

# Kocaeli Üniversitesi, Teknoloji Fakültesi

## Bilişim Sistemleri Mühendisliği Bölümü:

### Yazılım Geliştirme Laboratuvarı II 2024-2025

## Bahar Proje Raporu

Selin AKPOLAT  
Kocaeli Üniversitesi  
Bilişim Sistemleri  
Mühendisliği: Yazılım  
Geliştirme Laboratuvarı II  
İzmit/Kocaeli  
221307045@kocaeli.edu.tr

**Özet—** Bu çalışmada, farklı sınırlamalar altında çalışan otonom drone filoları için rota planlaması yapılmıştır. A\* arama algoritması ve Genetik Algoritma (GA) karşılaştırmalı olarak uygulanmış, enerji tüketimi, ihlal sayısı ve teslimat başarımı gibi metriklerle değerlendirilmiştir. Ayrıca Leaflet.js kullanılarak dinamik görselleştirme gerçekleştirilmiş ve başlangıç noktaları, teslimat konumları, uçuşa yasak bölgeler ve rotalar ayrıntılı şekilde harita üzerinde gösterilmiştir.

**Anahtar Kelimeler —** Drone optimizasyonu, Genetik Algoritma, A\* Arama, Leaflet.js, rota planlama, zaman karmaşıklığı, teslimat, uçuşa yasak bölge.

## I. GİRİŞ

Drone teknolojisi, dağıtım sistemlerinde devrim yaratmaktadır. Ancak, çok sayıda teslimat noktası, dinamik uçuşa yasak bölgeler, batarya kısıtları ve zaman pencereleri gibi gerçek dünya kısıtlamaları altında rota planlaması büyük bir zorluk oluşturmaktadır. Bu çalışma, çoklu drone sistemleri için optimum teslimat rotalarını bulmayı ve karşılaştırmalı olarak iki algoritmanın (A\* ve GA) performansını değerlendirmeyi hedeflemektedir.

## II. YÖNTEMLER

### A. Graf Oluşturma

Teslimat noktaları düğüm (node), hareket edilebilir yollar kenar (edge) olarak modellenmiştir. Her kenarın maliyeti şu şekilde tanımlanmıştır:

**Maliyet Fonksiyonu :**  $cost = distance \times weight + (priority \times 100)$

### B. A\* Algoritması ile Rota Bulma

A\* algoritmasında, her komşu nokta için bu değerler hesaplanarak öncelikli kuyrukta en düşük toplam maliyetli yol seçilmektedir. No-fly zone cezası aktif zaman aralığında olan bölgeleri kesen her rota için uygulanır.

**Heuristic=**  $distance + nofly\_zone\_penalty$

### C. Genetik Algoritma (GA)

GA, başlangıçta rastgele oluşturulmuş rota popülasyonlarıyla çalışır. Her birey (rota), şu fitness fonksiyonu ile değerlendirilir:

**Fitness Fonksiyonu (GA için):**

$Fitness = (tamamlanan\_teslimat \times 50) - (toplam\_enerji \times 0.1) - (ihlaller \times 1000)$

GA, aşağıdaki adımlardan oluşur:

- Popülasyon Üretimi:** random.shuffle ile teslimat ID'lerinin farklı sıralamaları oluşturulur.
- Çaprazlama (Crossover):** Order Crossover (OX) ile rota parçaları birleştirilir.
- Mutasyon:** Rastgele iki teslimat noktası yer değiştirir.
- Seçilim:** Elit %20 rota korunarak yeni popülasyon oluşturulur.
- Seçilim:** Elit %20 rota korunarak yeni popülasyon oluşturulur.

### D. Rastgele Veri Üretimi

Senaryo 2 için aşağıdaki rastgele üreticiler tanımlanmıştır:

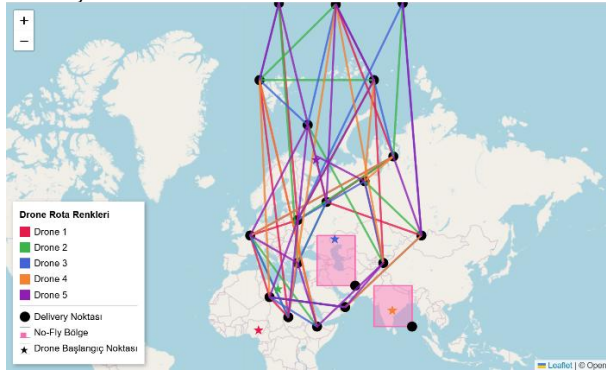
- generate\_random\_drones(n):** Her biri farklı başlangıç noktasına ve batarya/sürat/kapasite bilgisine sahip n drone üretir.
- generate\_random\_deliveries(m):** m adet rastgele koordinatta, farklı ağırlıklarda ve öncelik seviyelerinde teslimat üretir.
- generate\_random\_no\_fly\_zones(k):** Her biri (x,y) merkezli ve rastgele genişlik/yükseklik içeren, aktiflik zamanı belirli k adet no-fly bölge üretir.

### III. GÖRSELLEŞTİRME

Leaflet.js tabanlı interaktif HTML haritalar üretilmiştir. Harita üzerinde:

- **Drone rotaları:** Farklı renklerde çokgen çizgilerle gösterilir.
- **Teslimat noktaları:** Siyah daire (\u25CF)
- **Drone başlangıç noktaları:** Beyaz yıldız (★)
- **No-fly bölgeleri:** Şeffaf kırmızı dikdörtgen
- **Açıklama kutusu (legend):** Sol altta tüm semboller açıklar.

Tüm rotalar koordinat olarak generate\_leaflet\_html() fonksiyonuyla JSON'a gömülüp map\_template.html dosyasına entegre edilmiştir.



Şekil 1: Sabit veri kullanılarak yapılmış görselleştirme

### IV. KARŞILAŞTIRMALI ANALİZ: A\*, GA

#### A. Deneyisel Sonuçlar (Senaryo 1 - Sabit Veri)

Metrik	GA	A* Greedy
Tamamlanan	12/20 (60.0%)	29/100 (29.0%)
Enerji	3093.51	4259.85
İhlal	36	74
Süre	0.72 sn	0.92 sn

#### B. Deneyisel Sonuçlar (Senaryo 2 - Rastgele Veri)

Metrik	GA	A* Greedy
Tamamlanan	11/50 (22.0%)	10/500 (2.0%)
Enerji	24109.82	15222.01
İhlal	296	511
Süre	11.92 sn	37.80 sn

#### C. Yorumlar

- **A algoritması\***, her adımda en düşük maliyetli hamleyi seçse de, global optimuma ulaşamamakta ve çok sayıda ihlale neden olmaktadır.
- **Genetik algoritma**, ihlal ve enerji cezasını minimize eden daha optimal çözümler üretmektedir.
- GA, daha fazla hesaplama gerektirir ancak ihlalsiz teslimat yüzdesi açısından çok daha başarılıdır.

### V. ZAMAN KARMAŞIKLIĞI ANALİZİ

#### • A Arama:\*

- En kötü durumda  $O(b^d)$ , burada b: dal sayısı, d: hedef derinlik.

- Her teslimat noktası düğüm olduğundan ve her biri tüm komşularla bağlantılı olabileceğinden karmaşıklık hızla artar.

#### • GA:

- Her nesilde  $O(p.n)$  karmaşıklık gösterir. Burada p: popülasyon boyutu, n: rota uzunluğu.

- Tüm nesiller için toplam karmaşıklık:  $O(g.p.n)$

Örneğin: pop\_size=20, generations=10, n=20 için GA karmaşıklığı  $O(4000)$  düzeyindeyken, A\* her düğüm için yeniden çalıştırıldığında çok daha fazla işlem gerektirir.

### VI. KARŞILAŞILAN ZORLUK VE ÇÖZÜMLER

- **Uçuşa yasak bölge ihlali tespiti:** Shapely ile LineString.intersects(Polygon) kullanılarak zaman bazlı geçiş kontrolleri yapıldı.
- **Leaflet görselleştirme hataları:** Koordinat dönüşümü  $(x,y) \rightarrow (lat,lng)$  ve harita merkezleme ayarları yapıldı.
- **Matplotlib kurulum sorunları:** Yerine Leaflet.js tercih edilerek dinamik görsellik sağlandı.
- **GA'nın anlamsız çözümler üretmesi:** Drone kapasitesi ve zaman penceresi ihlalleri fitness fonksiyonuna dahil edildi.

### VII. SONUÇ

Bu çalışmada A\* ve GA algoritmaları, drone teslimat rotalarının planlanması için karşılaştırılmıştır. Sonuçlar, GA'nın ihlal oranı ve tamamlanan teslimat yüzdesi açısından daha üstün performans sergilediğini göstermektedir. GA'nın öğrenme kapasitesi sayesinde yüksek başarı elde edilirken, A\* yalnızca anlık en iyi adımı seçmesi nedeniyle sınırlı kalmıştır.

Görselleştirme aşamasında Leaflet.js kullanılarak, teslimat noktaları, no-fly bölgeleri ve rotalar interaktif şekilde harita üzerinde sunulmuştur.

### VIII. KAYNAKÇA

- [1] Drone Filo Optimizasyonu Yazlab-II Proje PDF Dokümanı (2425\_yazLab\_II\_drone\_filo.pdf)
- [2] Norvig, P., & Russell, S. (2010). *Artificial Intelligence: A Modern Approach*.
- [3] OpenStreetMap & Leaflet.js Belgeleri
- [4] Python shapely, random, json ve heapq kütüphane dokümantasyonları