



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



ARAMA KURTARMA FAALİYETLERİNDE
SÜRÜ İHA'LARIN OPTİMUM UÇUŞ
UYGULAMASI

Nisanur Kiraz PEKİN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Mayıs-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Nisanur Kiraz PEKİN tarafından hazırlanan “Arama Kurtarma Faaliyetlerinde Sürü İHA’ların Optimum Uçuş Uygulaması.” adlı tez çalışması 14/04/2022 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç. Dr. Mehmet Akif ŞAHMAN

Danışman

Dr. Öğr. Üyesi Sait Ali UYMAZ

Üye

Doç. Dr. İsmail BABAOĞLU

İmza

.....

.....

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Nisanur Kiraz PEKİN

Tarih:

ÖZET

YÜKSEK LİSANS

ARAMA KURTARMA FAALİYETLERİNDE SÜRÜ İHA'LARIN OPTİMUM UÇUŞ UYGULAMASI

Nisanur Kiraz PEKİN
Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Sait Ali UYMAZ

2022, 70 Sayfa

Jüri
Dr. Öğr. Üyesi Sait Ali UYMAZ
Diğer Üyenin Unvanı Adı SOYADI
Diğer Üyenin Unvanı Adı SOYADI

Her geçen gün gelişen teknoloji ile afet sonrası arama kurtarma faaliyetleri gibi zorlu görevlerde İnsansız Hava Aracı (İHA) teknolojisi kullanılmaya ve geliştirilmeye başlanmıştır. Gelişmeler devam ederken tek bir İHA yerine çoklu İHA'lerden oluşan bir çeşit sürü ile daha büyük alanlar daha kısa sürelerde izlenebilir hale gelmiştir. Tüm bu gelişmeler ile İHA sürülerinden daha akıllı rotalar oluşturularak otonom olarak çalışmalarını planlanmıştır. Bu sayede afet gibi zorlu görevlerde sonuçlara daha hızlı ulaşılmaktadır.

Bu çalışmada, önceden oluşturulmuş bir yangın afeti senaryosu üzerinde çalışılmıştır. Senaryo gereği İHA'ların uçuşa başladığı koordinat, yangın merkezi koordinatı ve arama uzayı olacak sınır koordinatları belirlenmiştir. Senaryo ortamının gerçekleştirilmesi için sensörler tarafından alınması planlanan veriler sanal olarak oluşturulmuştur. Başlangıç noktasından uçuşlarına başlayan İHA'lar yangın bölgesine sanal sensörlerden alınan veriler ile yaklaştırmaya çalışarak minimum sürede maksimum yaklaşma ile problemi optimize etmişlerdir. Tüm bölgenin İHA'lar ile ızgara olarak taranması ile Parçacık Sürü Optimizasyon (PSO) ve Yapay Alg algoritması (AAA) sezgisel algoritmaları kullanılarak oluşturulan rotalar ile gezilmesi kıyaslanarak sezgisel algoritmaların kullanılmasının önemi gerçek bir problem özelinde incelenmiştir. AAA ve PSO algoritmaları da belirlenen senaryo üzerinde birbirleri arasında kıyaslanarak sonuçları incelenmiştir. Yapılan projenin gerçek ortama uyumlu olması için Mission Planner isimli yer kontrol istasyonu yazılımının simülasyon ortamı kullanılmıştır.

Anahtar Kelimeler: İHA, Sürü İHA, Sezgisel Yaklaşım, PSO, AAA, Mission Planner, Arama Kurtarma

ABSTRACT

MS THESIS

OPTIMUM FLIGHT APPLICATION OF FLOCK UAVS IN SEARCH AND RESCUE ACTIVITIES

Nisanur Kiraz PEKİN
Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Nisanur Kiraz PEKİN

2022, 70 Pages

Jury
Asst. Prof. Dr. Sait Ali UYMAZ
Diğer Üyenin Unvanı Adı SOYADI
Diğer Üyenin Unvanı Adı SOYADI

With the technology developing day by day, Unmanned Aerial Vehicle (UAV) technology has started to be used and developed in challenging tasks such as post-disaster search and rescue activities. As developments continue, larger areas can be tracked in shorter times with a kind of swarm of multiple UAVs instead of a single UAV. With all these developments, it is planned to work autonomously by creating smarter routes than UAV swarms. In this way, results are achieved faster in difficult tasks such as disasters.

In this study, a preformed fire disaster scenario was studied. According to the scenario, the coordinates where the UAVs start their flight, the fire center coordinate and the border coordinates that will be the search space have been determined. In order to realize the scenario environment, the data planned to be received by the sensors was created virtually. UAVs, which started their flights from the starting point, tried to approach the fire area with the data received from the virtual sensors and optimized the problem with the maximum approach in the minimum time. The importance of using heuristic algorithms is investigated in a real problem by comparing the scanning of the entire region as a grid with UAVs and the routes created using Particle Swarm Optimization (PSO) and Artificial Algae algorithm (AAA) heuristic algorithms. AAA and PSO algorithms were also compared with each other on the determined scenario and the results were examined. In order for the project to be compatible with the real environment, the simulation environment of the ground control station software called Mission Planner was used.

Keywords: Intuitive approach, UAV, Swarm UAV, PSO, Mission Planner, Search and Rescue

ÖNSÖZ

Bu tez kapsamında artan teknolojik faaliyetlerinden faydalanarak yangın gibi olağanüstü durumlarda arama kurtarma faaliyetlerini gerçekleştirmek için en efektif yöntemlerin bulunması amaçlanmıştır. Arama kurtarma gibi kritik ve zaman açısından oldukça hızlı olunması gereken bir görevde İHA kullanımı hali hazırda gerçekleşmektedir. Bu tezin amacı hem birden fazla İHA ile kullanılan bu sistemin biraz daha zeka katılarak minimum süreye indirilmesine yöneliktir. PSO ve AAA sezgisel algoritmaları ile İHA görevleri tanımlanarak bu görevi insanların oluşturması değil makinanın kendisi oluşturarak otomatize edilmesi istenmektedir.

Bu çalışma gerçekleştirilirken bilgisini, sevgisini ve sabrını benden esirgemeyen ve hep yanımda olan sevgili eşim Mehmet Pekin'e, bu süreçte 7/24 yanımda olan minik bebeğim Ada İpek Pekin'e, bugün bu mesleği yapabiliyor olmamı sağlayan kıymetli ailem Reis Seyfi, Pınar Seyfi ve Mustafa Seyfi'ye, her zaman pozitif enerjisiyle yanımda olan dostum Arş. Gör. Beyza Kayhan'a ve yüksek lisans süreci boyunca en ufak detayla bile samimiyeti ve güler yüzü ile ilgilenen kıymetli danışman hocam Dr. Öğr. Üyesi Sait Ali Uymaz'a teşekkürlerimi sunuyorum.

Nisanur Kiraz PEKİN
KONYA-2022

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	viii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	3
3. TEORİK ALTYAPI	20
3.1. Sürü Zekası Optimizasyon Algoritmaları	20
3.1.1. Parçacık Sürü Optimizasyonu (Partical Swarm Optimization - PSO).....	22
3.1.2. Yapay Alg Algoritması (Artificial Algae Algorithm-AAA)	25
3.2. Sürü İHA	27
3.2.1. Sürü İHA Yapısı	28
3.2.2. Sürü İHA Kullanım Alanları.....	29
3.3. Mission Planner	30
3.4. Dronekit-Python.....	31
3.5. MAVLink Protokolü.....	32
3.6. Ardupilot.....	32
3.6.1. Ardupilot SITL	33
4. MATERYAL VE YÖNTEM.....	34
4.1. Ortam Oluşturulması ve Kurulumlar	34
4.2. Fiziki Donanım Planlaması.....	36
4.3. Afet Senaryosu.....	37
4.4. Proje Aşamaları.....	39
4.4.1. Dört Rotorlu İHA Dinamik Modelleme.....	39
4.4.2. İHA'ların bağlanması ve hazırlanması	41
4.4.3. Sezgisel yaklaşım ile görev belirleme	42
4.4.4. İHA'ların simülasyon ortamında uçuşlarının izlenmesi	45
5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	47
5.1. Araştırma Sonuçları	47
5.2. Tartışma	55
6. SONUÇLAR VE ÖNERİLER	57
KAYNAKLAR	58

SİMGELER VE KISALTMALAR

Kısaltmalar

AAA	– Artificial Algae Algorithm
GNSS	– Global Navigation Sattalite System
GPS	– Global Positioning System
İHA	– İnsansız Hava Aracı
PSO	– Partical Swarm Optimization
SITL	– Software-In-The-Loop
UDP	– User Datagram Protocol
XML	– Extensible Markup Language
WSL	– Windows Subsystem Linux

1. GİRİŞ

Afetler; insanların günlük işleyişlerini kesintiye uğratan, can ve mal kaybına sebep olan, bitki ve hayvanlara zarar veren olaylardır. Afetler ile çok sayıda insanlar ölüm, yaralanma ve maddi kayıplara uğramaktadır. Bu kayıpları en aza indirmek için afet yönetimi büyük önem taşımaktadır. (Öztürk ve Tekin, 2017)

Milyonlarca insan, dünyanın farklı bölgelerinde yangın, deprem, sel, tsunami, heyelan, kuraklık, volkanik patlama, kasırga, terör saldırıları, nükleer santral kazaları gibi farklı sebeplerle oluşan afetlerden etkilenmekte ve ciddi zararlar görmektedir. Büyük ölçekli afetlerde öncelikli olarak afetzedelerin yerlerinin belirlenmesi ve tıbbi müdahale yapılması elzemdir.

Afetle mücadele uygulamaları üç farklı faaliyete ayrılmıştır:

1. Afet öncesi faaliyetler
2. Afet sırasındaki faaliyetler
3. Afet sonrası faaliyetler

Afet öncesi faaliyetler için; gözetleme, izleme, risk haritaları oluşturma, lazer mesafe bulucu ve uydu gibi teknolojilerle üç boyutlu haritalar çıkarmak, İnsansız Hava Araçları (İHA) ile afet algılama kontrolü yapmak gibi örnekler verilebilir.

Afet sırasındaki faaliyetler için; afet ile mücadele ekiplerinin görevlerinde başarılı olmaları için İHA gibi işbirlikçi teknolojiler kullanılabilir. Afet sonrası değerlendirme için afet sırasındaki verilerin izlenmesi olayın gidişatı ve değerlendirilmesi için önemli olabilir. Hala devam etmekte olan yangın gibi bir afet sırasında hala yanan korların izlenmesi ve tespit edilmesi sağlanabilir.

Afet sonrasında ise, afetin bıraktığı izler tespit edilerek onarım çalışmalarına başlanabilir. Bu ve diğer tüm verilerle birlikte afetin bilançosu gözler önüne serilerek sonra gerçekleşebilecek afetler için analiz yapılarak önleme çalışmaları yapılabilir.

Arama kurtarma, bir afet ile karşılaşıldığında acil olarak yardıma ihtiyacı olan insanların konumunu tespit etme ve bu insanları yardım alabilecekleri güvenli bir alana ulaştırma faaliyetidir. Ülkemizde arama kurtarma çalışmalarıyla ilgili ilk düzenlemeler 1958 yılında Sivil Savunma Genel Müdürlüğü'nün kurulmasıyla başlamıştır. Marmara ve Düzce depremlerinin ardından 1999 yılında 11 ilde gerekli teknolojik araç ve gereçlerle oluşturulmuş Sivil Savunma Arama ve Kurtarma Birlik Müdürlükleri kurulmuştur. 2009 yılında ise bu müdürlükler İl Afet ve Acil Durum Müdürlükleri bünyesine dahil olmuşlardır (Yılmaz, 2020).

Bir afet olarak orman yangını düşünüldüğünde yangının erken tespiti ve hızlı söndürülmesi yangından kaynaklanan zararları en aza indirebilir. Yangın sırasında yetkililer yangın konumu, yangın boyutu, alev yüksekliği, sıcak noktalar ve konumları gibi parametreleri tahmin etmek zorundadır. Bu tahminlerin bir harita üzerinde incelenmesi yangın davranışını karakterize ederek değerlendirilmeye olanak sağlamaktadır. Bu tahminlerin gerçeğe en yakın ve en hızlı şekilde olması yangının nasıl ilerleyeceği ve nasıl bir yol izlenmesi gerektiği gibi önemli sonuçlara ulaşmayı sağlayacaktır. Böyle bir durumda yangın gibi hızla yayılan bir afet karşısında %100 doğru konum bilmek için uzun süre beklemek yerine en hızlı şekilde en yakın sonucu bulmak daha efektif sonuçlar sağlayacaktır (Shertjuk ve ark., 2018).

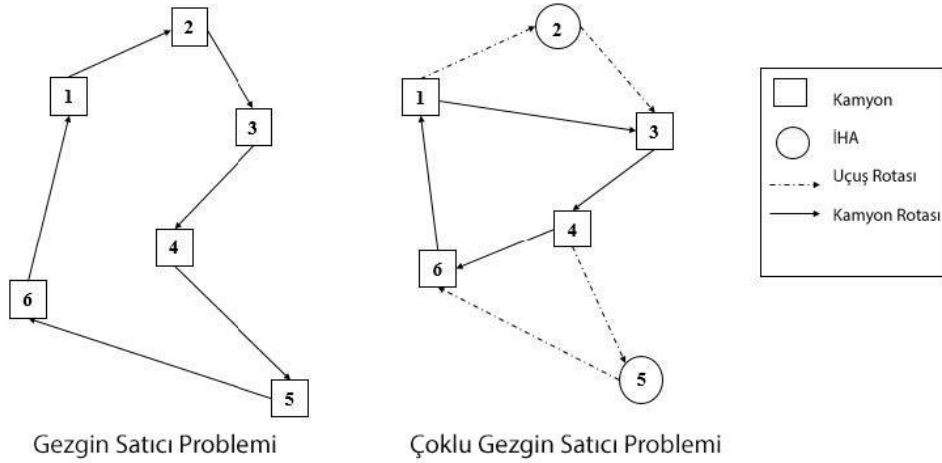
Yangın sıcaklığı yangın başlangıcından beş dakika sonra 500 °C, 90 dakika sonra yaklaşık 1000 °C'ye çıkmaktadır. Tuğlaların eriyerek damladığı bazı yangınlarda sıcaklığın 1700 °C'ye çıktığı görülmüştür (Kılıç, 2010).

Bu çalışmada, yaşanan yangın sırasında ve yangın sonrasındaki çalışmalarda koordinasyonu arttırmak amacıyla sürü şeklinde hareket edebilen insansız hava araçlarının önemi ve arama kurtarma faaliyetlerine faydası üzerinde durulmuştur. Yangınla mücadele büyük öneme sahip olmasına rağmen, doğası gereği zorluklarından dolayı çok fazla araştırma yapılmamıştır. Yangınla mücadele gibi bir afet durumunda İHA kullanımı denildiğinde akla iki konu gelmektedir. Yangını tespit etmek ve yangın söndürme çalışmalarında İHA'ların direkt olarak kullanılması. Bu çalışmada, yangını bulma, izleme ve doğrulama için tespit görevi üzerinde durulmuştur. Bu görev optimizasyon problemi olarak ele alınarak çözülmüştür. Birden fazla İHA sürü zekası optimizasyon algoritmaları kullanılarak yangın bölgesine yakınlaşmışlardır. Burada Parçacık Sürü Optimizasyonu (PSO) ve Yapay Alg Algoritması (AAA) optimizasyon algoritmaları kullanılarak yakınlaşma performansları değerlendirilmiştir. Sonuçları izlemek için İHA uçuşlarında kullanılan yer kontrol istasyonu yazılımı olan Mission Planner isimli yazılım kullanılmıştır. Bu yazılımın kullanılması gerçekleştirilen projenin gerçek ortam testlerinin yapılmasına da zemin hazırlamaktadır.

2. KAYNAK ARAŞTIRMASI

Özoğlu ve arkadaşları (2019), lojistik sektöründe kullanılması amaçlanan birden fazla insansız hava aracı için çoklu gezgin satıcı problemini ele almışlardır. Çoklu gezgin satıcı problemi ile gezgin satıcı problemi arasındaki fark Şekil 2.1'de görülmektedir. Çalışmada kamyon ve İHA'ların koordineli olarak teslimat yapılması üzerine çalışılmıştır. İHA'lar ile teslimat, kamyonlar ile olan geleneksel teslimat yöntemine bir alternatif veya tamamlayıcı olarak görüldüğünden dolayı Amazon gibi taşımacılık yapan firmalar tarafından teşvik edilmektedir. Kullanılan genetik algorithmada ilk nüfus rastgele oluşturulur ve popülasyon boyutu 100 olarak belirlenmiştir. Rulet tekerleği seçme yöntemi ile gen aktarım seçilimi yapılır. Çaprazlama oranı %80, mutasyon oranı ise %30 olarak verilmiştir. İterasyon sayısı olarak belirlenen 100 sayısı kadar tekrar eden bir algoritma ile işlem yapılmıştır. Genetik algoritmanın çözümü Clarke&Wright tasarruf algoritması (Pichpibul ve Kawtummachai, 2013) kullanılarak geliştirilmiştir. Clarke ve Wright tarafından önerilen bu yaklaşımda değerler büyükten küçüğe sıralanır ve tasarruf listesi oluşturulur. Bu listeye göre de rota planları oluşturulur. Bu çalışmada mesafeleri bulmak için genetik algoritma ve Clarke ve Wright tasarruf algoritması birleşiminin sonucu olarak yeni bir sezgisel algoritma geliştirilmiştir. Bu yeni optimizasyonda İHA ile kamyon arasındaki mesafeden birbirlerine yakın oldukları yerler tanımlanır. Kamyon güzergahı, İHA güzergahı ve bağlantıları Çizelge 2.1 ile gösterilir ve minimum paket teslimatı için mesafeler gösterilir.

Gezgin Satıcı Problemi (GSP) ve Çoklu Gezgin Satıcı Problemi (ÇGSP) arasındaki farklar Şekil 2.1'de gösterilmektedir. ÇGSP, kolilerin kamyon ve İHA kullanılarak teslim edilmesinden oluşur. Problem çözümünde İHA'ların kullanılması hareketlilik, esneklik ve düşük maliyet gibi belirgin avantajları vardır. Ancak uçuş süresi ve uçuş operasyonlarındaki çevreye bağlı diğer sınırlamaları ile geniş bir alan üzerinde kullanılmasının zor olması temel bir dezavantaj olarak görülebilir. Bu dezavantajı aşmanın bir yolu olarak, kamyon ve İHA'lardan oluşan hibrit bir dağıtım sistemi düşünülmüştür.



Şekil 2.1. Kamyon ve İHA'lerden oluşan sürü (Özoğlu ve ark., 2019)

Çizelge 2.1'deki çıktıları anlamak için problem 2'yi $n=9$ alınarak incelenir. 0 ile numaralandırılmış düğüm, teslimatların başladığı ve bittiği depoyu gösterir. Kamyon ve İHA'lar beraber kullanılır ve bir ÇGSP olarak değerlendirilir. Teslimat, kamyonla 0 düğümünde başlar ve İHA bu düğümde kamyonun içindedir. Kamyon 4. düğümde müşterinin siparişini teslim ederken, İHA 3. düğümde siparişi teslim eder ve aynı düğüme (4) geri döner. Kamyon, İHA aynı düğüme gelene kadar İHA'yı bekler ve bu da kamyonun bekleme süresinin artmasına neden olur. Bu süreç tüm müşteri siparişleri tatmin olana kadar devam eder.

Çizelge 2.1. Farklı problem setleri için güzergah ve teslimat türü atama (Özoğlu ve ark., 2019)

	Yineleme	Mesafe	Rota	Teslimat Türü
n=9, problem 1	100	376,68	0-2-6-7-4-5-1-8-9-3-0	1, 1, 1, 2, 2, 1, 2, 1, 1
n=9, problem 2	100	374,69	0-3-4-6-8-2-7-1-9-5-0	1, 2, 1, 1, 1, 2, 1, 2, 1
n=9, problem 3	100	435,36	0-5-3-7-4-8-2-1-9-6-0	1, 1, 2, 1, 2, 1, 2, 1, 1
n=9, problem 4	100	465,92	0-6-1-2-5-8-4-7-3-9-0	1, 2, 2, 1, 1, 1, 2, 1, 2
n=9, problem 5	100	381,56	0-1-6-4-8-2-7-3-5-9-0	1, 1, 1, 2, 1, 1, 1, 1, 1
n=9, problem 6	100	369,56	0-9-6-5-2-4-3-8-1-7-0	1, 2, 1, 2, 1, 1, 1, 2, 2
n=9, problem 7	100	333,65	0-1-5-8-4-7-3-2-6-9-0	1, 2, 2, 1, 2, 1, 2, 1, 1
n=9, problem 8	100	317,67	0-1-5-3-8-9-2-4-6-7-0	1, 2, 1, 1, 2, 1, 1, 1, 1
n=9, problem 9	100	461,24	0-9-3-4-5-7-2-8-1-6-0	1, 1, 2, 1, 2, 1, 1, 2, 2
n=9, problem 10	100	395,70	0-4-9-1-7-6-5-8-2-3-0	1, 2, 1, 2, 1, 1, 1, 1, 1

Yoon (2018), taşıt yönlendirme problemi için İHA çözümleri üzerine durmuştur. Bir kamyon ve birden fazla İHA ile hizmet vermenin maliyetini Karışık Tamsayılı

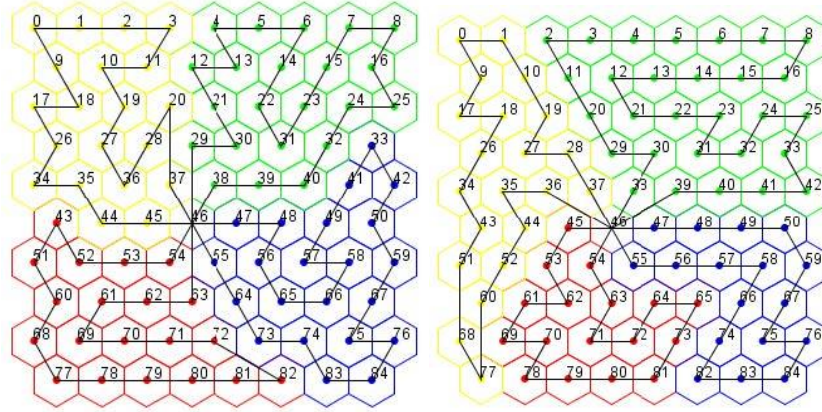
Doğrusal Programlama (Mixed Integer Linear Programming - MILP) optimizasyonu ile analiz etmiş ve modellenmiştir. Problemde her turda bir kamyon ve birden fazla İHA'nın çalışmasına izin verildiği varsayılır. Engelden kaçınma gibi durumlar için yeni rota oluşturma işlemlerinde Manhattan mesafesi kullanılarak uzaklıklar hesaplanmıştır. Amaç fonksiyonu toplam maliyeti en aza indirmeye yönelik çalışmaktadır. Toplam maliyet ise; kamyonun maliyeti, İHA'ların maliyetleri ve sabit maliyetlerin toplamından oluşur. Kamyon kapasitesi ile İHA sayısı arasındaki denge durumu, her İHA'nın birden fazla paket taşınması durumu ve aynı anda birden fazla müşteriye hizmet vermesi durumu için fizibilite çalışmasının önemi anlaşılmıştır. Kamyon ve İHA'lardan oluşan bu karma sistemin maliyetleri düşüreceği gözlemlenirken bu alandaki çalışmaların hızla artması beklenmektedir.

Chen ve arkadaşları (2017), popülerleşen İHA kullanımı için önemli sorunlardan biri olan İHA rotalarının seçilmesi davranışları için bir çalışma yapmışlardır. Batarya verimliliği, yük miktarı, zorlu hava koşulları gibi etkenler rota belirlemede dikkate alınmalıdır. Verimliliğin sağlanması için bazı kısıtlamalar altında rota optimizasyon problemleri matematiksel olarak ifade edilmektedir. İHA rotalarının belirlenmesi, Araç Yönlendirme Sorunu (Vehicle Routing Problem- VRP) ve Gezgin Satıcı Problemi (Traveling Salesman Problem-TSP) olarak iki farklı sorun olarak ele alınmış ve ikisi için de detaylı incelemeler yapılmıştır. Gezgin satıcı problemi ile İHA yol planlaması için genetik algoritma ve parçacık sürü optimizasyonu (PSO) tabanlı karınca kolonisi optimizasyonu algoritması paralel olarak kullanılmıştır. Yapılan deneyler sonucunda genetik algoritma ile gerçekleştirilen çözümün daha makul sonuçlar elde ettiği anlaşılmıştır. VRP, gezgin satıcı problemini genelleştiren bir optimizasyon ve tamsayı programlama problemidir. Bu problem Tam Sayı Doğrusal Program (Integer Linear Programming -ILP) olarak formüle edilip karınca kolonisi optimizasyonu ile meta-sezgisel bir yaklaşım ile çözüm sunmuştur. Detaylı olarak incelenen iki yöntem için İHA yol planlaması dinamik bir harita ile gösterilmiştir.

Kitjacharoenchai ve arkadaşları (2019), teslimat kamyonlarından uçurulan otonom İHA'lar ile hem kamyon hem de İHA'ların depoya varış sürelerini en aza indirmeyi amaçlayan bir problem üzerine çalışmışlardır. Bu senaryoya göre Mixed Integer Programming (MIP) ile sezgisel arama tabanlı yeni bir algoritma geliştirilmiş ve tek kamyon çok İHA sisteminde potansiyel kazanç gösterilmiştir. Bu operasyonların İHA'lara uygulanması Çoklu Gezgin Satıcı Problemi (ÇGSP) örneği olarak ele alınmıştır. Problem, hangi tur seçildiğine bakılmaksızın tüm satıcılar verilen görevi

yerine getirip başlangıç düğümüne geri gelmeli ve başlangıç düğümü dışında her düğüm satıcı tarafından bir kez ziyaret edilmelidir. ÇGSP’de amaç her satıcının belirlenen şehirleri gezerek başladığı yere gelmesi için gereken toplam en kısa yolu bulmaktır. Yani toplam teslimat tamamlama süresinin en aza indirilmesi amaçlanır. Çözüm için bir Uyarlamalı Ekleme Algoritması (Adaptive Insertion algorithm -ADI) önerilmiştir. Bu algoritma için sezgisel üç yöntem incelenmiştir. Bunlar Combined K-means, Nearest Neighbor and Random Cluster yöntemleridir. Daha sonra ise Genetik Algoritma kullanılarak hesaplama sürelerinde daha iyi çözümler elde edilmiştir. Birden fazla İHA kullanılarak gerçekleştirilmesi planlanan teslimat senaryosu için rota planlaması sonuçları oldukça iyileştirilmiştir. Optimum çözüme ise Genetik Algoritma/ ADI çözüm yöntemi ile ulaşılmıştır.

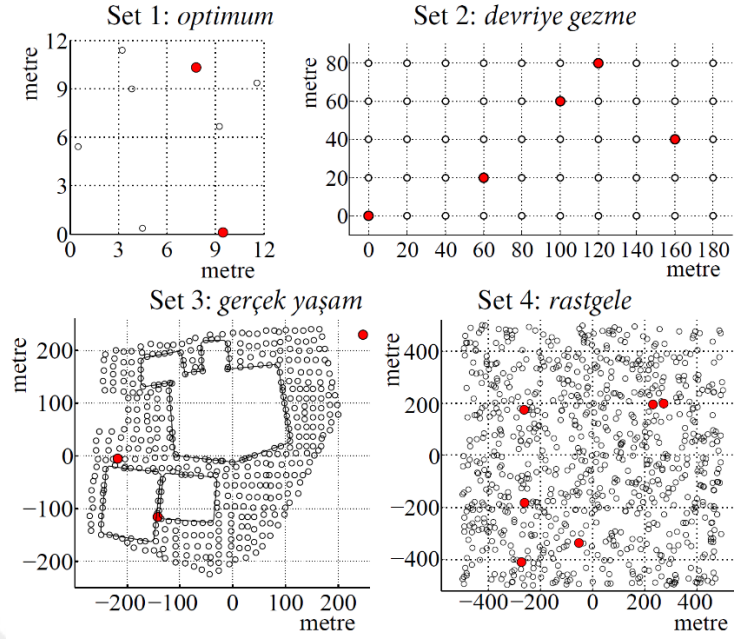
Khoufi ve arkadaşları (2016), birden fazla İHA ile alan izleme problemi üzerine çalışmışlardır. İzlenecek olan alanı hücrelere ayırmışlardır. Bir hücreye bir İHA sığacak şekilde İHA’lar yerleştirilir. Öncelikle her İHA için ziyaret edilmesi gereken hücreler atanır. Daha sonra ise İHA’nın ziyaret etmek için gittiği hücrelerden oluşan yollar optimize edilir. Planlanan senaryoya göre 4 adet İHA ve 200m*200m’lik bir alan belirlenmiştir. İHA’ların gidecekleri hücreleri seçmeleri için 2 farklı durumda hücre bulunur. Bunlar henüz gidilmemiş serbest hücreler ve bir İHA tarafından seçilmiş yani gidilmiş hücrelerdir. İHA yalnızca serbest hücreler arasından, daha önce gidilmiş hücrelerine bitişik bir hücre seçebilirler. Bu hücrelerin seçimi işleminde uzaklık hesabı yapılmaktadır. Şekil 2.2’de, küme tarafından elde edilen sonuçlar gösterilmektedir. 300m * 300m bir alan için sırasıyla oluşum ve koalisyon yöntemini göstermektedir. İHA’lar başlangıçta 54, 55, 37 ve 38 numaralı hücreleri işgal eder. Her iki şekilde de her bir İHA sadece bitişik hücreleri ziyaret eder ve yollar arasında herhangi bir örtüşme yoktur. Her İHA tarafından ziyaret edilen hücre sayısı ile ilgili olarak, küme oluşumu ile daha iyi bir sonuç sağlanır. İHA’lar tarafından ziyaret edilen hücreler için oluşturulan kümeler üzerinde işlemlerin yapıldığı bir algoritma çözümü sunmuşlardır. İkinci bir çözüm olarak da seçilen rotaların optimize edilmesi sunulmuştur. Her iki çözüm de uygulanarak sonuçları karşılaştırılmış ve ikisinde de başarılı sonuçlar elde edilmiştir. Bu iki çözüm için de temel amaç her İHA için pil ömrünü en verimli şekilde kullanmak için gezilen hücre sayısının en optimum düzeye çekilerek verilen hedeflere ulaşılmasıdır.



Şekil 2.2. a) Küme oluşturma yöntemiyle çözüm b) Koalisyon kurma yöntemiyle çözüm (Khoufi ve ark., 2016)

Mersheeva ve Friedrich (2015) sınırlı enerji kaynakları gibi bazı kısıtlamalarla birden fazla İHA'nın izlenme sorunu üzerine çalışmışlardır. Bu izlenme sorununu suç mahallerinin incelenmesi ve afet bölgelerinin izlenmesi olarak özelleştirmişlerdir. Bir felaket senaryosunda izleme sorunu için çözümler incelendiğinde yer robotlarının aksine mikro İHA'lar ortalama olarak daha kısa sürede verilen görevleri tamamlarlar. Ayrıca deprem, yangın gibi afetlerden sonra durum hakkında yorum yapabilmek adına havadan gözetleme gerekmesi de İHA'ları üstün kılmaktadır. Birden çok mikro İHA'nın eş zamanlı olarak aldığı görüntüler birleştirilerek istasyonda işlenir ve durum kısa sürede izlenebilir. Alan tam ağırlıklı bir graf olarak Denklem 2.1'deki gibi tanımlanmıştır. N düğüm ve E kenar anlamına gelir. Kenar ağırlıkları hesaplanıp güncellenerek planlama ve rota belirleme yapılmaktadır. Depolar Arası Yerleştirme Sezgisel Yöntemi (inter-depot insertion heuristic -IDIH) komşuluk arama yaklaşımına dayalı bir yöntem olarak öneriliyor. Bu yöntem ile dinamik olarak katsayıları değiştirilerek ayarlama yapılmaktadır. Dinamik olarak çalışabilmesi görev sırasında yeniden planlama yapılmasına olanak sağlamaktadır. Yöntemin kapsamlı olarak değerlendirilmesi için dört örnek seti oluşturulmuştur. Buna göre 10 küçük optimum örnek, 12 devriye gezme, 48 yaşam ve 60 rastgele örnek setleri Şekil 2.3'de gösterilmiştir.

$$G = (N, E) \quad (2.1)$$



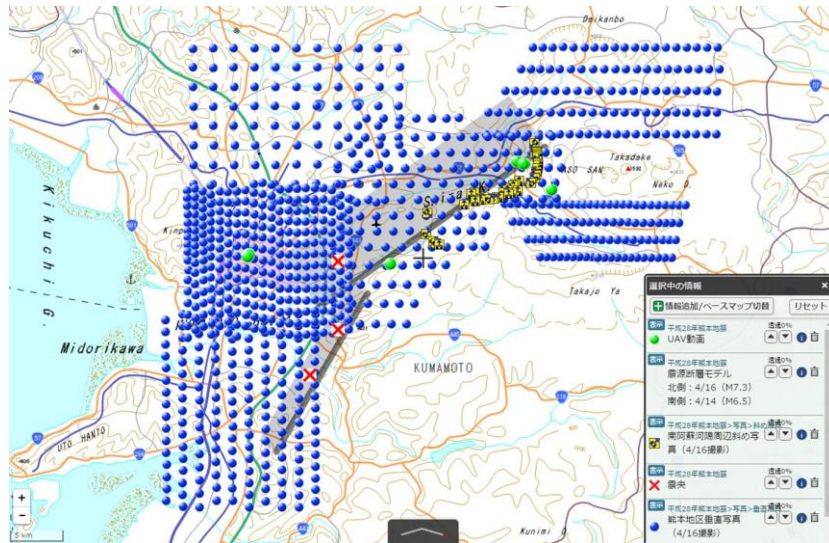
Şekil 2.3. Dört test seti için örnek durumlar (Mersheeva ve Friedrich, 2015)

Yakıcı (2016), insanlı uçuşlar için zor ve güvensiz durumlarda kullanılması amacıyla kullanılacak olan İHA'lar üzerinde çalışmalar yapmıştır. Bu tür görevlerde İHA'ların yönlendirmelerinin optimize edilmesi gerekmektedir. Sorunu, Konum ve Yönlendirme Problemi (Location and Routing Problem -LRP) olarak ele almıştır. Nispeten büyük boyutlu örneklere kısa sürede daha iyi çözümler sunmak amacıyla karınca kolonisi tabanlı yeni bir meta sezgisel yaklaşım oluşturmuştur. İHA sürüsünün aynı hızlara sahip benzer araçlardan oluşan homojen bir sürü olduğu varsayılmaktadır. Her İHA bir karınca olarak kabul edilir ve yollar karıncalar tarafından inşa edilir. Bırakılan izlerin güncellemesi ile algoritma karakterize edilir. Birden fazla feromon izi kullanılır ve bu izler ayırt edilerek davranışlar şekillenir. Sınırlı uçuş süresi olduğu kabul edilen İHA'lar ile sürünün ziyaret ettiği noktaların toplam önemi en üst düzeye çıkılması amaçlanmaktadır. Yani ziyaret edilen yol noktalarından toplanan toplam puan maksimize edilmektedir.

Nedjati ve ekibi (2016) tarafından yapılan çalışmada deprem sonrası müdahalede karayolu ulaşımındaki zorluklar sebebiyle alternatif seçeneklerin oluşturulması gerekmektedir. Yolların tıkanması durumunda zaman kısıtları nedeniyle gerekli ulaşımın sağlanmaması hayati önem taşımaktadır. Bu sorunun bir çözümü olarak mikro İHA helikopteri sunulmuştur. Tıkanmış olan kara ulaşım sistemine etki etmeden havadan hizmet sunması da ayrıca bir avantaj olarak görülmüştür. Deprem sonrası müdahalede deprem alanından görüntüleri toplamak ve harita oluşturmak için birden

fazla İHA helikopteri ile ızgara desen eşlemi ayrışmasına dayanan bir çözüm sunulmuştur. Uygulama, Tahran'da bir vaka çalışması olarak incelenmiştir. Önerilen helikopter çözümü ile deprem sonrası kayıplar ve yaralanmalar için ilk yardım ürünleri ve talep edilen malzemeler ulaşılması zor alanlara ulaştırılır. Yaralıların seviyesi ve farklı yerlerdeki yaralıların sayıları gibi bilgiler ile bölgeler sınıflandırılır. Böylece öncelik düzeyleri belirlenir. Yardım malzemesi taşıyan İHA'lar olduğu gibi keşif amaçlı görevleri olan İHA'lar da bulunmaktadır.

Yamazaki ve Liu (2016) Japonya Kumamoto depreminden sonra elde edilen uzaktan algılama verilerini tanıtmışlardır. Büyük çaplı gerçekleşen doğal afetler göz önüne alındığında uzaktan algılama sistemleri oldukça önemlidir. Optik uydu sistemleri gündüz çalışır ve bulut örtüsü karşısında istenilen sonuçları vermez. Sentetik Açıklıklı Radar (Synthetic Aperture Radar-SAR) olarak adlandırılan radar sistemi tüm hava koşullarında kullanılabilir. Gelişen teknoloji ile dijital hava kameraları, lazer mesafe bulucular, İHA'lar gibi hava araştırma araçları ortaya çıkmıştır. Kumamoto kalesinin İHA ile video çekimi sonucunda oluşturulan 3 boyutlu (3D) modeli Agisoft'un PhotoScan yazılımı ile gerçekleştirilmiştir. İHA, optik uydular, lazer mesafe bulucular ve SAR verileri çeşitli uluslararası uzay ajansları ve hava araştırma şirketleri tarafından alınmış ve Küresel Konum Belirleme Sistemi (Global Navigation Sattalite System - GNSS) gözlemi verileri ve sismik kayıtlarla karşılaştırılmıştır. Şekil 2.4'de Japonya Jeo-uzamsal Bilgi Kurumu (GSI) tarafından 16 Nisan 2016'da (ana şokun yaşandığı gün) gerçekleştirilen dikey havadan fotoğraf çekim noktalarını (mavi noktalar) göstermektedir.



Şekil 2.4. GSI tarafından gerçekleştirilen dikey havadan fotoğraf çekim noktalarını (mavi noktalar), eğik fotoğraf çekim noktalarını (sarı semboller) ve İHA uçuşlarını (yeşil noktalar) gösteren harita (Yamazaki ve Liu, 2016)

Ashraf ve arkadaşları (2020) tehlikesiz güvenli bir İHA uçuşu sağlanması için çözüm yolları aramışlardır. İHA'ların artan popülaritesi ile İHA kazaları da doğrusal olarak artması sorunu üzerine çalışmışlardır. İHA sürüleri için İHA'ların birbirleriyle çarpışmadan güvenli bir şekilde pilot kontrolünde uçuşması kaza riskini oldukça arttırmaktadır. Çarpışmaları öngörerek, sürü güvenliği ve çarpışmadan kaçınmak için en iyi yolların bulunmasına yönelik bir çalışmadır. İHA çarpışmalarının yüksek risk içerdiği senaryoları için bir çözümde bulunmuşlardır. Önerilen sistemde uçuş öncesi tüm engellerin bilinmesi ve bu bilgilere göre uçuş planı hazırlanmasına gerek duyulmaz. Sistem engeller hakkında önceden bilgiye sahip değildir. Uçuş sırasında karşılaşılan sabit veya hareketli engellere karşı önceden tahminde bulunup kaçınma hareketi yapmasını sağlar. Dis, güvenli mesafe, Sp hız, Cl iletişim gecikmesi, Pt işlem süresi olmak üzere güvenli mesafe hesabı için formül Denklem 2.1'deki kullanılmaktadır.

$$Diss = 2 \cdot Sp (2 \cdot Cl + Pt) \quad (2.1)$$

Örneğin; her biri saniyede maksimum 5 metre hızla ve 0,5 saniyelik bir engel algılama ve işleme süresi ve 0,2 saniyelik kablosuz iletişim gecikmesi ile birbirine doğru hareket eden iki İHA bulunursa, güvenli mesafe değerler yerine yazıldığında Denklem 2.2'deki gibi 9 olarak bulunur.

$$2.5(2 \cdot 0,2 + 0,5) = 9 \quad (2.2)$$

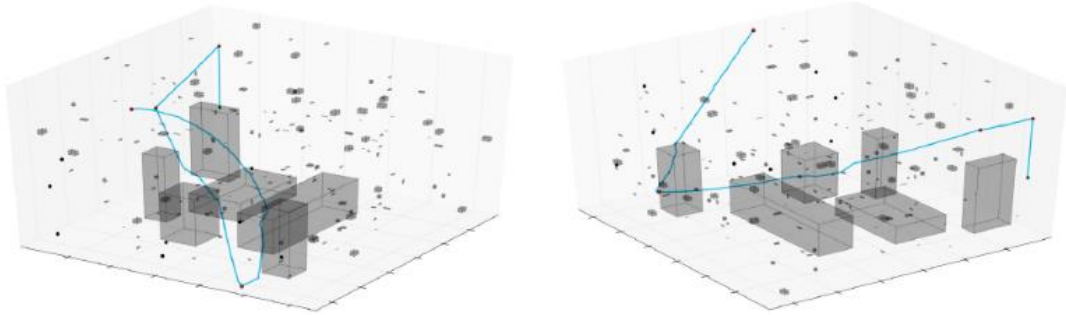
Goel ve arkadaşları (2018) insansız hava araçlarının üç boyutlu bir ortamda dinamik engellere çarpmadan yol bulmaları için bir çözüm geliştirmişlerdir. Verilen görev sırasında değişken ve rastlantısal olarak çevreden gelebilecek engellere karşı İHA'nın çarpışmadan manevra yapması amaçlanır. Bu doğrultuda optimum bir uçuş rotası belirlenmelidir. Bu çalışmada rota planlaması için Ateş Böceği Sürü Optimizasyonu (Glow-worm Swarm Optimization - GSO) kullanılmıştır. Python kullanılarak simülasyon üzerinde denemeleri yapılmış ve başarılı olunmuştur. Burada amaçlanan sürü zekası şekli doğada yaygın olarak görünen merkezi olmayan parçacıkların kendiliğinden organize olarak hareket etmesidir. Çevreyi keşfetmek için etrafında bulunan komşu parçacıklarla işbirliği yaparlar. Keşif davranışı gösterirken önemli noktalardan biri de komşu parçacıklara ulaşabilecekleri iletişim kanallarının olmasıdır. Arama işlemi yapılırken kendi yön bilgileri ile komşulardan gelen veriler kıyaslanır ve en iyi değer ile arama yönü devamlı güncellenerek sürü davranışı şekillenmiş olur. GSO, ateş böceklerinin yiyecek arama davranışlarından esinlenilerek geliştirilmiş bir optimizasyon algoritmasıdır. Bu böcekler lüser isimli kimyasal bir madde kullanarak ışık yayabilirler. Bu ışığın miktarı kontrol edilebilir ve ayarlanabilen

bu miktarlar bazı anlamlara gelir. Bir ateş böceği bulunduğu popülasyon içinde lusiferin yoğunluğu daha yüksek yani daha paralak olan ateş böceğine doğru yönelecektir. Öncelikle ortama ateş böcekleri rastgele dağılırlar. Her böcek için başlangıç lusiferin değeri I_0 ve aralığı r_0 'dır. Lusiferin güncelenmesi için Denklem 2.3 formülü kullanılır. Burada i , i . Ateşböceği anlamına gelmektedir. ρ , lusiferinin bozunma sabiti ve t zamanı ifade etmektedir. Daha sonra ateşböceği lusiferini yüksek olan komşulara doğru harekete geçecektir. Ateş böceği hareketine Denklem 2.4'de gösterilen olasılık denkleminde göre karar verir. Ateş böceğinin seçilen komşusu yönünde gittiği yol güncelleme denklemi ise Denklem 2.5'de verildiği gibidir. Burada st adım boyunu gösterir. $\|$ ise Öklid uzaklığı operatörü olarak gösterilmiştir. Yol planlama sorunu için arama alanının ızgara tabanlı bir yapıya bölündüğü düşünülebilir. Şekil 2.5'de Gri küboidler statik engellerdir. Siyah noktalar dinamik engellerdir, yeşil nokta kaynaktır ve kırmızı noktalar hedeflerdir.

$$li(t) = (1 - \rho)li(t - 1) + \gamma J(t) \quad (2.3)$$

$$pi_j(t) = (lj(t) - li(t)) / (\sum_{k \in N_t} (lk(t) - li(t))) \quad (2.4)$$

$$xi(t + 1) = xi(t) + st * ((xj(t) - xi(t)) / \|xj(t) - xi(t)\|) \quad (2.5)$$

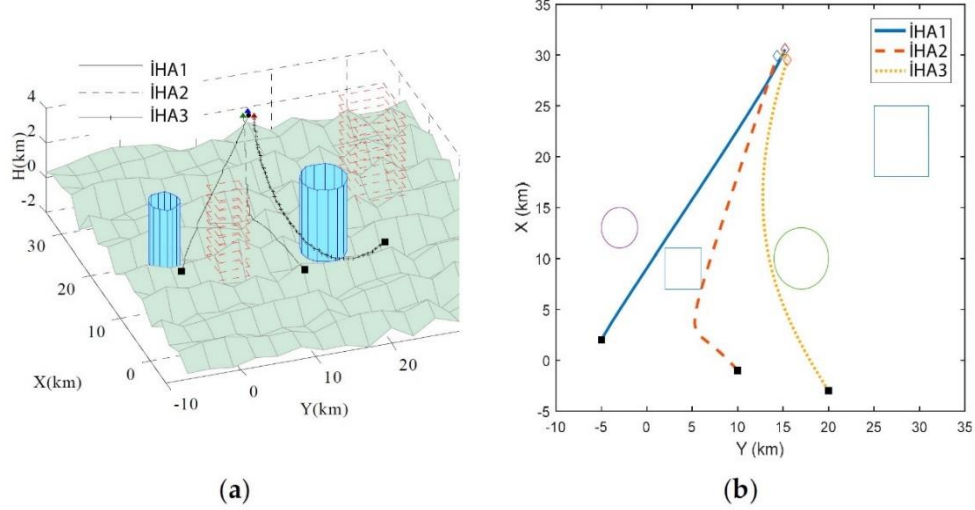


Şekil 2.5. Dinamik engeller ve hareket halindeki çoklu hedefler ile dinamik yol planlaması (Goel ve ark.,2018)

Hoang ve arkadaşları (2018), birden fazla insansız hava aracı için engellerden kaçarak yol planlama yapmışlardır. İHA'ların konumu için kullanılan açısız hız ve konum ile sürünün yakınsamasını hızlandırmak için Açık kodlamalı Parçacık Sürüsü Optimizasyonu (Angle-encoded Particle Swarm Optimization - θ -PSO) algoritması önerilmiştir. İncelenen yüzeyin uydu haritasından çıkarılan 3D modeli ile çevresel özelliklerin çıkarımı ile başlar. Daha sonra referans ara noktaları belirlenir. Birden fazla İHA'nın yol planlaması optimize edilerek yapılmaktadır. Bu yollar dinamik olarak

değişen çevre şartlarına göre güvenli bir uçuş sağlamalıdır. PSO için uygun maliyet fonksiyonu, toplam hareket mesafesini en aza indirmeye çalışırken, İHA'ların birbirleriyle ve engellerle çarpışmadan kaçınmasını da sağlamalıdır. Mission Planner uygulaması kullanılarak test ortamı Monorail köprüsü üzerinde oluşturulmuş ve görev planlayıcıya yüklenmiştir. Planlayıcıdaki haritaya dayanarak engeller belirlenir. Gerçekleştirilen saha testleri sonucunda İHA'ların köprü üzerine birbirlerine çarpmadan hareket edebilmeleri ve algoritmanın çalışması uygulanabilirliğini doğrulamıştır.

Shao ve arkadaşları (2019), çoklu İHA oluşumunun yol planlaması sorununu karmaşık birçok amaçlı optimizasyon problemi olarak ele almışlardır. Yol planı için evrim teorisinden esinlenilerek Dağıtılmış Kooperatif Parçacık Sürü Optimizasyonu (Distributed Cooperative Particle Swarm Optimization- DCPSO) algoritması önerilmiştir. Bu algoritma ile İHA'lar arasındaki iş birliği ihtiyacı karşılanırken optimal ve alt optimal yollar elde edilir. Aynı zamanda İHA'lar arasında çarpışmadan kaçınma da bu yöntemle sağlanır. Pisagor Hodografisi (Pythagorean Hodograph- PH) eğrisi teorisine göre bir yol, başlangıç ve son konumlarda yön vektörlerinin uzunluklarına göre belirlenir. İHA popülasyonunun PH yol çözümleri ile optimize etmek için DCPSO algoritması kullanılmıştır. Oluşturulan alt kümeler ve parçacıkların uygunluk değerlerinin hesaplanıp uygunluklarına göre yer değiştirmeleri temeline dayanmaktadır. DCPSO, Kooperatif Birlikte Evrimsel Genetik Algoritma (Cooperative Co-Evolutionary Genetic Algorithm- CCGA) ile karşılaştırılmıştır. Karşılaştırma analizinin daha iyi yapılması için her iki algoritma için aynı ortamda 30 deney yapılmıştır. DCPSO'nun başarı oranı 0,9 CCGA ise 0,433 olarak gözlemlenmiştir. DCPSO ile daha iyi bir kararlılık gösterdiği anlaşılmaktadır. Algoritma çalışmasını göstermek için 2 boyutlu ve 3 boyutlu ortamlarda simüle edilmiştir. Önerilen DCPSO tarafından 3 boyutlu ortamlarda iş birliği ile oluşturulan İHA'ların yolları Şekil 2.6'da gösterilmektedir.



Şekil 2.6. 3D ortamda İHA'ların buluşma noktasını oluşturma yolu (a) 3D görünüm (b) üstten görünüm (Shao ve ark.,2019)

Zhang ve Ali (2020), bilinmeyen karmaşık ortamlarda sürü İHA'lar tarafından tek bir hedef arama görevini simüle etmek için bir Robot Fasulye Optimizasyon Algoritması (Robot Bean Optimization Algorithm- RBOA) önermiş ve uygulamışlardır. Sürü optimizasyonuna dayalı algoritmalar genellikle hayvan hareketleri esas alınarak geliştirilmişlerdir. Burada incelenen algoritma bitki sürüsü gibi yeni bir alanda geliştirilmiştir. Bitki biyoniklerine dayalı sürü algoritmalarının robotik sistemler üzerindeki araştırmalar şu anda nadirdir. Bitkiler üzerine yapılan ekolojik araştırmalar ile bir bitki popülasyonunun milyonlarca yıllık evrimi bu alanda da kullanılmaya değer sürü zekası içerdiğini göstermiş ve daha fazla incelenmeye başlamıştır. Fasulye Optimizasyon Algoritması (Bean Optimization Algorithm BOA), tohumların doğal yayılma modundan ve nüfus dağılımlarının evriminden esinlenmiştir.

F(x) Algorithm		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
RBOA-30	Average	1.18×10^{-1}	3.93×10^{-2}	1.12×10^{-1}	2.66×10^{-1}	1.92	1.12×10^{-2}	4.37×10^{-3}	-7.55×10^2	9.7×10^{-1}	6.3×10^{-2}
	Stdev	1.24×10^{-1}	3.06×10^{-2}	2.32×10^{-1}	1.45×10^{-1}	3.63	6.78×10^{-3}	4.15×10^{-3}	9.4×10^1	1.02	4.6×10^{-2}
	Min	9.99×10^{-5}	6.55×10^{-4}	1.75×10^{-3}	4×10^{-2}	8.04×10^{-4}	1.36×10^{-4}	5.68×10^{-5}	-8.37×10^2	1.38×10^{-3}	4.48×10^{-3}
	Max	5.45×10^{-1}	1.64×10^{-1}	1.59	6.06×10^{-1}	15.32	3.72×10^{-2}	1.76×10^{-2}	-5.41×10^2	4.04	1.85×10^{-1}
ARPSO-30	Average	2.15×10^{-1}	9.3×10^{-2}	2.7×10^{-1}	4.7×10^{-1}	3.64	2.09×10^{-2}	1.11×10^{-2}	-7.10×10^2	1.19	1.16×10^{-1}
	Stdev	2.37×10^{-1}	5.05×10^{-2}	3.17×10^{-1}	2.8×10^{-1}	6.41	1.4×10^{-2}	8.9×10^{-3}	7.4×10^1	1.01	6.68×10^{-2}
	Min	3.9×10^{-3}	1.12×10^{-2}	4.66×10^{-3}	4.6×10^{-2}	2.23×10^{-3}	9.7×10^{-3}	2.7×10^{-4}	-8.24×10^2	7.2×10^{-3}	2.1×10^{-2}
	Max	1.48	2.18×10^{-1}	1.28	1.11	31.58	7.5×10^{-2}	4.05×10^{-2}	-5.39×10^2	4.35	2.7×10^{-1}

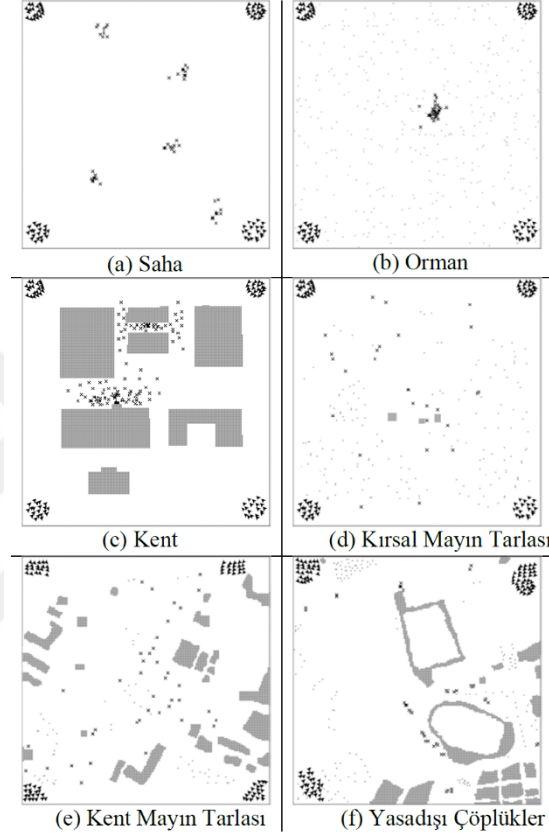
Şekil 2.7. 30 İHA için simülasyon deneyi sonuçları (Zhang ve Ali, 2020)

Önerilen algoritmanın performansını doğrulamak için Matlab'da birkaç simülasyon deneyi yapılmıştır. Deneysel sonuçlar, RBOA'nın performansının Uyarlanabilir Robotik Parçacık Sürüsü Optimizasyonu (Adaptive Robotic Particle Swarm Optimization-A-RPSO)'dan daha iyi olduğunu göstermiştir. İki algoritmanın 30 İHA için kıyaslamaları Şekil 2.7'de verilmiştir. Önemi ise karmaşık bir ortamda diğer yaklaşımlara kıyasla İHA sayısı ve nesil sayısı az olduğunda daha avantajlı olduğu anlaşılmıştır.

Wang ve arkadaşları (2020), Nesnelerin İnterneti (Internet of Things – IoT) uygulamalarında kullanmak üzere veri toplama platformu olarak İHA'ların kullanılması prensibine dayanarak bir çalışma gerçekleştirmişlerdir. İHA'ların veri toplarken konuşlandırılmalarını ise optimize etmek istemişlerdir. Temel amaç, İHA'lardan kaybedilen enerjisinin toplamını ve IoT uygulamasına gönderilecek veri iletim kaybını en aza indirmek. Burada her birey bir çözüm olarak kabul edilmiştir. Çiçek Tozlaşma Algoritması (Flower Pollination Algorithm - FPA), Salp Sürü Algoritması (Salp Swarm Algorithm – SSA), Sinüs Kosinüs Algoritması (Sine Cosine Algorithm - SCA) sürü zekası algoritmaları kullanılarak simülasyon ortamında testleri gerçekleştirilmiştir. Algoritmalar karşılaştırıldığında SCA'nın genel performansı en iyi olarak gözlemlenmiştir. Ancak bu çalışmada İHA'nın uçuş yüksekliği bilgisi kullanılmamıştır.

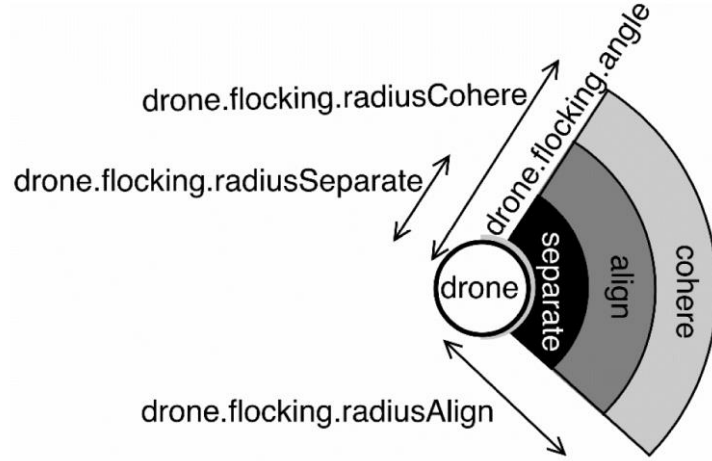
Cimino ve arkadaşları (2016), Diferansiyel Gelişim (Differential Evolution - DE) Algoritması aracılığıyla, sürüleme temelinde hedef tespiti gerçekleştiren bir İHA sürüsünün koordinasyonunu uyarlamak için bir algoritma çalışması yapmışlardır. Diferansiyel Evrim algoritması (DE), sayısal ve çok modlu optimizasyon problemleri için çok uygun olduğu düşüncesiyle bu algoritma üzerine çalışılmıştır. Bu yaklaşımda, yapısal olmayan ortamların seyrüseferinde kendi kendine koordinasyonu sağlamak için sürü mantığı kullanılmıştır. Yapısal olmayan ortamlardan kastedilen ağaç, bina gibi engellerden oluşan farklı ortamlardır. Sayısal optimizasyon çözümü Matlab üzerinde uygulanmış, simülasyon ortamı olarak ise ayrıca NetLogo kullanılmıştır. Toplam 80 İHA kullanılmıştır. Sürü algoritması altı farklı senaryoda test edilmiştir. Yaklaşımları karşılaştırmak için, dikkate alınan her yaklaşım için 10 deneme gerçekleştirilmiştir. Ayrıca ortaya çıkan performans göstergesi örneklerinin grafiksel normallik testi kullanılarak normal dağılımla iyi modellendiği gözlemlenmiştir. Sonuç olarak, yaklaşım tüm senaryolarda hakim olduğu gözlemlenmiştir. Şekil 2.8 (a) hiçbir şekilde engel içermez ve her biri 10 hedeften oluşan 5 küme içerir. Şekil 2.8 (b), ağaç engelleri olan 20 hedeflik 1 kümeden oluşan bir keresteyi temsil etmektedir. Şekil 2.8 (c), her biri bir

kirletici kaybını temsil eden 55 hedeflik 2 kümeden oluşan bir sanayi bölgesini göstermektedir. Sırasıyla Şekil 2.8 (d) ve Şekil 2.8 (e)'de gösterilen kırsal mayın tarlası ve kentsel mayın tarlası senaryoları, Bosna-Hersek'te Saraybosna yakınlarındaki gerçek dünya haritalarından türetilen 80.000 m²'lik iki mayınlı alandır. Son senaryo, İtalya'nın Paternò kasabası yakınlarındaki 80.000 m²'lik bir alandan alınan gerçek dünya yasa dışı çöp boşaltma örneğidir.



Şekil 2.8. 3 sentetik ve 3 gerçek dünya senaryosunun haritası (Cimino ve ark.,2016)

Cimino ve arkadaşları (2019), dağıtılmış hedef tespiti ve takibinde yer alan İHA sürüsünün koordinasyonu için biyo-esinlenmiş bir yaklaşım üzerine çalışmışlardır. Yapılandırılmamış ortamlarda statik veya dinamik hedefleri keşfetme ve izleme problemini, konumları ve engellerin yerleşimi hakkında önceden bilgi olmadan ele almışlar ve bu bağlamda üç farklı senaryoya göre işlemişlerdir. Sürecin kalitesi, verilen hedefleri keşfetmek için gereken süreyi en aza indirerek veya verilen sürede keşfedilen hedeflerin sayısını en üst düzeye çıkarmak olarak tanımlanmıştır. Sürü davranışını ortama uyarlamak için, sürecin kalitesi Diferansiyel Evrim (DE) algoritması kullanılarak optimize edilmiştir. Sayısal optimizasyon çözümü Matlab üzerinde uygulanmış, simülasyon ortamı olarak ise ayrıca NetLogo kullanılmıştır.



Şekil 2.9. Hizalama kuralı (Cimino ve ark., 2019)

Şekil 2.9’da gösterilen kural 1987’de Reynolds’ flocking rules ismi ile geçen hizalama kuralıdır. Hizalama İHA’ları yakındaki ajanların ortalama yönüne göre hizalar. Seperate, sürü eşlerin arasında minimum mesafeyi koruyarak büyük bir oluşumu korur. Cohere, her ajanı sürü eşlerinin merkezine doğru yönlendirir. Oluşan bu küresel görüş açısı ve farklı yarıçap aralıkları, üç etki alanının karakterize olmasını sağlar.

Shao ve arkadaşları (2020), çoklu İHA sistemlerinde engellerden kaçınma, hedef arama, hedefe hücum etme gibi görevleri yerine getirmek için sürü zekası algoritmalarının kullanılmasını düşünmüşlerdir. Her bir İHA tarafından elde edilen bilgi kaynaştırılmalıdır. Paylaşılan ortamda farklı anlamlara sahip feromonlar oluşturulur ve bu feromonlar entegre edilerek harita üzerinde güncellemeler yapılır. Buradaki sürü sisteminde dijital feromonları algılayarak bölgeye ulaştırmada diğer İHA’lara liderlik eden bir lider İHA kullanılmıştır. Bilinmeyen bir ortamda birden fazla bileşenin bulunduğu duruma bağlı olarak hedefleri, engelleri ve İHA rotalarını işaretlemek için üç tür anlamsal dijital feromon kullanılmıştır. Yol feromonlarının amaç, yakın zamanda geçilen yolu işaretleyerek tekrarlanan ziyareti önlemektir. Her bir İHA, yerel feromon haritasını elde etmek ve bilgileri entegre etmek için yer kontrol istasyonu ile iletişim kurar. Olasılık fonksiyonuna bağlı olarak hareket etmek için uygun bir yön seçer. Bu işlemler için karınca kolonisi algoritması kullanılmıştır. Algoritmanın sağlamlığını doğrulamak için farklı boyutlarda ve sayıda hedefe sahip çeşitli haritalar denenmiştir.

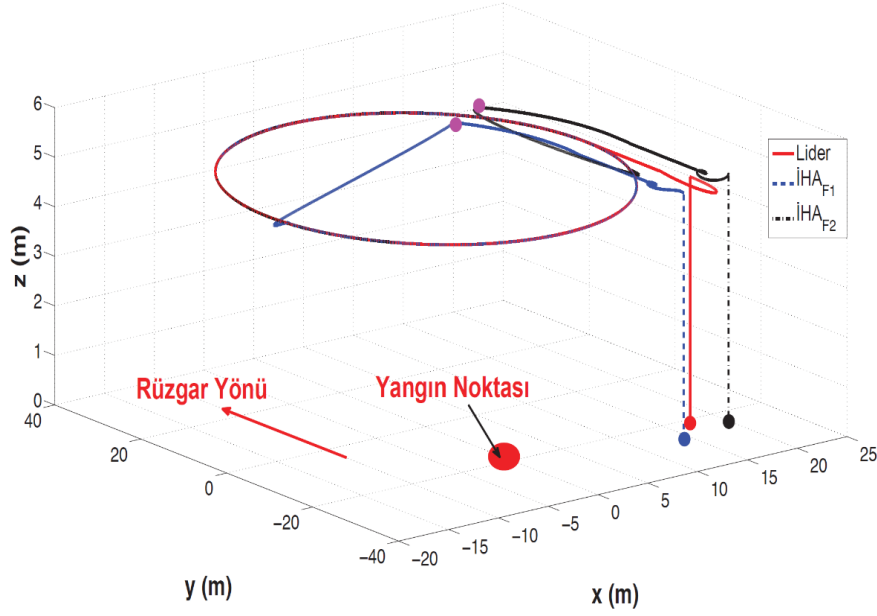
Ghamry ve arkadaşları (2016), orman yangınlarını izleme, yangın algılama ve yangın yayılmasının takibi için sürü şeklinde hareket eden insansız hava araçlarının

(İHA) kullanılmasını önermişlerdir. Önerilen algoritma, orman yangını problemlerini arama, doğrulama ve gözlem olmak üzere üç aşamayı dahil ederek çözmektedir. Arama aşamasında, İHA sürüsü lidere dayalı yaklaşım göstererek belirli bir diziliş şeklinde hareket eder, bu aşamada istenen diziliş şeklini korumak için dağıtık kayan modlu bir oluşum kontrolü tasarlanmıştır. Bir yangın algılandığında, yanlış alarmları önlemek için sürüdeki her İHA'da farklı tipte sensörler kullanılmıştır. Eliptik oluşturulan yangın çevresi için İHA'ları eşit olarak dağıtmak ve izlemek için Kayan Mod Kontrolü (Sliding Mode Control-SMC)'ne dayalı olarak yeniden yapılandırılabilen bir kontrolör tasarlanmıştır.

Bu çalışmada yangın şeklinin tahmini için bir noktada yanan bir ateşin sabit bir hızla genişleyeceği varsayılarak basit bir eliptik model üzerine çalışılmıştır. İHA'lardan alınan tüm sensör bilgileri yer kontrol istasyonuna gönderilir ve burada algoritma yeni eliptik yörüngeyi sensörlerden aldığı bilgilere göre tahmin eder. Belirlenen yeni yörünge lider İHA'ya görev olarak iletilir. Lider İHA'ya gelen yeni görev ile her İHA yeni pozisyonunu lidere göre hesaplayarak ilerler.

$$x = ct_{fire} + at_{fire}\cos\beta, \quad y = bt_{fire}\sin\beta \quad (2.6)$$

Orman izleme ve yangın algılama görevi sırasında ekibin oluşumu Şekil 2.10 ile gösterilmektedir. Görüldüğü gibi İHA ekibi havalanıyor ve arama görevine başlıyor. Yer kontrol istasyonu, İHA'ların izlemesi gereken referans yörüngesini Denklem 2.6'daki yangın yayılma modeline göre planlar, burada $0 \leq \beta \leq 2\pi$, t_{fire} yangının yayılma hızı, orijin ateşleme noktasıdır ve x eksenini rüzgar yönüdür. a ve b parametrelerinin her ikisi de doğrusal yangın yayılma oranlarıdır; 2a, yangının toplam uzunluğunun rüzgar yönünde (x eksenini) artış hızı ve 2b, rüzgara dik açılarda (y eksenini) yangın uzunluğundaki toplam artıştır. Elipsin merkezi rüzgar yönünde (x eksenini) sabit bir c hızında hareket eder. Elipsin büyük ve küçük eksenleri sırasıyla at ve bt'dir.

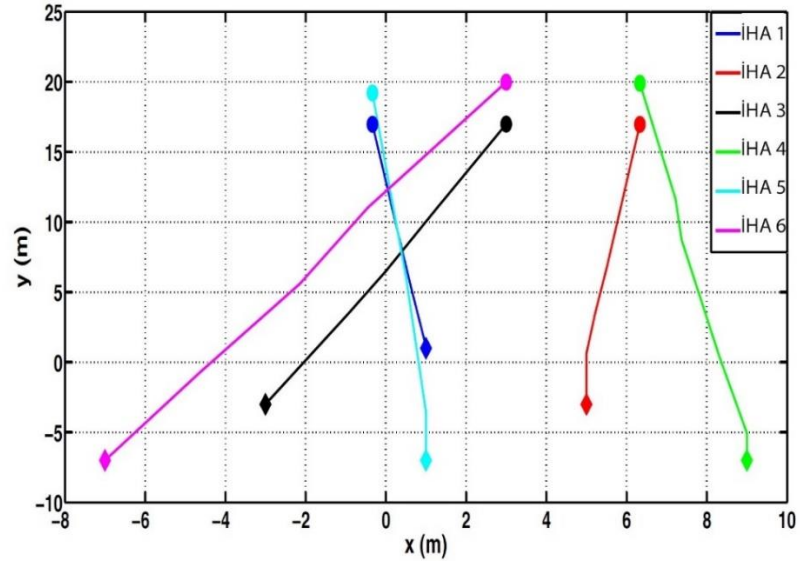


Şekil 2.10. İHA ekibinin oluşumu (Ghamry ve arkadaşları, 2016)

Ghamry ve arkadaşları (2017), orman yangınları ile mücadelede İHA sürüşü kullanılmasını ele almışlardır. Çalışmada kullanılan algoritmanın amacı, her bir İHA'nın ilk konumu ile atanan yangın noktası arasında kat edilen mesafeyi en aza indirmek için yeni koordinatlarını belirlemek ve rotalarını atamaktır. Bunun için PSO algoritması kullanılmış ve en uygun rota planlaması yapılmıştır. Bu planlama ile İHA'lar önceden belirlenen koordinata yangın söndürme sıvısıyla giderek iş birliği halinde yangına müdahale ettikleri varsayılmıştır. Önerilen yöntem sonuçları simülasyon ortamında incelenmiştir.

$$N = M_f \quad (2.7)$$

Yapılan çalışmada yangın tespiti aşaması ekibin önceki yaptıkları çalışma ile yapıldığı varsayılmıştır (Ghamry ve arkadaşları, 2016). Bu bağlamda her İHA gideceği yangın noktası bilgisine önceden sahiptir. Yangınları söndürmek için gerekli yangın söndürme sıvıları İHA'lara faydalı yük olarak eklenmiştir. N adet İHA ve M_f adet yangın noktası olarak Denklem 2.7'de verildiği üzere her yangına bir İHA ataması yapılmıştır. Şekil 2.11, görevin yürütülmesi sırasında İHA'ların yörüngelerini göstermektedir. Her yangın için en yakın İHA PSO ile seçerek görevlendirmeler yapılmıştır.



Şekil 2.11. Görev yürütme sırasında İHA'ların yörüngeleri (Ghamry ve arkadaşları, 2017)

3. TEORİK ALTYAPI

Bu bölümde sürü zekası optimizasyon algoritmaları ve sürü İHA hakkında temel bilgiler verilmektedir. Ardından projede kullanılan İHA bağlantı, simülasyon ve haberleşme yazılımları hakkında gerekli bilgiler detaylı olarak verilmiştir.

3.1. Sürü Zekası Optimizasyon Algoritmaları

Optimizasyon algoritmalarının gelişim süreci ve yeni algoritmaların oluşturulması çalışmalarına bakılacak olursa doğadan esinlenilerek birçok algoritma oluşturulmuştur. Doğadaki canlı davranışları incelendiğinde, toplu halde hareket ederken gerçekleştirdikleri eylemler ile minimum enerji ve maksimum verim olacak şekilde optimize edildikleri gözlemlenmiştir. Bu hareketleri esnasında her bir birey kendi deneyimlerini yorumlayarak diğer bireylere bunun sonuçlarını bir şekilde paylaşmaktadırlar. Tüm bu deneyimlerin farklı sonuçları paylaşıldıkça her birey tek tek her olasılığı denememiş ve diğer bireylerin deneyim sonuçlarını da alarak vakit kazanmıştır. Yani, sürü davranışı gösteren canlılarda belirli bir amaca yönelik davranışı optimum şekilde gerçekleştirmek ve hedefe en mantıklı yoldan ulaşmak için birlikte çalışma söz konusudur. Bu ve benzeri şekillerde karşılaşılan ortak hareketlerle birbirlerini etkileyen canlı davranışlarının her biri birer optimizasyon algoritması olarak karşımıza çıkmaktadır. Bilgisayar bilimlerinin doğayı taklit ilkesiyle beraber bu sürü davranışlarından esinlenilmesi ve optimize edilmesi çalışmalarının devam edeceği açıkça görülmektedir. (Akyol ve Alataş, 2012)

Doğadan ilham alınarak geliştirilen algoritmalara karınca kolonileri örnek verilebilir. Karıncaların toplu hareketlerindeki yuvaları ile yiyecek arasındaki en kısa yolu bulma kabiliyetleri izlenerek yola çıkılmış bir algoritmadır. Buradaki önemli noktalardan biri de karıncalar, yuva ve yiyecek arasındaki yolda çevre şartları değiştiğinde kolaylıkla adapte olup yeni en kısa yollar bulabiliyor olmalarıdır. Karıncaların bu davranışları incelendiğinde davranış temelinin görme üzerine değil kimyasal bir haberleşme yöntemlerinin olduğu anlaşılmıştır. Karıncalar birbirleriyle haberleşmelerinde feromon isimli kimyasal bir madde kullanmaktadırlar.

Karıncalar yuvadan çıktıklarında yürüdükleri yollar üzerine feromon maddesi bırakırlar. Her karınca yiyecek bulmak için rastgele bir doğrultuyu seçer. Zamanla tüm karıncalar aynı yolu takip etmeye başlarlar. Bunun sebebi tüm karıncaların hızları ve

yola bıraktığı feromon miktarı aynı kabul edilirse, birim zamanda daha kısa yollar üzerinde feromon miktarı daha fazla olacaktır. Yani yuva ile yiyecek arasındaki en kısa yolda feromon miktarı en fazla olacaktır. Bu sebeple feromon yoğunluğunu takip eden karıncalar hızla en kısa yolu tercih etmeye başlarlar.

Feromon yoğunluğu yüksek olan takip ettikleri yol üzerinde dış etkenlere bağlı olarak gerçekleşen bir değişiklik olduğunda yani yol üzerinde bir engel ile karşılaşıldığında çevredeki bu değişime adapte olabilirler. Takip ettikleri en kısa yol yeni durumdan sonra artık en kısa yol değilse, yeni en kısa yol bulunmaktadır. Böyle bir durumda gidebilecekleri yollardan birini rastgele seçerek devam ederler. Kısa zamanda yine aynı şekilde birim zamanda yol üzerindeki feromon miktarı artarak yeni en kısa yol bulunmuş olur.

Karıncalar kolonisi optimizasyon algoritması, gerçek karınca kolonilerinin doğal olarak yaptığı bu optimizasyonu taklit ederek yapılan yapay bir versiyonudur. Yapay feromon izleri ve yolların güncelleştirilmesiyle çalışan tekrarlı bir yapıları vardır. (Dalkılıç ve Türkmen, 2002)

Karıncalar kolonilerine benzer şekilde bir başka doğadaki davranışları incelenerek geliştirilen algoritmaya örnek ateş böcekleri verilebilir. Ateş böceklerinin birbirlerine ve çevresine parlak ışık göndermelerindeki amaçları izlenilerek esinlenilmiş sezgisel bir algoritmadır. Ateş böcekleri doğada birbirlerine kur yapmak, avlarını cezbetmek, düşmandan korunmak gibi amaçlarla parlak ışıklarını kullanırlar.

Ateş böceklerinin parlaklığa duyarlı davranışları incelenmiştir ve kur yapma davranışı üzerinde yoğunlaşmıştır. Ateş böcekleri içinde parlaklığı daha fazla olan ateş böceği çekicidir ve ateş böcekleri daha parlak olan ateş böceğine doğru yönelirler. Bu parlaklık etkisi uzaklık ile de doğru orantılıdır. Yani uzaktaki parlak ateş böceği o kadar parlak görünmediği için çekici değildir. Bir ateş böceği kendinden daha parlak ateş böceği bulamazsa rastgele hareket eder.

Ateş böceğinin parlaklığı amaç fonksiyonuna göre belirlenir. Ateş böceğinin çekiciliği parlaklığı ile doğru, aralarındaki mesafe ile ters orantılıdır. Öncelikle her bir ateş böceği diğer tüm ateş böceklerini kontrol eder. Parlaklığı kendinden fazla olan bir ateş böceği bulduğunda parlaklığa doğru hareket eder. Işık kaynağından uzaklaştıkça kaynağın uzaklığının karesi kadar ışık parlaklığını yitirmektedir. Bu sebeple ateş böceklerinin parlaklığı uzaktaki ateş böceklerini daha az etkilemektedir. (Gürgüze ve Türkoğlu, 2019)

Karınca kolonileri, ateş böcekleri gibi incelenen bir diğer hayvan sürüsü olarak guguk kuşları incelenmiştir. Guguk kuşları yumurtalarını yabancı kuşların yuvalarına bırakarak fark edilmeden başka bir yuvada büyüyerek hızlı bir çoğalma stratejisine sahiptirler. Bu hızlı çoğalma stratejisinin farklılığı optimizasyon algoritmalarına konu olmuştur.

Guguk kuşları etraftaki kuşları ve yuvalarını gözlemleyerek yumurtalarını uygun gördüğü farklı cins kuşların yuvalarına bırakırlar ve yuva içerisindeki bir yumurtayı da dışarı atarlar. Yuva sahibi kuş yabancı yumurtayı fark ettiğinde ise yuvadan bu yabancı kuşu atacaktır. Yuva sahibi kuş farklı olan yumurtayı fark etmezse bu guguk yumurtasını kuluçka eder ve diğer kuş yumurtalarıyla birlikte guguk kuşu yumurtası gelişir. Bu durum kuluçka asalaklığı olarak tabir edilmektedir.

Yuvada bulunan guguk yumurtaları yuva sahibi kuş yumurtalarından önce yumurtadan çıktıklarında diğer yumurtaları yuvadan dışarı atmaya çalışmaktadırlar. Yumurtadan yeni çıkmış olan guguk kuşu genellikle yuvadaki diğer yavrulardan daha büyük bir bünyeye sahip olmaktadır. Bu durum yuva sahibi kuşun getirdiği yiyeceklerin çoğunun guguk yavrusunun tüketmesini sağlar. Yuva sahibi kuş daha sonra bu yavrunun kendi yavrusu olmadığını anlasa bile annelik içgüdüğü ile bu yavruyu da beslemeye devam etmektedir. Guguk kuşları yumurtlama zamanları geldiğinde yumurtalarını benzetebileceği türlere sahip ve yiyecek kaynağı bol olan ortam arayışına girer ve bu amaçla göç ederler. (Abreo ve ark., 2021)

Karınca kolonileri, ateş böcekleri ve guguk kuşları sürülerinin davranışları incelenerek geliştirilmiş sürü zekası algortimalarından bahsedilmiştir. Bundan sonra bu projede kullanılan PSO ve AAA algoritmaları detaylı olarak incelenecektir.

3.1.1. Parçacık Sürü Optimizasyonu (Partical Swarm Optimization - PSO)

Bir optimizasyon algoritmasından beklenen, amaç fonksiyonunun mevcut kısıtlar ve sınırlar altında en uygun çözümü verecek karar değişken değerlerini bulmaya çalışmasıdır. Bu noktada doğadaki sürü davranışlarından ilham alınarak geliştirilmiş bir diğer algoritma da parçacık sürü optimizasyon algoritmasıdır. Sürüdeki bireyler diğer bireyleri etkileyerek sürünün amacına daha kolay ulaşmasını sağlar.

Bu algoritmadaki her bireye parçacık, parçacıkların oluşturduğu popülasyona da sürü ismi verilir. Her parçacığın çözüme yakınlığını hesaplayan formüle uygunluk fonksiyonu denilir. Uygunluk fonksiyonu gerçek çözüme ne kadar yaklaşıldığını

göstermektedir. Çözüm aranırken her bir parçacığın çözüme en yakın olduğu en iyi durumuna pbest, tüm sürüde çözüme en çok yaklaşan parçacığın durumuna ise gbest denilir.

Algoritmaya başlarken parametreler ve sürünün başlangıç pozisyonları ayarlanır. Her bir parçacık için ayrı ayrı uygunluk değeri hesaplanır. Yine her bir parçacık için pbest ve gbest değerleri hesaplanır ve güncellenir. Parçacıkların değişim hızları da tek tek hesaplanarak her biri için yeni pozisyonları belirlenir. Daha önce belirlenen sonlandırma kriterine ulaşıncaya kadar her parçacık için ayrı ayrı uygunluk değeri, pbest, gbest ve pozisyonları sırasıyla belirlenir.

Bu mantıkla çalışan bir algoritma ile herhangi bir nesnenin en az maliyetle en uygun yerlere yerleştirilmesi şeklindeki problemlere makul çözümler üretilebilir. Algoritmada parçacıklar en iyi çözümü takip etme eğiliminde olduklarından bir süre sonra parçacıklar bir yerde kümelenmiş şekilde rotalarına devam edeceklerdir. Bu durum aranacak bölge içinde detaylı arama gerektiren problemlerde olumsuz bir durum oluşturmaktadır. (Ghamry ve ark., 2017)

Optimizasyon problemlerinde en iyiye ulaşmak hedeflenir. Bu en iyi probleme ve çözümüne bağlı olarak minimum veya maksimuma yaklaşma olabilir. Yani kazançları en üst düzeye çıkarmak veya kayıpları en aza indirmek amaçlanıyor. Burada çözüme ulaşmak için matematiksel yöntemlerden ziyade sezgisel denilen yaklaşma şeklinde çalışan yeni yöntemler kullanılmaya başlanmıştır. Hayvan gruplarının sosyal davranışlarına dayanan bu yöntemler ile grup arasında hayatta kalma avantajlarının arttığı gözlemlenmiştir. Bu çalışmada PSO sezgisel algoritması incelenecektir.

PSO, var olan bir probleme karşı çözüm uzayında optimal bir çözüm aramak için doğadan ilham alınarak tasarlanmış bir algoritmadır. Algoritma mantığını anlamak için doğada gördüğümüz bir canlı sürüsü düşünmeliyiz. Balık ya da kuş sürüsü olabilir. Bir grup içinde hareket eden kuş sürüsünün sürünün diğer tüm üyelerinin deneyimlerinden faydalanabileceği düşüncesinden yola çıkılarak hazırlanmış bir algoritmadır. Başka bir deyişle, örneğin bir kuş rastgele uçup yiyecek ararken, sürüdeki tüm kuşlar keşiflerini paylaşabilir ve tüm sürünün en iyi avı elde etmesine yardımcı olabilir.

Sürü içerisindeki her bir birey parçacık ile ifade edilir. Temelde her bir parçacık sürü içerisinde sürünün en iyi konumuna sahip parçacığına yaklaşmayı amaçlar. Bir parçacığın çözüme olan yakınlığını anlamak için uygunluk fonksiyonu kullanılmaktadır. Bu fonksiyonun çözümüne yaklaşma hedeflenmektedir.

3.1.1.1. Particle Swarm Optimization (PSO) Algoritması

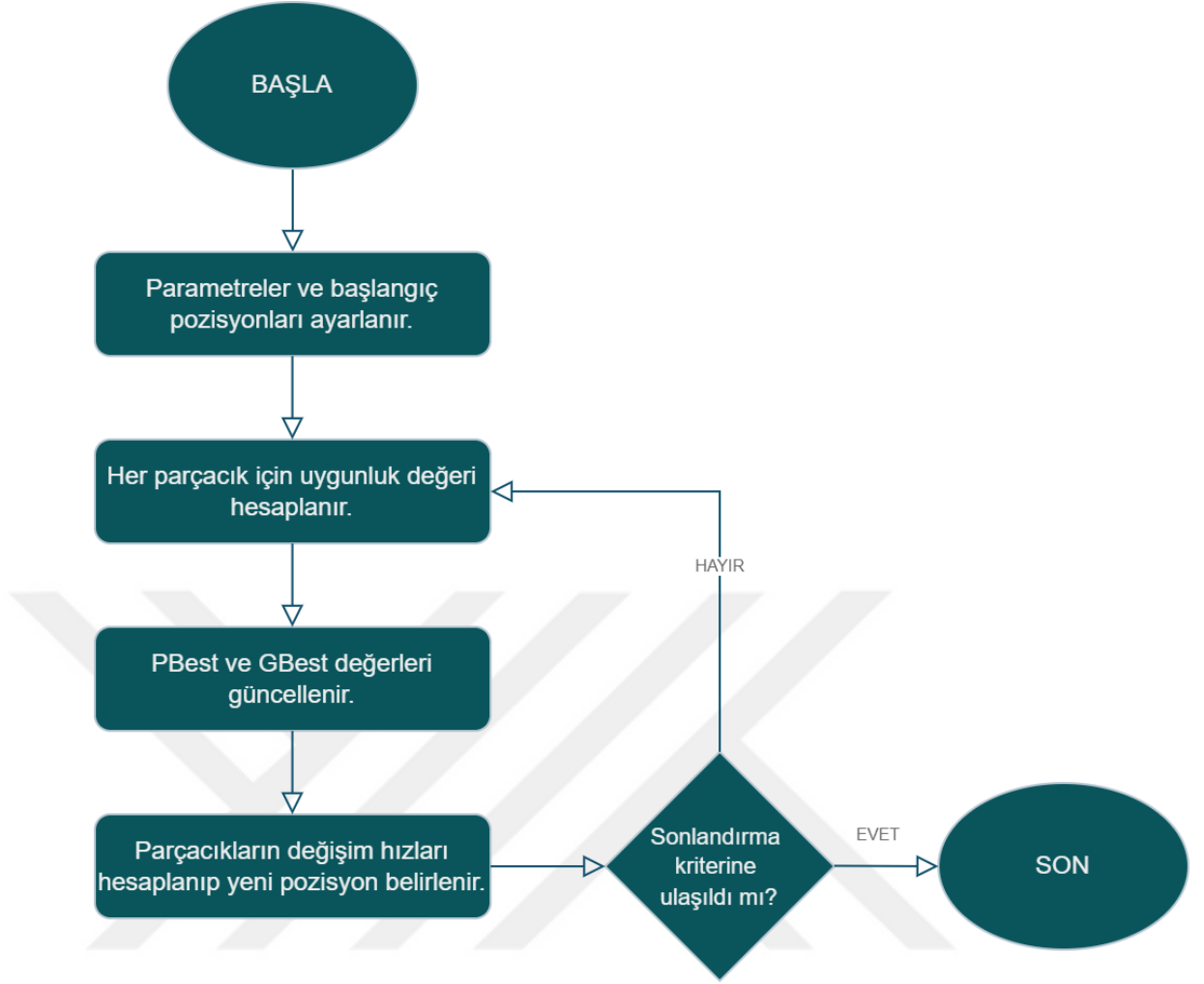
Algoritmada kullanılan değişken ve tanımlamalar şu şekilde olacaktır:

- Parçacık (Partical): Her bir bireye verilen isim.
- Sürü (Swarm): Parçacıkların bulunduğu popülasyona verilen isim.
- Uygunluk Fonksiyonu (Fitness Function): Bir bireyin çözüme ne kadar yakın olduğunu anlamak için kullanılan matematiksel ifadeye verilen isim. $f()$ şeklinde gösterilir.
- Pbest (Personal Best): Bir parçacığın kendisinin çözüme en yakın olduğu andaki en iyi durumuna verilen isimdir.
- Gbest (Global Best): Tüm sürüde arama boyunca çözüme en yakın olan parçacığın o andaki durumuna verilen isimdir.
- V_i : Parçacığın hızı
- X_i : Parçacığın konumu
- A : Popülasyon
- W : Eylemsizlik ağırlığı
- C_1 : Bilişsel sabit
- C_2 : Sosyal sabit
- R_1, R_2 : Rastgele sayılar

Parçacığın hız değişimi hesaplaması Denklem 3.1’de ve parçacığın yeni konum hesaplaması Denklem 3.2’de verildiği gibidir. Algoritmanın akış şeması Şekil 3.1’deki gibidir. (Tam, 2021)

$$V_{i+1} = W \cdot V_i + C_1 * R_1 * (pbest - X_i) + C_2 * R_2 * (gbest - X_i) \quad (3.1)$$

$$P_{i+1} = P_i + V_i \quad (3.2)$$



Şekil 3.1. PSO algoritması akış diyagramı

3.1.2. Yapay Alg Algoritması (Artificial Algae Algorithm-AAA)

Alglerin davranışlarının incelenmesi ise geliştirilmiş bir biyolojik yaklaşımlı algoritmadır. Problem uzayındaki her çözüm gerçek yaşayan bir alg ile aynı özellikleri gösteren yapay bir alg kolonisi ile ifade edilmektedir. Algoritma temelde üç bölümden oluşmaktadır. Bunlar;

1. Evrimsel süreç,
2. Adaptasyon,
3. Helisel hareket

Bir alg hücresi yeni bir hücre üretmek için bölündüğü zaman oluşan iki yeni hücre yan yana yaşarlar. Her bölünme işleminden sonra oluşan hücrelerin yan yana yaşaması ile koloni meydana gelmektedir. Kolonideki hücreler uygun olmayan koşullarda ölür veya koloni dağılabilir ve uygun koşullar sağlandığında her hücre yeni

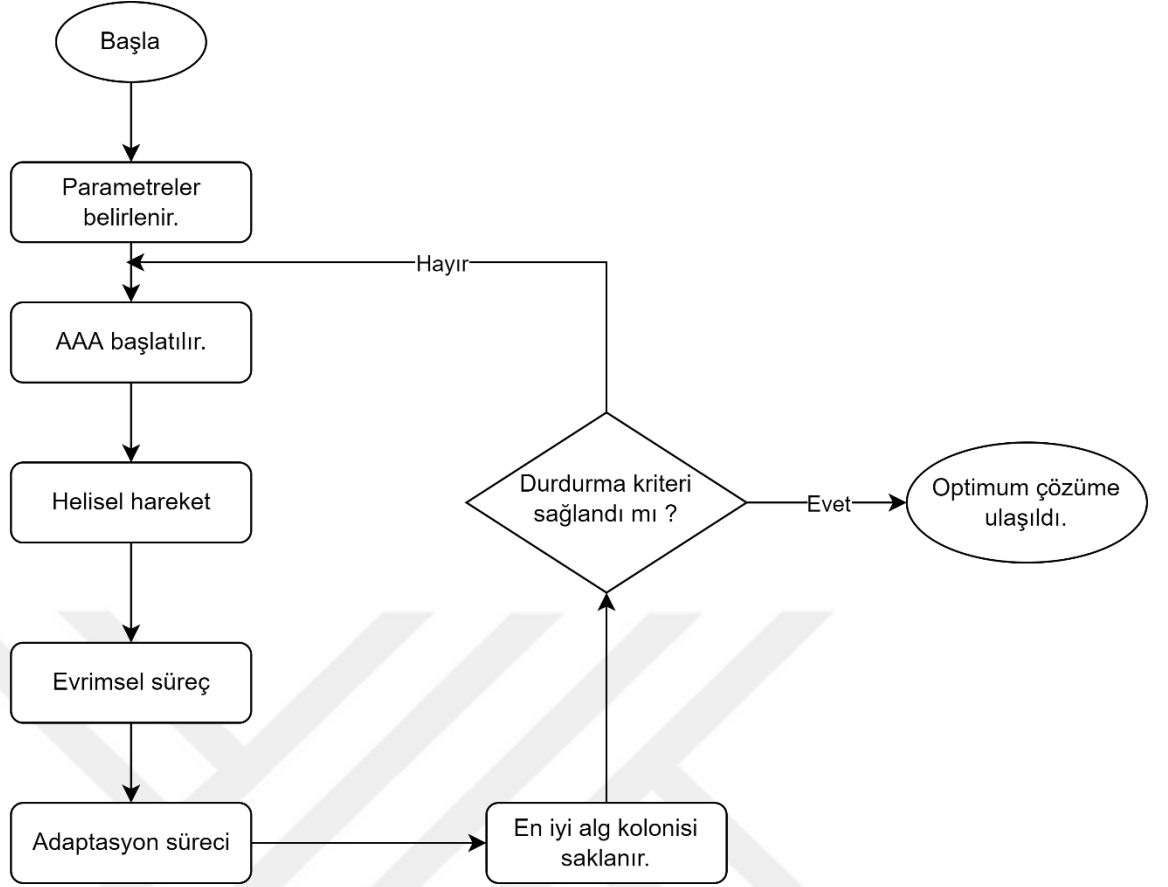
bir koloni oluşturabilir. Bunları yaparak her koloni optimum şekilde hayatını devam ettirmek ister.

Optimum alg hücrelerinin bir araya gelmesiyle oluşan koloni çözüme ulaştığında optimum elde edilmiş demektir.

Yapay alg kolonisinin evrimsel süreci incelendiğinde; ışık alan hücre gelişir, büyür ve çoğalır, yeterli ışık almayan hücre ise ölür. Başlangıçta tüm kolonilerin büyüklükleri bir olarak kabul edilir. Helisel hareket sonucunda iyi çözüm veren koloni daha fazla büyür ve tüm koloni için helisel hareket tamamlandığında evrimsel süreç başlar. En küçük boyuttaki koloniden ölen alg hücrelerinin yerine en büyük boyuttaki alg kolonisinin bir alg hücresi kopyalanır.

Bulunduğu ortamda yaşam koşulları iyi olmayan ve yeterince büyüemeyen koloni hayatta kalmak için ortama adapte olmaya çalışacaktır. Adaptasyonu sağlamanın yolu bulunduğu ortamdaki en büyük koloniye benzemeye çalışmak için baskın türü değiştirmektir. Bu süreçte kolonilerin açlık seviyeleri dikkate alınır. Başlangıçta her koloni için açlık seviyesi sıfır olarak kabul edilir. Helisel hareket sonucunda daha iyi bir çözüme ulaşan koloninin açlık seviyesi değişmez, daha kötü bir çözüme ulaşan koloninin açlık seviyesi artar. Tüm alg kolonileri için hareketler tamamlandıktan sonra, en kötü çözüme doğru giden yani en aç olan koloni adaptasyon işlemine tabi tutulur.

Alg kolonileri hayatta kalmaları için gerekli ışığı yeterince alabilmeleri için su yüzeyine yakın yerlerde yaşamaya çalışırlar. Sıvı içerisinde kamçıları ile hareket etmelerine helisel denilmektedir. Helisel hareketi, yer çekimi ve su ile arasında oluşan viskoz sürtünme ile engellenir. Büyük koloninin sürtünme yüzey alanı büyük olduğundan sıvı içerisinde daha yavaş hareket eder ve yerel arama yeteneği artar. Küçük kolonilerde ise durum tam tersidir ve global arama yetenekleri artar. Koloninin her bir hareketinde çözüm arama sayısı koloninin sahip olduğu enerjiye göre belirlenir. Her hareket başlangıcında sıfır ile bir arasında bir değer koloni büyüklüğü ile orantılı olarak hesaplanır. Her hareket ile enerji kaybı meydana gelir. İyi çözüme giden koloni için enerji kaybı daha az olmaktadır. Algoritmanın akış şeması Şekil 3.2'deki gibidir. (Uymaz, 2015)



Şekil 3.2. AAA algoritması akış diyagramı

3.2. Sürü İHA

Sürü İHA, belirli bir görevi gerçekleştirmek üzere birlikte koordineli olarak çalışabilen birden fazla İHA'dan oluşan sisteme verilen isimdir. Birden fazla İHA'nın koordinasyon sağlaması sebebiyle sürü davranışına benzetildiği için sürü İHA denilmektedir. İnsansız hava aracı sürüleri, verilen misyon hedefine ulaşmak için kendi kendilerini organize ederek görevleri yerine getiren bir çeşit araç filosudur. Sürü modeli düşünüldüğünde, her araç bir düğüm ile ifade edilir ve araçlar arasındaki bilgi alışverişinin sağlandığı bağ bir çeşit ağdır. (Bai ve arkadaşları, 2019)

Yüksek risk içeren İHA çalışmalarında birden fazla İHA kullanılması görevin başarı ihtimalini arttırmaktadır. Bir İHA görevini yerine getiremeyecek olursa görevi tamamlayacak başka İHA'ların olması başarı ihtimalini yüksek oranda arttırmaktadır. Birden fazla İHA'nın kullanılması aynı zamanda farklı özellikler gösteren İHA'ların aynı görev dahilinde yer almasına da olanak sağlamaktadır.

Sürü sistemleri; kuşlar, arılar gibi doğal sürü davranışlarını taklit ederek hayata geçirilirler. Gerçek bir sürüde canlılar, iş birliği yaparak görevlerini yerine getirirler ve

bunu yaparken de çevreye uyum sağlamak ve rekabet etmek için iletişim kurmaktadırlar. Yani bir sürü davranışının temelinde iletişim bulunmaktadır. (Engebraten ve arkadaşları, 2019)

3.2.1. Sürü İHA Yapısı

Çoklu İHA'ların sahip olduğu faydalı yükleri, otomasyonları, platform konfigürasyonları gibi özellikleri farklılık gösterebileceği gibi aynı da olabilir. Temeli bu ayrıma dayandırıldığında İHA sürüleri homojen ya da heterojen sürüler olarak sınıflandırılırlar.

İki sınıflandırma durumunun da verilen görevlere göre avantaj ve dezavantajları vardır. Homojen İHA'ların kullanıldığı bir sistemde birbirinin aynısı olan cihazlar sayesinde sistem yapısı daha basit olur, bakım çalışmaları ve yedek parça durumları daha kolay olur. Görev sırasında lider başarısız olursa homojen sürülerde yeni lider seçimi de kolay olacaktır. Faydalı yük çeşitliliği bulundurma ve esneklik gibi durumlarda ise daha kısıtlı kalmaktadır. Heterojen sürü İHA'lar daha karmaşık sistem yapılarına sahiptir. Ancak sürü içerisindeki farklı yeteneklere sahip olan İHA'lar sayesinde çok daha faydalı hale gelebilirler.

Tekli İHA'larda olduğu gibi çoklu İHA'larda da sistemin kontrolünün sağlandığı görev tanımlarının yapılması gerekmektedir. Bu konuda merkezi kontrol mimarisi ve merkezi olmayan kontrol mimarisi olarak iki farklı yaklaşımdan bahsedilmektedir.

Merkezi kontrol mimarisi yaklaşımında, İHA'lar kendi aralarında haberleşmezler. Merkezi bir noktadan her İHA'nın bilgileri alınır, koordinasyon sağlanır ve her İHA için görevler merkezde önceden tanımlanır. Kolay optimize edilir ve basit bir yaklaşımdır. Ancak İHA'da ya da haberleşmede oluşacak hatalara karşı dayanıksız bir ortam oluşturur.

Merkezi olmayan kontrol mimarisi yaklaşımında, İHA'lar kendi aralarında karşılıklı olarak haberleşebilirler. Kendi aralarında haberleşebilen bu İHA'lar sürü içerisindeki kararlarını da kendileri verebilirler. Merkezi sistemlere göre çok daha karmaşık yapıdadırlar. Görevler İHA'lar arasında paylaşıldığı için daha dayanıklı, esnek bir sistem sağlanmaktadır. Uçuş sırasında karşılaşılabilecek olumsuzluklara karşı daha dayanıklı ve yedekli bir sistemdir.

Sürü İHA çalışmalarının temelinde bu yaklaşımın otomatik çalışır hale gelerek insan kararından uzaklaştırılması, daha matematiksel ve optimum sonuçlar elde

edilmesi amaçlanmaktadır. Biyolojik sürü davranışında yer alan dağıtım, kendi kendine adaptasyon ve sağlamlık özellikleri, sürü İHA'larının koordineli özerk kontrolü için gereklilikler ile tutarlıdır. Biyolojik sürü davranışında yer alan iç mekanizmaları incelemek ve bunları sürü İHA'ların koordineli otonom kontrolüne uyarlamak, İHA'ların karmaşık ortamlarda karar verme ve planlama yeteneklerini geliştirebilir. (Zhang, 2020)

3.2.2. Sürü İHA Kullanım Alanları

Çok sayıda insansız hava aracının belirli bir düzen içerisinde uçarak verilen görevi otomatik veya bir pilot vasıtasıyla yerine getirebilir. Arama kurtarma çalışmalarında, orman yangını, sel felaketi gibi afetlerin takibinde ve mücadelesinde, köprülerin, baz istasyonlarının veya uzun hatlar gibi erişilmesi güç yerlerin gözlenmesinde, sınır kontrollerinde, güvenlik amacıyla izinsiz giriş izlenmesinde, telekomünikasyon hizmetlerinde, zirai ilaçlamalarda kullanılabilirler.

Çok sayıda İHA'nın aynı anda kullanılması geniş alanlarda olağandışı olaylar için verilen görevlerin yerine getirilmesinde mantıklı ve hızlı sonuçlar almayı sağlar. Örneğin, birden fazla faydalı yük ile aynı anda aynı alan farklı veriler elde edilerek taranabilir. Bir orman yangınında termal kamera, gündüz kamerası, lazer mesafe ölçer gibi farklı sensörler ile geniş alanlar taranabilmektedir. Ayrıca doğal afetler gibi hayati risk taşıyan durumlarda bu geniş alanların olabildiğince hızlı şekilde taranması oldukça önemlidir.

Çoklu hedef takibi gerektiren durumlar, arama kurtarma durumları gibi karmaşık görevlerde koordinasyon sağlanarak birden fazla insansız hava aracı sürü diye tabir edilen şekillerde uçabilmektedirler.

Ayrıca gösteri amaçlı görsel etkinlikler de sürü şeklinde yapılmaktadır. Bu gösteri uçuşlarının havai fişek gösterilerinin yerini alması gerektiği tartışmaları da başlamıştır. Böylece daha temiz enerjiyle daha kompleks gösteriler yapılabilir.

Genel olarak kaynak araştırmasında kullanılan çözümler incelendiğinde insansız hava araçlarının sürü hareketiyle ilgili temel sorunlar çarpışma ve rota oluşturma sorunları olarak karşımıza çıkmaktadır. Uçuş planı dahilinde oluşturulan rotaların zeka göstererek oluşturulması yani bir otonomi sunması istenilmektedir. Rota oluşturma sorunu düşünüldüğünde akla ilk olarak graf teorisi, komşuluklar, gezgin satıcı problemi gibi terimler gelmektedir. Gezgin satıcı problemi, bir adet satıcının birden fazla şehri (noktayı) gezmesi için optimum rotanın belirlenmesi problemidir. İHA'larda rota

oluşturmadaki sorun birden fazla satıcı birden fazla şehir olarak düşünüldüğü için çoklu gezgin satıcı problemi ile istenilen rota oluşturma problemine çözüm uygulanabilir.

3.3. Mission Planner

İnsansız hava araçlarının kontrolü için kullanılan farklı yer kontrol yazılımları bulunmaktadır. Güncel olarak kullanılan yer kontrol istasyonu yazılımlarını listeler isek;

- Mission Planner (Windows, Mac OS X)
- APM Planner 2.0 (Windows, Mac OS X, Linux)
- MAVProxy (Linux)
- QGroundControl (Windows, Mac OS X, Linux, Android ve iOS)
- UgCS – Universal Ground Control Station (Windows, Mac OS X, Ubuntu)
- Tower (Android telefonlar ve tabletler)
- DJI GS Pro (iOS tabletler)
- Pix4Dcapture (Android ve iOS)

Bunlardan en çok kullanılan ikisi “Mission Planner” isimli yazılım ve “QGroundControl” isimli yazılımdır. Bu yazılımlar harita üzerinde bir görev tanımlamak ve gerekli ayarlamaları yapmak için bir ara yüz kullanımı sağlamaktadır. Bu ara yüzde tanımlanan görevler bağlı olan cihazlara verilmekte ve uçuş boyunca da cihazlar bu yazılımlar üzerinde izlenebilmektedir. Aynı zamanda simülasyon desteği ile uçuş maliyeti olmadan geliştiriciler için denemelerin yapılabileceği bir ortam oluştururlar. (Ardupilot Dev Team, 2021)

Mission Planner Ardupilot ile tam uyumlu yer kontrol istasyonu yazılımıdır. 2010 yılında Michael Osborne tarafından yazılmıştır. Uçuş kontrol kartı ile usb üzerinden seri port haberleşmesi ya da kurulan udp ya da tcp portu üzerinden haberleşmektedir. Bağlantı kurulduktan sonra uçuş kontrol kartına yazılım yükleme/güncelleme, sensör kalibrasyonları, parametre güncellemeleri, anlık komut iletimi, anlık sensör ve konum takibi işlemleri yapılabilmektedir.

Yapılacak uygulama için gerçeğine en yakın şekilde yapılmasının istenmesi simülasyon ortamı için gerçek insansız hava araçlarının rota planlamasının yapıldığı bu yer kontrol yazılımlarından birinin simülasyon ortamı ile yapılması planlanmaktadır.

Böylece proje gerçek hayata geçirilirken yapılacak ekstra çalışma süresi minimumda olacaktır.

Açık kaynak kodlu olup GPLv3 lisansına sahip olması sebebi ile bu ortamlardan “Mission Planner” isimli yazılım kullanılacaktır. Yazılım ara yüzünün görüntüsü Şekil 3.3’te verilmiştir.



Şekil 3.3. Mission Planner arayüzü

Mission Planner kurulumu gerçekleştiğinde bu şekilde bir arayüz ile uçuş planı hazırlama, uygulamaya bağlanan araçlar için bu uçuş planını bir görev olarak verme ve uçuş esnasında da araçtan aldığı verileri anlık olarak izleyebilme imkanı sunmaktadır. Ayrıca uçuş log kayıtlarını da tutarak uçuş sırasında yaşanan aksilikler izlenebilir.

Yalnızca gerçek araçlar ile değil simülasyon ortamında tanımlı araçlar ile gerçeğiyle aynı işlemleri yaparak sanal olarak oluşan araçlar ile deneme ortamı da oluşturulmaktadır.

3.4. Dronekit-Python

Dronekit, 3DR firması tarafından geliştirilen, geliştiricilerin bilgisayarda Ardupilot uçuş kontrolörü ile iletişim kurarak uygulamalar yapmasına imkan sağlayan açık kaynak kodlu bir yazılımdır. Bu uygulamalar ile otopilot iyileştirilebilir, araç davranışlarına daha fazla zeka eklenebilir. Dronekit-Python açık kaynak kodlu ve topluluk odaklı bir projedir ve yazılımının tüm kaynak kodu <https://dronekit.io/> adresinden indirilebilir. Teknik destek ihtiyacı olduğunda Dronekit forumları oldukça

aktif olarak kullanılmaktadır. Dronekit MAVLink isimli bir protokol ile araçlarla iletişim kurar.

Dronekit, programcılar için İHA uygulama katmanında işlem yapabilmeye olanak sağlar.

3.5. MAVLink Protokolü

MAVLink, İHA'larla iletişim kurmak için bir mesajlaşma protokolüdür. Genişletilebilir İşaretleme Dili (Mesajlar Extensible Markup Language -XML) dosyaları içinde tanımlanır. Çoğu yer kontrol istasyonu ve otopilot tarafından uygulanan mesaj seti common.xml'de tanımlanır.

MAVLink, Massachusetts Teknoloji Enstitüsü(MIT) lisansı ile kullanılmaktadır. Kaynak kodunu yayınlamadan herhangi bir kapalı kaynaklı uygulamada kullanılabilme imkanı sunmaktadır.

Telefon ile haberleşmek için bir haberleşme alt yapısı kullanılmaktadır. Bu alt yapıyı kullanmak için standartlaşmış bir sistem olarak ülke kodu, yerel kod ve kişinin numarası çevrilmelidir. İHA iletişimi için MAVLink bu örneğe benzetilebilir. (Ardupilot Dev Team, 2019)

3.6. Ardupilot

Ardupilot, tıpkı Windows donanım yazılımının bilgisayar donanımıyla iletişim kurması gibi İHA donanımına komut verir. Ardupilot gibi uçuş yazılımları olmadan çok rotorlu İHA'ların uçurulması imkansız olurdu denilebilir. Çünkü Ardupilot İHA motorlarına saniyede yaklaşık 400 komut gönderebilir. Bu durum da İHA'ların sorunsuz ve sabit bir uçuş gerçekleştirebilmesi anlamına gelmektedir.

Ardupilot Berkeley merkezli 3DR şirketi tarafından açık kaynak kodlu olarak 2009 yılında yayımlanmıştır. Dünya genelinde birçok geliştirici sayesinde sürekli gelişerek ve güncellenerek sektörde lider yazılımlardan biri haline gelmiştir. 2016 yılında 3DR firması ve Dronecode ile ilişkisi kesilen proje GPLv3 lisansına sahiptir. Uçuş kontrol kartı donanımlarına yüklenen ardupilot yazılımları kullanım amacına göre ayrıca isimlendirilmektedir. Örneğin sabit kanatlı İHA'lar için arduplane, döner kanatlı İHA'lar için ise arducopter alt adı kullanılmaktadır. (Ardupilot Dev Team, 2019)

3.6.1. Ardupilot SITL

SITL, “Software-In-The-Loop (Döngü İçinde Yazılım)” anlamına gelir. SITL kullanmanın avantajları:

Aynı proje kodu gerçek bir otopilot için veya simüle edilmiş İHA için derlenebilir.

Gerçek üretici yazılımını doğrudan bilgisayardan test edilebilir. Örneğin, aniden GPS sinyali kaybolursa İHA'ya ne olur? Bu SITL'de test edilebilir.

Kodu sahada denemeden önce simüle edilmiş ardupilot, dronekit python komut dosyalarını test etmek de mümkündür.

SITL, GPS, pil gibi gerekli tüm donanımı modeller. Sensörlerdeki GPS'deki gürültü, rüzgar hızı ve yönü gibi çevresel olumsuzluklarla ilgili geniş bir parametre setinin kurulmasına izin verir. Tüm bu parametreler simüle edilen modeli gerçek duruma yaklaştırmaktadır. Akademik çalışmalarda deneylerde, test sonuçlarını etkilememek ve aynı koşullar altında tekrarlanabilir testlere sahip olmak için ortam koşulları genellikle gürültüsüz kabul edilir. Otonom bir uçuş gerçekleştirmek için uçuş kontrolörü bir görev dosyasını (yerel olarak depolanan) okur ve görevi başarıyla tamamlamak için gerekli komutları gönderir. Tanımlanan araç ile bağlantı açık olduğundan, sensör veya kamera verilerinin okumaları, GPS mesajları, aracın konumu gibi uçuşun her yönüne ilişkin ayrıntılı bilgileri uçuş günlüğüne kaydedilir. Her uçuş sonunda uçuş kayıtlarının indirilebilmesi için sistem bir yer kontrol istasyonuna bağlanır. Bunun yanı sıra, aracı başlatmak, görevi gerçekleştirmek, günlükleri toplamak ve süreçleri temizlemek için DroneKit API kullanılarak uygulanan bir Python betiği kullanılarak tüm deneysel süreç otomatikleştirilir, böylece tüm deneyler birbirinden bağımsız olarak çalıştırılabilir. (Mendes ve ark., 2018)

4. MATERYAL VE YÖNTEM

Bu bölümde bu tez kapsamında kullanılmış olan programların detaylı kurulum ve çalıştırma aşamaları ve gerçekleştirilen yazılımın işleyişinin aşamaları anlatılmaktadır. Son olarak ise simülasyon sonuçları ile görsel olarak çalışması gösterilmiştir.

4.1. Ortam Oluşturulması ve Kurulumlar

Açık kaynaklı İHA yazılımları hakkında yapılan araştırmalar sonucu Ardupilot yazılımlarının kullanılmasına karar verilmiştir. Ardupilot ekosisteminde farklı tip insansız araçlara uygun otopilot yazılımları, yer kontrol istasyonu yazılımları ve yardımcı yazılımlar yer almaktadır. Yardımcı yazılımlar içerisinde geliştiriciler için simülasyon ortamları bulunmaktadır.

İHA'ları uzaktan kontrol etmeye yarayan sistemlerin genel adı yer kontrol istasyonlarıdır. Yer kontrol istasyonunun ana bileşenlerinden biri de yer kontrol istasyonu yazılımlarıdır. Yer kontrol istasyonu sayesinde İHA'ların anlık konumları, sensör verileri uzaktan izlenebildiği gibi İHA'lara veri bağı üzerinden komut gönderilebilir.

Açık kaynak kodlu olması ve ArduPilot otopilot yazılımına uyumluluğu ile ön plana çıkan Mission Planner yer kontrol istasyonu yazılımı tez çalışması için seçilmiştir. Mission Planner uygulaması Windows işletim sistemli sistemlerde çalışmakta olup aynı zamanda seri port, udp, ve tcp protokolleri ile İHA bağlantısı yapabilecek kabiliyettedir. Mission Planner yazılımı kaynak web sitesinden ücretsiz şekilde indirilmiş ve Windows üzerine kurulumu yapılmıştır.

Ardupilot yazılımının SITL adında İHA'ların simüle edilmesine yarayan alt yazılımı bulunmaktadır. SITL sayesinde İHA'lar fiziki olarak bulunmasa dahi sanal olarak çalıştırabilir, geliştirilebilir ve kodları test edilebilir. SITL Windows işletiminde de çalışabilir fakat alt yapısı Linux tabanlı olduğu için Ubuntu üzerinde daha sağlıklı çalışmaktadır. Mission planner uygulaması Windows işletim sistemi üzerinde çalıştığından aynı anda ubuntu alt yapıyı yazılımları da kullanabilmek için Windows Subsystem Linux (WSL) kurulumu yapılmıştır. WSL kurulumu için Windows yazılım mağazasından Ubuntu'nun istenilen sürümü Ubuntu 20 indirilmiştir. Akabinde WSL özelliği Windows üzerinden aktif edilerek kurulum tamamlanmıştır. WSL kurulumu sonrasında Ardupilot için gerekli ortam kurulumları adımına geçilmiştir. Ardupilot

github üzerinden indirilmektedir. İndirme işlemi tamamlandıktan sonra WSL üzerinden kurulum gerçekleştirilmiştir. Ardupilot github reposu yerel depolama birimine kaydedilir ve kurulur. Ardından SITL kurulumu için gerekli adımlar yapılır. Bu adım içerisinde en önemli adımlardan birisi ağ uygulama çerçevesi (web application framework WAF) kurulumudur. WAF sayesinde istenilen versiyona sahip STM32 mikro denetleyicine sahip uçuş kontrolcü donanımının sanal kopyası oluşturulmaktadır. Yapılan çalışmada STM32 H7 mikro denetleyici mimarisine sahip Şekil 4.1’de gösterilen Cortex-M7 işlemcili Cube Orange uçuş kontrol kartı modelinin WAF konfigürasyonu yapılmıştır.



Şekil 4.1. Uçuş kontrol kartı

Sonrasında proje için kullanılacak tüm kurulum aşamaları aşağıda sıralanmıştır:

- Dronekit Python 2.7 sürümü ile tam uyumlu olduğu için Python 2.7 yüklenir.
- Python 2.7 üzerine pip kurulumu yapılır.
- Pip üzerinden hem python 3 için hemde python 2.7 için dronekit kurulumu yapılır.
- DroneKit-Python ve dronekit-sitl simülatorü tüm platformlarda pip'ten kurulur.
- Ardupilot github dan temin edilerek ubuntu üzerinde kurulur.
- Ardupilot kurulduğu için artık sanal uçuş kontrol kartı oluşturulan waf konfigürasyonu gerçekleştirilir.
- SITL kurulumu yapılarak sanal olarak oluşturulan uçuş kontrol kartı bir İHA olarak kullanılabilir hale gelir.
- Bu işlem sonrasında sanal İHA istenildiğinde çalışmaya hazır hale getirilmiştir.

- SITL başlatılırken İHA'nın istenilen koordinatta başlaması için -L parametresi eklenir.
- Sürü çalışmaları için birden fazla sanal İHA oluşturulması gerekmektedir.
- Komut dosyası düzenlenerek her İHA için ayrı ad ve parametre (.parm) tanımlanır ve dosyaya eklenir ve Copter parametre dosyasında tanımlanır.
- Her bir parametre dosyasının içinde ayrı ID tanımlaması yapılır ve parametre dosyasına eklenir.
- Bu işlemlerden sonra her İHA terminal ekranından ayrı ayrı çağırılarak çalıştırabilir ve Mission Planner uygulamasına bağlayarak takip edilebilir.
- Proje için 5 farklı İHA sanal olarak başlatılmış ve Mission Planner üzerinden farklı UDP portları ile bağlanabilir hale gelmiştir. Her bir İHA için açılan UDP portları sırası ile;

I0: 127.0.0.1:14550 ve 127.0.0.1:14551,
 I1: 127.0.0.1:14560 ve 127.0.0.1:14561,
 I2: 127.0.0.1:14570 ve 127.0.0.1:14571,
 I3: 127.0.0.1:14580 ve 127.0.0.1:14581,
 I4: 127.0.0.1:14590 ve 127.0.0.1:14591 olmuştur.

4.2. Fiziki Donanım Planlaması

Çalışmada kullanılacak İHA'ların afet alanından sensör verisi elde edebilmesi için gerekli faydalı yük araştırması ve planlanması yapılmıştır. Sıcaklık ölçümü için iki farklı varyasyon belirlenmiştir. Bu özellikler Çizelge 4.1'de gösterilmiştir. Optris marka Pi640 model termal kamera ana sıcaklık ölçüm sensörü olarak belirlenmiştir. 32Hz yenileme hızı ve radyometrik veri sağlama özellikleri ile öne çıkan bu sensörün yurt dışı satış fiyatı 6800 Euro'dan başlamaktadır. Çalışma kapsamında algoritmanın başarısı dahilinde daha düşük özellikli bir sıcaklık algılama sensörü ile başarıya ulaşma ihtimali üzerinde durulmuş ve hedefler arasına koyulmuştur. DF Robotics markasının SEN0263 model sıcaklık algılama sensörünün yurt dışı satış fiyatı yaklaşık 57 Euro'dur bu sensör ile yangınları tespit edebilmek sistemlerin yatırım maliyetini de düşüreceği için çalışmanın önemini artırmaktadır.

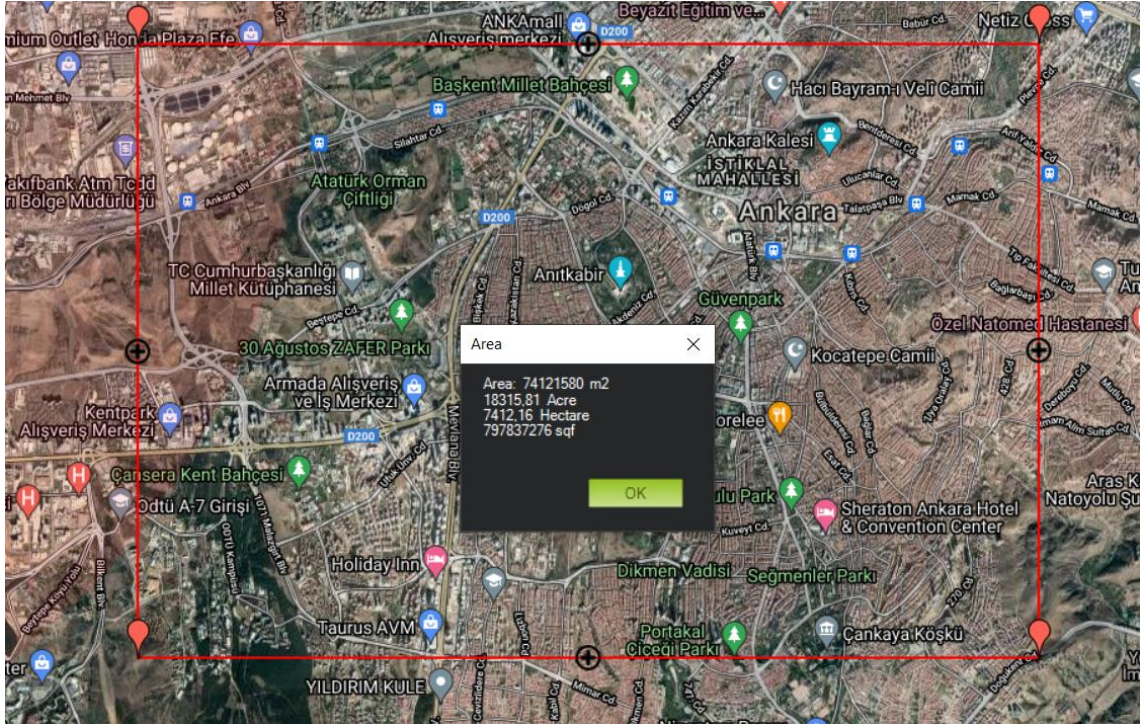
Çizelge 4.1. Sıcaklık ölçüm amaçlı faydalı yük özellikleri

	Termal Kamera - Optris Pi640	Termal Sensör - DF Robotics SEN0263
Görüş alanı	90° x 60°	5° x 5°
Ölçüm çıktısı	Çoklu nokta radyometrik	Tek nokta ortalama
Ölçüm Aralığı	-20° ve 900° arası	-70.01° ve 270° arası
Çözünürlük	0,075°	0.01 °
Fiyat	6800 Euro	57 Euro

Karbonmonoksit gaz ölçümü için MQ7 isimli sensör kartı planlanmıştır. Bu kart güncel olarak 33 TL'ye temin edilebilir ve hassasiyeti yüksektir.

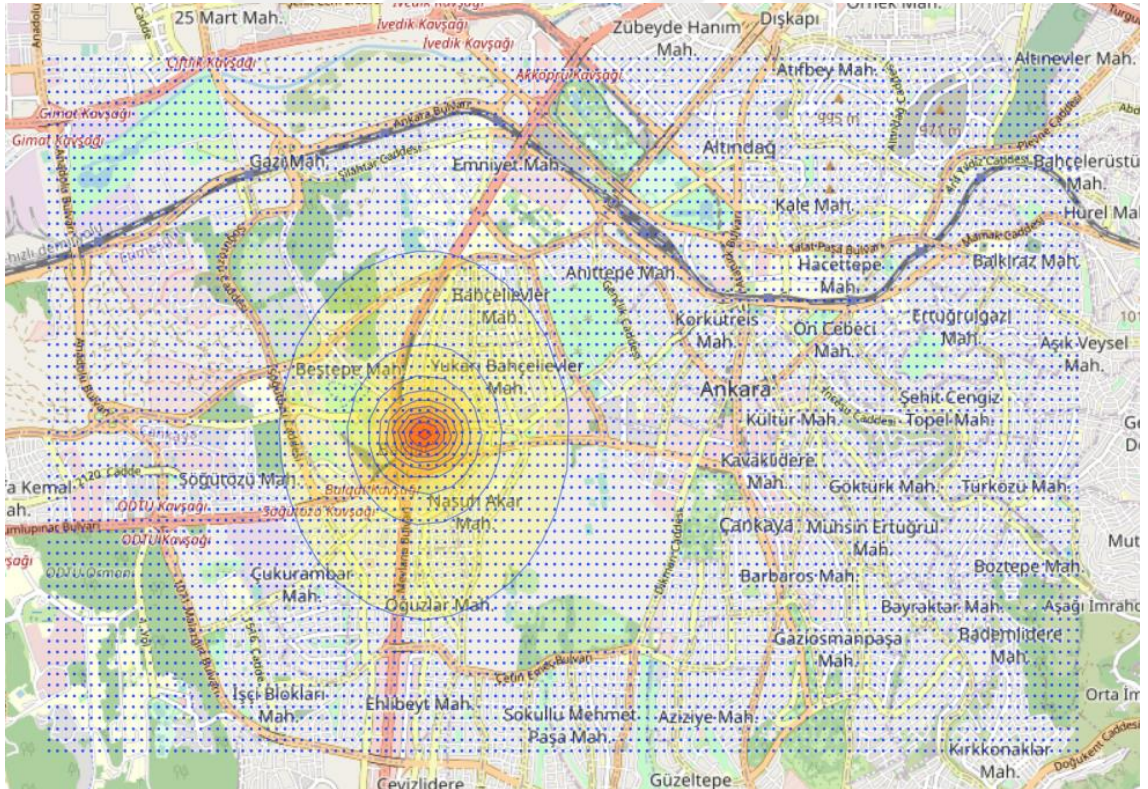
4.3. Afet Senaryosu

Yazılımın gerçekleştirileceği afet yangın senaryosu olarak belirlenmiştir. Simülasyon ortamında çalışıldığından kod üzerinde daha önce belirlenen yangın bölge ve sensör bilgileri oluşturulmuştur. Yangının aranacağı alan Şekil 4.2'de gösterildiği gibi 7412,16 hektardır.

**Şekil 4.2.** Taranacak alan

Senaryo çalışması sonucunda 4 bilgi içeren bir matris oluşturulmuştur. Oluşturulan matraste belirlenen sınırlar içerisindeki her koordinat değeri 0.001 aralıklarla alınmıştır. Yani belirlenen sınır içerisindeki 0.001 aralıklarla tüm koordinat olasılıkları latitude(enlem) ve longitude(boylam) olarak tabloya eklenmiştir. Daha sonra senaryo gereği belirlenen yangın koordinatına yaklaşıldıkça sıcaklık ve karbon monoksit değeri artacağı için sıcaklık ve karbon monoksit sensörlerini taklit edecek veriler üretilmiştir.

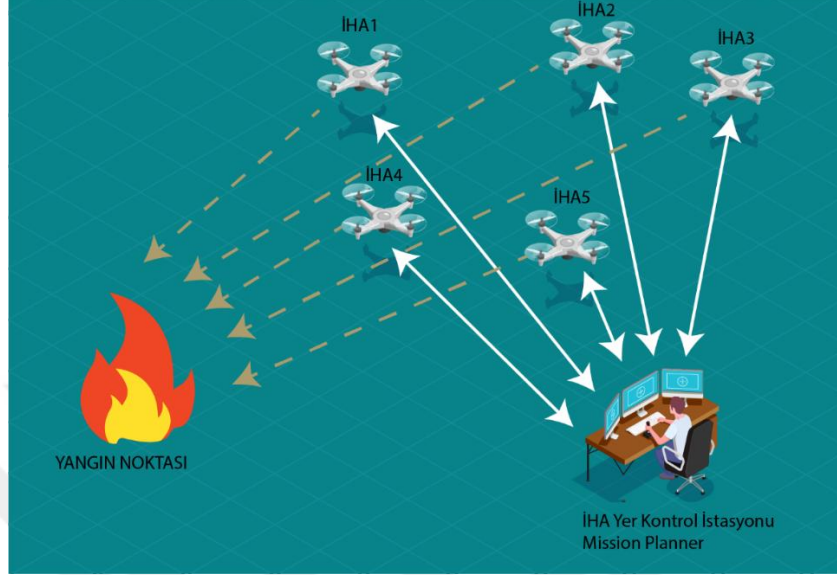
Senaryo verilerinin tutulduğu tablo içerisinde bulunan tüm koordinatlara karşılık gelecek şekilde sıcaklık ve karbon monoksit verileri de oluşturulmuştur. Bu veriler Şekil 4.3’de olduğu gibi her koordinat için belirlenen yangın koordinatına olan öklid mesafesi hesaplanarak sıcaklık ve karbon monoksit için yaklaştıkça artacak şekilde değerler üretilmiştir. Böylece alandaki hemen hemen her koordinata karşılık gelebilecek sensör verileri yaklaşık olarak oluşturulmuştur.



Şekil 4.3. Senaryo gereği taklit edilen sensör verilerinin tematik gösterimi

Son olarak, İHA’lar, yangın noktasını çevreleyen yeni duruma göre oluşumunu yeniden yapılandıracak ve alarmı teyit etmek için yangın çevresini takip edecek,

izleyecek ve yangının yayılması, yoğunluğu, yanan alan ve hala görülebilen yanan alanlar hakkında güncel çevrimiçi bilgiler sağlayacaktır. Tüm proje senaryosunun modellenmesi Şekil 4.4’de gösterilmiştir.

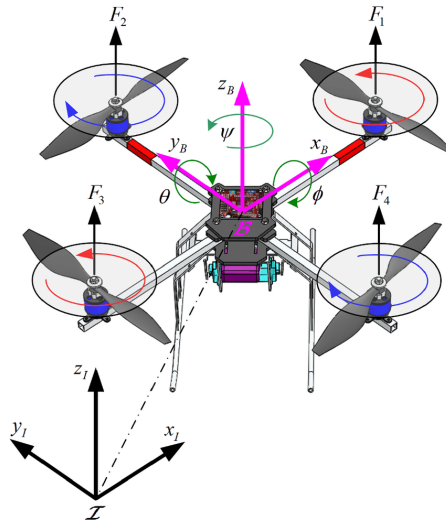


Şekil 4.4. Proje senaryosunun modellenmesi

Şekil 4.4’de görüldüğü üzere, tanımlanan 5 İHA yer kontrol istasyonu ile ayrı ayrı bağlanmıştır ve yangına doğru yönelim içerisindedir.

4.4. Proje Aşamaları

4.4.1. Dört Rotorlu İHA Dinamik Modelleme



Şekil 4.5. Dört rotorlu İHA'nın koordinasyon sistemi (Zhang, 2019)

Bu çalışmada ele alınan İHA'ların tamamı aynı özellikte olup Şekil 4.5'de gösterildiği gibi, merkezi etrafında simetrik olarak yerleştirilmiş dört rotorlu bir helikopterdir.

$$x = (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\frac{U_z}{M}, \quad \phi = \frac{U_\phi}{J_{xx}} \quad (4.1)$$

$$y = (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\frac{U_z}{M}, \quad \theta = \frac{U_\theta}{J_{yy}} \quad (4.2)$$

$$z = -g + (\cos\theta\cos\phi)\frac{U_z}{M}, \quad \psi = \frac{U_\psi}{J_{zz}} \quad (4.3)$$

Burada ϕ , θ ve ψ sırasıyla yuvarlanma, adım ve sapmayı temsil eden Euler açılarıdır. J_{xx} , J_{yy} ve J_{zz} sırasıyla x , y ve z eksenleri etrafındaki dört rotorlu eylemsizlik momentidir. Denklemdaki dört rotor eksenleri boyunca kuvvetler ve momentler Denklem 4.4 – 4.7 'deki tanımlamalar gibidir.

$$U_z = F_1 + F_2 + F_3 + F_4 \quad (4.4)$$

$$U_\phi = L(F_3 - F_4) \quad (4.5)$$

$$U_\theta = L(F_1 - F_2) \quad (4.6)$$

$$U_\psi = K_{yaw}(F_1 + F_2 - F_3 - F_4) \quad (4.7)$$

Burada $F_i, i=1, \dots, 4$ dört pervanenin i 'si tarafından üretilen itme kuvvetidir, L motor ile helikopter merkezi arasındaki mesafedir. K_{yaw} , pervane itişini yalpalama momenti ile ilişkilendiren bir sabittir.

Dört rotorlu bir İHA'nın iki boyutlu uzaydaki duruşu Denklem 4.1'deki gibi tanımlanır. Bu tanımdaki ifade edilen iki boyut x ve y koordinatları üzerindeki enlem ve boylamı, θ ise dönme açısını ifade eder.

$$q_i = [x_i, y_i, \emptyset]^T \in \mathbb{R}^3, \forall i \in \{1, 2, \dots, N\} \quad (4.1)$$

Bir İHA'nın basit bir matematiksel modeli Denklem 4.2'deki gibi temsil edilebilir.

$$q = \begin{bmatrix} x \\ y \\ \emptyset \end{bmatrix} = \begin{bmatrix} \cos\emptyset & 0 \\ \sin\emptyset & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (4.2)$$

Burada v ve w İHA'nın doğrusal ve açısal hızlarını göstermektedir. Kontrol girişleri vektörü ise Denklem 4.3'deki gibi tanımlanabilir. (Ghamry ve arkadaşları 2017)

$$u(t) = [v(t), w(t)]^T, u \in \mathbb{R}^m, m = 2 \quad (4.3)$$

Bu çalışmada projede şu varsayımlar kabul edilerek bir yazılım gerçekleştirilmiştir:

Varsayım 1: Hepsi aynı özelliklere sahip döner kanatlı 5 adet İHA tanımlanır.

Varsayım 2: Tüm İHA'ların başlangıç noktaları aynı olup Anıtkabir'in koordinatı olan [39.9250869,32.8369463] olarak, yangın merkez koordinatı ise [39.9166159,32.8170741] olarak belirlenmiştir.

Varsayım 3: İHA'ların yangın arayacakları ve taranacak alan sınır koordinatları [(39.8868211,32.7715302),(39.9508062,32.8937531)] olup 7412,16 hektardır.

Varsayım 4: İHA'ların uçuş boyunca iletişim sorunları olmadığı ve uçuşlarının çevre koşullarından etkilenmeden tamamlandığı varsayılmıştır.

Proje özeti Şekil 4.6'de gösterildiği gibi İHA'ların bağlanma, görev alma, uçuş işlemlerinin yapılması ve Mission Planner ekranından izlenmesini kapsamaktadır.



Şekil 4.6. Proje özeti

4.4.2. İHA'ların bağlanması ve hazırlanması

Projede kullanılan simülasyon ortamı gerçek bir yer kontrol istasyonunun simülasyon ortamı olduğu için ortam oluşturulması ve kurulumlar kısmında anlatıldığı gibi her bir İHA için terminalden bir SITL yani sanal bir İHA oluşturulur. Daha sonra ise kod içerisinde bu İHA ile bağlantı kurulur. Kurulan bağlantı dronekit kütüphanesinde kullanıldığı şekli ile yapılmaktadır. Her bir İHA için bir araç nesnesi oluşur ve bu nesne üzerinden hem işlemler yapılır hem de uçuş verisi kayıtları tutulur.

Eş zamanlı olarak projenin izlenebilmesi için Mission Planner üzerinden de oluşturulan SITL'ler ile bağlantı sağlanır ve bağlı olan İHA'lar ekranda izlenebilir. Şekil 4.7'de İHA'ların bağlantılarının terminal ekranından izlenmesi görülmektedir.

```

C:\Python27amd64\python.exe

Reading mission from file: b.txt

FINAL SOLUTION:
Connecting to vehicle1 on: udp:127.0.0.1:14550
udp:127.0.0.1:14550
Connecting to vehicle2 on: udp:127.0.0.1:14560
udp:127.0.0.1:14560
Connecting to vehicle3 on: udp:127.0.0.1:14570
udp:127.0.0.1:14570
Connecting to vehicle4 on: udp:127.0.0.1:14580
udp:127.0.0.1:14580
Connecting to vehicle5 on: udp:127.0.0.1:14590
udp:127.0.0.1:14590

```

Şekil 4.7. Her İHA ile yapılan bağlantı takibi

4.4.3. Sezgisel yaklaşım ile görev belirleme

Proje ile iki farklı sezgisel yaklaşım metodu denenerek iki algoritma için sonuçlar incelenerek karşılaştırma yapılmıştır. PSO ve AAA algoritmaları ile ayrı ayrı işlemler uygulanarak her bir İHA'nın bir sonraki koordinatı belirlenerek yangın noktasına yaklaşma rotası oluşturulmuştur.

4.4.3.1 PSO ile görev belirleme

Yazılan PSO kodu ile her bir yinelemede ortaya çıkan her parçacık için oluşan koordinatlar İHA'ların yeni koordinatları olarak belirlenmiştir. Belirlenen bu koordinatlar MAVLink protokolüne uygun şekilde proje konumu içerisinde .txt formatında kaydedilmektedir. Bu dosya Şekil 4.8'de görülmektedir.

missionADA1 - Not Defteri											
	Dosya	Düzen	Birim	Görünüm	Yardım						
0	1	0	16	0	0	0	0	39.5022981	32.500847	0.0	1
1	0	3	22	0.0	0.0	0.0	0.0	39.5019841102	32.5003904332	20.0	1
2	0	3	16	0.0	0.0	0.0	0.0	39.5019839532	32.5003902049	20.0	1
3	0	3	16	0.0	0.0	0.0	0.0	39.5019748817	32.4998941185	20.0	1
4	0	3	16	0.0	0.0	0.0	0.0	39.5019855235	32.4998489944	20.0	1
5	0	3	16	0.0	0.0	0.0	0.0	39.5019861443	32.4997017382	20.0	1
6	0	3	16	0.0	0.0	0.0	0.0	39.5019854994	32.4994937503	20.0	1
7	0	3	16	0.0	0.0	0.0	0.0	39.5019859635	32.4993387353	20.0	1
8	0	3	16	0.0	0.0	0.0	0.0	39.5019852662	32.4991991394	20.0	1
9	0	3	16	0.0	0.0	0.0	0.0	39.501985297	32.4989427756	20.0	1
10	0	3	20	0.0	0.0	0.0	0.0	39.5019852416	32.4988043114	20.0	1

Şekil 4.8. İHA için tanımlanan görev dosyası

PSO algortimasının çalışma adımlarının açıkça izlenip takip altına alınması için veriler kayıt altına alınmıştır. Bu veriler Şekil 4.9’da görülmektedir.

	A	B	C	D	E	F	G	H	I	J
1	iterasyon	drone	err	err_best	p_best_X	p_best_Y	positionX	positionY	velocityX	velocityY
2	0	0	0,040978	0,040978	39,92509	32,83695	39,92509	32,83695	0,661749	-0,18724
3	0	1	0,040978	0,040978	39,92509	32,83695	39,92509	32,83695	-0,57523	-0,84557
4	0	2	0,040978	0,040978	39,92509	32,83695	39,92509	32,83695	-0,98877	-0,99024
5	0	3	0,040978	0,040978	39,92509	32,83695	39,92509	32,83695	-0,53668	-0,08186
6	0	4	0,040978	0,040978	39,92509	32,83695	39,92509	32,83695	0,541458	-0,26698
7	1	0	0,045096	0,040978	39,92509	32,83695	39,9284	32,83601	0,003309	-0,00094
8	1	1	0,030209	0,030209	39,92221	32,83272	39,92221	32,83272	-0,00288	-0,00423
9	1	2	0,025255	0,025255	39,92014	32,832	39,92014	32,832	-0,00494	-0,00495
10	1	3	0,035314	0,035314	39,9224	32,83654	39,9224	32,83654	-0,00268	-0,00041
11	1	4	0,044096	0,040978	39,92509	32,83695	39,92779	32,83561	0,002707	-0,00133
12	2	0	0,023044	0,023044	39,914	32,83137	39,914	32,83137	-0,01439	-0,00464
13	2	1	0,025707	0,025707	39,92031	32,83228	39,92031	32,83228	-0,0019	-0,00044
14	2	2	0,025206	0,025206	39,92012	32,83197	39,92012	32,83197	-2,5E-05	-2,5E-05
15	2	3	0,019759	0,019759	39,91998	32,82768	39,91998	32,82768	-0,00243	-0,00885
16	2	4	0,023851	0,023851	39,91792	32,83503	39,91792	32,83503	-0,00987	-0,00058
17	3	0	0,030153	0,023044	39,914	32,83137	39,9251	32,82904	0,011104	-0,00233
18	3	1	0,019969	0,019969	39,91965	32,82822	39,91965	32,82822	-0,00066	-0,00406
19	3	2	0,022624	0,022624	39,92006	32,82994	39,92006	32,82994	-5,9E-05	-0,00203
20	3	3	0,019702	0,019702	39,91996	32,82764	39,91996	32,82764	-1,2E-05	-4,4E-05
21	3	4	0,007641	0,007641	39,91816	32,82113	39,91816	32,82113	0,000235	-0,0139
22	4	0	0,010037	0,010037	39,91782	32,82335	39,91782	32,82335	-0,00729	-0,0057
23	4	1	0,011762	0,011762	39,91823	32,82443	39,91823	32,82443	-0,00142	-0,00379
24	4	2	0,006084	0,006084	39,91947	32,81851	39,91947	32,81851	-0,00059	-0,01142
25	4	3	0,012378	0,012378	39,91762	32,82545	39,91762	32,82545	-0,00235	-0,00219

Şekil 4.9. PSO sonucu verilerin takibi

Hazırlanan yazılım ile öncelikle simülasyon olarak oluşturulmuş araçlarla tanımlanan portlarda bağlantı gerçekleştirilir.

Daha sonra tek tek bağlanan bu araçlara PSO sonucu oluşturulmuş .txt görev dosyaları yüklenir. Görev dosyaları her İHA’ya ayrı ayrı yüklenir ve bu işlem yine terminal ekranından Şekil 4.10 deki gibi izlenir.

```

C:\Python27amd64\python.exe

Reading mission from file: missionADA1.txt

Upload mission from a file: missionADA1.txt
Clear mission
Upload mission

Reading mission from file: missionADA2.txt

Upload mission from a file: missionADA2.txt
Clear mission
Upload mission

Reading mission from file: missionADA3.txt

Upload mission from a file: missionADA3.txt
Clear mission
Upload mission

Reading mission from file: missionADA4.txt

Upload mission from a file: missionADA4.txt
Clear mission
Upload mission

Reading mission from file: missionADA5.txt

Upload mission from a file: missionADA5.txt
Clear mission
Upload mission
Close vehicle1 object
Close vehicle2 object
Close vehicle3 object
Close vehicle4 object
Close vehicle5 object

```

Şekil 4.10. Tüm İHA'lar için görev dosyalarının yüklenmesinin takibi

Bu işlemler WSL da terminal ekranından takip edilerek gözlemlenebilir.

4.4.3.2. AAA ile görev belirleme

Projeye eklenen AAA kodu ile koloninin en iyi çözüme ulaşması için her bir yinelemede aldıkları yol İHA'lar için birer pozisyon olarak belirlenmiş olur.

Uygunluk fonksiyonu PSO ile aynı fonksiyon olup şu şekildedir:

$$f(x) = h.CO \quad (4.3)$$

Denklem 4.3'de x , yangına yakınlığı ifade etmektedir. h , sıcaklık ve CO karbon monoksit seviyesini göstermektedir. Yangına yakınlık sıcaklık ve karbon monoksit ile doğru orantılı olarak değişmektedir. Burada verilen uygunluk fonksiyonu ile yangına maksimum yakınlık bulunmak istenmektedir.

Uygunluk fonksiyonundaki bu değerler, gerçek sensörler ile çalışılmadığından dolayı önceden ayarlanan senaryo çalışması ile oluşturulan dosyadan yakınlığa göre sensör verilerini taklit eden veriler alınarak kıyaslanmış ve hesaplama yapılmıştır. Yine PSO ile aynı şekilde AAA sonucunda oluşan koordinatlar MAVLink protokolüne uygun

şekilde proje konumu içerisinde .txt formatında kaydedilmektedir. Projenin geri kalanı yukarıda anlatıldığı gibi aynı ilerlemektedir. Tek fark İHA koordinatlarının birinde PSO algoritması ile birinde de AAA algoritması ile hesaplanması olmuştur. Onun dışındaki tüm diğer koşullar aynı tutulmuş ve tüm işlemler aynı şekilde ilerlemiştir.

4.4.4. İHA'ların simülasyon ortamında uçuşlarının izlenmesi

Kod içerisinde her bir İHA'nın tüm koordinatları iki farklı sezgisel algoritma ile oluşturulmuş ve İHA'lar için görev olarak tanımlanmıştır. Bu koordinatlar eş zamanlı olarak dosyalara yazılarak kıyaslama ve ortalama alma gibi matematiksel hesaplamaları da yapılmıştır.

İHA'ların görevlerini gerçekleştirilmesi ve takibi için Mission Planner'a ayrı ayrı bağlantılar yapılır. Her bağlantı için belirlenen portlar seçilir. Bu işlemin yapılışı Şekil 4.11'de görülmektedir.



Şekil 4.11. İHA'ların Mission Planner'a bağlanması

Görevleri belirlenmiş İHA'ların önce motorları etkinleştirilir. Motorları çalıştırılan İHA uçuşa hazır duruma gelmektedir. Daha sonra belirlenen yüksekliğe kadar yükselir ve görevi ile verilen koordinata doğru ilerler.

Sonuç olarak tüm araçların Mission Planner'a başarılı şekilde bağlı olduğu Şekil 4.12'deki gibi görülmektedir.



Şekil 4.12. Başarılı İHA bağlantıları

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez kapsamında, yangın afeti sırasında yangının merkezini tespit ederek arama kurtarma faaliyetlerini hızlandırmak amaçlanmıştır. Bu amaç doğrultusunda çoklu İHA kullanılarak sürü zekası algoritmalarından elde edilen rota planlanmasında Python (2.7) programlama dili kullanılmıştır. Otopilot yazılımı olarak ArduPilot, araç iletişimi için Dronekit ve simülasyon ortamı için ise MissionPlanner kullanılmıştır. Sürü İHA'ların yangın alanını tespit etmek için kullandıkları rotalar PSO ve AAA sezgisel algoritmalarıyla oluşturulmuştur. Tüm arama uzayının yaklaşım algoritması kullanılmadan taranması ile yaklaşım yöntemleri ile yangın alanına yönelmeler süre olarak kıyaslandığında %300 fark gözlemlenmiştir.

5.1. Araştırma Sonuçları

Bu çalışmada gerçekleştirilen literatür taramasında incelenen benzer çalışmalar dikkate alındığında aradaki en büyük farklardan birinin kullanılan Mission Planner ve SITL araçları olduğu gözlemlenmiştir. Bu ortamlar İHA'ların bir sayıdan ibaret parçacıklar olmasından ziyade çok daha fazla parametreye bağlı gerçek birer araç olarak ele alınmasına olanak vermiştir. Bu ortamların geliştiricilere sunduğu simülasyon ve test ortamları gerçek İHA gömülü yazılımı alt yapısı ve parametreleriyle çalışması yapılan testlerin gerçek ortamda çalıştırılabilir olmasına olanak sağlamaktadır. Ayrıca kullanılan her yazılımın açık kaynak kodlu olmasıyla dünyanın her noktasından geliştiricilerle bu teknolojinin daha ileri seviyelere taşınması mümkün hale gelmektedir.

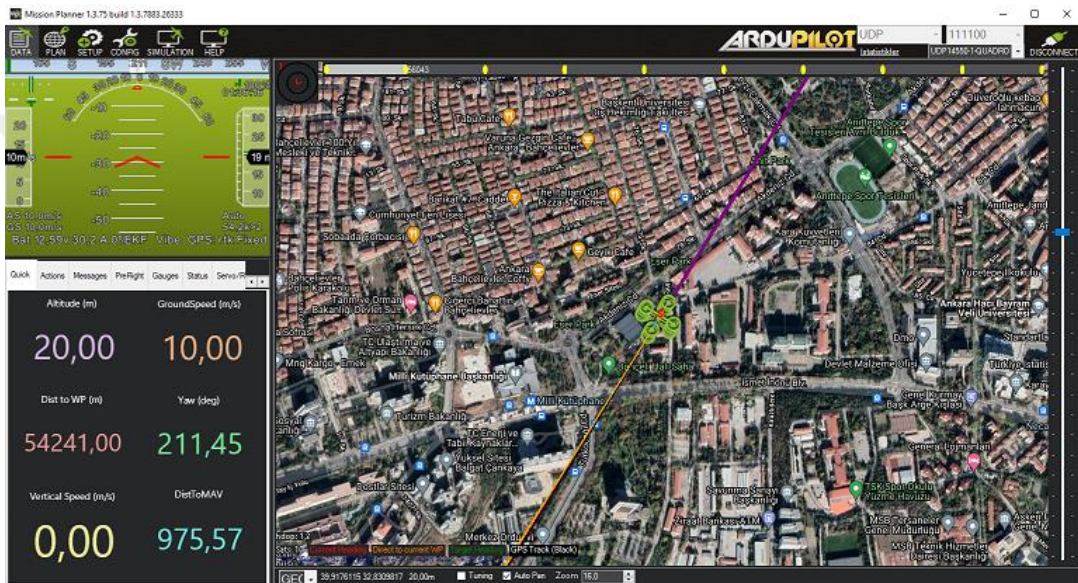
Gerçekleştirilen çalışmalar sonucunda tüm araçlara yüklenen görevler terminal ekranından başlatılır. Uçuşların takibi hem terminal ekranında Şekil 5.1'deki gibi hem de simülasyon ekranında Şekil 5.2'deki gibi detaylı olarak izlenebilir.

```

nisa@DESKTOP-1259FSQ: /mnt/c/Users/NisanurKirazPekin$ cd /mnt/c/Users/NisanurKirazPekin/desktop/PSOMission
nisa@DESKTOP-1259FSQ: /mnt/c/Users/NisanurKirazPekin/desktop/PSOMission$ sudo python 04_mission.py
Connecting...
Downloading mission
Downloading mission
Downloading mission
Downloading mission
Downloading mission
A valid mission has been uploaded: takeoff!
Home waypoint added to the mission
Arming motors
Taking Off
>>V1 Altitude = 0.0 m
>>V2 Altitude = 0.0 m
>>V3 Altitude = 0.0 m
>>V4 Altitude = 0.0 m
>>V5 Altitude = 0.0 m

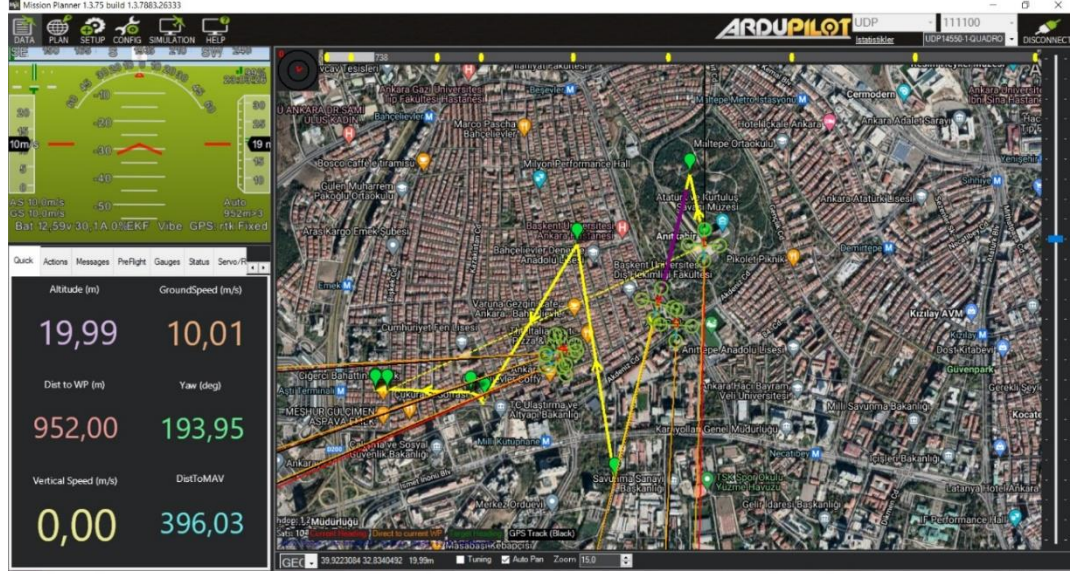
```

Şekil 5.1. Verilen görevi gerçekleştiren İHA'ların terminal ekranından takibi

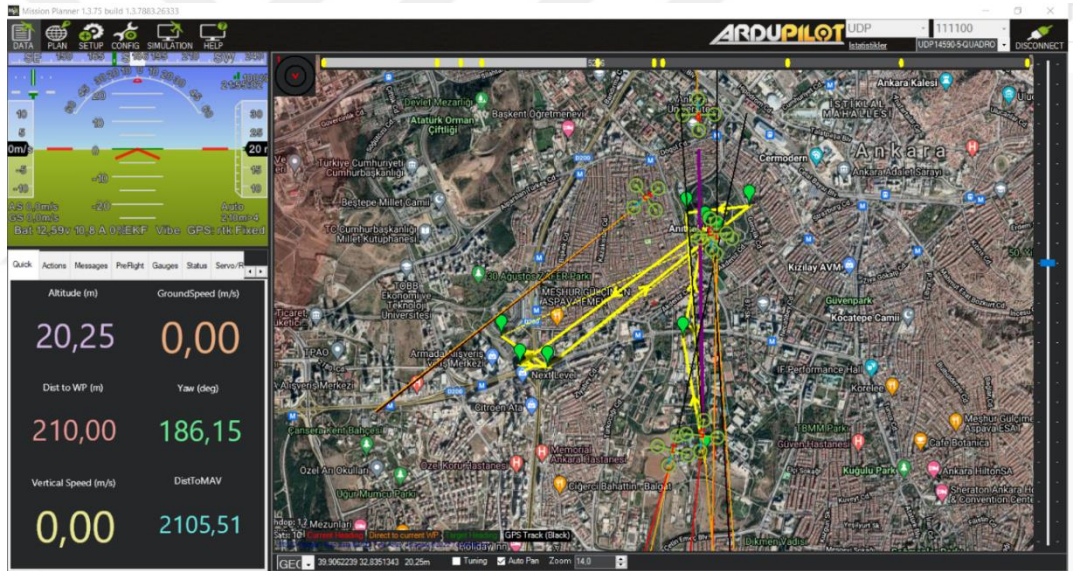


Şekil 5.2. Verilen görevi gerçekleştiren İHA'ların Mission Planner takibi

PSO ile oluşturulan rota örneği Şekil 5.3'te ve AAA ile oluşturulan rota örneği 5.4'de gösterilmiştir.

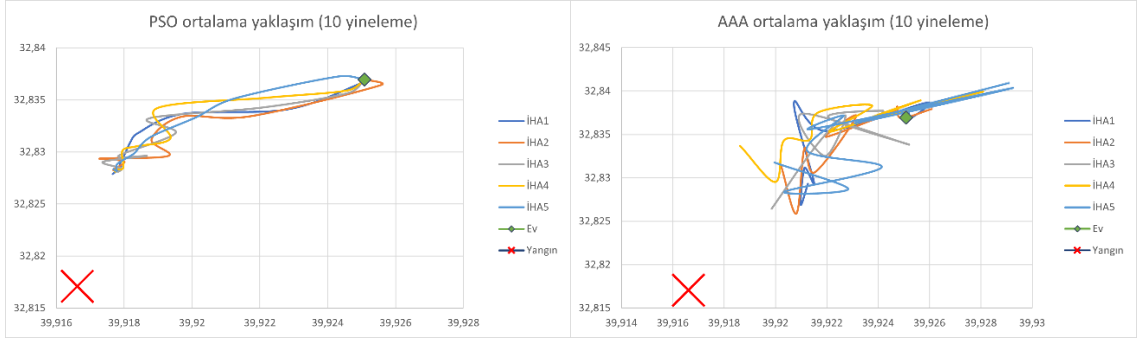


Şekil 5.3. PSO sonucu uçuş görüntüsü

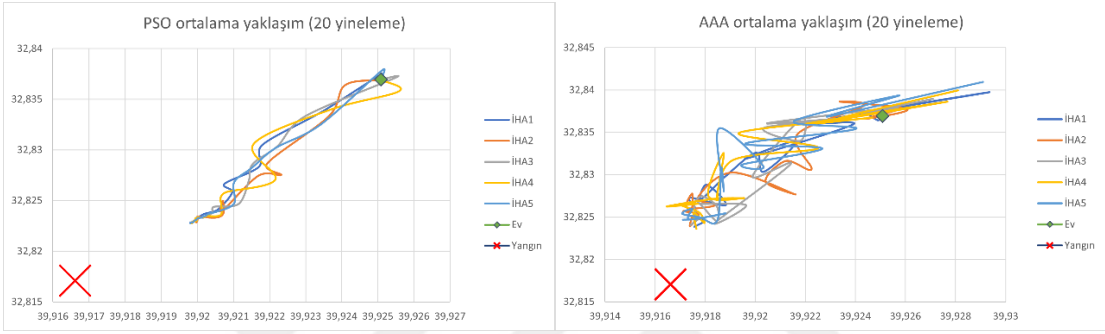


Şekil 5.4. AAA sonucu uçuş görüntüsü

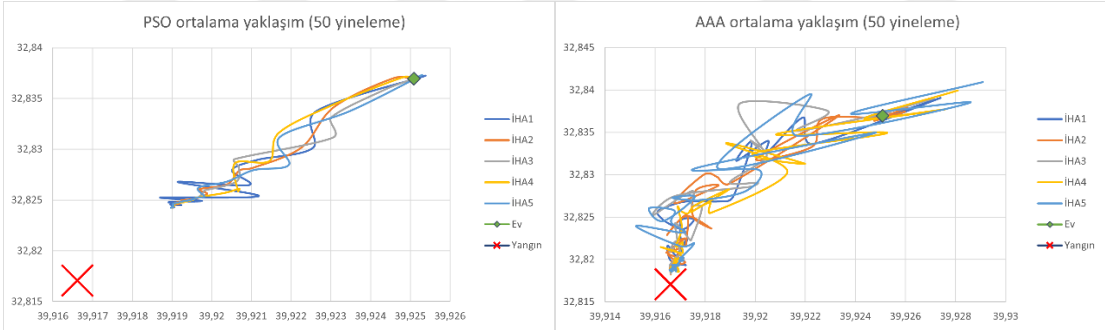
Bu çalışmada kullanılan her iki algoritma da sezgisel yaklaşım olduğu için her seferinde farklı sonuçlar elde edilmektedir. İki algoritmanın daha iyi kıyaslanabilmesi için 10, 20 ve 50 yinleme için ayrı ayrı her iki algoritma da yirmişer kez çalıştırılmış ve ortalamaları alınmıştır. Her iki algoritmanın 20 çalıştırma sonucu ortalama yaklaşma sonuçları grafiklenerek Şekil 5.5, Şekil 5.6 ve Şekil 5.7’de olduğu gibi görülmektedir. İki algoritma çalışmasında da verilen şekil ve grafikler incelendiğinde İHA’ların yangın alanına doğru yöneldiği bu sayede mantıklı yaklaşım sağladığı gözlemlenmiştir.



Şekil 5.5. PSO ve AAA için ortalama yaklaşım grafikleri (10 yineleme)



Şekil 5.6. PSO ve AAA için ortalama yaklaşım grafikleri (20 yineleme)



Şekil 5.7. PSO ve AAA için ortalama yaklaşım grafikleri (50 yineleme)

Yangın yeri ile İHA'ların yol noktaları arasındaki mesafeler Denklem 5.1'deki öklid mesafe bulma formülü ile hesaplanmıştır. Bu formülde A ile ifade edilen İHA'nın yol noktası, B ile ifade edilen yangın noktası olacak şekilde aralarındaki mesafeler hesaplanmıştır. PSO ve AAA algoritmaları için ayrı ayrı olacak şekilde her algoritma 20 kez çalıştırılmış ve her İHA'nın son yol noktaları alınarak Denklem 5.1'de verilen formül ile mesafeler hesaplanmıştır. Çizelge 5.1'deki veriler 10 yinelemeli PSO algoritmasıyla, Çizelge 5.2'deki veriler 10 yinelemeli AAA algoritmasıyla, Çizelge 5.3'deki veriler 20 yinelemeli PSO algoritmasıyla, Çizelge 5.4'deki veriler 20

yinelemeli AAA algoritmasıyla, Çizelge 5.5'deki veriler 50 yinelemeli PSO algoritmasıyla, Çizelge 5.6'deki veriler 50 yinelemeli AAA algoritmasıyla hesaplanan rotada İHA'ların son yol noktaları ile yangın noktasına mesafelerini metre cinsinden göstermektedir.

$$d(A,B) = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad (5.1)$$

Çizelge 5.1. PSO algoritması ile İHA'ların yangına olan mesafeleri 10 yineleme

Çalıştırma	İHA1	İHA2	İHA3	İHA4	İHA5	Ortalama
1	71,73	345,58	112,1	286,09	54,7	174,04
2	1355,52	1209,15	1251,18	1204,65	1210	1246,1
3	1355,13	1244,56	1301,83	1256,56	1290	1289,616
4	17,53	13,52	48,56	17,29	49,8	29,432
5	1370,68	1407,86	1361,7	1349,3	1360	1369,908
6	1718,8	1723,87	1723,99	1627,83	1710	1700,898
7	1655,04	1627,43	1633,05	1631,67	1630	1635,438
8	564,3	563,12	563,34	562,43	566,9	564,018
9	962,47	882,28	949,05	992,19	953,4	947,878
10	162,15	46,35	61,87	107,75	95,2	94,664
11	1265,97	1264,89	1263,91	1270	1290	1270,954
12	1504,13	1499,99	1503,2	1500	1500	1501,464
13	1308,17	1304,44	1309,22	1300	1300	1304,366
14	285,79	315,03	280,45	281,3	885,8	409,674
15	1276,53	1294,58	1300,61	1290	1260	1284,344
16	704,73	785,54	1073,58	805,3	552,6	784,35
17	802,07	818,04	809,53	808,7	788,1	805,288
18	1522,93	1519,48	1514,34	1520	1520	1519,35
19	1632,28	1629,53	1624,62	1650	1670	1641,286
20	775,13	767,89	776,79	764,2	772,4	771,282
Ortalama	1015,554	1013,18	1023,146	1011,263	1022,945	1017,218
Std. Sapma	543,487692	522,858631	535,676422	515,747789	523,166544	523,048112

Çizelge 5.2. AAA algoritması ile İHA'ların yangına olan mesafeleri 10 yineleme

Çalıştırma	İHA1	İHA2	İHA3	İHA4	İHA5	Ortalama
1	2120	1750	1280	920,2	814,7	1376,98
2	2440	2940	872	883	936,8	1614,36
3	2800	1650	1670	1690	1690	1900
4	482,5	736,8	365	469,9	559,5	522,74
5	2190	1690	2000	1850	1920	1930
6	1420	428,1	437,5	819,7	741,7	769,4
7	713	167,4	266,5	3870	524,2	1108,22
8	160,2	2120	556,7	4600	4980	2483,38
9	279,2	407,3	713,9	4190	293,4	1176,76
10	1380	1570	1260	731,4	128,8	1014,24

11	1820	1950	1720	1640	1840	1794
12	1270	1260	1280	1260	4150	1844
13	1630	1710	1620	1650	2520	1826
14	1480	1550	1580	1410	1520	1508
15	736,9	784,2	865,1	1120	963,7	893,98
16	1980	1310	795	859,8	637,4	1116,44
17	2070	1840	1900	1740	1710	1852
18	1760	1740	2110	1370	1500	1696
19	2280	2140	1260	1780	164,9	1524,98
20	1850	1800	1950	1290	1100	1598
Ortalama	1825,188	1801,333	1444,133	2022,867	2293	1477,474
Std. Sapma	723,41477	663,431369	571,14199	1128,12979	1224,12414	464,262338

Çizelge 5.3. PSO algoritması ile İHA'ların yangına olan mesafeleri 20 yineleme

Çalıştırma	İHA1	İHA2	İHA3	İHA4	İHA5	Ortalama
1	395	383,3	391,7	385,3	401,1	391,28
2	1421,3	1437,4	1425,8	1434,6	1413,2	1426,46
3	1857,1	1857	1857,3	1857	1857,2	1857,12
4	790,9	810,2	789	810,2	800,3	800,12
5	480,9	470,4	481,7	470,4	468,8	474,44
6	1198,7	1215,1	1199,7	1215	1203,4	1206,38
7	9	6,2	7,6	7	7	7,36
8	676,6	674	676,5	674,7	675,5	675,46
9	1639,8	1640	1639,5	1640	1639,7	1639,8
10	1664,2	1664,2	1662,8	1664,2	1663,5	1663,78
11	823,5	823,6	823,8	823,8	823,6	823,66
12	88,3	88,5	88,4	88,8	88,3	88,46
13	1512,2	1511,8	1512,1	1511,8	1512,1	1512
14	1607,8	1607,8	1607,9	1607,8	1607,9	1607,84
15	822,9	828,6	830,3	828,7	828,7	827,84
16	428,2	423,5	429	422,5	424	425,44
17	1369,6	1367,5	1368,9	1367,7	1366	1367,94
18	539	538,8	539,4	538,7	538,8	538,94
19	411,7	411,8	413,3	411,8	413,5	412,42
20	763,8	763	764,7	762,9	763,8	763,64
Ortalama	925,025	926,135	925,47	926,145	924,82	925,519
Std. Sapma	550,409329	552,691578	550,569021	552,412232	550,320171	551,267901

Çizelge 5.4. AAA algoritması ile İHA'ların yangına olan mesafeleri 20 yineleme

Çalıştırma	İHA1	İHA2	İHA3	İHA4	İHA5	Ortalama
1	21	112,1	91,8	21	21	53,38
2	739,6	630,1	664,5	649	739,6	684,56
3	773,5	443	317	714,5	773,5	604,3
4	771,8	1101,3	931,4	778,5	771,8	870,96
5	980	1068,9	902,8	1144,3	980	1015,2
6	883,6	811,2	1216,9	905,6	883,6	940,18
7	95,8	127,5	119,2	117,7	95,8	111,2
8	439,5	394	423,1	448	439,5	428,82
9	2111,2	1988,4	1936,4	1564,4	2111,2	1942,32
10	982,5	852,8	797,5	438,3	982,5	810,72
11	792,8	799,5	708,5	793,7	792,8	777,46
12	502,4	154,3	358,9	367,7	502,4	377,14
13	1776,4	1872,7	1431,3	986,6	1776,4	1568,68

14	1798,2	1946,6	1614,9	1281,1	1798,2	1687,8
15	1734,2	1664,9	1733,2	1728,7	1734,2	1719,04
16	1427,6	1426,8	1426,5	1721,2	1427,6	1485,94
17	80	76,2	75,6	334,5	80	129,26
18	1752,3	1825,2	1748,7	1751,4	1752,3	1765,98
19	278,8	1148,8	706,1	1557,8	278,8	794,06
20	1580	695,3	1097,4	249,7	1580	1040,48
Ortalama	976,06	956,98	915,085	877,685	976,06	940,374
Std. Sapma	633,251792	634,934518	576,101418	552,911237	633,251792	570,858821

Çizelge 5.5. PSO algoritması ile İHA'ların yangına olan mesafeleri 50 yineleme

Çalıştırma	İHA1	İHA2	İHA3	İHA4	İHA5	Ortalama
1	1319,4	1319,4	1319,4	1319,4	1319,4	1319,4
2	1048,9	1048,9	1048,9	1048,9	1048,9	1048,9
3	1139,8	1139,8	1139,8	1139,8	1139,8	1139,8
4	381	381	381	381	381	381
5	244,3	244,3	244,3	244,3	244,3	244,3
6	530,9	530,9	530,9	530,9	530,9	530,9
7	1112,2	1112,2	1112,2	1112,2	1112,2	1112,2
8	1341,4	1341,4	1341,4	1341,4	1341,4	1341,4
9	1318,8	1318,8	1318,8	1318,8	1318,8	1318,8
10	233,2	233,3	233,2	233,3	233,2	233,24
11	1681,6	1681,6	1681,6	1681,6	1468,4	1468,4
12	334	334	334	334	334	334
13	495,3	495,3	495,3	495,3	495,3	495,3
14	69,5	69,5	69,5	69,5	69,5	69,5
15	1256,3	1256,3	1256,3	1256,3	1256,3	1256,3
16	77,7	77,7	77,7	77,7	77,7	77,7
17	1178,4	1178,4	1178,4	1178,4	1178,4	1178,4
18	571,4	571,4	571,4	571,4	571,4	571,4
19	472	472	472	472	472	472
20	702,6	702,6	702,6	702,6	702,6	702,6
Ortalama	775,435	775,44	775,435	775,44	764,775	764,777
Std. Sapma	481,465926	481,460296	481,465926	481,460296	463,30236	463,300065

Çizelge 5.6. AAA algoritması ile İHA'ların yangına olan mesafeleri 50 yineleme

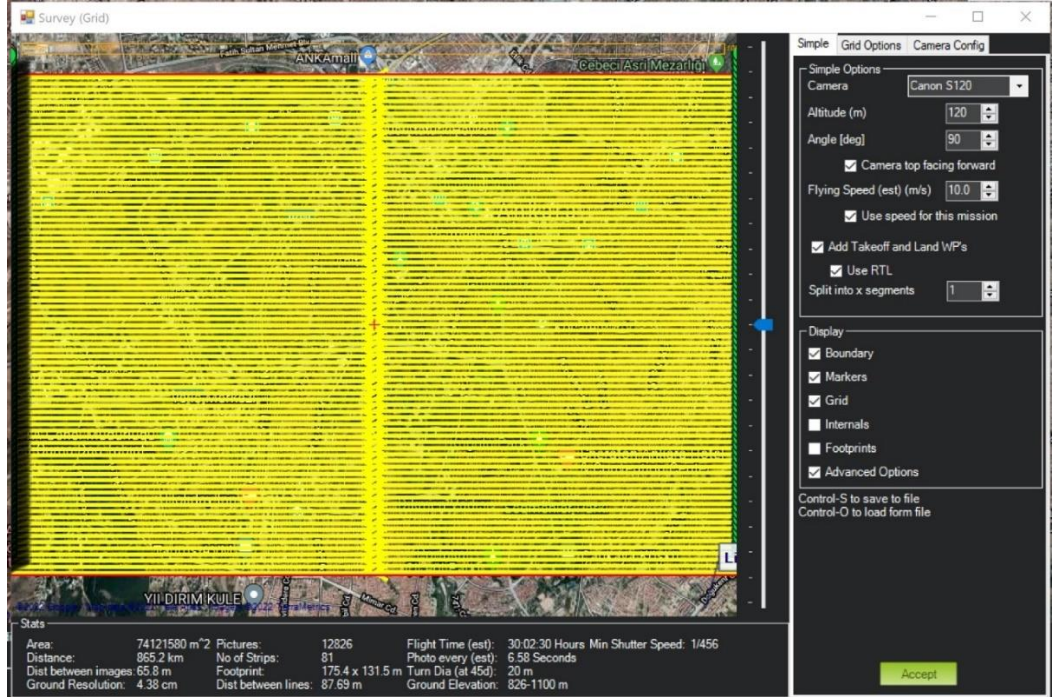
Çalıştırma	İHA1	İHA2	İHA3	İHA4	İHA5	Ortalama
1	22,6	22,1	22,2	42,7	22,9	26,5
2	10,5	9,3	9,4	37,7	11,1	15,6
3	18,2	18,2	18,3	18,1	18,1	18,18
4	18,4	18,3	18,5	18,3	18,2	18,34
5	1302,6	1025,2	1140,8	1644,6	1686,3	1359,9
6	1299,5	1022,6	1137,7	1641,5	1683,3	1356,92
7	852,3	966	896,4	896,4	896,3	901,48
8	861,4	973	875,6	962,6	901,3	914,78
9	18,9	19,7	28,5	63,6	18,7	29,88
10	10,1	11,5	9,9	8,6	9,3	9,88
11	6,5	15,5	9,7	7,3	9,4	9,68
12	7,6	7,9	0,6	2,6	0,5	3,84
13	12,3	1,6	4,2	1,2	1,9	4,24
14	12,9	3,1	4,8	2,4	1,6	4,96

15	4,2	4,2	4,2	4,2	4,2	4,2
16	13,5	1,4	1,2	1,5	1	3,72
17	14	2,7	2,5	2,9	45,3	13,48
18	2,4	2,8	2,3	2,5	45	11
19	0,3	0,2	0,2	0,2	0,7	0,32
20	0,2	0,2	0,3	0,3	0,4	0,28
Ortalama	224,42	206,275	209,365	267,96	268,775	235,359
Std. Sapma	438,701211	395,458866	405,687975	533,921334	541,014591	460,177121

Çizelge 5.7. AAA ve PSO uçuş için kat edilen yol karşılaştırma

	AAA(km)	Uçuş sür. (dk.)	PSO(km)	Uçuş sür. (dk.)
10 yineleme	18,1	30,1666667	7,2	12
20 yineleme	26,7	44,5	8,1	13,5
50 yineleme	33,2	55,3333333	8,3	13,8333333
Ortalama	26	43,3333333	7,86666667	13,1111111

Çizelge 5.1-5.2 de ortalama değerler incelendiğinde yangına yaklaşma konusunda PSO algoritması AAA'ya göre 1,45 kat daha iyi sonuç veriyor. Yineleme sayısı 20 olduğunda Çizelge 5.3-5.4'da incelendiğinde 0,98 kat daha iyi ve yineleme sayısı 50 olduğunda ise Çizelge 5.5- 5.6'da incelendiğinde 3,24 kat daha kötü sonuç verdiği gözleniyor. Çizelge 5.7'de yaklaşma için kat edilen toplam yollar incelendiğinde AAA algoritması çalıştırmalarının ortalamasında İHA'lar PSO algoritmasına göre yaklaşık 3 kat fazla yolu yine yaklaşık 3 kat fazla sürede almıştır.



Şekil 5.9 Tüm arama uzayını tarama

Bu çalışmada İHA'ların yangın tespiti için belirlenen arama uzayı 7412.16 hektardır. Bu alan Şekil 5.9'da görüldüğü gibi 120 metre irtifada %50 yan bindirme oranı ile tarama yapıldığında 865 km yol yapılacağı anlamına gelir. 865 km'lik bu yol 10 m/sn hızla giden 5 İHA'nın 6 saat uçuş yapacağı anlamına gelmektedir. Sezgisel algoritmalar kullanıldığında PSO ile ortalama 13 dakika, AAA ile ortalama 43 dakika uçuş ile yangın noktasına yaklaşılmıştır.

Sonuç olarak;

AAA algoritması kullanılarak hesaplanan yol noktalarından oluşan görevlerin İHA'lara yüklenmesi ve gönderilen komutlar ile İHA'lar yangına yaklaşırken 10 yinelemede yaklaşık 30 dakika, 20 yinelemede yaklaşık 44 dakika, 50 yinelemede yaklaşık 55 dakika uçuş süresi hesaplanmıştır.

PSO algoritması kullanılarak hesaplanan yol noktalarından oluşan görevlerin İHA'lara yüklenmesi ve gönderilen komutlar ile İHA'lar yangına yaklaşırken 10 yinelemede yaklaşık 12 dakika, 20 yinelemede yaklaşık 13 dakika, 50 yinelemede yaklaşık 14 dakika uçuş süresi hesaplanmıştır.

Standart devriye gezme yöntemi ile İHA'ların alanı tarama süresi ise Şekil 5.9'da 6 saat olduğu görüldüğünde sezgisel algoritmaların afet bölgesini bulmadaki hızı ve başarısı ortaya çıkmaktadır.

AAA ve PSO algoritmaları kıyaslandığında ise PSO ile daha kısa sürede bir yakınlaşma sağlanırken AAA ile daha yakın sonuç daha uzun sürelerde elde ediliyor. Bu noktada amaç doğrultusunda aranan yangın noktası nokta atışı bulunmak istenirse AAA, yaklaşık bir sonuç en hızlı şekilde bulunsun istenirse PSO tercih edilebilir. Ancak tüm alanın baştan sona taranması ile kıyaslandığında sezgisel algoritmalar tercih edilmelidir.

5.2. Tartışma

İHA teknolojisinin oluşturulması ve gelişim sürecinde insanlı uçuşlar için zor ve güvensiz durumlarda kullanılması prensibine dayandırılmıştır. İHA konusundaki gelişmeler ile birlikte akademik ve teknik çalışmaların birleştirilmeleriyle birlikte bu teknolojiye gerekli yapay zeka katkıları ile daha ileri bir seviyeye taşımak mümkün olacaktır. Bu gelişmelerin gerçek dünya problemleri olduğu unutulmadan gerçekleştirilecek çalışmalarla insanlı sistemler kadar hatta daha fazla güven ile uçuşlarının gerçekleştirilmesi mümkün hale gelecektir.

Akademik çalışmaların yalnızca algoritmaları test etmek ve verilen senaryodaki durumlarda kullanılabilir olduğunu söylemek problemin gerçek sonuçları düşünüldüğünde yetersiz kalmaktadır. Akademik çalışmanın sonucunun gerçeğe en yakın şekilde hatta gerçek ortamlarda yapılarak gözlemlenmesi ve algoritmaların bu şekilde ihtiyaç olunan doğrultuda çalıştırılması teknolojik ilerlemenin en önemli adımı olacaktır.



6. SONUÇLAR VE ÖNERİLER

İHA ve sürü İHA teknolojilerinin gelişimleri incelendiğinde otonom uçuş çalışmalarının ön plana çıktığı görülmektedir. İnsandan mümkün olduğunca bağımsız olarak otonom uçuşların gerçekleşmesi için uçuş rotalarının insan eli ile değil bir algoritma ile planlanması bu sürecin en makul şekilde gelişimi için gerek görülmüştür. İHA'ların sürü olarak arama kurtarma faaliyetlerinde kullanımının yaygınlaşması için yatırım maliyetlerinin düşmesi büyük önem taşımaktadır. Bu çalışmada çıkan en önemli sonuçlardan biri sezgisel algoritmaların kullanımı sayesinde daha düşük maliyetli faydalı yükler ve sistemler kullanılarak aynı sonuca ulaşmanın mümkün olduğunun gösterilmesidir.

Bu çalışmada gerçekleştirilen İHA teknolojisinin sürü zekası algoritmaları ile optimize edilmesi işlemi için PSO tabanlı bir yazılım ile AAA tabanlı bir yazılım diğer tüm koşullar sabit tutularak kıyaslanmıştır. Bu iki yazılım ile otonom uçuşlarda 5 İHA için verilen sınırlar içerisinde optimum sürede gezerek senaryo gereği hazırlanan yangın bölgesine en hızlı şekilde yaklaşması sağlanmıştır.

Bu projenin misyonu olan İHA teknolojisinin geliştirilmesi ve gerçek hayatta kullanılmasına yönelik çalışmalarda bulunulmasına en büyük katkıyı kullanılan yazılımların açık kaynak kodlu olarak tüm dünyada aktif olarak kullanılması ve ilerlemelerin kaydedilerek daha iyi seviyelere çıkarılabiliyor oluşudur. Bu bağlamda proje geliştiricileri ile akademik anlamda da açık kaynak kod kullanımının önemine tekrar değinilmiştir. Teknolojik gelişimin bir parçası olarak bu felsefenin benimsenmesinin elzem olduğu düşünülmektedir.

Daha sonraki çalışmalar için yangın söndürme problemi iki aşamalı olarak ele alınabilir. İlk aşamada bu çalışmada olduğu gibi yangın tespiti ikinci aşamada ise yangın söndürmek üzere faydalı yükü yangın söndürücüler olan İHA'lar ile yangına aktif şekilde müdahale edilmesi. Yangının kontrol altına alınması için öncelikle söndürülmesi gereken kritik noktalar ilk aşamada tespit edilerek sürüdeki söndürme işlemi yapacak olan diğer İHA'lara belirlenen koordinatlardaki görevler verilebilir.

KAYNAKLAR

- Abreo O. R., Rodriguez F. J. O., Pedraza A., R., Ramos J., B., H., Barbarosa J. G., 2021, "Backstepping control for a UAV-manipulator tuned by Cuckoo Search algorithm", Volume 147, January 2022, 103910
- Akyol S., Alataş B., 2012, "Güncel Sürü Zekâsı Optimizasyon Algoritmaları", Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi 1 (2012) 36-50
- Ardupilot Dev Team, 2021, [Choosing a Ground Station], <https://ardupilot.org/plane/docs/common-choosing-a-ground-station.html>, [Ziyaret Tarihi: 29 Mart 2022]
- Ashraf A., Majd A., Troubitsyna E., 2020, "Online Path Generation and Navigation for Swarms of UAVs", Article ID 8530763, 14 pages, doi.org/10.1155/2020/8530763
- Chen J., Ye F., Li Y., 2017, "Travelling salesman problem for UAV path planning with two parallel optimization algorithms", doi: 10.1109/PIERS-FALL.2017.8293250
- Cimino M. G. C. A., Lazzeri A., Vaglini G., 2016, "Using Differential Evolution to Improve Pheromone-based Coordination of Swarms of Drones for Collaborative Target Detection", Department of Information Engineering, Università di Pisa, Largo Lazzarino 1, Pisa, Italy
- Cimino M. G. C. A., Lega M., Monaco M., Vaglini G., 2019, "Adaptive Exploration of a UAVs Swarm for Distributed Targets Detection and Tracking", Department of Information Engineering, University of Pisa, 56122 Pisa, Italy
- Dalkılıç G., Türkmen F., 2002, "Karıncı Kolonisi Optimizasyonu", *Yüksek Performanslı Bilişim Sempozyumu*, Ocak 2002
- Engebraten S., Nummedal O. R., Gilbreath D., Yakimenko O., Glette K., 2019, "UAV swarm with mesh radios: development update", EasyChair Preprint No:1797
- Ghamry K., Zhang Y., 2016, "Cooperative Control of Multiple UAVs for ForestFire Monitoring and Detection", *12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Ağustos 2016
- Ghamry K. A., Kamel M. A., Zhang Y., 2017, "Multiple UAVs in Forest Fire Fighting Mission Using Particle Swarm Optimization", *International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA
- Goel U., Varshney S., Jain A., Maheshwari S., Shukla A., 2018, "Three Dimensional Path Planning for UAVs in Dynamic Environment using Glow-worm Swarm Optimization", *Procedia Computer Science*, Volume 133, 2018, Pages 230-239
- Gürgüze G., Türkoğlu İ., 2019, "Robot Sistemlerinde Kullanılan Algoritmalar", *Türk Doğa ve Fen Dergisi*, Cilt 8, Sayı 1, Sayfa 17-31, 2019
- Hoang V. T., Phung D. M., Dinh T. H., Ha Q., 2018, "Angle-Encoded Swarm Optimization for UAV Formation Path Planning", *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, At Madrid, Spain
- Khoufi I., Minet P., Achir N., 2016, "Unmanned Aerial Vehicles Path Planning for Area Monitoring", HAL Id: hal-01410071
- Kılıç A., 2010, Ateşi Tutan Eller- Ateş Kahramanları, *Teknik Yayıncılık*, İstanbul, Yangın Yayılımı ve Tehlikeleri
- Kitjacharoenchai P., Ventresca M., Brunese P. A., Javadi M. M., Lee S., Tanchoco J. M. A., 2019, "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach", doi: 10.1016/j.cie.2019.01.020
- Kumar T., J., Singamsetty P., 2019, "An Open Close Multiple Travelling Salesman Problem With Single Depot", DOI: 10.5267/j.dsl.2018.8.002

- Liu W., Yamazaki F., 2016, “Remote sensing technologies for post-earthquake damage assessment: a case study on the 2016 kumamoto earthquake”, *6th ASIA Conference on Earthquake Engineering (6ACEE)*, Cebu City, Philippines, 22-24 Sept 2016
- Mendes D., Ivaki N., Madeira H., 2018, “Effects of GPS Spoofing on Unmanned Aerial Vehicles”, *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*, Taiwan
- Mersheeva V., Friedrich G., 2015, “Multi-UAV monitoring with priorities and limited energy resources”, *Universitätsstraße 65–67, 9020 Klagenfurt, Austria*
- Nedjati A., Vizvari B., İzbirak G., 2015, “Post-earthquake response by small UAV helicopters”, *Natural Hazards*, Volume 80, Issue 3, pp 1669–1688
- Özoğlu B., Çakmak E., Koç T., 2019, “Clarke & Wright's savings algorithm and genetic algorithms based hybrid approach for flying sidekick traveling salesman problem”, *European Journal of Science and Technology Special Issue*, pp. 185-192, doi: 10.31590/ejosat.637816
- Öztürk Y., Tekin U., 2019, “Doğal Afetlerle Mücadelede İstanbul İli Sarıyer İlçesi Örneği”, *Dergi Park*, Cilt 2, Sayı 1, 29-41
- Parashar P., Singh S., 2017, “Socket Programming”, *Journal of Inventions in Computer Science and Communication Technology (JICSCT)*ISSN(O): 2455-5738
- Pichpibul T., Kawtummachai R., 2013, “A Heuristic Approach Based on Clarke-Wright Algorithm for Open Vehicle Routing Problem”, Article ID 874349, 11 pages, doi.org/10.1155/2013/874349
- Rauniyar A., Saputra O., Irfan M., Kim J. W., 2017, “Design and Development of a Real-Time Monitoring System for Multiple Lead–Acid Batteries Based on Internet of Things”, doi: 10.3390/fi9030028
- Sarıman G., 2011, “Veri Madenciliğinde Kümeleme Teknikleri Üzerine Bir Çalışma: K-Means ve K-Medoids Kümeleme Algoritmalarının Karşılaştırılması”, *Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü Dergisi*, 15-3(2011),192-202
- Shao Z., Yan F., Zhou Z., Zhu X., 2019, “Path Planning for Multi-UAV Formation Rendezvous Based on Distributed Cooperative Particle Swarm Optimization”, *School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China*
- Shao Y., Zhao Z., Li R., Zhou Y., 2020, “Target detection for multi-UAVs via digital pheromones and navigation algorithm in unknown environments”, *1College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China 2Zhejiang Lab, Hangzhou 311121, China*
- Shertjuk V., Zharikova M., Sokol I., 2018, “Forest Fire-Fighting Monitoring System Based on UAV team and Remote Sensing”, *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, Kyiv, Ukraine, Temmuz 2018
- Tam, A., 2021, A Gentle Introduction to Particle Swarm Optimization [online], *Machine Learning Mastery*, <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/> [Ziyaret Tarihi: 28 Mart2022].
- Uymaz S. A., 2015, “Yeni Bir Biyolojik İlhamlı Metasezgisel Optimizasyon Metodu: Yapay Alg Algoritması”, *Doktora, Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya, 101
- Wang B., Wang Z., Chen E., Chen J., Mohammed A. W., Chen Y., 2020, “Swarm Intelligence Application to UAV Aided IoT Data Acquisition Deployment Optimization”, *College of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China*

- Xu X., Yuan H., Liptrott M., Trovati M., 2017, “Two Phase Heuristic Algorithm For The Multiple-Travelling Salesman Problem”, *Soft Comput* 22, 6567–6581, doi.org/10.1007/s00500-017-2705-5
- Yakıcı E., 2016, “Solving location and routing problem for UAVs”, *Computers & Industrial Engineering* Volume 102, Pages 294-301
- Yaşar C., Temurtaş H., Özyön S., 2010, “Diferansiyel Gelişim Algoritmasının Termik Birimlerden Oluşan Çevresel Ekonomik Güç Dağıtım Problemlerine Uygulanması”, 6. *Ulusal Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu (ELECO 2010)*, Bursa, Aralık 2010
- Yoon J. J., 2018, “The traveling salesman problem with multiple drones: an optimization model for last-mile delivery”, Massachusetts Institute of Technology
- Zhang X. ve Ali M., 2020, “A Bees Optimization-Based Cooperation Method for Target Searching by Swarm UAVs in Unknown Environments”, School of Information and Computers, Anhui Agricultural University, Hefei 230031, China
- Zhang J., Ren Z., Deng C., Wen B., 2019, “Adaptive fuzzy global sliding mode control for trajectory tracking of quadrotor UAVs”, Scientific Figure on ResearchGate. https://www.researchgate.net/figure/Configuration-of-the-quadrotor-UAV_fig1_333465006 [Ziyaret Tarihi: 29 Mart 2022]