ise amaç fonksiyonu olarak tanımlanır. Arı kolonisi algoritmasının adımları aşağıdaki gibidir:

Başlangıç, hedef yiyecek kaynaklarının yerlerinin bulunmasıdır. Keşif arılar ilk önce bu kaynakları rastgele ararlar.

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (3.4)$$

Burada kaynak sayısı  $i$ , parametre sayısı ise  $j$ -dir.  $Rand(0,1)$ , 0 ve 1 arasında rastgele sayı üretmektedir.  $xmin$  ve  $xmax$  alt ve üst sınır değerleridir.

İşçi arılar besin kaynaklarından birini rastgele olarak seçerler. Seçtikleri bu kaynağın çözüm değerini hesaplarlar. Yeni kaynaktaki değer, eski kaynaktaki değerden daha iyiye hafızaya alır ve eski değeri hafızalarından silerler. Kaynağın bulunması denklemi:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (3.5)$$

ile gerçekleşir. Kaynağın değeri hesaplandıktan sonra kaynağa uygunluk değeri ( $fitness_i$ ) uygulanır. Devamında  $x_i$  ve  $v_i$  arasında uygunluk değerine göre açgözlü (greedy) seçimi uygulanır. Eğer bulunan  $v_i$  çözümü daha iyiye bu çözümü yeni kaynak olarak kullanır, eski kaynağı hafızasından siler ve geliştirme sayacı sıfırlanır. Aksi takdirde arı eski kaynağa yeniden gider ve geliştirememeye sayacı bir artır. Gözcü arılar elde ettikleri bilgilerle orantılı bir kaynak seçerler ve (3.6) ile hesaplanmış değeri (0,1) arasında rasgele bir sayı ile karşılaştırırlar. Eğer hesaplanmış değer bu rastgele değerden büyük ise gözcü arılar denklem (3.5) yardımıyla yeni bir çözüm oluşturur ve oluşturduğu çözümün kalitesini hesaplar. Eski çözüm ile yeni oluşturulmuş çözüm, aralarında açgözlü seçim yöntemi ile karşılaştırılır.

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (3.6)$$

$p_i$ , i'nci çözümün olasılık değeri,  $fitness_i$  ise i. çözümün uygunluk değeridir. Arıların hepsi görevlerini tamamladıktan sonra çözüm geliştirme sayacı kontrol edilir. Bu sayaç sayesinde arının besin kaynağında nektarın tükenip tükenmediği, kaynağa arının gidip gitmediği belirlenir. Eğer bu sayaç belirli bir değerin üzerindeyse keşif arıları (3.4) denklemi ile yeni rastgele bir yiyecek kaynağı ararlar. Bu aramalar önceden tanımlanan kriter sayısına ulaşıncaya kadar devam eder ve algoritma sonlandırılır [110].

Çavuş ve Tuncer'in "İnsansız Hava Araçları İçin Yapay Arı Kolonisi Algoritması Kullanarak Rota Planlama" çalışmasında [111] amaç fonksiyonuna ek olarak engel maliyeti de eklenmiştir. İki nokta arasında mesafeyi hesaplarken "Haversine Algoritması" yardımıyla dünyanın geometrik şeklini göz önünde bulundurmuşlardır. Rota üzerinde  $(x_1, y_1)$  ve  $B(x_2, y_2)$  engelin olup olmadığını belirlemek için ise

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_2}{x - x_2} \quad (3.7)$$

denklemini kullanılmıştır. İHA'ların rota planlamasında arayüz geliştirilmesi için C# programlama dili kullanılmıştır. Geliştirilen bu arayüzde haritaları kullanmak için ise Google gMapControl kullanılmıştır. Daha sonra başlangıç ve bitiş noktaları, engeller haritada belirlenmiştir. Engellere rastlanıldığında İHA'lar kendi etrafında dönerek yükselmiş ve engelleri geçmiştir. Yapay arı algoritması yardımıyla geliştirilen algoritmanın İHA'lar için en uygun rota olduğu belirlenmiştir [111].

Li ve diğerleri “Engeller olduğunda Yapay Arı Algoritması ile İHA'ların optimal yörüngesinin bulunması” konusunda çalışmışlardır [112]. Burada deneyler MATLAB program ile uygulanmıştır. Algoritmanın verimliliğini ve sağlamlığını araştırmak için üç farklı simülasyon durumu araştırılmıştır. Her bir deney farklı rastgele 50 kaynağa gitmiştir. Simülasyon sonuçlarının diğer geleneksel arı algoritmalarından daha verimli sonuç elde ettiği gösterilmiştir [112].

Dorling ve diğerlerinin yaptıkları bir çalışmada iki problemin konusuna bakılmıştır. Bunlar gönderim sınırı ile ilgili minimum maliyet ve bütçe kısıtlamaları dikkate alınarak minimum teslim süresidir. Makalede enerji kullanımı için doğrusal bir model önerilmektedir. Burada amaç önce masrafları en aza indirmektir. Bunun için, toplam süreyi ele almak mümkündür. Örnek olarak, tüm teslimatların toplam süresine bakılmıştır [11].

### **3.3.4. Süpürme algoritması**

Süpürme algoritması ilk defa 1971 yılında Gilllet ve Miler tarafından geliştirilmiştir. Süpürme algoritmasında müşterilerin ve deponun koordinatları önceden belli olması gereklidir. İki aşamalı bir yöntemdir. Süpürme algoritmasında depo polar koordinat sisteminin merkezinde, gidilecek noktalar ise polar koordinat sistemi üzerinde gösterilmektedir [113].

### **3.3.5. Tabu arama algoritması**

Tabu arama algoritması 1986 yılında Fred Glove tarafından geliştirilmiştir. Yerel arama algoritmasından farkı ise yerel algoritma sadece pozitif yönlü hareket etse de tabu algoritması hem pozitif hem de negatif yönlü hareket etmektedir. Tabu algoritması her gittiği yolu hafızaya kaydeder ve geçtiği yerlerden bir daha geçmez [114]. Tabu algoritmasının

temeli yasaklı listedir. Bu liste tekrarlanmayacak şekilde sürekli olarak en iyisini seçene kadar güncellenir. Tabu algoritması araç rota planlama probleminde araştırmacılar tarafından en çok kullanılan algoritmadır [40]. Tabu algoritmasının birkaç temel elemanlarından birisi de hafızadır. Her aramada ortaya çıkan sonuçlar “kısa dönemli hafıza” adı verilen hafızaya kaydedilir. Yapılması yasaklanan hareketler’ ise “tabu listesi” adı verilen listeye kaydedilir. Ancak, belirli bir süre geçtikten sonra, tabu listesindeki hafızalar silinir ve yapılmasına izin verilir. Tabu arama algoritması birçok alanlarda kullanılmaktadır. Bunların arasında karakter tanımlama, eleman listeleme, şebeke topolojisi tasarımı, gezgin satış problemleri, telekomünikasyon yolu görevlendirme gibi alanlar da bulunmaktadır.

### **3.3.6. En yakın komşu algoritması**

En yakın komşu algoritması depoya en yakın noktayı seçer. Daha sonraki adımlarda bir sonraki en yakın noktaya gider ve bu şekilde son noktaya ulaşana kadar bu tekrarı devam eder. 1966 yılında gezgin satıcı probleminin çözümü için Bellmore ve Nemhauser tarafından bu algoritma geliştirilmiştir [115].

### **3.3.7. Genetik algoritma**

“Genetik algoritma” terimi ilk defa John Holland tarafından 1975 yılında kullanılmıştır ancak 1960 yılında Genetik Algoritma (GA) ismi kullanılsa da bununla ilgili birçok çalışmalar yapılmıştır. Mesela, Bremermann, Fogel ve diğerleri Darwin teorisindeki mutasyon ve seçim kavramını kullanarak evrensel programlama fikrini hayata geçirdiler. Daha sonra 1989 yılında John Holland’ın öğrencisi David Goldberg gaz boru hattında genetik algoritma optimizasyonu ile ilgili doktora tezi hazırladı [116].

Genetik algoritma doğal seleksiyon ve doğal genetik mekaniğine dayanan bir arama algoritmasıdır. Genetik algoritma bir arama probleminin karar değişkenlerini belirli sonlu uzunluktaki alfabe dizilerine kodlar. Burada dizilere kromozom, alfabelere gen, genlerin değerlerine ise alel denir. Genetik Algoritma doğada bulunan canlıların uyumunu yapay sistemlere uygulayarak geleneksel yollarla amaca ulaşamayan problemleri çözebilen algoritmadır. Bu uyumu yapay sistemlere uygulamak için birkaç temel özellik vardır:

- Başlangıç: Başlangıç popülasyonu belirler.
- Değerlendirme: Her bir kromozomun uygunluk değeri hesaplanır.
- Seçim: Yeni iterasyonlar için yeni kromozomlar seçilir.
- Birleşme: Daha iyi çözümler oluşturmak için iki veya daha fazla ebeveynleri birleştirir.
- Mutasyon: Bir bireyin genlerinin değiştirilerek yeni bir birey haline getirilmesine mutasyon denir.
- Yer değişme: Seçme, genetik çeşitlenme ve mutasyonla yaratılmış yavru popülasyon, orijinal ebeveyn popülasyonunun yerine geçer [117].

Genetik algoritmanın çalışma adımları:

Adım 1. İlk adımda popülasyondaki birey sayısı belirlenir.

Adım 2. Uygunluk fonksiyonu yardımıyla her bir kromozomun ne kadar iyi olduğu belirlenir. Bu fonksiyon ne kadar verimli ve hassas olursa, genetik algoritma sonucu da bir o kadar başarılı olacaktır.

Adım 3. Burada yeniden kopyalama ve değişim işlemleri uygulanır ve sonuçta yeni popülasyon oluşur. Değişim turnova seçimi ve rulet tekerleği sayesinde gerçekleştirilir. Turnova seçimi rastgele seçilen iki bireyden en iyisini seçerek bir sonraki popülasyona aktarır. Rulet tekerleği seçiminde ise her bir bireyin toplam bireylerin değerine bölünerek seçim olasılığı uygulanır. En yüksek değeri olan birey yeni popülasyona aktarılır.

Adım 4. Eski kromozomlar popülasyondan çıkartılarak yeni kromozomlar eklenir.

Adım 5. Yeni popülasyonun başarısını bulmak için tüm kromozomlar yeniden hesaplanır.

Adım 6. Genetik algoritma birkaç defa çalıştırılıp yeni popülasyon oluşturularak hesaplanır.

Adım 7. Bu ana kadar bulunmuş en iyi kromozom, algoritma için en iyi sonuçtur.

Genetik algoritma ile İHA'ların başlangıç noktasından son noktasına en uygun yolun bulunması ile ilgili Reagan ve diğerlerinin hazırlamış olduğu "Path Planning for Quadrotor UAV Using Genetic" [118] makalesinde çalışılmıştır. Bu makalede engeller olduğunda verilmiş başlangıç noktasından bitiş noktasına en kısa yolu belirlemek, enerji ve zamandan tasarruf etmek için genetik algoritma kullanılmıştır. Geliştirilen algoritma ile C++ programlama dilinde deney yapılmıştır. Başlangıç noktası ve hedef noktası kromozomlarda belirtilmiştir. Algoritma başlangıcında popülasyon üretilir ve daha sonra engeller tespit edilir. Kromozomda engel varsa ceza işlevi uygulanır. Bu da mesafenin büyümesine neden

olur. Seçim sürecinde, nesildeki yüksek kondisyon kromozomunun ilk yarısı kopyalanır. Daha sonra yeni nesil üretmek için bir ebeveyn kromozomu kullanılır. Ortaya çıkan sonuç, en yüksekten en düşük değere göre sıralanır. En uygun kromozom, gidilen en kısa mesafe değerine sahip kromozom olacaktır. Daha sonra gelecek nesiller için geçiş uygulanır. Burada geçiş noktasına bağlı olarak iki ana kromozomun genleri değiştirilir. Rastgele geçiş noktası kullanılarak her ebeveyn kromozomu farklı geçiş noktalarına sahip olmuştur. Bu parametrenin quadrotor olarak bilinen dört motorlu İHA tarafından kat edilen mesafeyi iyileştirdiği gösterilmiştir.

İvanov, İHA rotasını planlamak için seyahat eden satıcı problemine GA çözümlerinin kullanımını ele alarak, planlanan rotayı en aza indirmiştir. Rota üç koordinatlı sistemde olup, her nokta bir koordinatla tanımlanmıştır. Önerilen algoritma İHA için uygun rotayı belirler. Fakat gerçekte İHA uçuşu başlatıldığında algorithmadan çıkan sonuçla gerçek sonuçlar farklı olacaktır çünkü İHA kırık çizgi boyunca uçamaz, kendi etrafında sorunsuz şekilde dönebilir. Programlaştırılan İHA uçuşu simülasyonunun normal rotadan %60 kısa olduğu belirtilmiştir. Bu sorunların gelecek makalelerde çözüleceği belirtilmiştir [119].

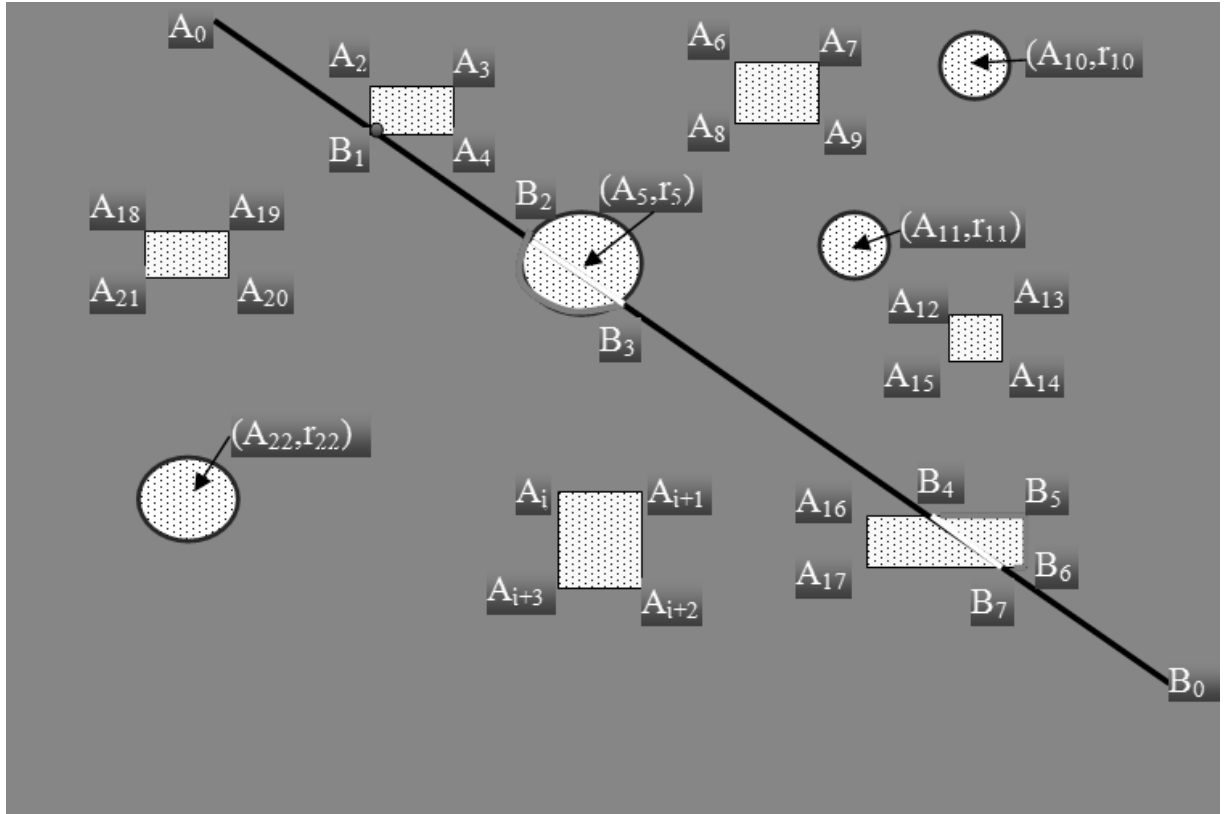
### **3.4. Literatürde Yer Alan İnsansız Hava Araçları'nın Rota Optimizasyonu ile İlgili Diğer Çalışmalar**

İHA'ların teknik özellikleri İlhan Ercan'ın "İnsansız hava sistemleri rota planlaması dinamik çözüm metotları ve literatür araştırması" çalışmasında gösterilmiştir. Buradan anlaşılıyor ki, İHA için, havada kalma süresi 2 ila 85 saat arasında olabilmektedir. Uçuş mesafesine göre İHA'lar 90.000 fit ve ağırlık olarak 481,2 kg olabilir. Stratejik İHA'lar ortalama 24-48 saat uçuş süresi, 65.000 fit uçuş yüksekliği, 500 km'den büyük uçuş menzili ve 1200 kg maksimum kalkış ağırlığına sahiptir [120]. Bu göstergeler, yukarıdaki amaçlar için kullanılmasını sağlar. Büyük şehirlerde İHA'nın kullanılması, depolar ve müşteriler arasında malların dağıtımı ile ilgili problemlere odaklanılmaktadır [121, 122]. Büyük organizasyonlar da İHA'lar ile teslimata ilgi göstermektedir. Örneğin 2013 yılında Amazon Prime Air [11], müşterilerinin Amazon'dan satın aldığı ürünleri göndermek için İHA'ları kullanmıştır. 2014 yılında insansız hava araçlarının üretimi için Google Project Wing tasarlandı. Bu İHA'lar Prime Air ve Parcelcopter'den daha fazla yükü taşıyabilmektedirler. Sunghun Jung'un "Analysis of Amazon Prime Air UAV Delivery Service" çalışmasında, Amazon teslimatları için kullanılan İHA'ların ideal donanım özellikleri olarak motor, akü

grubu, kontrolör ve tasarımı araştırılmıştır. Özellikle Amazon'un İHA teslimatlarının özellikleri, teslimatta bilgi açıklığı, teorik hesaplamalar ve xcopterCalc hesaplamalarından elde edilen bilgilerle bu üç olanak karşılaştırılmıştır. 2021 yılında Amazon Prime Air servisi ile maliyetin %8,6'ya çıkacağı belirtilmiştir [123]. İHA'ların rota planlamasında Xing Chen Karınca Kolonisi Algoritmasını kullanmıştır [8]. Puanlama tabanlı hareket algoritması, dar geçitlerde dinamik kısıtlamalar dikkate alındığında birçok robotik problemde başarılı olmuştur. Başka bir çalışmada ise film dinamik hareket algoritması çalışılmış ve iki adımlı bir algoritma önerilmiştir. İlk aşamada, yaklaşık doğru parçası modeli ile "Çabuk Araştıran Rastgele Ağaçlar Algoritmasını" kullanılarak rota ortaya çıkarılmaktadır. Daha sonra gördüğü noktaları birleştirerek görüş alanındaki uçuş yolu bulunur. İkinci aşamada çıkış noktaları B-Spline eğrisi algoritması ile birleştirilir ve dinamik uçuş kriterleri dikkate alınarak noktalar optimize edilmeye çalışılır. Bu iş tek ve çoklu insansız hava araçlarının tehlikeli ve engelleri olan ortamlarda güvenli rota planlanmasına yöneliktir [124]. Genetik algoritma (GA) kullanılarak, İHA'nın kinematik sınırlamaları ve toprak şartları dikkate alınarak en uygun yol planlaması araştırılmıştır. NASA tarafından kullanılan arazi verileri gerçek üç boyutlu uydu verilerinden oluşmaktadır, yükseklik verilerine erişilmiş ve arazilerdeki dağlar belirlenmiştir. Gerçek coğrafi koordinat sistemini ve dünyanın geometrisini hesaba katmak için yüksek hassasiyetli hesaplamalar kullanılmıştır. Küresel rota planlaması, GA ile belirlenen noktaların yerel rota planlaması üzerine kurulmuştur. Elde edilen sonuçlara bakıldığında, İHA'ların, uygun görevleri yerine getirmesinde zaman ve performans açısından çok yararlı olduğu görülmüştür. Bu çalışma ile otonom İHA'ların kullanılmasının avantajları sayısal verilerle ortaya çıkartılmıştır. Bir İHA yerine birden fazla İHA kullanımı, daha büyük alan üzerinde daha çabuk kontrol yeteneği sağlamıştır. Bir İHA'nın görevini yerine getirme zamanının ve maliyetinin, çoklu İHA'lardan daha yüksek olduğu kanıtlanmıştır. Bu şekilde, silahlı kuvvetlerin pilot ihtiyaçları ve eğitimli bir operatöre duyulan ihtiyaç azalacaktır. Bir büyük İHA yerine birlikte çalışan birden fazla İHA'nın kullanımı etkili bir çözüm olarak görülmektedir. Bu çalışmanın devamında, faktör sistemleri kullanılarak, gerçek zamanlı rota planlamasında daha fazla kısıtlama kullanılması amaçlanmıştır. Dağlar arasında bağlantı sağlamak ve engelleri tespit etmek için daha geniş bir alanın taranması gerekir. B-spline veya Bezier eğrileri kullanılarak, iki nokta arasında hareket algoritmaları olmadan hareket planları yapılır [125, 126].

#### 4. ÇALIŞMANIN AMACI ve PROBLEMİN TANIMLANMASI

Rota planlama problemi, belirli amaçlar için araçların en uygun şekilde izlenmesi ile rotaların hazırlanması problemidir [127]. Bu çalışmada amaç fonksiyonu, istenmeyen bölgelerin üzerinden uçmaktan sakınmaktır (Resim 4.1).



Resim 4.1. Engeller olduğunda İHA'ların en optimal uçuş yörüngesinin belirlenmesi

Bu çalışma, tek bir İHA için yapılmıştır. Çok sayıda İHA olduğunda her bir İHA'nın rotasını tek bir İHA için rota planlama algoritmasını kullanarak tasarlanabilir. Belirtmelidir ki, büyük şehirlerde, uçuşların yasak olduğu yerlerde İHA'ların optimal uçuş yörüngesinin belirlenmesi problemleri ile ilgili çalışmalar pek fazla bulunmamaktadır. Bu çalışmada Google haritası üzerinden engellerin olduğu arazilerde önceden belirlenen engellerin, İHA'ların hareket ettiği hava mekanlarından uzaklaştırılması ve onların en uygun yörünge ile uçuşunun sağlanması için yöntemler ve algoritmalar geliştirilmiştir.

Problemi çözmek için Google haritası üzerinde engelleri belirleyip bunları daire ile veya dörtgen ile tamamlayabiliriz. Yasaklanmış nesnelerin şekilleri ve koordinatları bilinmektedir

ve deđiřmeden kalmaktadır. Problemi sadeleřtirmek iin ilk olarak İHA'nın Google haritasında bir uuř operasyonu gerekleřtirdiđini kabul edeceđiz. řehirlerde birok engelin (anaokulları, okullar, zel tesisler) bulunduđunu bilerek, İHA'nın bu engellerin etrafından gemesi gerekir. Bu problem matematiksel řekilde ařađıda verilmiřtir. Belirli bir geometrik alanda nceden verilmiř sabit parametrelere sahip daireler veya drtgenlerden teđet geilerek, bu alandaki iki noktayı bađlayan dođru paralarından oluřan en kısa mesafeli kırık izginin kurulması gerekir.

Optimal rotada dođru paralarının, engellerin kenarları ile kesiřim noktalarından bařlayarak kenarın, kısa mesafeli kısmında hareket eder. Benzer problemlerin zm konusunda ok sayıda alıřma bulunmaktadır. Bu ynde en popler algoritma Bentley-Ottman algoritmasıdır. Bařka bir algoritma ise sprme algoritması olarak tanımlanır ve her dz izgi parasının kesiřmesine dayanır [117].

Bu algoritmalarda bir dođru parasının diđer dođru parası ile kesiřimi kontrol edilir. Dzlem zerinde x eksenini (enine) ve y eksenini (boyuna) řeritlere blnerek her řeritte koordinatların kesiřim kořulları bulunur. Bu, hesaplama sayısını ve gerekli bellek miktarını artırır. Bu tez alıřması kapsamında nerilen algoritmada ise iki dođru parası kesiřtiđi zaman, kenar noktalarında kurulmuř bir dıřbkey drtgenin zelliklerine dayanmaktadır. Bu hesaplama, hesaplama sayısının ve gerekli bellek miktarı hacminin az olması ile mevcut algoritmalarından farklılık gsterir.

#### 4.1. Matematiksel Modelleme

Eđer  $(A_0B_0)$  ve  $(AB)$  dođru paraları kesiřirse, bu dođru paraları zerinde kurulmuř  $(A_0B B_0A)$  dıřbkey drtgenin diyagonalleri olacaktır. Aksi takdirde  $A_0B B_0A$  dıřbkey olamayacak ve dođru parasının diyagonal olması řartı gerekleřmeyecektir. Bylece  $A_0B B_0A$  drtgenin alanı:

$$S_1 = \frac{1}{2} |AB| * |A_0B_0| * \sin \angle A_0B_0A \quad (4.1)$$

ve

$$S_2 = S_{A_0AB_0} + S_{A_0BB_0} \quad (4.2)$$

Böylece 2 doğru parçasının kesişimi halinde;

$$S_1 = S_2 \quad (4.3)$$

olacaktır.

$S_1 \neq S_2$  olduğunda ise  $(AB)$  ve  $(A_0B_0)$  doğru parçaları kesişmez.

Bu belirtilen koşullar için düzlem üzerinde  $n$  sayıda koordinatları önceden belirlenmiş doğru parçalarıyla koordinatları değişebilen verilmiş doğru parçasının kesişimini belirleyen algoritma geliştirilebilir.

Düzlem üzerinde  $n$  sayıda doğru parçasıyla, verilmiş doğru parçasının kesişimini belirleyen algoritma şu şekildedir;

*İşaretlemeler:*

Doğru parçaları koordinatlarla verilir:

Düzlem üzerinde verilmiş doğru parçaları:  $A_iB_i$   $i = \overline{1, n}$

Doğru parçalarının koordinatları:  $A_i(x_i^0, y_i^0); B_i(x_i^1, y_i^1), i = \overline{1, n}$  ;

Verilmiş doğru parçası  $(A_0B_0)$ , koordinatları ise:  $A_0(x_0^0, y_0^0); B_0(x_0^1, y_0^1)$

$J(n)$ - $n$  ölçülü kesişim noktalarından oluşan vektör:

*Input*  $(x_0^0, y_0^0), (x_0^1, y_0^1) \& n$

*for*  $i=1$  *to*  $n$

// $n$ -engellerin sınırlarından oluşmuş doğru parçalarının sayısıdır.

*Input*  $(x_i^0, y_i^0)(x_i^1, y_i^1)$

*end for*

$$K_0 = (y_0^1 - y_0^0)/(x_0^1 - x_0^0); \quad (4.4)$$

$m=1$

*for*  $i=1$  *to*  $n$ ;

$$K_i = (y_i^1 - y_i^0)/(x_i^1 - x_i^0); \quad (4.5)$$

If  $K_0 = K_i$  then  $i$ 'nci doğru parçası verilmiş doğru parçası ile paralel olduğu için o listeden çıkartılır:  $(x_i^0, y_i^0)$  ve  $(x_i^1, y_i^1)$  değerleri yeniden ayarlanabilir,  $n=n-1$  kabul edilir,  
else

End if

End for

//Kesişim koşulları kontrol edilir.

$$|A_0B_0| = d_0 = \sqrt{(x_0^1 - x_0^0)^2 + (y_0^1 - y_0^0)^2} \quad (4.6)$$

$$|A_iB_i| = d_1 = \sqrt{(x_i^1 - x_i^0)^2 + (y_i^1 - y_i^0)^2} \quad (4.7)$$

$$\cos(A_iB_i \wedge A_0B_0) = \frac{(A_iB_i)(A_0B_0)}{|A_iB_i||A_0B_0|} = \frac{(x_0^1 - x_0^0)(x_i^1 - x_i^0) + (y_0^1 - y_0^0)(y_i^1 - y_i^0)}{d_0 d_1} \quad (4.8)$$

$$\sin(A_iB_i \wedge A_0B_0) = \sqrt{1 - \cos^2(A_iB_i \wedge A_0B_0)} = D_1 \quad (4.9)$$

$$S_1 = \frac{1}{2} d_0 * d_1 * D_1 \quad (4.10)$$

//hesaplanır.

For  $i=1$  to  $n$

$$S_{A_0A_iB_0} = \frac{1}{2} |A_iA_0| * |A_iB_0| (\sin A_iA_0 \wedge A_iB_0) \quad (4.11)$$

// $|A_iA_0|$  ve  $|A_iB_0|$  uzunlukları (4.6) ve (4.7) formüllerine uygun olarak uç noktaların koordinatları dikkate alınarak hesaplanabilir.

//Aynı formül ile  $S_{A_0B_iB_0}$  alanı da hesaplanabilir.

$$S_i = S_{A_0A_iB_0} + S_{A_0B_iB_0} \quad (4.12)$$

*End for*

If  $S_0 = S_i$  then  $J(m) = \arg[S_0 = S_i]$ ,

//arg- argüman değeridir,  $S_0 = S_i$  olduğu durum için  $J(m) = i$

// $J(i)$ 'nci doğru parçası verilmiş doğru parçası ile kesişir.

$m=m+1$

else doğru parçası verilmiş doğru parçası ile kesişmez.

*End if*

*End for* (adım 2)

//Kesişen düz çizginin listesi ve kesişim noktaları yazılır.

For  $i=1$  to  $m$ ;

Print  $J(i)$ ;

//Kesişim noktalarının koordinatları yazdırılır:

Print

$$X = \frac{(y_{J(i)} - b_{J(i)}) - (y_0 - b_0)}{(k_{J(i)} - k_0)} \quad (4.13)$$

$$Y = k_0 X + b_0 \quad (4.14)$$

*End for*

Algoritmanın sonu;

Bu algoritma, engellerin çokgen olduğu durumlarda kullanılması uygundur. Engeller bir daire şeklinde verildiğinde algoritma kolayca uygulanabilir, bu durumda, nesnenin çevresi bir engel olarak kabul edilir.

Bunun için engellerin merkezi  $A_0(x_0, y_0)$  ve yarıçap  $(r_i)$  bilinmelidir. Engeller bir daire şeklinde verildiğinde, düz çizgi parçası ile kesişimin belirlenmesi koşulu, düz çizgi denkleminin ve dairenin denkleminin birlikte çözümüyle ortaya çıkar.

$$\frac{b_0}{r_i} \leq \sqrt{1 + k_0^2} \quad (4.15)$$

Eğer (4.15) durumunda eşitlik koşulu sağlanırsa doğru, daireye teğet geçer. Küçük olduğu durumda ise iki noktada kesişim olur.

$$\begin{cases} y_{12} = k_0 x_{12}^i + b_0 \\ x_{12}^i = k_0 b_0 \pm \sqrt{r_i^2 (1 + k_0^2) - b_0^2} / (1 + k_0^2) \end{cases} \quad (4.16)$$

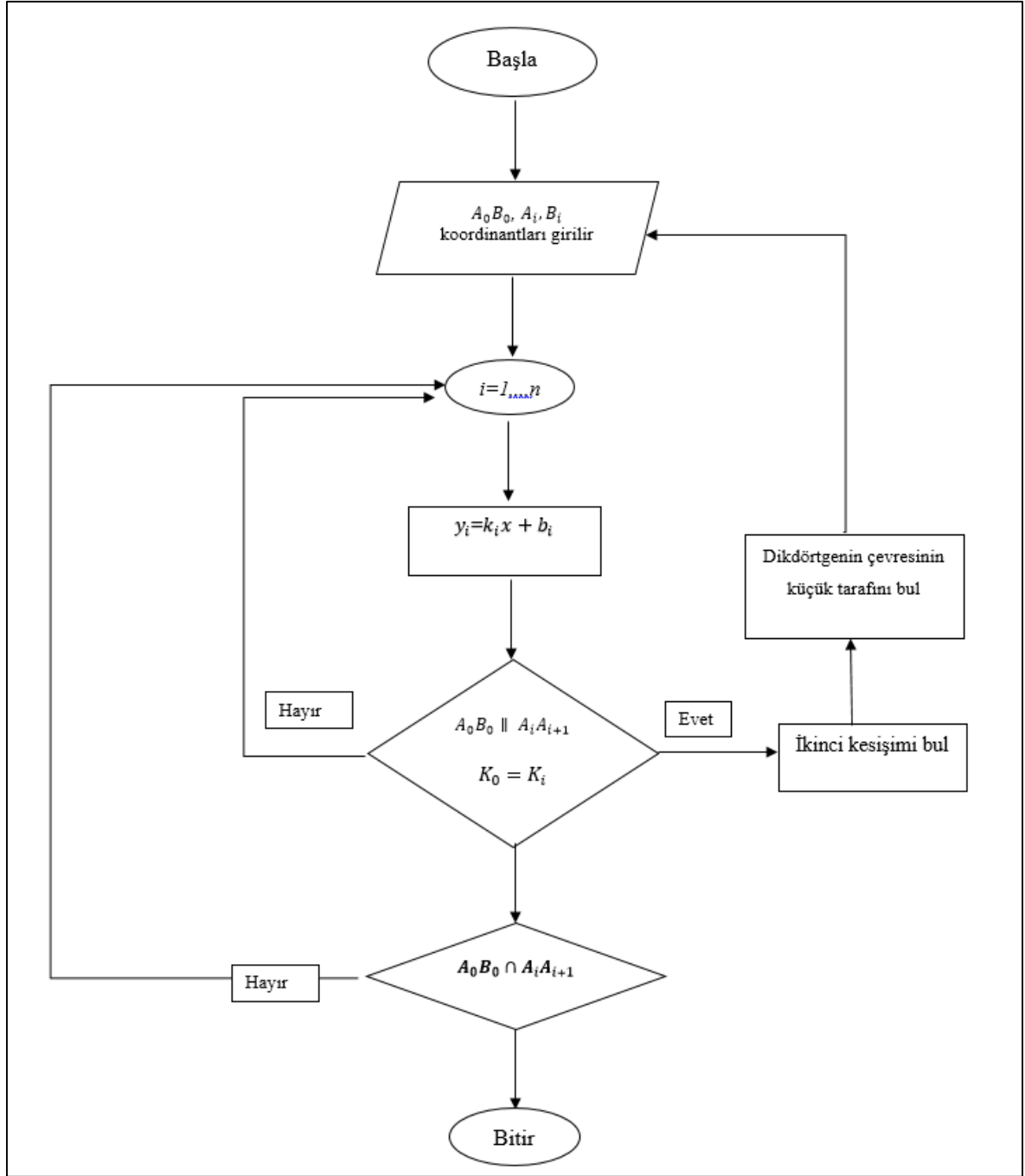
$(A_0 B_0)$  kesişim koşulları aşağıdaki gibi olur:

$$\begin{cases} x_0^0 \leq x_{12}^i \leq x_0^1 \\ y_0^0 \leq y_{12}^i \leq y_0^1 \end{cases} \quad (4.17)$$

Böylece engeller daire biçiminde verildiğinde, (4.15) ve (4.17) koşulları kontrol edilir ve kesişim koordinatları formülü (4.16) ile hesaplanır. Her iki durumda da İHA, düz çizgi üzerindeki rotasında engel olduğunda, (çokgen veya dairesel) ikinci kesişim çizgisine kadar en kısa yoldan hareket ederek tekrar düz çizgi ile hareket eder.

Algoritma;

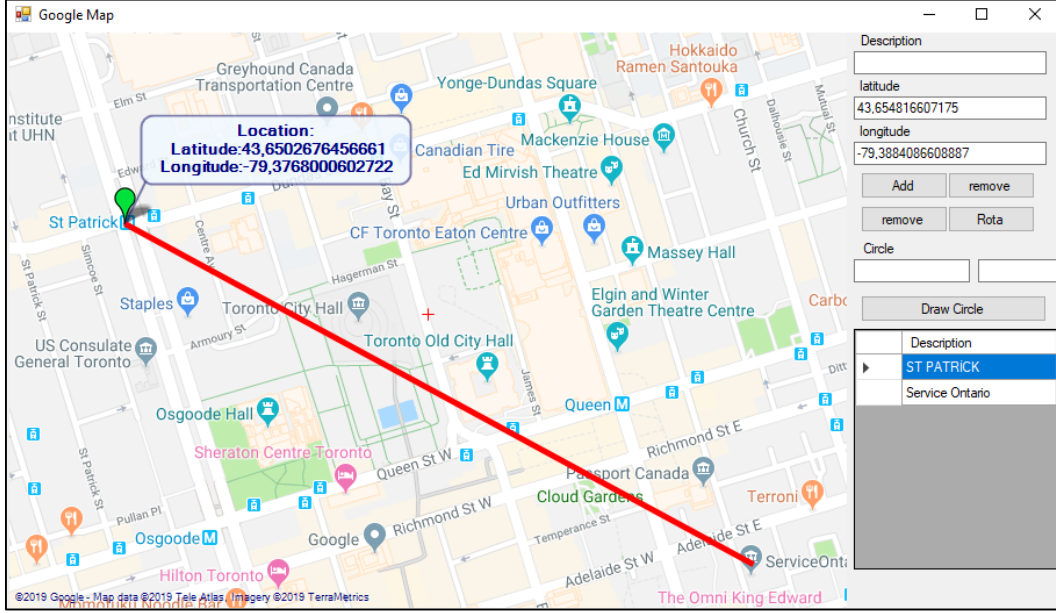
1.  $A, B, A_i$  koordinatları veri tabanına girilir.
2.  $A_i A_{i+1}$  düz çizgi denklemini oluşturuyoruz:  $y_i = k_{ix} + b_i$ . Bu, tarafların denklemdir.
3.  $AB$  d \ x ile  $A_i A_{i+1}$  kesişimlerini bulunur.
4. Figürün çevresi boyunca (en az iki birim) şekil noktaları arasındaki mesafeyi saat cinsinden belirlenir.
5. Eğer bu mesafe, şeklin çevresinin yarısından daha az ise, olduğu gibi kalır, aksi takdirde diğer tarafa doğru hareket edilir (bu durum noktalarla gösterilir) (Şekil 4.1) [128].



Şekil 4.1. Düzlem üzerinde  $n$  sayıda doğru parçasıyla, verilmiş doğru parçasının kesişimini belirleyen algoritma

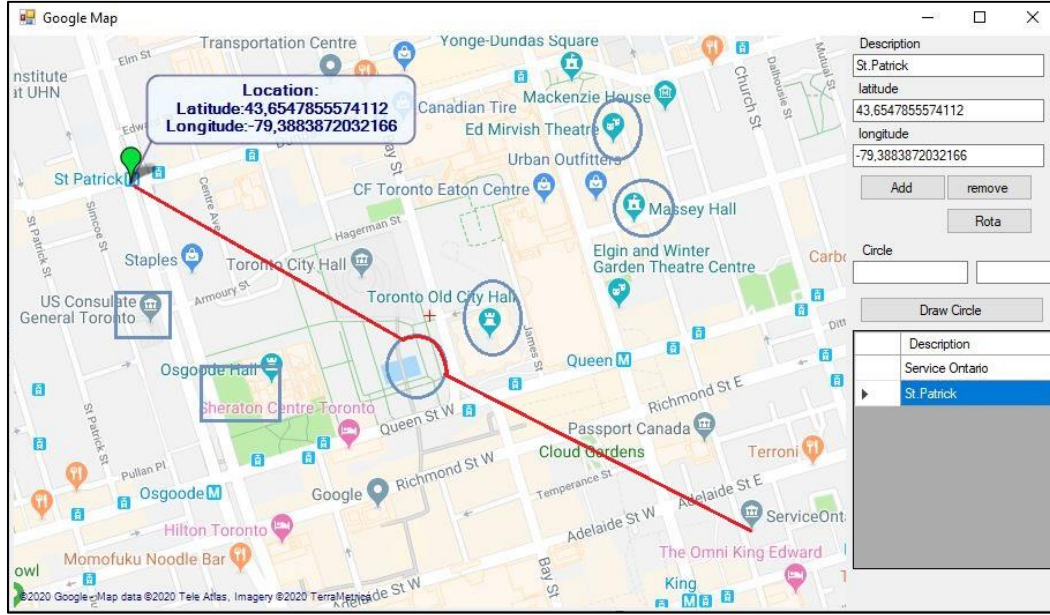
## 4.2. Hesaplama Deneyi

Önerilen algoritmalar üzerinde bir hesaplama deneyi yapılmıştır. Bunu yapmak için Toronto şehri St. Patrick metro istasyonundan Service Ontario'ya bir İHA uçuşu başlatmak için yazmış olduğumuz uygulama ile Google Maps uygulaması kullanılmıştır. (Resim 4.2).



Resim 4.2. St. Patrick'ten Service Ontario'ya İHS uçuşunun ilk talimatı

İHA'nın başlangıç ve bitiş noktaları düz bir çizgi ile birleştirilmiştir. Yukarıdaki algoritmaları uygulamak için uçuş yolu boyunca engeller belirleyip bu algoritma ile rota denklemini belirlenmiştir (Resim 4.3).



Resim 4.3. St. Patrick'ten Service Ontario'ya İHA'nın uçuş kısıtlı bölgelerde en kısa uçuş mesafesi

Engeller bir daire içinde gösterilir. En iyi rotayı tanımlamak için bir yazılım geliştirilmiştir. Program başlangıçta Google Map uygulamasında İHA başlangıç ve bitiş hedeflerini kaydetmeye başlar. Bunu yapmak için, bu noktalar işaretlenir ve koordinatlar not edilir. Bu durumda koordinatlar, x eksenini için 43,6536 ve y eksenini için - 79,3828 olacaktır. Aynı şekilde, engellerin koordinatları ve şeklin sınırı (bizim durumumuzda dairenin sınırı) belirtilmiştir. Engellerin kaydedilmesinden sonra, program bir parçanın  $n$  sayıda parçayla kesişimini belirlemek için bir algoritma kullanır ve devamında en iyi rotayı tanımlamak için önerilen algoritmayı rapor eder. Önerilen yazılım paketi, engeller dikdörtgenler veya daireler şeklinde hesaplandıktan sonra uçuşun optimal yörüngesini otomatik olarak belirler.

#### 4.3. Deneylerden Çıkan Sonuçlar

1. Uçuşların yasak olduğu kısıtlı alanlarda İHA'nın en uygun uçuş rotasını belirlemek için bir algoritma verilmiştir. Algoritma düzlem üzerinde koordinatları önceden belirli olan doğru parçalarının, koordinatları her zaman değişen, verilmiş doğru parçasının kesişiminin üzerinde, kurulmuş dışbükey dörtgen olması fikrine dayanmaktadır. Algoritma mevcut algoritmalarından daha az sayıda hesaplama ve bellek hacmi gerektirir.
2. Önerilen algoritma, engellerin sayısını sınırlandırmamaktadır. Algoritma, diğer algoritmalarından daha az hesaplama kaynağı gerektirir.

3. Algoritmayı herhangi bir sayıda İHA ile uygulamak için ek olarak, yörüngelerin kesişim noktalarında, İHA'ların uçuş yöntemini (hız ve irtifalar için) ayarlamak için işlemler yapılmalıdır. Bu işlemler, bilinen yörünge boyunca İHA'ların uçuşunu sağlamak veya havada İHA'ların hareketini gösteren denklemleri çözmek için mevcut araştırma sonuçlarını kullanarak elde edilebilir.

## 5. İNSANSIZ HAVA ARAÇLARININ KARINCA KOLONİSİ ALGORİTMASI İLE OPTİMAL ROTASININ BULUNMASI

Engeller rastgele olduğu için, bazı durumlarda (örneğin birbirine yakın olduğu durumlarda) problemin çözümü optimal olmayabilir. Bu durumda tüm engellerin durumu dikkate alınarak bu tezde Karınca Kolonisi Algoritması uygulanarak, problemin çözümü verilmiştir.  $A$  ve  $B$  noktalarını içeren dikdörtgen alan ve alanın içindeki engeller haritada işaretlenmiştir.  $AB$  doğru parçası dikdörtgenin köşegeni olacak şekilde seçilmiştir.  $A$  tepesinden başlanarak dikdörtgen  $m \times n$  karelere bölünür ve bu karelerin merkezlerinin koordinatları olan  $L(m \times n)$  kümeleri oluşturulur. Her karıncanın bir merkezden diğerine sabit zamanda hareket edebileceği varsayılmaktadır. Başka bir deyişle, dikdörtgen alan, karıncanın tek bir zamanda gidebileceği mesafeye göre bölümlere ayrılmıştır. Karıncaların başlangıç noktasını ( $A$  noktası)  $L_{i0j0}$ , bitiş noktasını ( $B$  noktası)  $L_{imaxjmax}$  olarak gösterelim. Engellerin içinde olan hücrelerden oluşan dikdörtgenleri not edelim. Engellerde bulunan hücrelerin ve sınır çizgilerinin koordinatlarından oluşan “tabulist” yasaklı veri kümesini  $TL(i, j)$  oluşturalım. Engeller olmadığı zaman optimal rota başlangıç ve son noktayı birleştiren  $AB$  doğru parçası olacaktır. Yukarıdaki çalışmadaki yöntemle engeller olduğunda  $L_{i0j0}$  ve  $L_{imaxjmax}$  noktalarını birleştiren düz bir çizgi (engellerin dahilindeki dikdörtgenlerin ana hat boyunca) belirtilmiştir, sonuç olarak,  $[L_{i0j0}L_{imaxjmax}]$  kırık çizgisi oluşmuştur. Düzlemin herhangi  $(i, j)$  noktasından bu kırık çizgiye kadar olan en kısa mesafeyi  $d_{ij}$  ile işaretlendiğinde,

$$\mu_{i,j} = 1/d_{ij} \quad (5.1)$$

karıncanın amacına ulaşmak isteğini veya gidiş yolunun çekiciliği gösterilmiştir. Algoritmanın başlangıcında  $K$  süresince dahil olan karıncaların sayısı ve  $T$ -iterasyonun maksimum değeri verilir. Her bir  $i$ 'nci karıncanın  $i = (1, K)$   $j$ 'nci iterasyonunda  $j = (1, T)$  durumu,

$$S_i(j) = \{X_{ik,jk}^{i,j}, \tau_{ik,jk}^{i,j}, \theta_{ik,jk}, ik = 1, N, jk = 1, M\} \quad (5.2)$$

gibi üç parametreyle gösterilecektir: Birinci parametre hücrede olan karıncaların sayısını, ikinci parametre hücrede olan feromonun mevcut miktarını, üçüncü parametre ise hücreye girişinin serbest veya yasak olduğunu belirtir.

$$\theta_{ik,jk} = \begin{cases} 1 & i_k j_k \neq 0 \cup M \cup N, (i_k j_k) \in L(N \times M) \setminus TL(i, j), \\ 0 & i_k \neq 0 \cup M, j_k \neq 0 \cup N, (i_k j_k) \in TL(i, j) \\ \infty & i_k = 0 \cup M, j_k = 0 \cup N \end{cases} \quad (5.3)$$

Bunu da belirtmek gerekir ki, algoritmanın çalıştığı durumlarda  $\theta_{ik,jk}$  sabittir. Başlangıçta  $(i_0 j_0)$  hücresinden başka bütün hücreler boştur, yani

$$X_{i_1, j_1}^{i, 1} = \begin{cases} 1 & i_1 = i_0, j_1 = j_0 \\ 0 & i_1 \neq i_0, j_1 \neq j_0 \end{cases} \quad i = 1, K \quad (5.4)$$

karıncaların hepsi  $(i_0 j_0)$  hücresindedir. Her bir  $j$ 'nci iterasyonunda  $(i_1 j_1)$  hücresinde olan karıncaların sayısı

$$X_{i_1, j_1}^{i, j} = \begin{cases} \sum X_{i_1, j_1}^{i, j} + 1 & \text{eğer } (i_1, j_1) \text{ hücresi seçildiyse} \\ \sum X_{i_1, j_1}^{i, j} & \text{eğer } (i_1, j_1) \text{ hücresi seçilmediyse} \end{cases} \quad (5.5)$$

denklemleri ile hesaplanır. Aynı anda herhangi bir hücrede bulunan karıncaların sayısı  $K$ 'yı geçemez.

$$N_{i_1, j_1, j} \leq K \quad (5.6)$$

Burada toplanma  $j$ 'nci iterasyonunda  $(i_1 j_1)$  hücresini seçen karıncalar üzerinde gerçekleştirilir. Öte yandan ardışık iki iterasyonda aynı durum elde edilemez:

$$\arg(X_{ik, jk}^{i, j}) \neq \arg(X_{ik, jk}^{i, j+1}) \quad (5.7)$$

Başka bir deyişle,  $i$ 'nci karınca  $j$ 'nci iterasyonunda hücreden çıkarsa,  $j+1$ 'nci iterasyonunda oraya giremez. Başlangıçta tüm hücrelerdeki feromonun miktarı 0'dır böylece, başlangıç durumunda

$$S_i(1) = \{X_{i_0, j_0}^{i, 1}, 0, \theta_{ik, jk} \quad i = (1, K)\} \quad (5.8)$$

durumu oluşur.  $i$ 'nci iterasyonunda  $(i_k, j_k)$  hücresinden  $(l, k)$  hücresine, onun komşuluğunda olan 8 hücreden birine geçme olasılığı

$$p_{ik,jk}^{l,k,i} = \frac{\alpha_{ik,jk} \tau_{l,k}^i + (1 - \alpha_{ik,jk}) \mu_{l,k}}{\sum_{(l,k) \in TL(i,j)} \alpha_{ik,jk} \tau_{l,k}^i + (1 - \alpha_{ik,jk}) \mu_{l,k}} \theta_{l,k} \quad (5.9)$$

olarak hesaplanır.

Başlangıçta,  $p_{ik,jk}^{l,k,0} = 1/8$  olur ve 8 hücreden herhangi biri rastgele olarak seçilir. Olasılığın 0 veya  $\infty$  olması, bu hücrenin seçilmediği anlamına gelir.

$$\alpha_{ik,jk} = \frac{1}{\sum [L_{i0j0} L_{ikjk}]} \quad (5.10)$$

Başlangıçta  $(i_k j_k)$  hücresine giden yolun uzunluğunun ters değeri olarak hesaplanır. (5.7) formülü karıncaların hareketlerini engellere ve  $[L_{i0j0} L_{ikjk}]$  kırık çizgiye göre ayarlar.

Karınca (5.9) formülü ile her bir  $(i_k j_k)$  hücresi etrafında olan hücreler için (5.7) durumu göz önüne alınarak olasılıkları hesaplar ve onlardan en büyüğünü seçer.

Feromonun değişimi her bir  $i$ 'nci karıncanın  $(i_k j_k)$  hücresine girdiğinde oluşur.

$$\tau_{ik,jk}^i(t+1) = (1 - \alpha) \left( \tau_{ik,jk}^i(t) + \sum_{t=1}^t \sum_{lk=1}^{X_{i1,j1}^{i,j}} \tau_{lk,jk}^{lk}(t) \right) \quad (5.11)$$

$$\alpha = \frac{\alpha_{ik,jk}}{\sum_{(l,k) \in TL(i,j)} \alpha_{l,k}} \quad (5.12)$$

Burada  $\alpha$  -feromonun buharlaşma hızıdır.

Algoritma;

Begin

Haritada tepe noktaları  $A$  ve  $B$  olmak üzere bir dikdörtgen oluşturulur,  $L(m \times n)$  kümesi,  $TL(i, j)$  engeller kümesi ve  $[L_{i_0j_0} L_{imaxjmax}]$  kırık çizgi oluşturulur.

Karıncları  $K$  sayısı ve iterasyonun maksimum sayısı,  $T$  verilir.

$t=0$

Tüm karıncalar  $A$  hücresine yerleştirilir.

(5.1-5.4) ve (5.8) formülü ile başlangıç duruma getirilir.

repeat

$t=t+1$

for her  $i=1, 2, \dots, K$  karınca için

do

$i$ 'nci karınca için  $(i_0j_0)$ , başlangıç durumunda rota  $l_i = 0$  alınır.

repeat

(5.9) formülü ile her bir  $(i_kj_k)$  hücrenin etrafında bulunan hücreler için (5.7) koşulunu göz önünde bulundurarak olasılıklar hesaplanır ve onlardan en büyüğü seçilir.

Seçilmiş hücrenin koordinatları rotaya eklenir. Rotanın uzunluğu (boylam)  $l_i(t) = l_i(t) + 1$  hesaplanır.

(5.11) ve (5.12) formülleri ile feromon değiştirilir.

until seçilmiş hücre sınır hücresi değil

end do

If karınca  $L_{imaxjmax}$  hücresine varmışsa

then karınca başlangıç durumuna geri döner ve rotanın uzunluğu kaydedilir.

end if

If karınca sınırdaysa

then karınca başlangıç durumuna geri döner ve rotanın uzunluğu sıfırlanır.

end if

end for

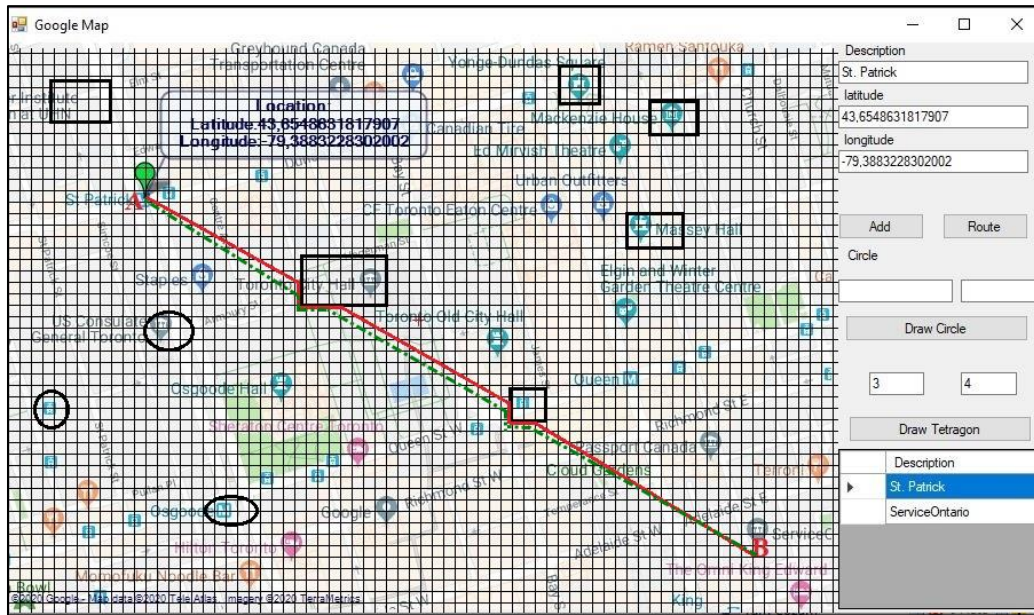
Kaydedilen rotalardan en kısa rota ekrana verilir

until iterasyonun sayısı  $t \leq T$  [129].

### 5.1. Hesaplama Deneyi

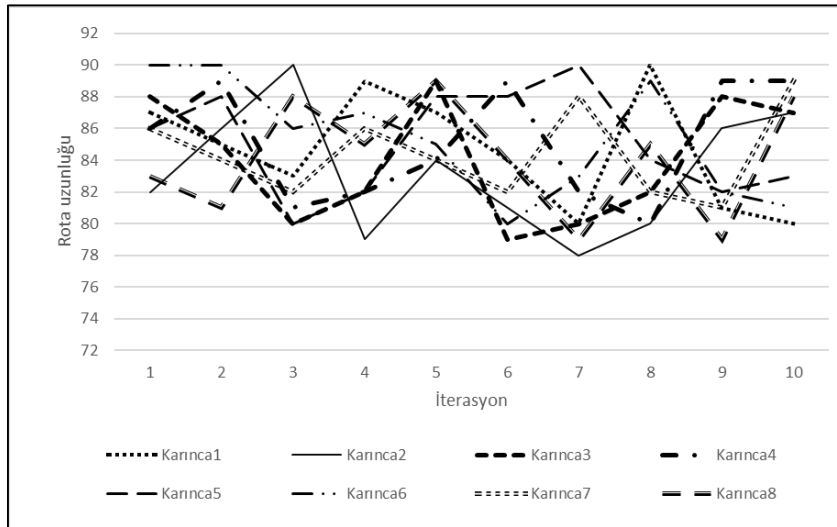
Önerilen algoritma üzerinden basit bir simülasyon yapılmıştır. Bu simülasyon C# programlama dilinde yazılmıştır. Bunu yapmak için Toronto şehri haritası üzerinde St. Patrick noktasından (A noktası) Service Ontario'ya (B noktası) bir İHA uçuşu başlatmak için simülasyonda Google Maps uygulaması kullanılmıştır (Resim 5.1). Öncelikle Google Map uygulamasında İHA'nın başlangıç ve bitiş noktalarını not ederek, haritanın bu bölümünü küçük karelere bölünmüştür. Yukarıda belirtilen koşullarda bu karelerin sayılarından oluşan  $L(53 \times 33)$  matrisini ve  $TL(i, j)$  engeller kümesi oluşturulmuştur. Bu durumda 9 engel vardır. Her bir engel  $3 \times 3$  birim boyutunda karedir. Sınır noktalar ve engeller göz önünde bulundurularak  $53 \times 33$  birim boyuta sahip  $\theta_{ik,jk}$  matrisi oluşturulmuştur. Hesaplamalarda karıncaların sayısı  $K=8$ , iterasyonun maksimum sayısı  $100$  dür ( $T=100$ ). Bu bilgilere dayanarak ilk durum matrisi oluşturulur. İlk aşama için hesaplama deneyleri yapılır.

C# programlama dili ile yazılmış bir programda, kesişen engelin tepe noktaları hesaplanır ve çevre boyunca en kısa yol işaretlenir. Bu çizgi düz çizginin aksine kırık bir çizgi olacaktır. Bu rotanın harita uzunluğu 78 birimdir. Bu rotayı ana rota olarak kabul edilip, bir sonraki aşamaya geçilir.



Resim 5.1. St. Patrick'ten Service Ontario'ya İHA'nın engeller olduğunda en kısa uçuş mesafesi

İkinci aşamada Karınca Kolonisi Algoritması ile hesaplama deneyleri yapılır. İterasyon başlangıcında karıncalar sırayla A noktasından harekete başlar. Her karınca, A hücresinin etrafındaki diğer hücrelerden birini rastgele seçer. Bunu yapmak için, program ilk önce iterasyon 1-8 aralığında rastgele bir tam sayı üretir ve bu sayıya karşılık gelen hücreyi saat yönünde seçer. Sonraki iterasyonlarda, en yüksek istek katsayısına sahip durum için hücre seçilir. Böylece, her iterasyonda karınca A noktasından hareket eder. Her yeni hücrenin seçimi istek katsayısını hesaba kattığından, bu rota ana rotaya yakın olacaktır. Ancak bu seçimde karınca sınır hücresini ve engellerin içinde kalan hücreleri seçmeyecektir. Bu hücrelere yönelik istek katsayısı "0" olduğundan, seçilme olasılığı da "0" olacaktır. Diğer hücrelerin istek katsayıları "0" dan farklı olduğu için karıncaların seçeceği yollar da rastgele olacaktır. İterasyonların sayısı sınırlı olduğundan, bu karıncaların tümü B noktasına ulaşamayacaktır. İşlem sırasında istek katsayısının "100" olduğu durumlar seçilir ve bu rotaların uzunlukları kaydedilir. Karıncalar için bu koşulu karşılayan 10 rota oluştuğunda işlem durdurulur. Böylelikle her bir karınca için farklı uzunluklarda 10 farklı rota elde edilecektir. Bu rotaların uzunluklarının dinamikleri Şekil 5.1'de gösterilmektedir.



Şekil 5.1. Karıncaların iterasyonlara bağlı olarak rotalarının uzunluk dinamiği

Görüldüğü gibi rotaların değişim hızları farklı olmakla birlikte belirli aralıklarla değişmektedir. Bunun nedeni bu rotaların ana rotaya göre seçilmesidir. Bu rotalar arasından her bir karınca için en kısa ve en uzun rotayı seçebiliriz. Örneğin, 3. karınca için en kısa yol 6. iterasyonda yaklaşık 79 iken en uzun rota 9. iterasyonda yaklaşık 88'dir. Karınca algoritmasının verimini hesaplamak için, Şekil 5.1'de oluşturulan yolların uzunluklarının ana rotadan bağıl hatası hesaplanmıştır (Şekil 5.2).

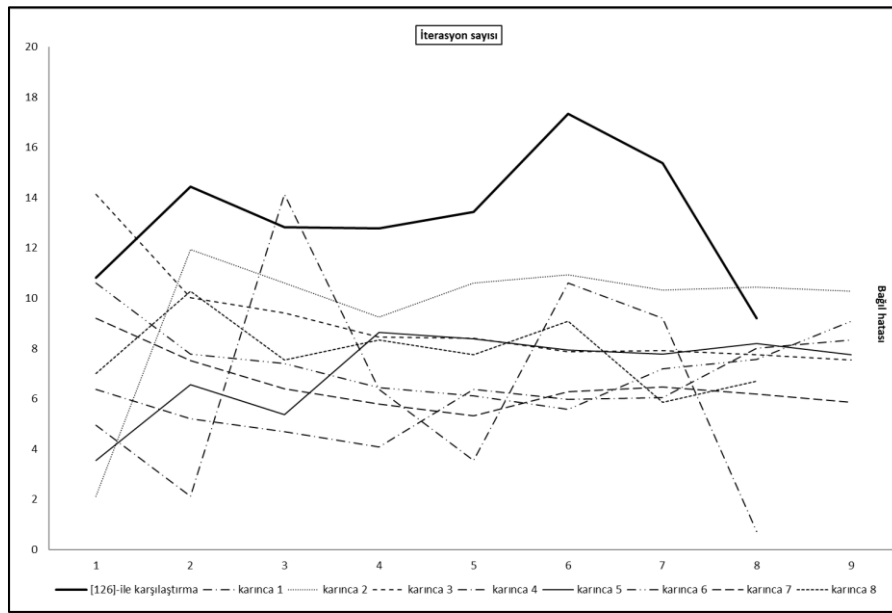
$$N_{ij} = \frac{100 \cdot ABS(M_{ij} - M_b)}{M_b} \quad (5.13)$$

Burada,

$M_{ij}$ -i-nci karıncanın j'nci iterasyonda A noktasından B noktasına gittiği rotanın birimlerle uzunluğudur.

$M_b$ -ana rotasının birimlerle uzunluğudur. Bu durumda da  $M_b = 78$ .

$N_{ij}$ -i-nci karıncanın j'nci iterasyonda A noktasından B noktasına gittiği rotadaki ana rotadan bağlı hatasıdır.



Şekil 5.2. Karıncaların belirlediği rotanın uzunluklarının diğer çalışmada verilmiş yöntemle bulunmuş rotasındaki bağlı hatası

Şekilden de görülebileceği gibi, bağlı hatanın en düşük değeri yaklaşık % 2'dir. Şekil 5.1'de düz siyah çizgi ile her iterasyonda tüm karıncaların rotalarının ortalama uzunluğu ile ana rotanın bağlı hatası verilmiştir. Her iterasyon, tüm karıncaların yollarının ortalama uzunluğu ile bağlı hatasını gösterir. Bağlı hata ile ana rota arasındaki fark, genel bir kalite göstergesidir ve karınca algoritmasının çalıştığını gösterir.

Şekil 5.1'e göre önerilen algortmada en kısa yol 1. karınca için 10. iterasyonda, 2. karınca için 4. iterasyonda ve 7. karınca için 9. iterasyonda elde edilir. Karınca algoritması kullanılarak oluşturulan en kısa rotanın uzunluğu 74 birimdir (Resim 5.1, yeşil çizgi). Bu

rota, temel rotadan 4 birim kadar kısadır. Bu, karınca algoritması ile daha kısa yolun bulunduğunu gösterir.

## 6. SONUÇ VE ÖNERİLER

Bu çalışmada İHA'ların engel olarak belirlenen bölgelerde uçuş rotasının optimizasyonu konusuna odaklanmıştır. İHA'ların şu anda sivil ve askeri alanlarda çeşitli amaçlarla kullanımı giderek artmaktadır. Amaçlara bakılmaksızın İHA'ların etkili ve güvenilir kullanımı her durumda en önemli konulardan birisidir. Bu durum öncelikle uçuş ve enerji olanaklarının sınırlılığı ile diğer yandan çözülmesi gereken rota optimizasyonu sorunun özelliklerinden kaynaklanmaktadır.

İHA'ların uçuş ve enerji kısıtlamaları ile problemten kaynaklanan sınırlamalar dikkate alınarak çözülen problemler genellikle matematiksel bir programlama problemi olarak ele alınır. Burada İHA'ların özelliklerinden kaynaklanan sınırlamalardan bahsedildiğinde uçuş mesafesi, rakım, hız, yük ağırlığı, uçuş süresi vb. kısıtlamalar anlaşılır. Bu durumda, İHA'ların kullanımında verimliliği, en kısa uçuş mesafesi, yükün maksimum ağırlığı, minimum uçuş süresi vb. koşullar esas alınarak belirlenir.

Bu tür konularda fiziki harita göz önüne alındığında, rota üzerinde engeller olduğunda çeşitli konuların çözülmesi son yıllarda araştırmacılar tarafından daha çok dikkat çekmektedir. Sorunu çözmek için ilk önce İHA'nın rotasının engellerin sınırı ile kesişim noktalarının bulunması gerekmektedir. Bu konular araştırmacılar tarafından fiziki olarak ölçüm, görüntüleme, radar vb. yöntemlerle, matematiksel modelleme olarak da imitasyon, dekompozisyon, ağ yöntemleri vb. çözümlerle üzerinde çalışılmıştır.

Bu tezde gerekli problemin çözümü için iki yönde çalışma yapılmıştır:

- Engellerin sınırı ile uçuş yörüngesinin bulunması için yeni bir algoritma önerilmiştir.
- Önerilen algoritmaya dayanarak, en kısa uçuş yörüngesinin bulunması problemi çözülmüştür.

Engellerin sınırı ile uçuş yörüngesinin bulunması problemi araştırmacılar tarafından farklı yaklaşımlarla çözülmüştür. Çoğu durumda uçuşun gerçekleştirilebileceği tüm bölge küçük parçacıklara ayrılarak, her bir parçacıkta engellerin olup olmadığı konusuna bakılmıştır ve sonuçlar birleştirilmiştir. Bu çözüm yöntemleri hem işlemlerin hacmi nedeniyle, hem de

kullanılan algoritmanın karmaşıklılığı nedeniyle, önerilen algoritmadan bir kaç kat daha büyüktür.

Tezde önerilen algoritma iki doğru parçası kesişiminde elde edilen dışbükey dörtgenin özelliklerine dayanmaktadır. Bu özelliği kullanarak iki noktayı birleştiren doğru parçası ile kesişen doğru parçasının (engelin sınırı) kesişim noktası kolayca tanımlanır. Sonuç olarak, engellerden kaçınarak iki noktayı birleştiren ve engellerin sınırları boyunca hareket eden bir rota elde edilir. Engeller olmadığında bu rota İHA'nın uçuşunun başladığı ve bittiği noktayı bağlayan doğru parçası olacaktır. Engeller olduğunda ise bu doğru çizgiden ve engellerin sınırından oluşan kırık çizgilerden oluşacaktır.

Önerilen algoritma hesaplama miktarına göre engellerin sayısında doğrusal bağıntı olmakla, aynı zamanda zorluk derecesine göre birinci derleme olduğundan algoritmik dillerde daha kolay programlanabilir.

Önerilen algoritmaya dayanarak en kısa uçuş rotasının belirlenmesi konusu karınca kolonisi algoritması uygulanarak çözülmüştür. Bu tezde kullanılan Karınca Kolonisi Algoritması klasik karınca algoritmasından aşağıdaki özelliklerine göre farklılıklar göstermektedir:

1. Optimal rota, engelleri belirleyen algoritmaya dayalı rotaya göre tanımlanır. Bu rota temel alınarak Karınca Kolonisi Algoritması uygulanır.
2. Karıncaların davranışları üzerindeki kısıtlamalar tabu listesi tarafından uygulanır. Tabu listesi, alan kabartma haritasına dayanan özel kurallara göre düzenlenmiştir.
3. Hareket yönünün ve feromonun azalma hızının değişmesi mevcut hareket rotası ana rotaya yakınlığı ile belirlenir.

Her iki algoritma Google Maps ve Google.map tabanlı herhangi arazide uygulama için arayüz oluşturulmuş ve hesaplama deneyleri ile yazılım pratik kullanımının doğruluğu test edilmiştir.

Bu nedenle sonuçlar aşağıdaki gibi sıralanabilir:

1. İHA'ların kısıtlı olarak belirlenen bölgelerde uçuş rotasının oluşturulmasına ilişkin çalışmalar araştırılmış, onların olumlu ve olumsuz yönleri gösterilmiştir.

2. İncelenen arařtırmalar bilimsel-teorik temeller, problem belirleme, çözüm yöntemi, kullanılan yöntemler, pratik uygulanabilirlik düzeyine göre sıralanmıştır.
3. Engel olarak belirlenen araziler olduėunda İHA'ların bir noktadan diėer noktaya uçuşunu gerçekleřtiren engellerin sınırları boyunca geen kırık izgilerden oluřmuř rotanın belirlenmesinin yeni algoritması önerilmiştir.
4. Engel olarak belirlenen araziler olduėunda İHA'ların bir noktadan diėer noktaya uçuşunu gerçekleřtiren en kısa rotayı belirleyen ve Karınca Kolonisi Algoritmasına dayalı güncelleřtirilmiř algoritma önerilmiştir.
5. Önerilen algoritmaya göre yazılım geliřtirilmiř ve hesaplama deneyleri yapılmıştır.
6. Hesaplama deneylerinin sonuçları algoritmaların pratikliėini ve Google Haritalar üzerinde istenilen araziler için uygulama yapılabildiėini göstermiştir.

Gelecekte İHA'ların askeri ve sivil alanlarda kullanımının daha da artması nedeniyle önerilen algoritmaların ok sayıda İHA olduėunda uygulanması öngörölmelidir.

Önerilen algoritmaların, fiziki haritalar kullanılmadan, engel olarak belirlenen arazilerin özellikleri göz önünde bulundurularak, sinirsel modelleme ile İHA'lara bu bilgileri öğretilmesi konusu ele alınmalıdır.

Önerilen ve gelecekte düşünölen algoritmalar temelinde belirli fiziksel ve aerodinamik özelliklere sahip İHA'ların yönetimi için algoritmalar oluřturmak gerekir.



## KAYNAKLAR

1. Shakhathreh, H., Sawalmeh, A., Fuqaha, A., Dou, Z., Almatia, E., Issa Khalil, I., Othman, S. N., Khreishah, A. and Guizani, M. (2019). Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *Digital Object Identifier*, 48572–48634.
2. Randjelovic, A., Radovanović, M. and Milic, A (2019). Use of drones in operations in the urban environment. *Conference: 5 th International Scientific conference Safety and crisis management - Theory and practise Safety for the future*, 124-130.
3. Яковлев, К.С., Макаров, Д.А. и Баскин, Е.С. (2014). Метод автоматического планирования траектории беспилотного летательного аппарата в условиях ограничений на динамику полета. *Искусственный интеллект и принятие решений*, 3-17.
4. Яковлев, К.С., Петров, А.В. и Хитыков, В.В. (2013). Программный комплекс навигации и управления беспилотными транспортными средствами. *Информационные технологии и вычислительные системы*, 72-83.
5. Яковлев К.С., Макаров Д.А., Панов, А.И. и Зубарев, Д.В. (2013). Принципы построения многоуровневых архитектур систем управления беспилотными летательными аппаратами. *Авиакосмическое приборостроение*, 10-28.
6. Моисеев, В.С. (2017). Групповое применение беспилотных летательных аппаратов:– Казань: Редакционно- издательский центр Школа, 572.
7. Рэндал, У. и Тимоти, Б.У. (2015). Мак Лэйн, Малые беспилотные летательные аппараты: теория и практика. *Москва: ТЕХНОСФЕРА*, 312.
8. Chen,X., Xu, R. and Zhao, J. (2017). Multiobjective route planning for UAV. *4th International Conference on Information Science and Control Engineering*, 1023-1027.
9. İnternet: İnsansız Kara Aracı. URL: [https://tr.wikipedia.org/wiki/%C4%B0nsans%C4%B1z\\_kara\\_arac%C4%B1](https://tr.wikipedia.org/wiki/%C4%B0nsans%C4%B1z_kara_arac%C4%B1), Son Erişim Tarihi: 04.04.2020.
10. Yakut, M., Yılmaz, S., İnce, S., Murat, O. ve Aygün, E. (2015). Derinlik ve Yön Kontrol Uygulamaları için Sualtı Aracı Tasarımı. *Gazi Üniversitesi Fen Bilimleri Dergisi Part:C, Tasarım ve Teknoloji*, 3(1), 343-355.
11. Dorling, K., Heinrichs, J., Messier, G. and Magierowski, S. (2017). Vehicle Routing Problem for Drone Delive. *The Institute of Electrical and Electronics Engineers Transactions on Systems, Man, and Cybernetics: Systems*, 47(1), 70-85.
12. Prisacariu, V. (2017). The History and The Evolution of UAV's from the Beginning Till the 70s Air Force Academy. *Journal of Defense Resources Management*, 8(1), 181-189.

13. İnternet: How drone strikes work. URL: <https://science.howstuffworks.com/drone-strikes.htm>, Son Erişim tarihi: 07.12.2020
14. İnternet: The Early Days Of Drones – Unmanned Aircraft From World War One And World War Two. URL: <https://www.warhistoryonline.com/military-vehicle-news/short-history-drones-b.html>, Son Erişim Tarihi: 07.12.2020.
15. Arjomandi, M., Agostino, S., Mammone, M., Nelson, M. and Zhou, M. (2006). Aeronautical Engineering Classification of Unmanned Aerial Vehicles *Mechanical Engineering 3016 Aeronautical Engineering*, 1-49.
16. İnternet: General Atomics GNAT. URL: [https://en.wikipedia.org/wiki/General\\_Atomics\\_GNAT](https://en.wikipedia.org/wiki/General_Atomics_GNAT), Son Erişim Tarihi: 07.12.2020.
17. Kahvecioğlu, S. ve Oktal, H. (2014). Turkish UAV capabilities as a new competitor in the market. *International Journal of Intelligent Unmanned Systems*, 2(3), 183-191.
18. Akyürek, S., Yılmaz, M.A. ve Taşkiran, Y. (2012). *İnsansız Hava Araçları Muharebe Alanında ve Terörle Mücadelede Devrimsel Dönüşüm*. Türkiye: Bilgesam Yayınları, 1-52.
19. İnternet: ANKA İnsansız Hava Aracı. URL: <https://www.tusas.com/urun/anka>, Son Erişim Tarihi: 04.04.2020.
20. Karaağaç, C. (2016) İHA Sistemleri Yol Haritası Geleceğin Hava Kuvvetleri 2016-2050. *STM Savunma Teknolojileri Mühendislik ve Ticaret A.Ş.*, 1-80.
21. İnternet: Bayraktar TB'2 İHA URL: [https://tr.wikipedia.org/wiki/Baykar\\_Bayraktar\\_TB2](https://tr.wikipedia.org/wiki/Baykar_Bayraktar_TB2), Son Erişim Tarihi: 12.10.2019.
22. İnternet: Bayraktar TB'2 İnsansız Hava Aracı, URL: <http://www.millisavunma.com/bayraktar-tb2-insansiz-hava-araci/>, Son Erişim Tarihi: 07.12.2020.
23. İnternet: Vestel Karayel İnsansız Hava Aracı. <http://www.millisavunma.com/vestel-karayel-insansiz-hava-araci>, Son Erişim Tarihi: 07.12.2020.
24. Odom, E. (2002). Future Employment of Unmanned Aerial Vehicles. *Aerospace Power Journal*, 16(2), 86-97.
25. Karakuş, C. ve Katman, F. (2019). Male Sınıfı İnsansız Hava Aracı (İHA) Teknolojisi ve Konvansiyonel (geleneksel) Savaştaki Yeri. *Akademik Tarih ve Düşünce Dergisi*, (6)2, 882-897.
26. Ceylan, A. (2019). *İrtifa Uzun Menzil İHA Tasarımı Ve Uçuş Testleri*, Yüksek lisans tezi, T.C. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri.
27. Mathew, L., Bansod, B. and Singhal, G. (2018). Unmanned Aerial Vehicle classification, Applications and challenges: A Review. *Preprints 2*. 1-19.

28. İnternet: Drone capabilities. URL : <https://www.911security.com/learn/airspace-security/drone-fundamentals/drone-capabilities-endurance-range>. Son Erişim Tarihi: 15.12.2019.
29. Krishna, K.R. (2019). *Unmanned Aerial Vehicle Systems in Crop Production*, Florida: A Compendium Apple Academic Press Published, 710.
30. Çoban, S. ve Oktay, T. (2018). Unmanned Aerial Vehicles (UAVs) According to Engine Type. *Journal of Aviation*, 2 (2), 177-184.
31. Korchenko, A.G. and Illyash, O.S. (2013) The Generalized Classification of Unmanned Air Vehicles. *The Institute of Electrical and Electronics Engineers 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings*, 28-34.
32. İnternet: Argrow, B. M., Axisa, D., Chilson, P., Ellis, S., Fladeland, Frew, M., E., Jacob, W. J., Lord, M., Moore, J., Oncley, S., Roberts, G., Schoenung, S., Vömel, H. and Wolff, C. (2017). Unmanned Aircraft Systems for Atmospheric Research. *National Center for Atmospheric Research NCAR Earth Observing Laboratory EOL Workshop*. URL: <https://www.eol.ucar.edu/eol-uas-workshop>. Son Erişim Tarihi 23.07.2020.
33. Bale, D.L.T., Ugwu, C. and Nwachukwu, E.O. (2016). Route Optimization Techniques: An Overview. *In Proceedings of International Journal of Scientific and Engineering Research*, 7(11), 1367-1372.
34. Родзин, С.И. и Курейчик, В.В. (2017). Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации (обзор). *Кибернетика и программирование*, 3, 51-79.
35. Мехтиев, Ш. и Мехдиев, А.(2018). Некоторые вопросы безопасности применения дронов. *Материалы IV республиканская конференция Актуальные научно практические мультидисциплинарные проблемы информационные безопасности*, 91-94.
36. Christofides, N., Mingozzi, A. and Toth, P. (1981). Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations. *Math Program* 20, 255–282.
37. Christofides, N., Mingozzi A. and Toth, P. (1981). State-space relaxation procedures for the computation of bounds to routing problems. *Networks* 11, 145–164.
38. Лебедев, Г.Н. и Мирзоян, Л.А. (2005). Нейросетевое действий по облету наземных объектов группой летательных аппаратов. *Авиакосмическое приборостроение*, 12, 41-47.
39. İnternet: Araç Rota Planlama. URL: <http://sabri.suyunu.com/tag/rota-planlama/>, Son Erişim Tarihi: 07.12.2020.
40. Keskindürk, T. (2015). Araç Rotalama Problemleri ile Çözüm Yöntemlerinin Sınıflandırılması ve Bir Uygulama. *İşletme Bilimi Dergisi*, 3(2), 77-107.

41. İnternet: Dijkstra's Algorithm. URL: <http://www.questionsolves.com/Website-Content/DijkstraAlgorithm.php>, Son Erişim Tarihi: 07.12.2020.
42. Şahin, Y. ve Eroğlu, A. (2015). Sipariş Toplama ve Kapasite Kısıtlı Araç Rotalama Problemlerinin Hiyerarşik Çözümü. *Süleyman Demirel Üniversitesi Mühendislik Bilimleri ve Tasarım Dergisi*, 3(1), 15-28.
43. Göçken, T., Yaktubay, M. ve Kılıç, F. (2018). Zaman Pencerele Araç Rotalama Problemi Çözümü İçin Çok Amaçlı Genetik Algoritma Yaklaşımı. *Gazi Üniversitesi Fen Bilimleri Dergisi, Part C*, 6(4), 774-786.
44. Erol, V. (2006). *Araç Rotalama Problemleri İçin Popülasyon Ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı Ve Uygulaması*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
45. Çeyrekoğlu, S. (2017). *Araç Rotalama Problemine Genetik Algoritma Yaklaşımı Ve Örnek Bir Uygulama*, Yüksek Lisans Tezi, T.C. İstanbul Üniversitesi, Fen Bilimleri Enstitüsü.
46. Sural, H. (2003). Gezgin satıcı problemi. *Matematik dünyası dergisi*, 37-40.
47. Jain, C. and Kumawat, J. (2013). Processing Delay Consideration in Dijkstra's Algorithm, *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(8), 407-411.
48. Gupta, N., Mangla, K., Jha, A.K. and Umar, A. (2016).Applying Dijkstra's Algorithm in Routing Process. *International Journal of New Technology and Research (IJNTR)*, 2(5), 122-124.
49. Bayzan, Ş. (2005). *Araç Rotalarının En Kısa Yol Algoritmaları Kullanılarak Belirlenmesi ve .Net Ortamında Simülasyonu*, Yüksek lisans tezi, T.C. Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü.
50. Нечаев, В.В. и Миняйло, Д.В. (2019). Некоторые аспекты группового управления многокомпонентными системными объектами на основе биоинспирированных алгоритмов. *International Journal of Open Information Technologies*, 7(6), 94-100.
51. Huang, H. and Chung, S. (2004). Dynamic visibility graph for path planning. *The Institute of Electrical and Electronics Engineers International conference on intelligent robots and systems*, 2813–2818.
52. Janet, J.A., Luo, R.C. and Kay, M.G. (1995). The essential visibility graph: An approach to global motion planning for autonomous mobile robots. *The Institute of Electrical and Electronics Engineers International conference on robotics and Automation*, 1958–1963.
53. Habib, M.K. and Asama, H. (1991). Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach. *The Institute of Electrical and Electronics Engineers international workshop on intelligent robots and systems*, 2, 563–567.

54. Wallgrun, J.O. (2010). Voronoi graph matching for robot localization and mapping. *Transactions on computational science IX*, Springer, 6290, 76–108.
55. Ladd, A.M. and Kavraki, L.E. (2004). Measure theoretic analysis of probabilistic path planning. The Institute of Electrical and Electronics Engineers. *Transactions on Robotics and Automation*, 20(2), 229–242.
56. Amato, N.M. and Wu, Y.A. (1996). Randomized roadmap method for path and manipulation planning. *The Institute of Electrical and Electronics Engineers International conference on robotics and Automation*, 113–120.
57. Sleumer, N.H. and Tschichold-Gurman, N. (1999). Exact cell decomposition of arrangements used for path planning in robotics. 329, *Technical Report*, Zurich, 1-14.
58. Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 46–57.
59. Yahja, A., Stentz, A., Singh, S. and Brumitt, B.L. (1998). Framed-quadtree path planning for mobile robots operating in sparse environments. *The Institute of Electrical and Electronics Engineers International Conference on robotics and Automation*, 1, 650–655.
60. Kitamura, Y., Tanaka, T., Kishino, F. and Yachida, F. (1995). M. 3-D path planning in a dynamic environment using an octree and an artificial potential field. *The institute of electrical and electronics engineers international conference on intelligent robots and systems*, 2, 474–481.
61. Redding, J., Amin, J., Boskovic, J., Kang, Y., Hedrick, K., Howlett, J. and Poll S. A. (20-23 August 2007). Real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters. *American Institute of Aeronautics and Astronautics, navigation and control conference and exhibition*. 1-22.
62. Chazelle, B. and Palios, L. (1990). Triangulating a nonconvex polytope. *Discrete and Computational Geometry*. 5(5), 505–526.
63. Russell, S.J. and Norvig, P. (2010). *Artificial intelligence: A modern approach* (third edition) New Jersey: Upper Saddle River: Prentice Hall, 1132.
64. Ferguson, D. and Stentz, A. (2006). Using interpolation to improve path planning: The field D\* Algorithm. *Journal of Field Robotics*, 23(2), 79–101.
65. Koenig, S., Likhachev, M. and Furcy, D. (2004). Lifelong planning A\*. *Artificial Intelligence*, 155(1-2), 93–146.
66. Koenig, S. and Likhachev, M. (2002). D\* lite. *18th National conference on artificial intelligence*, 476–483.
67. Reddy, H. (2013). Path Finding - Dijkstra's and A\* Algorithm's. *International Journal in IT and Engineering*, 1-15.

68. De Filippis, L., Guglieri, G. and Quagliotti, F. (2011). A minimum risk approach for path planning of UAVs. *Journal of Intelligent and Robotic Systems*, 61(1–4), 203–219.
69. De Filippis L., Guglieri G. and Quagliotti, F. (2012). Path planning strategies for UAVs in 3D environments. *Journal of Intelligent and Robotic Systems*, 65(1–4), 247–264.
70. De Filippis, L. (2012). *Advanced path planning and collision avoidance algorithms for UAVs*: (Doctoral dissertation Torino: Ist. Politecnico di Torino) 142.
71. Alvarez, D., Gomez, J.V., Garrido, S. and Moreno, L., (2015). 3D robot formations path planning with fast marching square. *Journal of Intelligent and Robotic Systems* 80(3–4), 507–523.
72. Osher, S. and Sethian, J.A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1), 12–49.
73. Ge, S.S. and Cui, Y.J. (2000). New potential functions for mobile robot path planning. *The Institute of Electrical and Electronics Engineers Transaction on Robotics and Automation*, 16(5), 615–620.
74. Jing, R., McIsaac, K.A. and Patel, R.V. (2006). Modified Newton's method applied to potential fieldbased navigation for mobile robots. *The Institute of Electrical and Electronics Engineers Transaction on Robotics*, 22(2), 384–391.
75. Ferrara, A. and Rubagotti, M. (2008). Real-time obstacle avoidance-mode control of a mobile robot based on a harmonic potential field. *The Institute of Engineering and Technology Control Theory and Applications*, 2(9), 807–818.
76. Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots *International Journal of Robotics Research*, 5(1), 90–98.
77. Borenstein, J. and Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *The Institute of Electrical and Electronics Engineers Transaction on Robotics and Automation*, 7(3), 278–288.
78. Fujimura, K. and Samet, H. (1989). A hierarchical strategy for path planning among moving obstacles (mobile robot). *The Institute of Electrical and Electronics Engineers Transaction on Robotics and Automation*, 5(1), 61–69.
79. Zou, X. and Jing, Z. (2003). Virtual local target method for avoiding local minimum in potential field based robot navigation. *Journal of Zhejiang University. - Science A*, 4(3), 264–269.
80. Ulrich, I. and Borenstein, J. (1998). VFH+: Reliable obstacle avoidance for fast mobile robots. *The Institute of Electrical and Electronics Engineers conference on robotics and automation processing. The Institute of Electrical and Electronics Engineers* 2, 1572–1577.

81. Ulrich, I. and Borenstein, J. (2000). VFH\*: Local obstacle avoidance with look-ahead verification. *The Institute of Electrical and Electronics Engineers International conference on robotics and automation* the Institute of Electrical and Electronics Engineers, 3, 2505–2511.
82. Betts, J.T. (1998). Survey of numerical methods for trajectory optimization. *Jorunal of Guidance, Controland Dynamics*, 21(2), 193–207.
83. Ross, I.M. and Fahroo, F. A. (2002). Perspective on methods for trajectory erenceoptimization. *Astroynamics specialist conference and exhibition*. 1–7.
84. Fliess, M., Levine, J., Martin, P. and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6), 1327–1361.
85. İnternet: Gill, P.E., Murray, W. and Saunders, M.A. (June 2008). User's guide for SNOPT Version 7: software for large scale nonlinear programming. URL: [https://www.researchgate.net/publication/228974536\\_User%27s\\_Guide\\_for\\_SNOPT\\_Version\\_7\\_Software\\_for\\_Large-Scale\\_Nonlinear\\_Programming](https://www.researchgate.net/publication/228974536_User%27s_Guide_for_SNOPT_Version_7_Software_for_Large-Scale_Nonlinear_Programming), Son Erişim Tarihi: 07.12.2020.
86. Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1998). User's guide for NPSOL (version 5.0): a Fortran package for nonlinear programming, Technical Report NA 98-2. California 1-45.
87. Culligan, K., Valenti, M., Kuwata, Y. and How, J.P. (2007). Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization. *The Institute of Electrical and Electronics Engineers, American Control Conference*, 5322–5327.
88. Schouwenaars, T., De Moor, B. and Feron, E. (2001). How J. Mixed integer programming for multivehicle path planning. *Europe Control conference: ECC. The Institute of Electrical and Electronics Engineers*, 2603–2608.
89. Earl, M.G. and D'Andrea, R. (2005). Iterative MILP methods for vehicle-control problems. *The Institute of Electrical and Electronics Engineers*, 21(6), 1158–1167.
90. Kuwata, Y. (2003). *Real-time trajectory design for unmanned aerial vehicles using receding horizon control*, Master's thesis, Massachusetts Institute of Technology Department of Aeronautics and Astronautics, Cambridge, 151.
91. Habibi, G., Masehian, E. and Beheshti, M.T.H. (2007). Binary integer programming model of point robot path planning. *33rd annual conference of the The Institute of Electrical and Electronics Engineers Industrial Electronics*, 2841–2845.
92. İnternet: Kabalcı, E. Yapay Sinir Ağları, URL: <https://ekblc.files.wordpress.com/2013/09/ysa.pdf>
93. İnternet: Denetleme ve Düzenleme Sistemi, Sinir Sistemi ve Endokrin Sistem. URL: <https://www.biyolojidersnotlari.com/denetleme-ve-duzenleme-sistemi-sinir-sistemi-ve-endokrin-sistem.html>, Son Erişim Tarihi: 07.12.2020.

94. Öztemel, E. (2003). *Yapay Sinir Ağları*. (Üçüncü baskı). İstanbul: Papatya yayıncılık, 1-44.
95. Кореванов, С.В. (2014). Моделирование Групповых Действий Беспилотных Летательных Аппаратов. *Научный Вестник Мзту Га*, 138-141.
96. Eberhart, R. C. and Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science Journal*, 39-43.
97. Jung, L., F., Knutzon, J., S., Oliver, J..H. and Winer, E.H. (2006). Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization. *11th Multidisciplinary Analysis And Optimization Conference*, 992–1001.
98. Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*, (Doctoral dissertation, C, Politecnico di Milano, Italia).
99. Stützle, T. and Hoos, H. (2000). MAX MIN Ant System. *Future Generation Computer Systems*. 16(8), 889—914.
100. Lučić, P. and Teodorović, D. (2003). Computing with bees: Attacking complex transportation engineering problems. *International Journal on Artificial Intelligence Tools*, 12(3), 375–394.
101. Pham, D.T. and Castellani, M. (2009). The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimisation Problems. *Proceedings of the Institution of Mechanical Engineers, Part C*, 223(12), 2919-2938.
102. Курейчик В.М. и Кажаров А.А. (2010). Муравьиные алгоритмы для решения транспортных задач. *Известия РАН. Теория и системы управления*, 32–45.
103. Кочегурова Е.А., Мартынов Я.А., Мартынова Ю.А. и Цапко С.Г. (2014). Алгоритм муравьиных колоний для задачи проектирования рациональных маршрутных сетей городского пассажирского транспорта. *Вестник СибГУТИ*, 3(27), 89–100.
104. Леванов Т.В. и Усько, О.В. (2010). Алгоритм муравьиной колонии для одной задачи размещения с ограничениями на мощность производства. *Вестник УГАТУ*, 2(37), 202–208.
105. Козуб, А.Н. и Кучеров, Д.П. (2013). Интеграционный подход к задаче выбора маршрута группы БПЛА. *Искусственный интеллект*, 4(62), 333-343.
106. İnternet: Karakaya, M. (2016). En Az Sayıda İnsansız Hava Aracı Kullanarak Sabit Hedeflerin Gözetlenmesinin Planlanması, URL: [http://user.atilim.edu.tr/~murat.karakaya/publications/TOK2013\\_En\\_Az\\_IHA.pdf](http://user.atilim.edu.tr/~murat.karakaya/publications/TOK2013_En_Az_IHA.pdf), Son Erişim Tarihi: 06.12.2020.
107. Jevti, A. and Andina, D. (2010). Unmanned Aerial Vehicle Route Optimization Using Ant System Algorithm. *5th International Conference on System of Systems Engineering, The Institute of Electrical and Electronics Engineers*, 1-6.

108. Ma, G., Duan, H. and Liu, S. (2007). Improved Ant Colony Algorithm for Global Optimal Trajectory Planning of UAV under Complex Environment. *International Journal of Computer Science & Applications*. 4(3), 57-68.
109. Bansal, J., C., Sharma, H. and Jadon, S.S. (2013). Artificial bee colony algorithm: A survey. *Article in International Journal of Advanced Intelligence Paradigms*, 5(1-2), 123-159.
110. Karaboğa, D. ve Akay, B. (2007). Artificial Bee Colony (ABC) Algorithm on Training Artificial Neural Networks. *Signal Processing and Communications Application*, 1(3), 818-821.
111. Çavuş, V. ve Tuncer, A. (2016). İnsansız Hava Araçları İçin Yapay Arı Kolonisi Algoritması Kullanarak Rota Planlama. *Karaelmas Fen ve Mühendislik Dergisi Araştırma Makalesi*. 7(1), 259-265.
112. Li, B., Gong, L. and Yang, W. (2014). An Improved Artificial Bee Colony Algorithm Based on Balance-Evolution Strategy for Unmanned Combat Aerial Vehicle Path Planning. *The Scientific World Journal. Article I*, 1-10.
113. Nurcahyo, G.W., Shamsuddin, S.M., Alias, R.A. and Sap, M.N.M. (2003). Selection of defuzzification method to obtain crisp value for representing uncertain data in a modified sweep algorithm. *Journal of Computer Science and Technology*. 3(2), 22-28.
114. Gendreau, M. (2002). *An Introduction To Tabu Search* Boston: Springer *Handbook of Metaheuristics* 37-54.
115. Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine* 4(11), 218-21.
116. Sastry, K., Goldberg, D. and Graham, K. (2014). *Genetic Algorithms*, Boston: Springer *Search Methodologies*, 97-125.
117. Reeves, C. R. (2010). *Genetic Algorithms Chapter* (second edition) In *Handbook of Metaheuristics International series in operation research management science*, 109-139.
118. R. L. Galvez, E. P. Dadios and A. A. Bandala. (2014). Path Planning for Quadrotor UAV Using Genetic. *7th Path Planning for Quadrotor Unmanned Aerial Vehicle (UAV) Using Genetic Algorithm The Institute of Electrical and Electronics Engineers International Conference*, 1-6.
119. Иванов, А. (2016). Генетический алгоритм решения задачи коммивояжера для планирования маршрута беспилотного летательного аппарата, *Национальный аэрокосмический университет им. Н. Е. Жуковского Харьковский авиационный институт Открытые информационные и компьютерные интегрированные технологии*, 202-205.
120. Ercan, C. ve Gencer, C. (2013). İnsansız Hava Sistemleri Rota Planlaması Dinamik Çözüm Metotları ve Literatür Araştırması. *Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi*, 1(2), 51-72.

121. Toth, P. and Vigo, D. (2001). *The Vehicle Routing Problem*, Philadelphia, PA: SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics Publishing, 109-128.
122. Ozalp, N., Sahingoz, O.K. and Ayan, U. (2013). Autonomous Unmanned Aerial Vehicle Route Planning, *In Processing of The Institute of Electrical and Electronics Engineers 21st. Signal Processing and Communications Applications Conference (SIU)*, 228-231.
123. Jung, S. and Kim, H. (2017). Analysis of Amazon Prime Air UAV Delivery Service, *Journal of Knowledge Information Technology and Systems*, 12(2), 15-20.
124. Ozalp, N., Sahingoz, K. ve Ayan, A. (2014). 3 Boyutlu Arazi Üzerinde Çoklu Otonom İnsansız Hava Aracı Rota Planlaması. *The Institute of Electrical and Electronics Engineers 22nd Signal Processing and Communications Applications Conference*, 228-231.
125. Ergezer, H. ve Leblebicioğlu, H. (2012). Çoklu İnsansız Hava Araçları için Görev Planlaması. *20. The Institute of Electrical and Electronics Engineers Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, 49(1), 502-520.
126. Mehlhorn, K. and Naher, S. (1994). Implementation of a Sweep Line Algorithm for the Straight Line Segment Intersection Problem. *Technical Report MPI-I-94-160, Max-Planck-Institut für Informatik, Germany*, 1-41.
127. Daniel, K., Nash, A., Koenig, S. and Felner, A. (2010). Theta\*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 1177–1183.
128. Ага-Заде, С. и Джийлан, Б. (2019). Метод Оценки Параметров Траектории Блa В Мегaполисе При Наличии, *Проблеми інформатизації та управління Національного Авіаційного Університету в журнале*, 2 (62), 5-11.
129. Aghazada, S. ve Ciylan, B. (2019). Engel Olarak Belirlenen Araziler Olduğunda İnsansız Hava Araçlarının Karınca Kolonisi Algoritması ile Optimal Rotanın Bulunması. *ICEST'19 I Uluslararası Bilimsel-Uygulamalı Konferansı III. Bölüm*. 220-225.

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : Aghazada Sabina  
 Uyuğu : Azerbaycan  
 Doğum tarihi ve yeri : 01.09.1995  
 Medeni hali : Bekâr  
 Telefon : 0 (537) 683 19 74  
 E-mail : sabina.aghazada@gazi.edu.tr



### Eğitim

| Derece        | Eğitim Birim                                                    | Mezuniyet Tarihi |
|---------------|-----------------------------------------------------------------|------------------|
| Yüksek lisans | Gazi Üniversitesi / İleri Teknolojiler                          | Devam ediyor     |
| Lisans        | Milli Havacılık Akademisi/<br>Bilişim Tek. ve Sis. Mühendisliği | 2017             |
| Lise          | Bakü 261 numaralı lise                                          | 2013             |

### İş Deneyimi

| Yıl | Yer | Görev |
|-----|-----|-------|
| —   | —   | —     |

### Yabancı Dil

Azerbaycan Türkçesi, İngilizce, Rusça.

### Yayınlar

1. Aghazada, S. (Ага-Заде, С.), Ciylan, B. (Джийлан, Б.) (2019). Метод Оценки Параметров Траектории Блa В Мегалополисе При Наличии (Method of Uav Route Parameters When There are Obstacles in Megalopolis), *Проблеми інформатизації та управління Національного Авіаційного Університету в журнале* (Problems of Informatization and Management, National Aviation University Ukraine). 2(62), 5-11.

2. Aghazada, S., Ciylan, B. (2019). Engel Olarak Belirlenen Araziler Olduğunda İnsansız Hava Araçlarının Karınca Kolonisi Algoritması ile Optimal Rotanın Bulunması. *İCEST'19 I Uluslararası Bilimsel-Uygulamalı Konferansı*, 220-225.
3. Aghazada S., Ciylan, B. (2018). Üzerindən Uçuşları Qadağan Edilən Məhdud Ərazilərdə Dronların Optimal Marşrutunun Müəyyən Edilməsi (Determining the Optimal Route of Drones in Restricted Areas). *İnformasiya Sistemləri və Texnologiyaları Nailiyyətlər və Perspektivləri Beynəlxalq Elmi Konfransın Materialları (Materials International Scientific Conference for the Information Systems and Technologies Achievements and Perspectives)*, 258-260.
4. Aghazada S., Ciylan, B. (2020). Optimal Route Planning for Unmanned Aerial Vehicles in Flight Restricted Zones with Ant Colony Algorithm. *International Conference on Advanced Materials Science & Engineering and High Tech Devices Applications ICMATSE 2020*. Paper ID: 93.

### **Hobiler**

Spor, kitab okumak, dil öğrenmek



*GAZİ GELECEKTİR..*