

# Drone Filo Optimizasyonu: Çok Kısıtlı Ortamlarda Dinamik Teslimat Planlaması

Miray Gürbüz  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
221307031

Zeynep Yılmaz  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
221307012

Hilal Yüce  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
221307070

**Özet**—Bu rapor, Yazılım Geliştirme Laboratuvarı – II dersi kapsamında gerçekleştirilen çok kısıtlı ortamlarda dinamik teslimat planlaması projesinin detaylarını içermektedir. Projede, farklı ağırlıklara ve öncelik seviyelerine sahip teslimatların, çok sayıda drone ile en kısa sürede ve en verimli biçimde tamamlanması hedeflenmiştir. Geliştirilen sistem, operasyonel kısıtları dikkate alarak rotaları belirleyen A\* algoritması ile genetik algoritmanın performansını karşılaştırmalı olarak değerlendirmektedir. Simülasyon ortamı Python dili ile oluşturulmuş, algoritmalar çeşitli senaryolarda test edilerek başarı oranı, enerji verimliliği, çalışma süresi ve kısıt uyumluluğu gibi metrikler üzerinden analiz edilmiştir.

**Anahtar Kavramlar**— drone teslimatı, rota optimizasyonu, genetik algoritma, A\* algoritması, CSP, enerji verimliliği, uçuş yasağı bölgeleri, zaman karmaşıklığı

## I. GİRİŞ

Bu proje kapsamında farklı ağırlık ve öncelik seviyelerine sahip teslimatları, sınırlı batarya kapasitesine sahip çok sayıda drone ile en verimli şekilde gerçekleştirecek bir rota optimizasyon sistemi tasarlanmıştır. Tasarlanan sistem, uçuş yasağı bölgeleri zaman pencereleri, enerji tüketimi ve taşıma kapasitesi gibi çeşitli dinamik kısıtları göz önünde bulundurmaktadır.

Projede iki farklı yaklaşım karşılaştırmalı olarak uygulanmıştır:

- A\* algoritması ve *Constraint Satisfaction Problem (CSP)* ile görev atama doğruluğu ve ihlal analizi yapılarak en kısa ve geçerli rotaların aranması.
- Genetik Algoritma* ile karmaşık durumlarda yaklaşık çözüm üretimi.

Bu iki yaklaşım iki farklı test senaryosunda değerlendirilmiş ve çalışma süresi, enerji tüketimi, tamamlanma oranı gibi performans metrikleri üzerinden analiz edilmiştir.

## II. PROBLEM TANIMI

Teslimat hizmeti sunan bir lojistik firma, farklı ağırlık, öncelik ve zaman aralığına sahip onlarca paketi, belirli bir bölgedeki alıcılara drone'lar aracılığıyla ulaştırmak istemektedir. Ancak bu sistemin sorunsuz işlemesi için çeşitli operasyonel ve çevresel kısıtların dikkate alınması gerekmektedir. Bu kısıtlar, geleneksel rota planlama algoritmalarının yeterli olmamasına neden olur. Bu projede ele alınan temel zorluklar şunlardır:

- Enerji kısıtları:** Drone'lar belirli bir batarya kapasitesi ile çalışır. Her uçuşta harcanan enerji, drone'un taşıdığı yük ve mesafeye bağlı olarak

değişir. Enerjisi yetmeyen dronelar, görev öncesinde yeniden şarj olmalıdır.

- Ağırlık sınırlamaları:** Her drone, maksimum taşıma kapasitesine sahiptir. Bu kapasiteyi aşan teslimatlar, atama sırasında doğrudan elenir.
- Zaman pencereleri:** Teslimatların kabul edilebilir olduğu belirli zaman aralıkları bulunur. Drone'lar, bu pencerelere uymayan teslimatları gerçekleştiremez veya ceza alır.
- Uçuş yasaklı bölgeler:** Bazı bölgelerde dronelerin uçuşması güvenlik, yasal ya da teknik nedenlerle yasaktır. Bu alanlara giren rotalar sistem tarafından cezalandırılır veya engellenir.

Bu çoklu kısıtlar altında en fazla teslimatı başarıyla gerçekleştirecek ve toplam enerji tüketimini minimumda tutacak rotaların ve görev atamalarının planlanması gerekmektedir.

## III. TEMEL VERİ YAPILARI VE MODELLER

Bu projede üç temel veri yapısı kullanılmıştır: *Drone*, *Teslimat Noktası (Delivery)* ve *Uçuş Yasaklı Bölge (No-Fly Zone)*. Tüm yapılar Python `dataclass[1]` yapısıyla tanımlanmış ve algoritmalara doğrudan entegre edilmiştir.

### A. Drone Yapısı

Her drone, başlangıç noktasından hareket ederek belirli bir batarya kapasitesi ve maksimum taşıma ağırlığı ile teslimat yapmaktadır.

Alan	Drone Özellikleri	
	Açıklama	Veri Tipi
id	Benzersiz kimlik	int
max_weight	Taşıma kapasitesi	float
battery	Batarya kapasitesi	int
speed	Uçuş hızı	float
start_pos	Başlangıç konumu	Tuple[float, float]

Tablo 1: Drone Özellikleri Tablosu

Ayrıca; *current\_battery*, *current\_pos*, *recharge\_rate* gibi dinamik alanlar da takip edilir.

### B. Teslimat Noktası Yapısı

Her bir teslimat noktası bir paketi ifade etmektedir. Her biri; konuma, ağırlığa, önceliğe ve zaman aralığına sahiptir.

Alan	Teslimat Noktası Özellikleri	
	Açıklama	Veri Tipi
id	Benzersiz kimlik	int
weight	Paket ağırlığı	float
priority	Teslimat önceliği	int
time_window	Kabul edilebilir zaman aralığı	int
pos	Konum	Tuple[float, float]

Tablo 2: Teslimat Noktası Özellikleri Tablosu

### C. Uçuşa Yasaklı Bölge (No-Fly Zone) Yapısı

Bu yapılar, dronelerin geçemeyeceği alanları tanımlamaktadır. Aktif olduğu zamanlarda bu bölgelerin içine giren yollar geçersiz sayılır veya cezalandırılır.

Alan	Uçuşa Yasaklı Bölge Özellikleri	
	Açıklama	Veri Tipi
id	Benzersiz kimlik	int
active_time	Bölgenin aktif olduğu zaman aralığı	int
coordinates	Köşe noktaları listesi	Tuple[float, float]

Tablo 3: Uçuşa Yasaklı Bölge Özellikleri Tablosu

No-fly zone kesişimleri shapely[2] kütüphanesiyle polygon-intersect şeklinde kontrol edilmektedir.

Bu veri yapıları, algoritmaların ( $A^*$  + CSP, GA) ortak kullanabileceği biçimde tasarlanmıştır. Böylece hem rota planlama hem de kısıt kontrolü süreçleri tek bir yapı üzerinden sağlanmıştır.

## IV. ALGORİTMA TASARIMI

Projede iki farklı yaklaşım uygulanmış ve karşılaştırılmıştır:

- $A^*$  + CSP
- Genetik Algoritma

Algoritmalar; aynı veri yapıları ve kısıtlar üzerinde çalışacak şekilde tasarlanmış, sonuçları ortak metriklerle analiz edilmiştir.

### A. Graf Modeli

Teslimat sistemi graf modeli olarak temsil edilmektedir.

- **Düğüm**ler: Teslimat noktaları.
- **Kenar**lar: Drone'un bir teslimat noktasından diğerine gidebileceği yollar.

Her kenar için bir maliyet fonksiyonu hesaplanmaktadır:

$$maliyet = uzaklık \times yük + (öncelik \times 100)$$

Bu fonksiyon, hem fiziksel yükü hem de teslimatın önceliğini hesaba katmaktadır.

### B. $A^*$ Algoritması ile Rota Planlama

$A^*$  algoritması, başlangıç noktasından başlayarak en uygun teslimat yolunu bulmayı hedefler.[3][4]

Her bir rota arama işlemi *find\_path()* fonksiyonu aracılığıyla yürütülmüştür. Bu fonksiyon, başlangıç konumu ve hedef teslimat noktalarını giriş olarak almış, ve hedef noktaların tümüne ulaşan en uygun ziyaret sırasını üretmiştir.

Her olası geçiş için, önce *is\_feasible\_delivery()* fonksiyonu çağrılarak ilgili teslimat noktasının drone için uygunluğu kontrol edilmiştir. Bu uygunluk değerlendirmesi; ağırlık limiti, teslimat zaman aralığı, batarya yeterliliği ve aktif uçuş yasağı bölgeleri göz önünde bulundurularak yapılmıştır. Geçerli olan teslimatlar için *calculate\_actual\_cost()* fonksiyonu kullanılarak geçiş maliyeti (mesafe, zaman ve enerji tüketimi) hesaplanmış; ayrıca teslimat önceliğine bağlı olarak maliyette düzeltme uygulanmıştır.

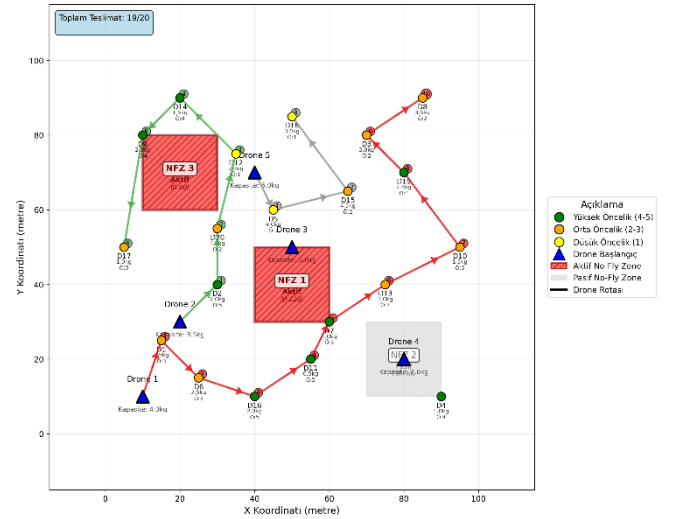
$f(n) = g(n) + h(n)$  biçimindeki klasik  $A^*$  değerlendirme formülünde;

- $g(n)$  değeri, o ana kadar kat edilen toplam maliyeti temsil etmiş,
- $h(n)$  değeri ise, heuristic() fonksiyonu ile kalan teslimatlar arasındaki en düşük mesafe ve aktif no-fly zone cezası dikkate alınarak tahmin edilmiştir.

Açık liste (*open\_set*) ve kapalı liste (*closed\_set*) yapıları kullanılarak düğüm genişletme işlemi gerçekleştirilmiş, aynı duruma daha düşük g-score ile ulaşılması durumunda güncellemeler yapılmıştır. Algoritma, genişletilen düğüm sayısı önceden belirlenmiş bir eşik değerini aştığında hedefe ulaşamamışsa *\_greedy\_fallback()* fonksiyonu devreye alınmış ve bu sayede en yakın geçerli teslimat noktalarına yönelen sezgisel bir çözüm uygulanmıştır.

Algoritmanın sonunda elde edilen rota üzerinden *get\_path\_statistics()* fonksiyonu aracılığıyla toplam mesafe, toplam süre, enerji tüketimi, ortalama öncelik skoru gibi performans metrikleri hesaplanmıştır. böylece geliştirilen sistem, hem operasyonel kısıtları hem de teslimat önceliklerini dikkate alan kısıt temelli bir rota planlama çözümü sunmuştur.

Drone Teslimat Rotaları - A\_STAR Algoritması



Görsel 1:  $A^*$  Algoritması - Örnek Veri Seti Drone Teslimat Rotaları

Görsel 1'de örnek veri seti kullanılarak (5 drone, 20 teslimat, 3 no-fly zone)  $A^*$  algoritması ile elde edilen teslimat rotaları gösterilmektedir. Toplamda 20 teslimattan 19'u başarıyla tamamlanmıştır. Her bir drone için başlangıç noktası, rotası ve teslimat noktaları gösterilmiştir. Dronelerin rotaları uçuşa yasaklı bölgelerle çakışmadan planlanmış olup, algoritmanın no-fly zone kısıtlarını dikkate aldığı gözlemlenmiştir.

### C. CSP (Constraint Satisfaction Problem) [5]

CSP, teslimat-drone eşleşmelerini kısıtlara göre gerçekleştirir:

- **Ağırlık:** Drone taşıma kapasitesini aşamaz
- **Zaman Penceresi:** Drone zaman aralığı dışında varamaz
- **Batarya:** Yetersiz enerji varsa teslimat yapılamaz

Bu projede, CSP yaklaşımı A algoritması ile entegre edilerek\* teslimat planlamasında hem rota hem de görev atamalarının kısıt temelli biçimde gerçekleştirilmesi sağlanmıştır.

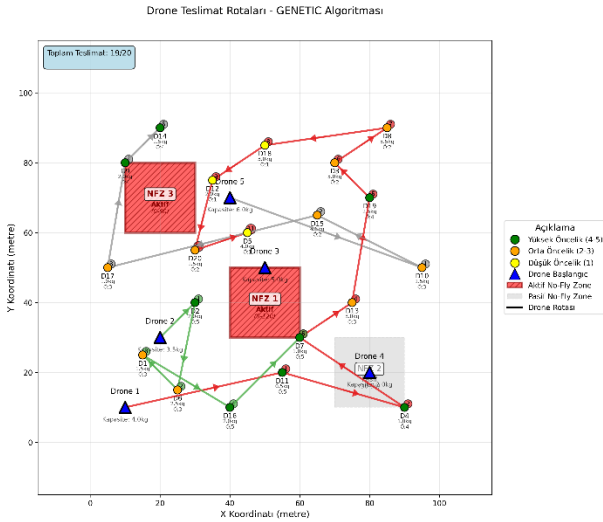
### D. Genetik Algoritma (GA)

Genetik Algoritma (GA), biyolojik evrim sürecinden esinlenen sezgisel bir optimizasyon ve arama algoritmasıdır. Karmaşık ve büyük çözüm uzaylarına sahip problemlerde, klasik algoritmaların zorlandığı durumlarda etkili çözümler sunar. Bu algoritma, doğal seçim, çaprazlama (crossover) ve mutasyon (mutation) gibi evrimsel mekanizmaları taklit eder. [6]

- **Birey**
- **Başlangıç Popülasyonu**
- **Çaprazlama (Crossover):** İki bireyin rotaları karıştırılarak yeni birey oluşturulur.
- **Mutasyon:** Rastgele teslimat değişimi yapılır. (swap/relocate)
- **Fitness Fonksiyonu:**  

$$fitness = (teslimat\ sayısı \times 50) - (enerji \times 0.1) - (kısıt\ ihlali \times 100)$$

Yüksek teslimat sayısı ödüllendirilir, ihlaller ve fazla enerji tüketimi cezalandırılır.



Görsel 2: Genetik Algoritma - Örnek Veri Seti Drone Teslimat Rotaları

Görsel 2'de örnek veri seti kullanılarak (5 drone, 20 teslimat, 3 no-fly zone) genetik algoritma ile elde edilen teslimat rotaları gösterilmektedir. Genetik algoritma sonucunda elde edilen teslimat rotalarının A\* algoritmasına göre daha karmaşık ve yoğun bir yapıya sahip olduğu gözlemlenmiştir. Bu farkın temel nedenleri aşağıda özetlenmiştir:

- **Çözüm Yaklaşımı:** Genetik algoritma, çözüm uzayında global bir optimum arayışı içindedir ve bu süreçte birçok drone'un rotasını eşzamanlı olarak optimize etmeye çalışır. buna karşılık, A\* algoritması yalnızca tekil bir başlangıç ve varış noktası arasında en kısa veya en düşük maliyetli rotayı bulur.
- **Rotadaki Rastlantısallık:** Genetik algoritma içinde kullanılan crossover ve mutasyon işlemleri, bazı rotaların belirli ölçüde rastlantısal veya dolaylı olmasına neden olabilmektedir. Bu da rotaların A\*'a kıyasla daha çok kesişmesine ve üst üste binmesine sebep olmaktadır.
- **No-Fly Zone (NFZ) Duyarlılığı:** Genetik algoritma ihlalleri ceza fonksiyonlarıyla yumuşak bir şekilde engeller, bu da rotaların NFZ sınırlarına daha yakın veya karmaşık olmasına neden olabilmektedir.

### V. TEST SENARYOLARI VE PERFORMANS ANALİZİ

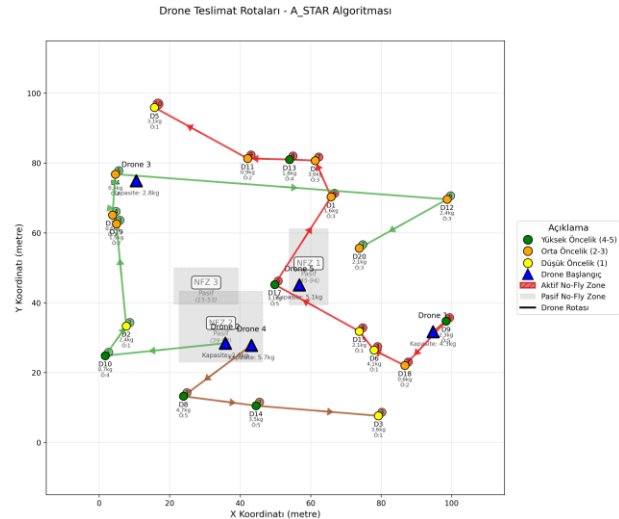
#### A. Test Senaryoları

Bu bölümde, geliştirilen algoritmalar iki farklı senaryo üzerinden test edilmiştir. Kullanılan veriler, sistem içinde yer alan veri üretici *generate\_scenario* aracılığıyla rastgele oluşturulmuştur. İki senaryoda da hem A\* algoritması hem de genetik algoritma (GA) ayrı ayrı çalıştırılmış ve elde edilen performans metrikleri karşılaştırmalı olarak analiz edilmiştir.

##### 1) Küçük Ölçekli Teslimat

- Drone Sayısı: 5
- Teslimat Noktası: 20
- No-Fly Zone: 3

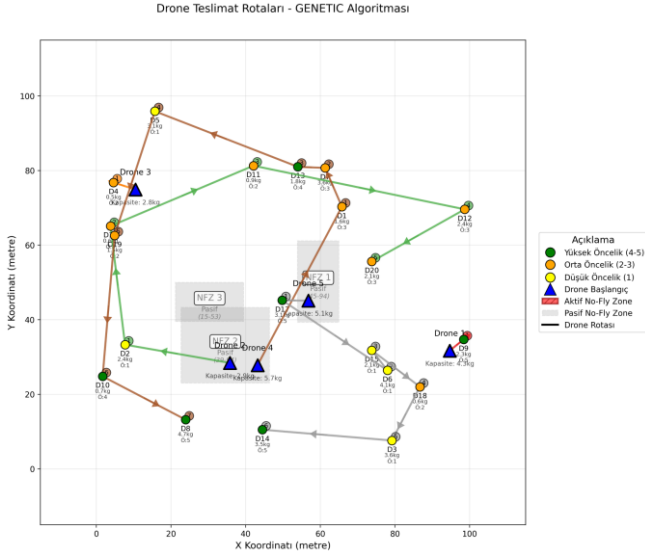
##### a) A\* Algoritması Sonuçları:



Görsel 3: Senaryo 1 - A\* Algoritması Drone Teslimat Rotaları

- Tamamlanan Teslimat: 20 / 20
- Ortalama Enerji: 330.51 mAh
- Toplam Mesafe: 447.91 metre
- Çalışma Süresi: 0.84 saniye

## b) Genetik Algoritma Sonuçları:



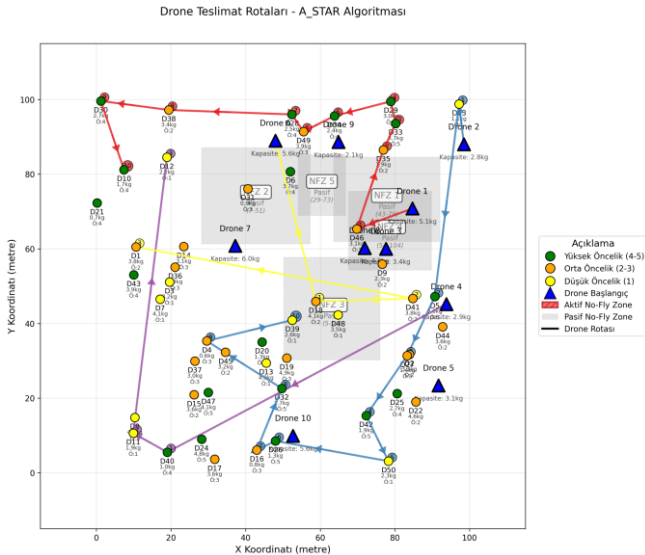
Görsel 4: Senaryo 1 - Genetik Algoritma Drone Teslimat Rotaları

- Tamamlanan Teslimat: 20 / 20
- Ortalama Enerji: 375.97 mAh
- Toplam Mesafe: 520.32 metre
- Çalışma Süresi: 2.71 saniye

## 2) Büyük Ölçekli Teslimat

- Drone Sayısı: 10
- Teslimat Noktası: 50
- No-Fly Zone: 5

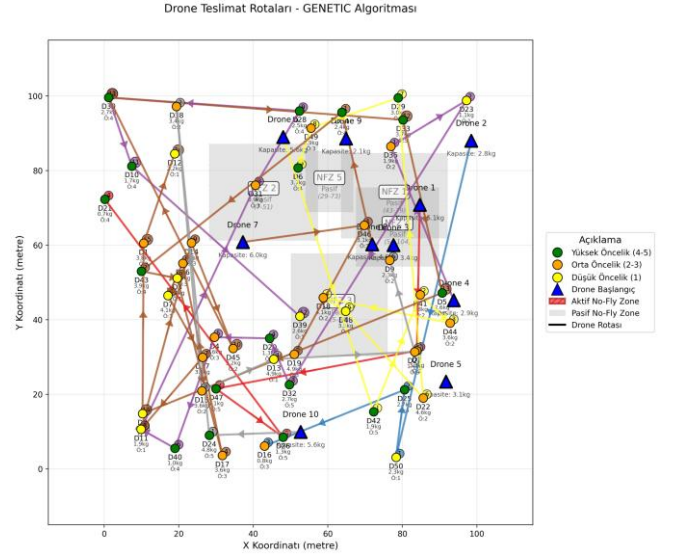
## a) A\* Algoritması Sonuçları:



Görsel 5: Senaryo 2 - A\* Algoritması Drone Teslimat Rotaları

- Tamamlanan Teslimat: 26 / 50
- Ortalama Enerji: 391.08 mAh
- Toplam Mesafe: 683.74 metre
- Çalışma Süresi: 7.56 saniye

## b) Genetik Algoritma Sonuçları:

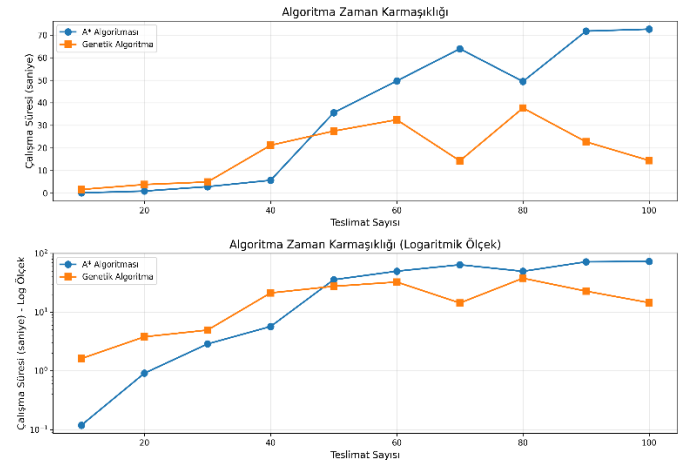


Görsel 6: Senaryo 2 - Genetik Algoritma Drone Teslimat Rotaları

- Tamamlanan Teslimat: 50 / 50
- Ortalama Enerji: 681.35 mAh
- Toplam Mesafe: 2167.81 metre
- Çalışma Süresi: 8.04 saniye

## B. Performans Analizi – Algoritma Zaman Karmaşıklığı

Görsel 7’de iki algoritmanın zaman karmaşıklığı karşılaştırması gösterilmiştir. Genetik algoritma daha büyük senaryolarda ölçeklenebilirlik açısından avantaj sunmaktadır. Logaritmik grafikte  $A^*$ ’ın üstel artış eğilimi net şekilde gözlemlenirken, GA daha yatay bir seyir izlemektedir. Bu durum;  $A^*$ ’ın çözüm kalitesine odaklı detaylı arama yapısına karşın, GA’nin yaklaşık çözümlerle daha hızlı sonuç üretmesinden kaynaklanmaktadır.



Görsel 7: Algoritma Zaman Karmaşıklığı Karşılaştırması

## VI. SONUÇ VE DEĞERLENDİRME

Bu çalışmada,  $A^*$  + CSP ve genetik algoritma iki farklı senaryo üzerinden karşılaştırmalı olarak değerlendirilmiştir.

Metrik	Algoritmalar (Senaryo 1 ve 2)			
	A* (S1)	GA (S1)	A* (S2)	GA (S2)
Tamamlanan Teslimat	20 / 20	20 / 20	26 / 50	50 / 50
Tamamlanma Oranı (%)	100.00	100.00	52.00	100.00
Toplam Enerji (mAh)	6610.29	7519.40	10168.09	34067.46
Toplam Mesafe (metre)	447.91	520.32	683.74	2167.81
Ortalama Enerji / Teslimat	330.51	375.97	391.08	681.35
Çalışma Süresi (saniye)	0.8409	2.7178	7.5596	8.0403
Zaman Karmaşıklığı	$O(20 \cdot 400 \cdot \log 20)$	$O(75 \cdot 30 \cdot 20)$	$O(50 \cdot 2500 \cdot \log 50)$	$O(75 \cdot 30 \cdot 50)$

Tablo 4: A\* ve Genetik Algoritmaların Senaryo Bazlı Performans Karşılaştırması

Küçük ölçekli senaryolarda A\* algoritması daha hızlı ve daha enerji verimli sonuçlar üretmiştir. Büyük ölçekli senaryolarda ise genetik algoritma kapsam açısından üstünlük sağlamış, ancak yüksek enerji tüketimi ile verimlilikten ödün vermiştir. Zaman karmaşıklığı analizleri, A\* algoritmasının problem boyutu büyüdükçe daha hızlı bozulduğunu, GA'nın ise daha dengeli performans gösterdiğini ortaya koymuştur. Algoritma seçiminin senaryo ölçeği ve öncelikli performans kriterine göre belirlenmesi gerektiği sonucuna varılmıştır.

#### REFERANSLAR

- [1] "Python 3 Data Classes." [Çevrimiçi]: <https://docs.python.org/3/library/dataclasses.html> [Erişim: 01.06.2025].
- [2] "Shapely documentation." [Çevrimiçi]: <https://shapely.readthedocs.io> [Erişim: 01.06.2025].
- [3] "A\* Search Algorithm," GeeksforGeeks. [Çevrimiçi]: <https://www.geeksforgeeks.org/a-search-algorithm/>. [Erişim: 01.06.2025].
- [4] "A\* Algorithm Tutorial," Brilliant.org. [Çevrimiçi]: <https://brilliant.org/wiki/a-star-search/>. [Erişim: 01.06.2025].
- [5] "Constraint Satisfaction Problem (CSP)," GeeksforGeeks.. [Çevrimiçi]: <https://www.geeksforgeeks.org/constraint-satisfaction-problems-csp-in-artificial-intelligence/>. [Erişim: 01.06.2025].
- [6] "Genetic Algorithm," GeeksforGeeks. [Çevrimiçi]: <https://www.geeksforgeeks.org/genetic-algorithms/>. [Erişim: 01.06.2025].