

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Uçaklar için Rota ve Yakıt Optimizasyonu

**PROJE YAZARI**

Burak Beyazıt

**DANIŞMAN**

Dr. Eyüp Emre Ülkü

**İstanbul,2025**



**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Uçaklar için Rota ve Yakıt Optimizasyonu

**PROJE YAZARI**

Burak Beyazıt

170119021

**DANIŞMAN**

Dr. Eyüp Emre Ülkü

**İstanbul,2025**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencisi ……………………… nın “…………………………” başlıklı bitirme projesi çalışması, …./…./….. tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Prof. Dr. Adı SOYADI (Danışman)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Doç. Dr. Adı SOYADI (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Dr. Öğr. Üyesi Adı SOYADI (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

İçindekiler

[1. GİRİŞ 5](#_Toc193234256)

[1.2 Bitirme Projesinin Amacı 5](#_Toc193234257)

[2. Materyal ve Yöntem 6](#_Toc193234258)

[2.1 Veri Setlerinin Genel Yapısı 6](#_Toc193234259)

[2.2 Kullanılan Algoritmalar 7](#_Toc193234260)

[2.3 Başarı Kriterleri 16](#_Toc193234261)

[3. Bulgular ve Tartışma 16](#_Toc193234262)

[4. Kaynakça 16](#_Toc193234263)

Şekiller

[Şekil 1 18](#_Toc193235022)

**ÖZET**

Bu projede, sivil havacılıkta yakıt tüketimini minimize etmeyi amaçlayarak, uçuş rotalarının belirli bir hava muhalefeti noktasını dışarıda bırakacak şekilde optimize edilmesi için detaylı bir literatür taraması yapıldı; son on yılda yayınlanan çalışmalar incelenerek, grafik teorisi tabanlı algoritmalar (Dijkstra, A\*) ve meta-sezgisel yöntemler (Genetik Algoritmalar, Parçacık Sürü Optimizasyonu, Simulated Annealing) temel alınarak, mevcut gerçekleşen ve planlanan uçuş verileri üzerinden her segment için yakıt tüketimi, mesafe ve seyir süresi gibi maliyet fonksiyonları tanımlanıp bir grafik modeli oluşturuldu; Python ortamında geliştirilen ilk prototip ile simülasyonlar gerçekleştirildi ve elde edilen sonuçlar doğrultusunda modelin parametre kalibrasyonu, dinamik rota güncelleme yeteneğinin entegrasyonu ve gerçek uçuş verileri ile kapsamlı karşılaştırma yapılması gibi adımlar gelecekteki çalışmalar için planlanmıştır.

**ABSTRACT**

In this project, aiming to minimize fuel consumption in civil aviation, a detailed literature review was conducted to optimize flight paths by excluding a designated air defense point; by examining studies published over the last decade, graph theory-based algorithms (Dijkstra, A\*) and metaheuristic methods (Genetic Algorithms, Particle Swarm Optimization, Simulated Annealing) were utilized to establish a graph model that defines cost functions such as fuel consumption, distance, and flight time for each segment based on actual and planned flight data; an initial prototype was developed in Python and simulations were performed, and based on the results obtained, future work will focus on calibrating the model’s parameters, integrating dynamic route update capabilities, and conducting comprehensive comparisons with real flight data.

# GİRİŞ

Günümüz havacılık sektöründe yakıt tüketiminin minimizasyonu ekonomik ve çevresel sürdürülebilirlik açısından kritik önem taşımaktadır. Bu çalışma, yapay zekâ temelli yakıt tüketimi tahmini ve meta-sezgisel optimizasyon yöntemleriyle yakıt tüketimini minimize eden uçuş rotalarını belirlemeyi amaçlamaktadır. Çalışmanın birincil hedefi, uçuşun kalkıştan inişe kadar toplam yakıt tüketimini minimuma indirmektir. Yakıt tüketimini etkileyen parametreler oldukça karmaşık ve doğrusal olmayan ilişkiler içerdiğinden, bu hesaplamalarda derin öğrenme yöntemleri kullanılmıştır. Alternatif güzergâhların yakıt tüketimi açısından değerlendirilmesi sonucunda en uygun rota belirlenmektedir. Model, uçuş mesafesi, uçuş süresi ve hava trafik kısıtlamalarını dikkate almaktadır. Mevcut literatür ve gerçek operasyonel veriler ışığında, rotalarda esneklik sağlanmasının %5-10 arasında yakıt tasarrufu sağladığı gösterilmiştir. Kullanılan yöntemler arasında Genetik Algoritma (GA) ve Simüle Tavlama (SA) öne çıkmaktadır. GA, popülasyon tabanlı küresel arama yaparak, uçuş rotası seçeneklerini geniş ölçekte değerlendirirken; SA yöntemi, mevcut rotaları yerel olarak iyileştirmekte etkilidir. Bu çalışma sonucunda elde edilen optimizasyon modeli, havayolu şirketlerinin operasyonel karar destek sistemlerinde kullanılarak yakıt tüketimini azaltabilir ve sürdürülebilir havacılık hedeflerine katkıda bulunabilir.

## 1.2 Bitirme Projesinin Amacı

Bu projenin temel amacı, havacılık sektöründe uçuş rotalarının belirlenmesinde kullanılan geleneksel yöntemlerin ötesine geçerek, uçakların yakıt tüketimini minimize eden ve operasyonel verimliliği maksimize eden yenilikçi bir rota optimizasyon sisteminin geliştirilmesidir. Projede, uçuş planlamasında göz önünde bulundurulması gereken uçuş süresi, uçak performans parametreleri gibi çok sayıda dinamik faktör, gelişmiş matematiksel modelleme ve yapay zeka destekli algoritmalar aracılığıyla entegre edilecektir. Bu sistem, geçmiş havacılık verileri ile eğiten bir model içermekte olup, gerçek zamanlı veri akışı ve ileri düzey simülasyon teknikleri kullanarak, her uçuş için en ideal güzergahı belirleyecek, böylece yakıt tüketiminde önemli oranda tasarruf sağlarken aynı zamanda uçuş süresini ve maliyetleri düşürecektir. Ayrıca, geliştirme sürecinde kullanılacak olan makine öğrenimi ve derin öğrenme teknikleri, sistemin geçmiş verilerden öğrenerek zaman içinde daha isabetli ve öngörülebilir sonuçlar üretmesine olanak tanıyacak; böylece hava koşullarındaki ani değişikliklere ve beklenmeyen durumlara anında adaptasyon gösterebilecektir. Proje kapsamında, sürdürülebilir havacılık uygulamalarına katkıda bulunmak adına karbon emisyonlarının azaltılmasına yönelik çalışmalar da gerçekleştirilecek, bu sayede çevresel etkiler minimize edilirken, havacılık sektöründe rekabet avantajı sağlayan ekonomik bir çözüm sunulması hedeflenecektir. Bu kapsamlı yaklaşım, yalnızca uçuş rotası optimizasyonuna değil, aynı zamanda havacılık endüstrisinde sürdürülebilirlik ve çevresel duyarlılık konularına da yeni bir bakış açısı getirerek, akademik ve endüstriyel çalışmalara ilham kaynağı olacak, gelecekteki araştırmalara sağlam bir temel oluşturacak niteliktedir.

# Materyal ve Yöntem

Bu bölümde, projede kullanılan veri seti, yakıt tüketimi hesaplama yöntemi ve rota optimizasyon algoritmalarının detayları açıklanmaktadır. Veri kaynağı olarak EUROCONTROL Belgium tarafından sağlanan 2021-2023 arası EUROCONTROL denetimine giren bütün uçuşlar kullanılmıştır. EUROCONTROL, Avrupa havacılığını desteklemeye adanmış, pan-Avrupa çapında sivil-askeri bir kuruluştur. Sağlanan uçuş verilerinde, kalkış ve varış koordinatları, uçak tipi, havayolu, sefer tipi, sekans numarası, kuyruk numarası, irtifa alanları bulunmaktadır. Ortalama yakıt tüketimi aşağıdaki formül ile hesaplanmaktadır.

Yakıt Tüketimi= Base Takeoff Cost + Yakıt Çarpım Sayısı × Mesafe

Bu formül ile 2 koordinat arası yakıt tüketimi hesaplanabilmektedir ve optimizasyon çalışmalarında ana kriter haline gelmektedir.

## 2.1 Veri Setlerinin Genel Yapısı

a. Uçuş Konum Verileri

* ECTRL ID: Her uçuşa ait benzersiz tanımlayıcı.
* Sequence Number: Uçuş sırasında elde edilen verilerin sıralı numarası.
* Time Over: Belirli konum noktasının geçildiği zaman.
* Flight Level: Uçuş sırasında ölçülen irtifa (bazı kayıtlar “0” olarak görünmekte; bu, uçuş başlangıcı veya iniş aşaması gibi durumları işaret edebilir).
* Latitude & Longitude: Uçuş sırasında ölçülen coğrafi konum koordinatları.

Bu veri seti, özellikle uçuşun zaman ve mekânsal hareketlerinin izlenmesi için kullanılır. Böylece uçuş takibi, iniş-kalkış ve seyir rotası detaylı bir şekilde analiz edilebilir.

b. Uçuş Operasyon ve Planlama Verileri

Değişkenler:

* ADEP ve ADES: Sırasıyla kalkış ve varış havalimanlarının kodları.
* ADEP/ADES Latitude & Longitude: Havalimanlarının coğrafi koordinatları.
* FILED OFF BLOCK TIME & FILED ARRIVAL TIME: Uçuş planında belirtilen kalkış ve varış saatleri.
* ACTUAL OFF BLOCK TIME & ACTUAL ARRIVAL TIME: Gerçekleşen kalkış ve varış zamanları.
* AC Type, AC Operator, AC Registration: Kullanılan uçak tipi, operatör ve kayıt bilgileri.
* ICAO Flight Type, STATFOR Market Segment: Uçuşun türü (örneğin charter, all-cargo, business aviation vb.) ve pazar segmenti.
* Requested FL & Actual Distance Flown (nm): Talep edilen uçuş seviyesi ve uçuş mesafesi (deniz mili cinsinden).

Bu veri seti, uçuş planlaması ile operasyon arasındaki farkların incelenmesine olanak sağlar. Planlanan kalkış-saati ile gerçekleşen kalkış arasındaki fark, varış gecikmeleri, uçuş mesafesi ile hesaplanan seyir performansı gibi unsurlar detaylı olarak analiz edilecektir. Farklı uçuş tipleri ve operatörler arasında performans karşılaştırmaları yapılabilir. Örneğin, “All-Cargo” ile “Business Aviation” uçuşları arasındaki farklar; uçuş süresi, gecikme oranları veya seyir irtifası farklılıkları incelenecektir.

Planlanan Uçuşların dosyalanan kalkış ve varış saatleri ile gerçek zamanlar arasındaki farklar analiz edilerek gecikme sebepleri, hava trafik yoğunluğu ya da operasyonel aksaklıklar belirlenebilir.

## 2.2 Kullanılan Algoritmalar

**Grafik Tabanlı Rota Optimizasyon Yöntemleri**

Graf teorisi tabanlı algoritmalar, hava sahasını düğümler (yol noktaları veya ızgara hücreleri) ve kenarlar (iki nokta arasındaki uçuş geçişleri) şeklinde modelliyerek en kısa veya en az maliyetli yolu arar. Bu alanda en öne çıkan iki klasik yöntem Dijkstra ve A\* algoritmalarıdır.

Dijkstra Algoritması, ağırlıklı bir graf üzerinde tek kaynaklı en kısa yol hesaplayan klasik bir yöntemdir. Uçuş planlamada, her kenarın ağırlığı mesafe, tahmini yakıt tüketimi veya zaman gibi metriklerle temsil edilerek en düşük toplam maliyetli rota bulunabilir. Dijkstra algoritması her koşulda optimal çözümü bulur; ancak arama alanı çok büyükse (örneğin tüm hava sahasını ince taneli bir ızgara olarak düşünürsek) hesaplama maliyeti yüksektir. Özellikle gerçek zamanlı değişen hava şartlarında Dijkstra’yı her güncellemede sıfırdan çalıştırmak pratik olmayabilir. Yine de, literatürde Dijkstra’nın hava akımları ve sıcaklık gibi meteorolojik verileri de ağırlık hesaplarına katarak yakıt tasarrufu için kullanıldığı örnekler vardır. Örneğin, Murrieta ve arkadaşları rüzgâr bileşenlerini göz önüne alarak Dijkstra ile optimal yanal (enlem-boylam düzleminde) trajektoryayı hesaplamışlardır. Dijkstra algoritmasının temel avantajı basitliği ve optimal sonuç garantisidir; dezavantajı ise büyük ölçekli veya çok boyutlu problemler için yavaş olabilmesi ve esnek olmamasıdır. [1]

A (A-Star) Algoritması\*: Dijkstra’nın genişletilmiş bir versiyonu olan A\*, bir heuristic (sezgisel) fonksiyon kullanarak aramayı hedefe doğru yönlendirir ve gereksiz düğüm incelemelerini azaltır. A\* algoritması ilk kez 1968’de Hart, Nilsson ve Raphael tarafından tanıtılmış ve en kısa yol aramada devrim niteliğinde olmuştur. Uçuş rotası planlamada A\* sıkça tercih edilir; çünkü uygun bir heuristik ile, en iyi rotayı bulurken çok daha az düğüm inceler ve bu da hesaplamayı hızlandırır. Örneğin Xie ve Zhong (2016) çalışmalarında algoritmalarını A\* temeliyle geliştirmiş, maliyet fonksiyonunu uçuş yakıt tüketimini ve hava şartlarını dikkate alacak şekilde özelleştirerek “en az enerji harcayan” rotayı bulmayı hedeflemişlerdir​ [2]

A\* algoritması, doğru tasarlandığında olası en güvenli ve yakıt verimli güzergâhı bularak olumsuz hava hücrelerinden ve yasaklı bölgelerden başarıyla kaçınabilmektedir​. [2]

Bir çalışmada A\* ile hesaplanan rotanın, kalkış-varış arasında fırtına hücrelerini dolanırken toplam mesafeyi ve yakıt harcamasını en aza indirdiği gösterilmiştir. A\*’nın heuristic fonksiyonu tipik olarak hedefe kalan düz mesafe gibi bir değer kullanır; ancak uçuş planlamada daha sofistike heuristic’ler de kullanılabilir (örneğin kalan mesafe + beklenen rüzgâr etkisi). A\* algoritmasının en büyük avantajı hız ve optimuma yakın çözümler sunmasıdır. Dezavantajı ise, heuristic fonksiyonunun her zaman “tutarlı” olması gerektiğidir – eğer heuristic hedefe kalan maliyeti gerçekte olduğundan küçük tahmin edemezse, A\* bazen optimal olmayan bir çözüm üretebilir veya arama verimi düşer. Ayrıca, A\* da bir graf modeline ihtiyaç duyar; hava sahasını sürekli bir alan yerine sonlu düğümlerle temsil etmek gerekir ki bu da çözümün doğruluğunu ızgara çözünürlüğüne bağımlı kılar. Xie ve Zhong’un çalışmasında, A\*’yı uçuş ortamına uyarlamak için maliyet fonksiyonuna farklı terimler eklendi: toplam yol mesafesi, uçağın dönüş açılarının keskinliği ve planlanan rotadan sapma miktarı gibi faktörler maliyete dahil edilerek, hava sahası kısıtlarıyla uyumlu ve daha uçulabilir rotalar elde edilmiştir. Bu sayede, A\* yalnızca en kısa değil, aynı zamanda güvenli dönüş limitlerini aşmayan ve orijinal plana çok uzaklaşmayan rotalar önerecek şekilde düzenlenmiştir. [2]

Graf tabanlı yöntemler Python ortamında kolaylıkla uygulanabilir. Örneğin, Python’un networkx kütüphanesi Dijkstra ve A\* gibi algoritmaları grafik üzerinde çalıştırma imkânı sunar. Hava sahasını bir düğüm-ağ modeline indirgeyip kenar ağırlıklarını (mesafe, yakıt veya risk) tanımladıktan sonra bu algoritmalar ile en iyi rotayı bulmak mümkündür. Özetle, Dijkstra ve A\* gibi algoritmalar optimal çözümler sunmalarıyla değerli olup, doğru kurgulandıklarında hava durumunu ve uçuş kısıtlarını da göz önüne alacak şekilde uyarlanabilirler​. Ancak problem boyutu büyüdüğünde, belirsizlik unsurları eklendiğinde ve sivil havacılık regülasyonları hesaba katıldığında, bu yöntemlerin hesaplama maliyeti yükselebilir ve daha esnek yöntemlere ihtiyaç duyulabilir.

**Meta-Sezgisel Rota Optimizasyon Yöntemleri**

Klasik graf arama yöntemleri karmaşık ve çok kısıtlı problemleri çözmede zorlandığında, meta-sezgisel algoritmalar devreye girer. Meta-sezgisel yöntemler, kesin optimal garantisi olmasa da geniş çözüm uzaylarında iyi çözümleri makul sürede bulabilirler. Son 10 yılda, uçuş rotası optimizasyonu problemlerine Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Karınca Kolonisi Optimizasyonu (ACO) ve Benzetilmiş Tavlama (SA) gibi tekniklerin başarılı uygulamaları görülmüştür. Bu yöntemler Python ile kolaylıkla prototiplenebilir.

Genetik Algoritmalar, doğadan esinlenen, rotaları bir “birey” (kromozom) şeklinde kodlayıp evrimsel işlemlerle iyileştirir. Bir uçuş rotası genellikle bir dizi coğrafi noktayı temsil edecek şekilde dizilenebilir. GA, seçilim, çaprazlama ve mutasyon operatörlerini kullanarak yeni rota çözümleri üretir ve bunları yakıt tüketimi, mesafe, risk gibi bir uygunluk değeri ile değerlendirir. Literatürde GA, özellikle çok kriterli optimizasyon gerektiren durumlarda öne çıkmaktadır. Örneğin, Ahn ve Ramakrishna değişken uzunlukta rotaları kromozom olarak kodlayan bir GA geliştirmiş ve farklı yakıt tüketimi ve güvenlik kriterlerini aynı anda optimize etmiştir. GA’nın önemli bir avantajı, küresel optimuma yakın çözümler bulmadaki başarısıdır – yerel takılmaları aşmak için popülasyon çeşitliliğini kullanır. [3] [4] [4] Nitekim, bir çalışmada dinamik ortamlarda GA ile planlama yapıldığında, GA’nın yakıt tüketimini ve irtifa değişimlerini azaltıp uçuş güvenliğini artıran rotalar bulabildiği, klasik yöntemlerin (örn. sadece rüzgârı hesaba katan bir planlama) göremediği alternatifleri keşfedebildiği belirtilmiştir. Hatta mobil robotlar üzerinde yapılan bir karşılaştırmada GA, geleneksel A\* algoritmasına kıyasla hem yol mesafesi hem de hesaplama süresi bakımından daha iyi performans göstermiş; GA’nın yerel minimumlara takılmadan daha kısa rotalar bulabildiği rapor edilmiştir. Uçuş planlamada GA’nın esnekliği, hava sahası kısıtlarını (örn. belirli noktalardan geçme zorunluluğu veya yasaklı bölgeler) genoma dahil etme imkânı sunar. GA tabanlı yaklaşımlar, uygun biçimde paralelleştirildiğinde gerçek zamanlıya yakın çözümler dahi üretebilir. Dezavantaj olarak, GA’nın performansı parametre ayarlarına hassastır – popülasyon boyutu, mutasyon oranı gibi değerlerin problemi iyi yansıtacak şekilde seçilmesi gerekir. Ayrıca GA kesin optimal garanti vermez; yeterli süre verilmezse bulunan çözüm alt-optimal kalabilir. Son yıllarda literatürde GA’nın yerel optimuma sıkışma eğilimini azaltmak için çeşitli iyileştirmeler önerilmiştir. Örneğin, bir çalışmada yeni bir mutasyon operatörü ile GA’nın çeşitliliği artırılmış ve uçuşların yeniden rotalanmasında yolcuların uğrayacağı rahatsızlık maliyeti de göz önüne alınmıştır. Bu iyileştirilmiş GA, standart GA’ya göre daha iyi çözümlere daha hızlı ulaşabilmiştir. GA yöntemleri Python ile kolayca uygulanabilir ve karmaşık uçuş kısıtlarını dahil etmek için uygundur; ancak sonuçların güvenilir olması için kapsamlı test ve parametre ayarı yapmak gerekir.

Parçacık Sürü Optimizasyonu (PSO): PSO da doğadan (kuş sürüsü hareketlerinden) esinlenmiş bir yöntemdir. Bir çözümü temsil eden “parçacıklar”, çözüm uzayında uçarak hareket eder ve her adımda kendi tecrübeleri ve sürünün en iyi tecrübesine göre konumlarını günceller. Uçuş rotası optimizasyonunda PSO, örneğin rota noktalarının enlem-boylam koordinatlarını sürekli bir uzayda optimize etmek için kullanılabilir. Murrieta-Mendoza ve Botez gibi araştırmacılar, uçuşun yanal ve dikey profilini optimize etmek üzere PSO’yu yapay arı kolonisi algoritması ile birleştirerek hibrit yaklaşımlar da denemişlerdir. [1] PSO’nun avantajı, az sayıda parametre ile hızlı bir şekilde iyi sonuçlara ulaşabilmesidir; özellikle sürekli parametre optimizasyonunda verimli olduğu bilinmektedir. Karmaşık yakıt tüketimi modelleri ve zaman kısıtlarıyla uğraşırken, PSO çok sayıda senaryoda denenmiş ve kombinatoryel problemlerde iyi sonuçlar verdiği belirtilmiştir. Ayrıca PSO, dinamik ortamlara adaptasyonu da nispeten kolay bir yöntemdir; parçacıklar ortam değişimine (örneğin yeni bir hava durumu güncellemesi) topluca tepki verebilir. Dezavantaj olarak, PSO’nun da küresel optimum garantisi yoktur ve eğer tüm parçacıklar benzer bir noktaya yakalanırsa (sürünün çeşitliliği düşerse) yerel optimumda kalabilirler. Uçuş rotası gibi rota uzunluğunun değişken olabildiği problemleri PSO ile çözmek de zordur; çünkü PSO sabit boyutlu vektörler üzerinde çalışır. Bu nedenle literatürde PSO genellikle sabit sayıda ara noktaya sahip rotaların sürekli koordinatlarını optimize etmek şeklinde uygulanır. Yine de, PSO tabanlı yaklaşımlar yakıt tüketimi optimizasyonunda başarılı sonuçlar sunmuştur. Örneğin, belirli bir sabit irtifada rüzgâr ve sıcaklık koşullarına göre en uygun Mach hız profilini seçmek için PSO kullanılarak önemli yakıt tasarrufları elde edildiği bildirilmiştir.

Karınca Kolonisi Optimizasyonu (ACO): ACO algoritması, birçok “yapay karıncanın” graf üzerinde yollar keşfedip feromon izleri bırakmasına dayanır. Bu yöntem, özellikle yol bulma problemleri için doğal bir benzetim olduğundan uçuş rota optimizasyonunda da uygulanmıştır. ACO’nun uçuş planlamaya uyarlanması, hava sahasının bir ızgara veya düğüm ağı olarak modellenmesi ve karıncaların kötü hava bölgelerini dolaşacak yollar araması şeklinde yapılır. Dijkstra veya A\* gibi tek seferde deterministik arama yapmak yerine, ACO pek çok deneme ile yolları keşfeder ve iyileştirir. Avantajı, karmaşık maliyet fonksiyonlarını rahatlıkla kullanabilmesidir; karıncalar yol alırken mesafe, yakıt, risk gibi faktörlerin birleşiminden oluşan bir maliyeti minimize etmeye çalışabilir. Ayrıca kolektif davranış sayesinde yerel engelleri aşma yeteneği yüksektir. Bazı karıncalar rastgele dolaşarak sıra dışı rotaları dener ve feromon mekanizması iyi rotaları güçlendirir. Nitekim, şiddetli hava koşullarından kaçınmak için uçaklara yeniden rota belirleme probleminde ACO uygulanmış ve toplam gecikme süresi ile yakıt tüketimi arasında dengeli çözümler elde edilmiştir. Bu çalışmada ACO ile tasarlanan alternatif uçuş yolları, mevcut yer hizmeti gecikme (ground delay) programlarına kıyasla toplam uçuş süresi ve yakıt tüketimi arasında daha iyi bir denge sağlamıştır. ACO’nun dezavantajı, çok sayıda iterasyon ve yapay karınca simülasyonu gerektirdiği için hesaplama olarak ağır olabilmesidir. Parametre ayarları (ör. feromon buharlaşma oranı, keşif davranışı) sonuca büyük etki eder; bu değerleri probleme göre kalibre etmek gerekir. Ayrıca, ACO da garantili optimum bulamaz ve yanlış ayarlarda tüm karıncalar benzer (belki suboptimal) bir rotada yoğunlaşabilir. Yine de son yıllarda ACO’nun hava trafik uygulamalarında (örn. birden fazla uçağın rotalarının birlikte optimizasyonu gibi) kullanımı artmakta ve umut verici sonuçlar alınmaktadır.

Benzetilmiş Tavlama (Simulated Annealing, SA): SA, bir çözümü adım adım rasgele değiştirip her adımda kabul/ret kararını kontrollü bir olasılıkla veren bir yöntemdir. Metal tavlama sürecini taklit ederek, başlangıçta büyük değişikliklere izin verip zamanla aramayı “soğutur”. Uçuş rotası problemlerinde SA, özellikle elde mevcut bir rotayı kısmen değiştirmek için kullanılabilir. Örneğin, halihazırdaki bir uçuş planını alıp, rotaya küçük bir sapma ekleyerek maliyet değişimini hesaplamak, SA’nın tipik bir hamlesi olabilir. Eğer değişiklik yakıt maliyetini veya gecikmeyi azaltıyorsa yeni rota kabul edilir; kötüleştiriyorsa belirli bir olasılıkla yine de kabul edilip arama çeşitliliği korunabilir. SA’nın avantajı, yöntemsel basitliği ve çok karmaşık kısıtları dahi doğrudan mevcut rotayı değiştirerek ele alabilmesidir. Ayrıca diğer metaheuristic yöntemlere göre daha az parametreye (sadece sıcaklık/soğuma parametreleri) ihtiyaç duyar. Literatürde SA, olumsuz hava koşulundan etkilenen uçuşlar için alternatif rotalar üretmede kullanılmış; başlangıçta rastgele eklenen/düzenlenen yol noktaları ile farklı rotalar test edilmiştir. Bu yaklaşım, operasyon maliyetindeki değişimi her adımda değerlendirerek daha iyi rotalara doğru ilerler. Dezavantajı ise, uygun bir soğuma planı seçilmezse algoritmanın ya çok erken tatmin olması (yerel minimumda kalması) ya da çok yavaş yakınsaması olasıdır. Ayrıca tek bir çözüm üzerinden yürüdüğü için, GA veya ACO’daki gibi popülasyon bazlı geniş arama yapmaz; bu da arama uzayını tam tarayamama riskini artırır. Bununla birlikte, pratikte SA ile hava durumu etrafından dolaşan uygulanabilir rotalar kısa sürede elde edilmiş ve bazı çalışmalar bunu gerçek zamanlı yeniden planlama sistemlerinin bir parçası olarak önermiştir.

Genel olarak meta-sezgisel yöntemlerin ortak özelliği, kesin optimum garantisi vermeden, karmaşık ve çok boyutlu problemleri çözebilme kabiliyetleridir. Bu algoritmalar, klasik yöntemlerin başarısız olduğu veya yavaş kaldığı durumlarda kabul edilebilir çözümleri hızlı üretirler. Örneğin, dinamik programlama veya A\* gibi yöntemlerin binlerce olası yön noktası ve kısıt altında çözüme ulaşamadığı durumlarda, GA veya ACO yaklaşımı ile daha kısa sürede makul bir rota bulunabilir. Nitekim, Ayo yaptığı çalışmada Dijkstra gibi kesin yöntemlerin çok sayıda ara nokta ve kısıt içeren gerçek zamanlı senaryolar için uygun olmadığını, bu yüzden alt-optimal de olsa hızlı çözüm veren meta-sezgisellerin daha pratik olduğunu vurgulamıştır. [5] Metaheuristic’lerin Python ile uygulanması da oldukça yaygındır; literatürdeki pek çok çalışma MATLAB veya Python ortamında bu algoritmaları prototiplemiştir. Örneğin, bir Python betiği ile GA kullanarak belirli bir uçağın rota noktalarını evrimleştirmek ve yakıt tüketimini simüle ederek hesaplamak mümkündür. Önemli olan, bu algoritmaların sonuçlarını doğrulamak ve gerekli olduğunda klasik yöntemlerle çapraz kontrol etmektir. Son olarak, meta-sezgisel bir yaklaşım ile elde edilen rotalar, eğer belirli kısıtları hafif ihlal ediyorsa (örneğin uçuş yolu hava sahası kurallarına tam uymuyorsa), bir uygunluk düzeltme aşaması ile operasyonel olarak uygulanabilir hale getirilebilir. Bu tür kombinasyon stratejileri, örneğin GA çıktısını A\* ile yerel olarak düzeltmek gibi, literatürde mevcuttur ve pratikte daha iyi sonuç verebilmektedir.

Uçuş rotası planlaması sadece kalkış öncesi yapılan bir işlem olmaktan çıkmıştır. Modern yaklaşımlar, uçuş boyunca rotanın dinamik olarak güncellenmesini ve anlık koşullara uyum sağlamasını öngörmektedir. Özellikle konvektif hava (fırtına hücreleri) hızla gelişip hareket edebildiğinden, kalkıştan saatler önce hazırlanan bir plan uçuşun ilerleyen safhalarında geçerliliğini yitirebilir. Bu sorunu adreslemek için araştırmacılar, gerçek zamanlı veri akışını kullanan ve uçuş esnasında çalışabilen algoritmalar geliştirmişlerdir. Dinamik rota güncellemesi için en yaygın yaklaşım, belirli aralıklarla (veya sürekli) uçuş rotasını yeniden optimize eden bir sistem tasarlamaktır. NASA tarafından geliştirilen Dynamic Weather Routes (DWR) konsepti bu alandaki öncü çalışmalardandır. DWR, havada seyreden uçuşları ve güncel radar hava verilerini sürekli analiz ederek, önceden belirlenmiş hava kaçınma rotalarının gereksiz uzun kısımlarını tespit eder ve daha kısa alternatif rota önerileri üretir​. Bu sistem her 12 saniyede bir, ilgili hava sahasındaki uçuşların güzergâhlarını kontrol edip eğer mümkünse uçağı emniyetten ödün vermeden daha kısa yoldan hedefe yönlendirecek yeni bir yol noktası dizisi hesaplar​. Aşağıdaki görsel, DWR konseptine ait bir örneği göstermektedir. Solda, kalkış öncesi fırtınaya geniş bir tampon bırakarak çizilmiş orijinal uçuş planı görülürken; sağda fırtınanın hareketinden sonra ortaya çıkan boşluğu değerlendiren ve uçuş esnasında önerilen dinamik rota (yeşil kesikli hat) görülmektedir. Bu yeni rota, uçağın fırtınaya güvenli mesafede yaklaşarak daha doğrudan uçmasını sağlamakta ve yaklaşık 5 dakika kazanç sağlamaktadır​. Bu örnek, gerçek operasyonlarda sıkça karşılaşılan “rota bayatlaması” problemini ve dinamik güncellemenin faydasını net biçimde gösterir.   
Örnek bir Dinamik Hava Rota (DWR) güncellemesi: MD-82 tipi bir uçağın Dallas – New Orleans uçuşunda, planlanan rota (sarı çizgi) fırtınayı büyük bir açıyla dolanırken, DWR sistemi tarafından uçuş sırasında önerilen yeni rota (yeşil kesikli çizgi) fırtınaya güvenli bir mesafeden daha doğrudan geçerek yaklaşık 31 dakika ve 3600 lb yakıt tasarrufu sağlamıştır. Gerçek zamanlı rota düzeltmeleri yapmak için, algoritmaların hızlı ve otonom çalışması gerekir. Grafik tabanlı yöntemlerden özellikle A\*, dinamik güncellemeye uygun olacak şekilde uyarlanabilir. Örneğin, gelen her yeni hava durumu güncellemesinde A\* algoritmasını tekrar çalıştırmak mümkündür; ancak daha verimli yaklaşımlar da vardır. D\* ve D Lite\* gibi A\* türevleri, önceki arama bilgisini kısmen koruyarak çevrim içi ortamdaki değişikliklere daha hızlı yanıt verebilir. Bu tarz algoritmalar insansız kara ve hava araçlarında dinamik engel kaçınmada başarıyla kullanılmıştır. Uçaklar için, D\* algoritması ile fırtına hücrelerinin hareketine uyum sağlayan rotalar üzerine araştırmalar mevcuttur. Bunlara ek olarak, günümüzde öğrenen yaklaşımlar da dinamik planlamaya entegre edilmektedir. Makine öğrenmesi teknikleri, belirli hava durumlarında uçuşların nasıl tepki verdiğini öğrenerek yeni rota önerilerinde bulunabilir. Örneğin, Wang ve çalışma arkadaşları terminal sahadaki tipik uçuş yollarını Random Forest, Boosting Tree gibi algoritmalarla hava durumuna bağlı olarak tahmin etmiş ve olası rotaları önceden kestirmeye çalışmışlardır. [6] Bu tür öğrenilmiş modeller, uzun süreli bir planlamayı (örneğin bir havalimanı yaklaşma paterni) gerçek zamanlı optimizasyonla birleştirmek için kullanılabilir. Ancak genel olarak bakıldığında, dinamik rota optimizasyonunda en somut kazanımlar, klasik arama algoritmalarının hızlı donanım ve güncel veri ile desteklenmesiyle elde edilmiştir. Ramée ve arkadaşlarının 2020 tarihli çalışması, dinamik güncellemenin faydasını ortaya koyan bir çerçeve sunmaktadır. Bu çalışmada araştırmacılar, çeşitli veri kaynaklarını (konvektif hava durumu, rüzgâr tahminleri, yasaklı hava sahaları vb.) otomatik olarak bir araya getiren ve sürekli güncellenen bir trajektorya optimizasyon sistemi önermiştir. [7] Çekirdek optimizasyon motoru olarak A\* algoritması kullanılmış ve hedef fonksiyon kruvaziyer safhadaki yakıt tüketimini en aza indirecek şekilde tanımlanmıştır. Böylece uçak havadayken, her yeni veri akışında A\* algoritması mevcut rotayı revize edip daha iyi bir alternatif olup olmadığına bakmaktadır. Simülasyon sonuçları oldukça olumludur: Gerçek uçuş verileriyle karşılaştırıldığında, bu sürekli güncellenen rotalar yakıt tüketiminde %8’e varan tasarruflar sağlamıştır. Bu, özellikle uzun mesafeli uçuşlarda ciddi bir kazançtır ve dinamik optimizasyonun pratik değerini gösterir. Dinamik güncelleme yaklaşımları sadece yakıt ekonomisi için değil, aynı zamanda emniyet için de kritiktir. Örneğin, rotanın uçuş esnasında kötü hava nedeniyle güncellenmesi, türbülanslı veya tehlikeli bölgelerden kaçınmayı sağlar ve bu da yolcu konforunu ve güvenliğini arttırır. Ancak bu yaklaşımların operasyonel uygulaması bazı zorluklar barındırır: Öncelikle, uçuş esnasında rota değişikliği yapmak hava trafik kontrol onayı gerektirir ve koordinasyon şarttır. Bu yüzden DWR gibi sistemler, yalnızca potansiyel tasarrufu hesaplamakla kalmaz, aynı zamanda önerilen yeni rotayı basit bir düzeltme şeklinde vermektedir. Gerçek bir uygulama olan American Airlines denemesinde, DWR sistemi her 12 saniyede bir arka planda çalışmış, tasarruf potansiyeli gören uçuşları listelemiş ve kullanıcı onayına sunmuştur​.İki yılı aşkın süre devam eden bu denemede, DWR’nin önerdiği rota düzeltmelerinin %64’ü operasyon birimlerince uygun bulunup uygulanabilir olarak değerlendirilmiştir – reddedilenlerin çoğu ise sistem kısıtlarından değil, hava trafik sıkışıklığı veya varış yaklaşma prosedürleri gibi harici faktörlerden kaynaklanmıştır​. Bu sonuç, teknolojik olarak dinamik optimizasyonun mümkün ve faydalı olduğunu, ancak entegrasyonun dikkatli yönetilmesi gerektiğini gösterir. Python ile dinamik rota güncellemesi için gerekli altyapı da mevcuttur. Gerçek zamanlı hava durumu verileri API’lar üzerinden alınabilir, uçuşun FMS (Flight Management System) planı dijital olarak takip edilip bir optimizasyon algoritmasına beslenebilir. Python’da çok iş parçacıklı veya asenkron programlama ile belirli aralıklarla (saniyelik veya dakikalık) optimizasyon tetiklenip sonuçlar değerlendirilebilir. Örneğin, networkx ile A\* hesaplanırken threading modülü ile belirli periyotlarda güncel radar verisi çekilebilir. Bu şekilde prototip bir dinamik planlama aracı akademik seviyede oluşturulabilir. Nitekim literatürdeki birçok çalışma MATLAB ortamında benzer simülasyonları gerçekleştirmiştir. Kritik olan, algoritmanın yeterince hızlı sonuç vermesi ve yeni rotanın mevcut rotaya yakın, uygulanabilir bir düzeltme olmasıdır – aksi takdirde sürekli değişen hedef noktalar hem pilotları hem kontrolörleri zorlar. Bu dengeyi sağlamak için genellikle “değişim eşiği” kullanılır; örneğin önerilecek rota mevcut rotadan en az X dakika avantaj sağlamıyorsa değişiklik önerilmez​. Sonuç olarak, dinamik rota güncelleme yaklaşımları hava taşımacılığının geleceğinde önemli bir rol oynayacak gibi görünmektedir. Gerçek zamanlı optimizasyon, 4D trajektorya yönetimi konseptinin de bir parçası olup geleceğin akıllı hava trafik sistemlerinde (SESAR ve NextGen programları gibi) kilit bir bileşen olacaktır.

Elimizde yalnızca gerçekleşmiş ve planlanmış uçuş verilerinin bulunduğu bir senaryoda, optimizasyon modellerini bu verilere dayalı geliştirmek ve doğrulamak gerekir. Veri güdümlü optimizasyon olarak adlandırılabilecek bu yaklaşımda, algoritmalar gerçek dünyadan elde edilen uçuş rotaları ve sonuçlarına göre ayarlanır. Bu adımlardan biri olan, veri Analizi ve maliyet modeli oluşturma: Gerçekleşen uçuş verileri , uçağın her fazdaki yakıt tüketimi ve uçuş zamanı hakkında zengin bilgi içerir. Bu veriler kullanılarak, mesafe, hız, irtifa ve rüzgâr gibi etkenlere bağlı yakıt tüketim modelleri oluşturulabilir. Örneğin, belirli bir uçak tipi için farklı irtifa ve hız kombinasyonlarında saatte ne kadar yakıt yakıldığı bu verilerden çıkarılabilir. Planlanan uçuş verileri (uçuş planları) ise, dispatcher veya otomatik sistemlerin başlangıçta öngördüğü rotayı gösterir. Gerçekleşen-planlanan karşılaştırması yapılarak, sapmaların nedenleri anlaşılabilir: Hava muhalefeti nedeniyle mi rota uzamış, yoksa ATC kısıtları nedeniyle mi farklı bir yol izlenmiş? Bu bilgiler optimizasyon modeline beslenir. Örneğin, eğer belirli bir bölgede çoğu uçuş planlanandan sapmışsa, muhtemelen orada bir hava sorunu yaşanmış demektir; model, bu bölgeyi yüksek maliyetli veya geçilmez olarak değerlendirmelidir.

## 2.3 Başarı Kriterleri

Projenin başarısı, belirlenen rotanın en az riskli ve en verimli hale getirilmesi, yani alternatif rotaların karşılaştırılarak optimum seçeneğin tespit edilmesine bağlıdır. Bu bağlamda, simülasyon ortamında hava savunma sistemlerinin (örneğin radar menzilleri ve füze müdahale alanları) gerçekçi ve doğru bir şekilde modellenmesi, elde edilen sonuçların gerçek dünya verilerine yakınlık göstermesi büyük önem taşımaktadır. Ayrıca, kullanılan algoritmanın, gerçek zamanlı ya da yakın gerçek zamanlı senaryolarda yeterli çözüm süresi, hesaplama verimliliği ve ölçeklenebilirlik sunması, projenin pratikte uygulanabilirliğini ortaya koyacaktır. Rotanın, belirlenen tehlike bölgelerinden ve hava savunma sistemlerinin etkilerinden etkin bir şekilde kaçınması, potansiyel risklerin doğru analiz edilip minimize edilmesi de kritik başarı kriterleri arasındadır. Bunun yanı sıra, projenin esnek yapıda olması, farklı senaryolara (örneğin değişen hava koşulları veya hareketli tehdit unsurları) adaptasyon sağlayabilmesi ve parametrelerde yapılacak değişikliklerin performansı olumlu yönde etkilemesi de değerlendirilecektir. Son olarak, kullanıcı arayüzünün anlaşılır, etkileşimli ve işlevsel olması ile projenin kapsamının, kullanılan yöntemlerin ve elde edilen sonuçların kapsamlı biçimde dokümante edilmesi de başarı kriterleri arasında yer almaktadır.

# Bulgular ve Tartışma

Bu çalışma kapsamında, son 10 yılda uçuş rota optimizasyonu ve hava muhalefeti simülasyonu alanında yapılan akademik araştırmalar detaylı bir literatür taramasına tabi tutulmuştur. Tarama sonucunda, sivil havacılıkta yakıt tüketiminin minimize edilmesi amacıyla, uçuş güzergahlarının belirli tehlike bölgelerinden kaçınarak optimize edilmesinde kullanılan grafik teorisi tabanlı yöntemler (Dijkstra, A\* algoritmaları) ile meta-sezgisel yöntemler (Genetik Algoritmalar, Parçacık Sürü Optimizasyonu, Karınca Kolonisi Optimizasyonu, Simulated Annealing) hakkında kapsamlı bilgiler elde edilmiştir. Bu yöntemlerin her birinin avantajları ve dezavantajları, çalışma boyunca ayrıntılı olarak değerlendirilmiş, özellikle mevcut uçuş verileri üzerinden model oluşturma sürecinde hangi algoritmaların uygulanabilir olduğu belirlenmiştir.

Şu ana kadar, elde edilen gerçekleşen ve planlanan uçuş verilerine dayanarak, hava muhalefeti noktası dışında en optimal rotayı hesaplayacak temel modelin kavramsal tasarımı gerçekleştirilmiştir. Python ortamında geliştirilen prototip sistem, uçuş güzergahlarının graf modeli üzerinden temsil edilmesi, maliyet fonksiyonlarının (yakıt tüketimi, mesafe, seyir süresi) tanımlanması ve belirlenen tehlike bölgesinin modele dahil edilmesi yönünde önemli adımlar atılmıştır. Böylece, mevcut uçuş planları analiz edilerek, modelin temel parametreleri ve kısıtları belirlenmiş, literatürde önerilen yöntemler ışığında ilk simülasyonlar yapılmıştır.

Gelecekte yapılacak çalışmalar kapsamında, geliştirilen modelin parametre kalibrasyonunun gerçekleştirilmesi, simülasyon sonuçlarının detaylı olarak analiz edilmesi ve elde edilen çıktının gerçek uçuş verileri ile karşılaştırılması planlanmaktadır. Ayrıca, modelin dinamik rota güncelleme yeteneğinin güçlendirilmesi amacıyla, gerçek zamanlı veri akışı entegrasyonu ve otomatik adaptasyon mekanizmalarının eklenmesi hedeflenmektedir. Bu aşamada, optimize edilen rotaların operasyonel uygulanabilirliğinin artırılması ve sivil havacılık operasyonlarında yakıt verimliliği ile güvenlik kriterlerinin daha etkin bir şekilde sağlanması amaçlanmaktadır.

metin, ekran görüntüsü, yazılım, harita içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 1

Yukarıdaki şekilde Londra – New York uçuşunda bir hava mualefeti noktasına karşın bir C noktası oluşturarak, yakıt tüketimi testleri sağlanmıştır. Bu C noktasında %1,2 lik fazlalık yakıt tüketimi sağlanarak mualefet noktasından kırılım sağlanmıştır.

# Kaynakça

|  |  |
| --- | --- |
| [1] | A. R. C. &. B. R. M. Murrieta-Mendoza, «3D Cruise Trajectory Optimization Inspired by a Shortest Path Algorithm,» *Aerospace,* p. 7, 2020. |
| [2] | Z. &. Z. Z. W. Xie, «Aircraft Path Planning under Adverse Weather Conditions,» %1 içinde *MATEC Web of Conferences*, 2016. |
| [3] | R. S. R. C. Ahn, «A genetic algorithm for shortest path routing problem and the sizing of populations,» no. 1583 - 1584, 2002. |
| [4] |  |
| [5] | B. S. Ayo, «An Improved Genetic Algorithm for Flight Path Re-Routes with Reduced Passenger Impact.,» *Journal of Computer and Communications,* no. 5, pp. 65-75, 2017. |
| [6] | G. W. a. b. d. e. ,. P. L. b. d. e. ,. H. W. c. ,. M. Z. a. ,. X. L. a. Zhigang Sun a b d e, «An improved random forest based on the classification accuracy and correlation measurement of decision trees,» *Expert Systems with Applications,* 2024. |
| [7] | C. K. J. D. M. J. C. B. S. &. M. D. Ramée, «Aircraft Flight Plan Optimization with Dynamic Weather and Airspace Constraints,» %1 içinde *Proceedings of ICRAT 2020*, 2020. |