

Drone ile Teslimat Sistemi

Muhammed Abdullah Acar
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
221307038
acar167257@gmail.com

Hakan Aytuğ Fırat
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
221307015
hakanaytugfirat@gmail.com

Mehmet Kordon
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
221307022
mehmetkordon09@gmail.com

Özetçe—Bu rapor, drone ile teslimat optimizasyonu problemini ele almıştır. Amaç birden fazla drone kullanarak teslimatların en verimli şekilde gerçekleştirilmesidir. Teslimatlar sırasında uçuşa yasak bölgeler(No-Fly Zone), drone batarya kapasiteleri, yük taşıma kısıtları ve teslimat öncelikleri ele alınmıştır. Çalışmada iki farklı algoritma uygulanmıştır. A* algoritması ve Genetik Algoritma (GA). Her iki algorithma aynı şartlarda farklı senaryolar ile test edilip tamamlanan teslimat yüzdesi, ortalama enerji tüketimi ve algoritma çalışma süresi gibi metrikler karşılaştırılmıştır. Karşılaştırmalar rastgele veri üretimi ve hazır veri seti üzerinden yapılmıştır. Teslimat rotaları matplotlib librarysiyle görselleştirilmiştir.

I. GİRİŞ

Günümüzde e-ticaretin yaygınlaşmasıyla beraber hızlı teslimat ihtiyacı önemli bir noktaya ulaşmıştır. Drone tabanlı teslimat sistemleri araştırma gittikçe artmıştır. Özellikle şehir içinde kısıtlı ortamlarda dronelerin verimli yönetimi, rotaların planlanması ve uçuş güvenliğinin sağlanması gerekmektedir. Bu proje kapsamında, **drone filo optimizasyonu** problemi ele alınmıştır. Çalışmada, **teslimat noktalarının konumu, paket ağırlıkları, drone ağırlık kapasiteleri, uçuşa yasak bölgeler(No-Fly Zone)** dikkate alınarak, iki algoritma modeline göre de teslimat rotalarının bulunması hedeflenmiştir. Projede hem **A* algoritması** hem de **Genetik Algoritma(GA)** uygulanarak, farklı algoritmaların performansları karşılaştırılmıştır. Ayrıca batarya kapasitesi, yük kısıtları ve zaman pencereleri gibi ek kısıtlamalar da göz önünde bulundurulmuştur. Geliştirilen sistemde **random veri üretici ve sabit veri seti(dataset.json)** kullanılarak farklı senaryolar üzerinde testler gerçekleştirilmiştir. Elde edilen rotalar görselleştirilip, algoritmaların sonuçları karşılaştırılmıştır. Uygulama dili olarak **Python** kullanılmıştır. Görselleştirmeler için **Matplotlib** kütüphanesinden yararlanılmıştır.

II. PROBLEM TANIMI

Drone filo optimizasyonu, problemi birden fazla drone kullanarak belirli bir coğrafi bölgede yer alan teslimat noktalarına en uygun rotalarla ve belirlenen kısıtlamalarla teslimat yapılmasını hedeflemektedir. Projede ele alınan temel problem, **birden fazla drone'un başlangıç noktalarından çıkarak**, teslimat noktalarına paketi ulaştırması ve mümkünse başlangıç noktasına geri dönmesidir. Bu süreçte aşağıdaki kısıtlamalar ve değişkenler dikkate alınmaktadır.

- **Drone Batarya kapasitesi:** Her drone'un belirli bir batarya kapasitesi vardır ve toplam uçuş mesafesi buna bağlıdır.
- **Taşıma kapasitesi:** Her drone'un kendine has taşıma kapasitesi vardır. Eğer paketin ağırlığı drone'un taşıma kapasitesi ağırlığını geçerse teslimat gerçekleştirilemez.
- **Teslimat Önceliği:** Her teslimat noktasına bir öncelik değeri atanmıştır ve algoritmanın önce bu noktalara teslimat yapması beklenir.
- **Uçuşa Yasak Bölgeler(No-Fly Zone):** Günümüzde nasıl askeri bölgeler gibi özel alanlar uçuşa yasaksa projede de bu noktalar göz ardı edilmemiştir. Drone bu noktalardan geçtikçe ceza puanı yemiştir.
- **Teslimat zaman penceresi:** Her teslimat için belirli bir zaman aralığı tanımlanmıştır ve teslimatın bu aralıkta yapılması istenir.

Problemin çözümünde amaç, yukarıdaki kısıtlamalara uyan ve şu hedefleri optimize eden rotaları belirlemektir:

- **Tamamlanan teslimat yüzdesini en üst düzeye çıkarmak.**
- **Toplam enerji tüketimini (batarya kullanımı) minimize etmek.**
- **Algoritma çalışma sürelerini optimize etmek.**

Bu bağlamda, farklı algoritmaların (A* ve GA*) performansı analiz edilmiştir.

III. KULLANILAN ALGORİTMALAR

III.I A* ALGORİTMASI

A* algoritması, ağırlıklı grafikler üzerinde en kısa yol bulmak için kullanılan bir yol bulma algoritmasıdır. Proje kapsamında A*, her drone için bağımsız olarak uygulanmış ve **mevcut teslimat noktaları arasında en uygun (en az maliyetli) rota** bulmaya çalışmıştır.

A* algoritmasında kullanılan **heuristic fonksiyon olarak öklidyen mesafe tercih edilmiştir**. Böylece drone'un mevcut konumdan hedef konuma teslimatı düz mesafe ile tahmin edilmiştir.

Her adımda algoritma, **drone batarya kapasitesini, yük taşıma kapasitesini ve no-fly zone bölgelerini de dikkate alarak** uygun teslimat noktalarına yönelmiştir.

III.II CSP KISITLAR

Kısıtlı Tatmin Problemi (Constraint Satisfaction Problem-CSP) yaklaşımı, çözümün yalnızca hedef fonksiyon değil, aynı zamanda belirli kısıtları da sağlaması gerektiği durumlarda uygulanır.

Proje kapsamında aşağıdaki CSP kısıtları uygulanmıştır.

- **Drone Maksimum Taşıma Kapasitesi:** Her teslimat için paket ağırlığı kontrol edilir.
- **Drone Batarya Durumu:** Uçuş için gerekli enerji kontrol edilir.
- **No-Fly Zone:** Teslimat hedefinin veya rotanın yasak bölgede olup olmadığı kontrol edilir.
- **Teslimat Zaman Penceresi:** Teslimatın Belirtilen zaman aralığında yapılması sağlanır.

A* ve GA algoritmaları bu kısıtlar doğrultusunda çalıştırılmış ve çözüm uzayı geçerli (feasible) çözümler ile sınırlandırılmıştır.

III.III Genetik Algoritma (GA)

Genetik Algoritma (GA), doğadan esinlenen popülasyon tabanlı bir optimizasyon algoritmasıdır.

Proje kapsamında GA, **teslimat sırasını global olarak optimizasyon etmek için uygulanmıştır**.

GA'da kullanılan temel yapı şunlardır:

- **Chromosome (Birey):** Teslimat noktalarının sıralanması.
- **Popülasyon:** Birden fazla olası teslimat sıralaması.
- **Fitness fonksiyonu:** Toplam yol maliyeti, teslimat öncelikleri ve no-fly zone geçiş cezalarını dikkate alarak hesaplar.
- **Crossover:** Farklı teslimat sıralamalarının kombinasyonu.
- **Mutasyon:** Teslimat sıralamasının rastgele değiştirilmesi.

GA, proje kapsamında **sadece bir drone üzerinden global teslimat sırasını optimize edecek şekilde uygulanmıştır**. Çoklu drone optimizasyonu için daha ileri seviye multi-agent GA teknikleri tercih edilebilir.

IV. SİSTEM TASARIMI VE UYGULAMA YAPISI

Projede modüler ve okunabilir bir yapı oluşturmak amacıyla sistem çeşitli dosyalara ve sınıflara ayrılmıştır. Bu sayede algoritmaların geliştirilmesi, veri yönetimi ve görselleştirme işlemleri bağımsız ve esnek bir şekilde yapılabilmektedir. Sistem **Python** dili kullanılarak geliştirilmiş ve aşağıdaki temel yapı üzerine kurulmuştur.

IV.I Proje Dosya Yapısı

- Main.py → Ana çalışma dosyası (senaryolar ve performans sonuçları)
- utils.py → Random veri üretici ve dataseti içeri aktarma
- ga.py → Genetik Algoritma implementasyonu
- a.py → A* algoritması ve CSP kısıtlarının içeri aktarılması
- Data_classes.py → Veri sınıfları (Drone, DeliveryPoint, NoflyZone)
- Dataset.json → Paylaşılan Veri Seti

IV.II Veri Üretici

Sistemde senaryoda bazı testler gerçekleştirmek için utils.py dosyasında

generate_random_drones, generate_random_deliveries, ve generate_random_noflyzones fonksiyonları geliştirilmiştir.

Bu fonksiyonlar ile her çalıştırmada farklı büyüklükte ve farklı yapıda test senaryoları oluşturulabilmektedir. Bu özellik sayesinde algoritmalar farklı durumlar için test edilip karşılaştırılmıştır.

IV.III Sabit Veri Seti (dataset.json)

Ayrıca projede hazır veri seti kullanmak için dataset.json dosyası paylaşılmıştır. Bu dosya içerisinde dronelar, teslimat noktaları ve uçuşa yasak bölgeler tanımlanmıştır. **Load_dataset()** fonksiyonu yardımıyla veri seti programa yüklenmektedir. Bu sayede algoritmalar sabit veri üzerinden de test edilip sonuçlar karşılaştırılmıştır.

V. TEST VE PERFORMANS ANALİZİ

Projede algoirtmaların başarımlarını değerlendirmek amacıyla farklı senaryolar üzerinde testler gerçekleştirilmiştir. Bu testlerde aşağıdaki metrikler kullanılmıştır.

- **Tamamlanan teslimat yüzdesi**
- **Ortalama enerji tüketimi (mAh)**
- **Algoritma çalışma süresi**

Her algoritma için 3 farklı test grubu çalıştırılmıştır.

- **Senaryo 1: 5 drone, 20 teslimat, 2 no-fly zone**
- **Senaryo 2: 10 drone, 50 teslimat, 5 no-fly zone**
- **Sabit veri seti (dataset.json)**

Testler sırasında her algoritma aynı veri üzerinde çalıştırılmış ve elde edilen sonuçlar Tablo 1., Tablo 2. ve Tablo 3'de sunulmuştur.

TABLO 1. Senaryo 1: 5 drone, 20 teslimat, 2 no-fly zone

ALGORİTMA	TAMAMLANAN	ORTALAMA ENERJİ	ÇALIŞMA SÜRESİ
A*	%53	885.81 MAH	0.01 SN
GA	%100	250.45 MAH	0.69 SN.

TABLO 2. Senaryo 2: 10 drone, 50 teslimat, 5 no-fly zone

Algoritma	Tamamlanan	Ortalama Enerji	Çalışma Süresi
A*	%26.80	705.88 mAh	0.08 saniye
GA	%100	403.13 mAh	4.13 saniye

TABLO 3. Senaryo 3: Dataset Üzerinden sonuçlar

Algoritma	Tamamlanan	Ortalama Enerji	Çalışma Süresi
A*	%71	809.11 mAh	0.01 saniye
GA	%100	313.26 mAh	0.15 saniye

Tamamlanan teslimat yüzdesi, random veri üretildiği ve **senaryoya göre no-fly zone ve drone kapasiteleri değişken olduğu** için her çalıştırmada farklı sonuçlar verebilmektedir. Bu durum beklenen ve normaldir. Bu metrik, **algoritmaların göreceli başarımlarını kıyaslamak için** kullanılmıştır. Aynı veri üzerinde her iki algoritma çalıştırıldığı için karşılaştırma anlamlıdır.

B) Enerji ve Süre Değerlendirmesi

GA algoritması, teslimat sırasını global olarak optimize ettiği için genellikle **daha yüksek tamamlanan % ve daha düşük enerji tüketimi** sağlayabilmektedir. Ancak **GA'nın çalışma süresi** A*'ye göre daha uzundur çünkü popülasyon tabanlı çalışmaktadır. A* algoritması, her drone için ayrı ayrı çalıştığı için daha hızlı sonuç verebilir ancak teslimat başarımları **GA kadar yüksek olmayabilir**. Test sonuçları bu genel davranışları doğrulamıştır.

VI. ZAMAN KARMAŞIKLIĞI VE ANALİZİ

Algoritmaların işlem süreleri, hem algoritma yapısına hem de problem büyüklüğüne göre değişmektedir. Bu projede A* ve GA algoritmaları zaman karmaşıklığı açısından farklı davranış göstermektedir.

A) A* Algoritması Zaman Karmaşıklığı

A* algoritması, her bir drone için bağımsız çalışmaktadır. Drone başına, her adımda en uygun teslimat noktası bulunmakta ve **öncelikli kuyruk (min-heap)** yapısı bulunmaktadır.

- **A* algoritmasının her adım için karmaşıklığı: $O(E \log V)$ (E: kenar sayısı V: düğüm sayısı)**
- **Ancak bu projede komple graf modellemesi yerine uçuş mesadesi bazlı değerlendirme yapıldığı için daha basit bir yapı kullanılmaktadır. ($O(n \log n)$ civarında çalışmaktadır. (n=teslimat sayısı)**
- **Ayrıca her drone için bağımsız çalıştığı için toplam karmaşıklık: $O(d^* n \log n)$ (d= drone sayısı)**

B) GA Algoritması Karmaşıklığı

Generik Algoritma (GA), popülasyon tabanlı bir algoritmadır. Her jenerasyonda mevcut popülasyon üzerinde çaprazlama, mutasyon ve fitness hesaplamaları

yapılmaktadır.

- **Bir jenerasyon için karmaşıklık: $O(p*n)$** (p : popülasyon büyüklüğü, n =teslimat sayısı)
- **Toplam çalışma karmaşıklığı: $O(g*p*n)$** (g : jenerasyon sayısı)

A* daha hızlı ancak daha lokal kararlar verir → **her drone bağımsız çalışır.**
GA daha uzun süre çalışır ancak **global optimizasyon sağlar.**

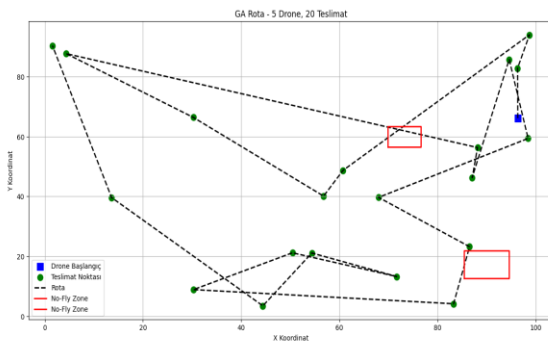
VII. GÖRSELLEŞTİRME SONUÇLARI

Projede elde edilen teslimat rotaları, algoritmaların çalışmasının daha iyi anlaşılması için Matplotlib kütüphanesi kullanılarak görselleştirilmiştir.

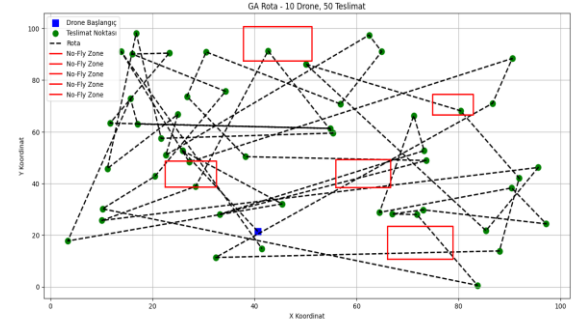
- **Drone başlangıç noktaları** → mavi kare (square) ile gösterilmiştir.
- **Teslimat noktaları** → yeşil daire (circle) ile gösterilmiştir.
- **Uçuşa yasak bölgeler (No-Fly Zone)** → kırmızı çizgilerle kapalı alan olarak gösterilmiştir.
- **Drone rotaları:**
 - **A*** için → her drone'un rotası ayrı renk ve farklı çizgi stili ile gösterilmiştir.
 - **GA** için → optimize edilmiş teslimat sırası bir rota olarak gösterilmiştir.

Görselleştirmeler, algoritmaların teslimat sıralarını ve uçuş yollarını açıkça göstermekte ve **No-Fly Zone** etkisini analiz etmeye de yardımcı olmaktadır.

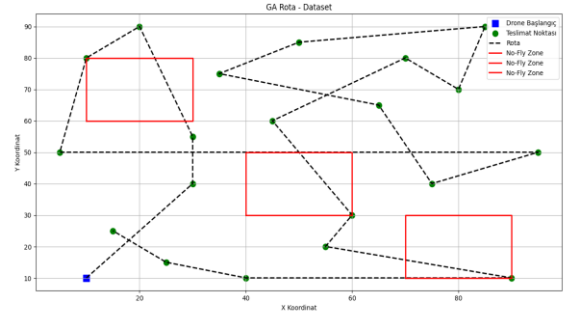
A) GA Algoritması Görselleştirilmesi



Resim 1 - Senaryo 1.

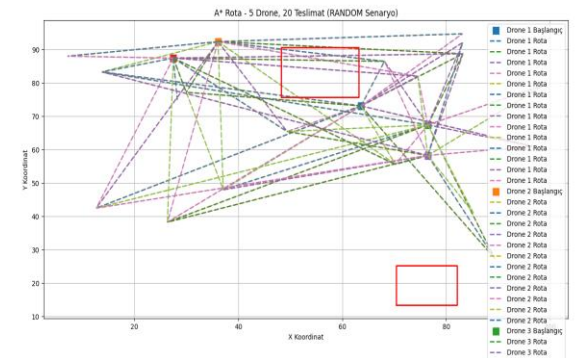


Resim 2 - Senaryo 2.

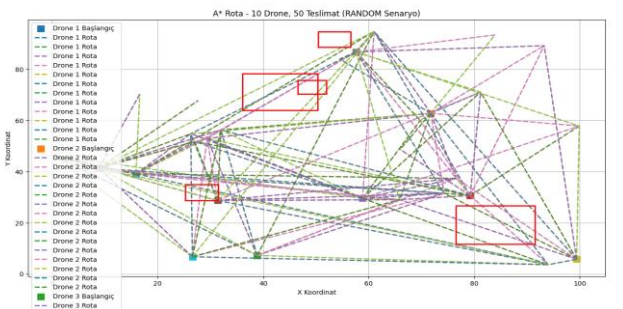


Resim 3 - Senaryo dataset verisi

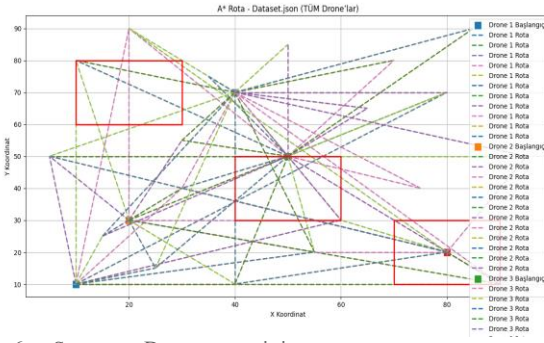
B) A Algoritması Görselleştirilmesi



Resim 4 - Senaryo 1



Resim 5 - Senaryo 2



Resim 6 – Senaryo Dataset verisi



Resim 7 – Harita üzerinden teslimat

VIII. SONUÇ

Bu proje kapsamında, **drone filo optimizasyonu** problemi ele alınmış ve farklı algoritmalar (A*, GA) ile çözüm yolları geliştirilmiştir.

Yapılan testlerde:

- **Genetik Algoritma (GA)**, daha yüksek teslimat tamamlama yüzdesi ve daha düşük enerji tüketimi sağlamış; ancak çalışma süresi daha uzun olmuştur.
- **A* algoritması** ise daha hızlı çalışmış, ancak her drone için lokalde karar verdiği için bazı senaryolarda daha az teslimat tamamlayabilmiştir.

Ayrıca:

- **No-Fly Zone kısıtlarının** rotaları önemli ölçüde etkilediği görülmüştür.
- **Batarya ve taşıma kapasitesi kısıtlarının** da teslimat başarı oranını azalttığı gözlemlenmiştir.

Görselleştirme sonuçları algoritmaların çalışma mantığını açıkça ortaya koymuş ve farklı senaryoların etkisini analiz etmede yardımcı olmuştur.

IX. KULLANILAN KÜTÜPHANELER

Proje geliştirilirken aşağıdaki Python kütüphanelerinden yararlanılmıştır:

matplotlib → Görselleştirme için (rota ve no-fly zone çizimleri)

math → Öklidyen mesafe ve matematiksel işlemler için

heapq → A* algoritmasında öncelikli kuyruk (min-heap) için

random → Random veri üretimi için

time → Performans ölçümleri için

json → Sabit veri seti yükleme için (dataset.json)

typing → Tip ipuçları ve veri yapılarını tanımlamak için (List, Tuple vs.)

dataclasses → Veri sınıfları oluşturmak için (Drone, DeliveryPoint, NoFlyZone)

VIII. KAYNAKÇA

[1] Python Dökümantasyonu: <https://docs.python.org/3/>
<https://medium.com/dev-bits/python-minis-1-data-classes-6b795c6d8306>
<https://forum.yazbel.com/t/oklid-uzunluk-formulunu-python-ile-kodlamak/10448>
<https://www.geeksforgeeks.org/python-time-module/>
<https://medium.com/@moraneus/exploring-the-power-of-pythons-typing-library-ff32cec44981>

[2] Matplotlib: <https://matplotlib.org/>,
<https://github.com/matplotlib/matplotlib>

[3] Heap yapısı: <https://medium.com/@kevsr/stack-heap-18301ae671ec>
<https://bilgisayarkavramlari.com/2008/08/09/yigin-agaci-heap/>

[4] A* Algoritması: <https://github.com/eraykisabacak/A-Star-Algoritması-8-Bulmacası>
<https://medium.com/@adem.akdogan/a-yol-bulma-algoritması-C4%B1-ve-python-ile-uygulaması-C4%B1-fd7d89f38cb1>
<https://bilgisayarkavramlari.com/2009/03/02/a-yildiz-arama-algoritması-a-star-search-algorithm-a/>
<https://bilgisayarkavramlari.com/2009/03/02/a-yildiz-arama-algoritması-a-star-search-algorithm-a/>

[5] GA Algoritması: <https://www.geeksforgeeks.org/genetic-algorithms/>
<https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>
<https://efecanxrd.medium.com/genetic-algorithm-nedir-ne-i-CC%87%C5%9Fe-yarar-cac86be1dea4>

Github link: <https://github.com/mehmetkrdn/DroneTeslimat>

