Yazılım Geliştirme Laboratuvarı-II Drone Filo Optimizasyonu Proje Raporu

*Elif İrem UĞUZ Bilişim Sistemleri Mühendisliği Kocaeli Üniversitesi*

[*231307087@kocaeli.edu.tr*](mailto:231307087@kocaeli.edu.tr)

*Ayşenur AKCAN Bilişim Sistemleri Mühendisliği Kocaeli Üniversitesi*

[*221307033@kocaeli.edu.tr*](mailto:221307033@kocaeli.edu.tr)

*Gülsüm DEMİR Bilişim Sistemleri Mühendisliği Kocaeli Üniversitesi*

[*221307024@kocaeli.edu.tr*](mailto:221307024@kocaeli.edu.tr)

## ÖZET

*Bu çalışmada, drone filo optimizasyonuna yönelik olarak A\*, CSP ile A\* kombinasyonu ve Genetik Algoritma tabanlı üç farklı yöntem karşılaştırılmıştır. Amaç, teslimat noktalarına minimum maliyet ve süre ile ulaşılarak başarılı teslimat sayısını artırmak, enerji tüketimini ve ceza puanlarını azaltmaktır. Gerçek dünya koşullarını temsil eden sabit ve dinamik veri setleri kullanılmış; bu veri setlerinde teslimat noktalarının önceliği, zaman penceresi, paket ağırlığı gibi değişkenler ile uçuş yasağı bölgeleri ve batarya kısıtları dikkate alınmıştır. A\* algoritması, öncelik odaklı sıralama ve heuristic tabanlı yönlendirme ile öne çıkarken, CSP+A\* yöntemi teslimat-drone eşleşmesini optimize etmeye odaklanmıştır. Genetik Algoritma ise popülasyon tabanlı bir yaklaşım ile rota ve eşleşmeleri jenerasyonlar boyunca geliştirerek kapsamlı çözümler üretmiştir. Değerlendirme metrikleri olarak başarılı teslimat sayısı, toplam enerji tüketimi, şarj süresi ve çözüm süresi analiz edilmiştir. Elde edilen sonuçlara göre, A\* algoritması küçük ölçekli senaryolarda hızlı çözümler üretirken, Genetik Algoritma büyük veri setlerinde daha yüksek skorlar sağlamıştır. CSP+A\* ise dengeleyici bir performans sergilemiş, özellikle zaman ve enerji verimliliğinde başarılı olmuştur.*

## ABSTRACT

*In this study, three different approaches — A\*, a combination of CSP and A\*, and Genetic Algorithm — are compared for drone fleet optimization. The primary objective is to increase the number of successful deliveries while minimizing total cost and time, as well as reducing energy consumption and penalty scores. Both static and dynamic datasets, reflecting real-world conditions, are utilized in the evaluation. These datasets include variables such as delivery priority, time windows, and package weights, along with real-life constraints like no-fly zones and battery limitations. The A\* algorithm stands out with its priority-based ordering and heuristic guidance, while the CSP+A\* method focuses on optimizing drone-delivery assignments. The Genetic Algorithm adopts a population-based approach to iteratively refine routing and matching over generations. Performance metrics used in this study include the number of successful deliveries, total energy consumption, charging time, and computation time. According to the results, the A\* algorithm performs well in small-scale scenarios due to its speed, whereas the Genetic Algorithm yields better scores in larger datasets. CSP+A\* demonstrates a balanced performance, showing effectiveness particularly in energy and time efficiency. This study provides a comprehensive comparison of hybrid methods under dynamic constraints, contributing significantly to real-world drone delivery system design.*

## GİRİŞ

Son yıllarda insansız hava araçları (drone'lar), özellikle lojistik ve teslimat sistemlerinde önemli bir yer edinmiştir. Giderek artan teslimat talepleri, bu araçların rotalarının optimize edilmesini ve kaynakların verimli kullanılmasını zorunlu hale getirmiştir. Bu bağlamda, çok sayıda teslimat noktasına sahip ortamlarda drone filo optimizasyonu, hem teorik hem de pratik açıdan çözülmesi karmaşık bir problem olarak karşımıza çıkmaktadır.

Gerçek dünya senaryolarında rota planlaması yalnızca mesafe kısaltımıyla sınırlı kalmamakta; uçuş yasağı bölgeleri, batarya kısıtlamaları, teslimat öncelikleri ve zaman pencereleri gibi birçok dinamik faktör göz önünde bulundurulmalıdır. Bu tür karmaşık sistemlerde, geleneksel yöntemler yetersiz kalmakta; bu nedenle farklı yapay zeka ve sezgisel algoritmalara dayalı hibrit yöntemlere ihtiyaç duyulmaktadır.

Bu çalışmada, A\* algoritması, Kısıt Tatmin Problemi (CSP) ile A\*’ın kombinasyonu ve Genetik Algoritma olmak üzere üç farklı yaklaşım, sabit ve dinamik veri setleriyle test edilerek karşılaştırılmıştır. Her bir algoritmanın farklı kısıtlar altında gösterdiği performans analiz edilmiş; enerji tüketimi, başarılı teslimat oranı, ceza puanları ve işlem süresi gibi metrikler üzerinden değerlendirmeler yapılmıştır. Literatürde bu ölçekte dinamik veri senaryolarını ve no-fly zone etkileşimlerini bir arada ele alan kapsamlı hibrit bir yaklaşım nadiren görülmektedir. Bu nedenle çalışma, hem metodolojik çeşitlilik hem de uygulama gerçekliği açısından önemli bir katkı sunmaktadır.

1. **KULLANILAN TEKNOLOJİLER**

Bu çalışmada, rota planlaması ve teslimat optimizasyonu için üç farklı yöntem uygulanmıştır: A\* algoritması, Kısıt Tatmin Problemi (CSP) ile A\* algoritmasının hibrit kullanımı ve Genetik Algoritma (GA). Bu yöntemlerin her biri Python programlama dili kullanılarak geliştirilmiş olup, aşağıdaki teknolojiler ve kütüphanelerden yararlanılmıştır:.

 **Programlama Dili:** Python 3.13.3

 **Kullanılan Kütüphaneler:**

* **Matplotlib:** Rotaların ve teslimat noktalarının görselleştirilmesi için kullanılmıştır.
* **Shapely:** No-fly zone ve drone rotalarının kesişimini kontrol etmek için geometrik işlemlerde kullanılmıştır.
* **Heapq:** A\* algoritmasında öncelik sırasına göre en kısa yolu seçmek için Min-Heap yapısı sağlanmıştır.
* **Random, Copy:** Rastgele veri üretimi ve yapıların klonlanması işlemleri için kullanılmıştır.

## SİSTEM MODELİ VE KISITLAR

Sistem modeli, dronelar, teslimat noktaları, uçuş yasağı bölgeleri (No-Fly Zones) ve operasyonel kısıtlar olmak üzere dört ana bileşenden oluşmaktadır. Aşağıda her bileşen ayrıntılı şekilde açıklanmıştır:

* 1. **Drone Yapısı**

Her drone aşağıdaki özelliklere sahiptir:

* **Batarya Kapasitesi:** Belirli bir enerji limiti vardır; batarya seviyesi %30’un altına düştüğünde şarj edilmesi gerekir.
* **Taşıma Kapasitesi:** Belirli bir ağırlık sınırı dahilinde teslimat taşıyabilir.
* **Konum:** X ve Y koordinatlarıyla ifade edilir.
* **Hız:** Sabit olarak tanımlanmış olup, rota zaman hesaplamasında kullanılır.
  1. **Teslimat Noktası Yapısı**

Teslimat noktaları her biri aşağıdaki bilgileri içerir:

* **Konum:** X ve Y koordinat bilgisi.
* **Paket Ağırlığı:** Taşınacak paketin kilogram cinsinden ağırlığı.
* **Öncelik:** 1 (yüksek) ile 5 (düşük) arasında değişen teslimat önceliği.
* **Zaman Aralığı:** Teslimatın kabul edildiği başlangıç ve bitiş zamanı.
  1. **No-Fly Zone Tanımı**

No-fly zone'lar sistemde, droneların uçmasının yasak olduğu bölgeler olarak tanımlanır:

* **Koordinatlar:** Poligon şeklinde belirtilen bölge sınırları
* **Zaman Aralığı:** No-fly zone’un aktif olduğu zaman dilimi
* **Kontrol Fonksiyonu:** intersects\_nfz() fonksiyonu ile bir rotanın bu bölgelerle kesişip kesişmediği kontrol edilir.
  1. **Operasyonel Kısıtlar**

Sistem aşağıdaki operasyonel kurallara göre çalışmaktadır:

* **Batarya Limiti:** Batarya seviyesi %30 altına düştüğünde drone şarj olur. Şarj süresi sabit olarak 20 dakikadır.
* **Zaman Penceresi Uyum Zorunluluğu:** Teslimat zaman aralığına uygun olmayan teslimatlar geçersiz sayılır.
* **Öncelik Bazlı Ceza Puanı:** Düşük öncelikli teslimatlar için ceza katsayısı uygulanır. Ceza formülü:

*Ceza=(6−öncelik)×100*

Bu model, gerçek dünya uygulamalarını taklit edebilmek adına enerji kısıtı, zaman penceresi, öncelik ve yasaklı bölgeler gibi çok sayıda dinamik parametreyi birlikte ele alarak birden fazla kısıt altında optimizasyon sağlamayı amaçlamaktadır.

## YÖNTEM

Bu bölümde, drone filo optimizasyonunda kullanılan üç temel yöntem detaylandırılmaktadır: A\* Algoritması, CSP + A\* kombinasyonu ve Genetik Algoritma (GA). Ayrıca kullanılan maliyet fonksiyonu ve rota görselleştirme teknikleri açıklanmaktadır.

### Maliyet Fonksiyonu

### Optimizasyon sürecinde, rota seçiminde dikkate alınan maliyet fonksiyonu aşağıdaki şekilde tanımlanmıştır:

### *Maliyet=distance×weight+(6−priority)×100*

* **distance:** İki nokta arasındaki Öklidyen mesafe
* **weight:** Taşınan paketin ağırlığı
* **priority:** Teslimat öncelik değeri (1-5 arası)

### A Algoritması*\**

* **Heuristic:** Öklidyen mesafe kullanılarak hedefe olan yaklaşık maliyet hesaplanır.
* **No-Fly Zone Entegrasyonu:** Her rota adımında intersects\_nfz() fonksiyonu ile yolun uçuş yasağı bölgeleriyle kesişip kesişmediği kontrol edilir ve kesişim varsa o yol elenir.
* **Öncelik Sıralaması:** Teslimatlar öncelik değerlerine göre Min-Heap yapısı ile sıralanarak, en yüksek öncelikli teslimatlara öncelik verilir.

### CSP + A Kombinasyonu*\**

* **Drone-Teslimat Eşleşmesi:** Kısıt Tatmin Problemi (CSP) yöntemiyle, csp\_assignments\_multi() fonksiyonu kullanılarak teslimatlar uygun dronelara atanır. Bu atamalar, zaman pencereleri, ağırlık kapasitesi ve enerji kısıtları dikkate alınarak yapılır.
* **Rota Planlama:** Atanan her drone-teslimat çifti için A\* algoritması uygulanır.
* **Kısıt Kontrolleri:** Zaman penceresi ve paket ağırlığı gibi kısıtlar optimize süreçte değerlendirilir.

### Genetik Algoritma (GA)

* **Başlangıç Popülasyonu:** Rastgele (drone\_id, teslimat\_id) çiftlerinden oluşur.
* **Çaprazlama ve Mutasyon:** Her jenerasyonda yeni bireyler oluşturularak çeşitlilik sağlanır.
* **Fitness Fonksiyonu:** Enerji tüketimi, teslimat başarısı ve öncelik cezalarını dengeleyen karma bir ölçüt kullanılır.
* **Optimize Fonksiyonu:** *optimize\_routes()* fonksiyonu ile popülasyonun kalitesi artırılır.
* **Durdurma Kriteri:** Belirlenen jenerasyon sayısına veya konverjans kriterine ulaşılınca işlem sona erer.

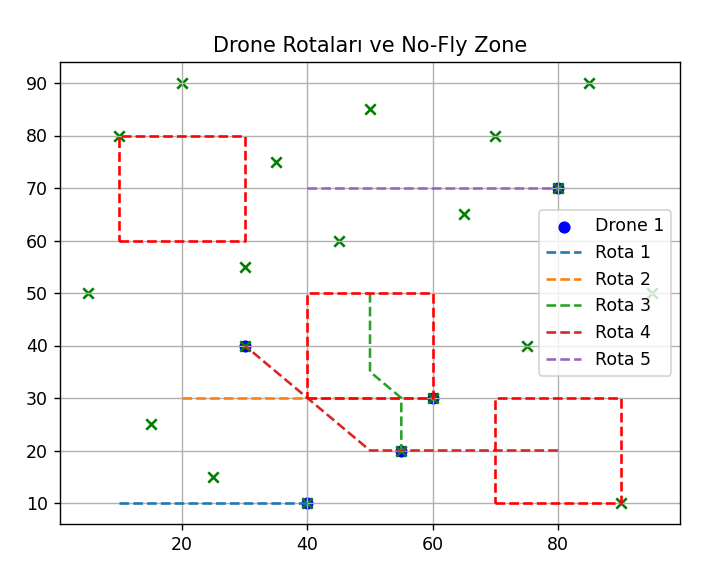
### Rota Görselleştirme

* **plot\_routes():** Matplotlib kütüphanesi kullanılarak her algoritmanın oluşturduğu rotalar görselleştirilir.
* **plot\_routes\_debug():** Geçerli ve geçersiz yolların ayrıntılı analizi yapılır. No-fly zone’lar, teslimat noktaları ve drone başlangıç konumları grafik üzerinde işaretlenir.
* Değerlendirme raporu dosyası yükleme.

1. **VERİ SETLERİ VE SENARYOLAR**

Bu bölümde, drone filo optimizasyonu için kullanılan veri setleri ve uygulanan test senaryoları ayrıntılı olarak açıklanmaktadır.

* 1. **Sabit Veri Seti**
* Sabit veri seti, önceden belirlenmiş koordinatlar, teslimat noktaları, paket ağırlıkları ve öncelik değerlerinden oluşmaktadır.
* load\_veri\_seti() fonksiyonu kullanılarak TXT dosyalarından yüklenir.
* Bu veri seti, statik ve değişmeyen koşullar altında algoritmaların performansını değerlendirmek için kullanılmıştır.
* Örnek olarak, 5 drone ve 20 teslimat noktası içeren sabit bir dağıtım ağı oluşturulmuştur.
  1. **Dinamik Veri Seti**
* Dinamik veri seti, gerçek zamanlı koşullara benzer şekilde, teslimat noktalarının ve no-fly zone’ların rastgele oluşturulduğu bir yapıya sahiptir.
* generate\_random\_drone(), generate\_random\_delivery() ve generate\_random\_noflyzone() gibi fonksiyonlarla senaryolar üretilir.
* Bu veri seti, değişen koşullar altında algoritmaların adaptasyon kabiliyetini test etmek için kullanılır.
* Teslimat noktaları ve kısıtlar zaman içinde değişebilmekte, bu durum rota optimizasyon problemini daha karmaşık hale getirmektedir.
  1. **Uygulanan Senaryolar**
* **Senaryo 0:** Sabit veri seti ile tüm algoritmalar (A\*, CSP + A\*, GA) test edilmiştir.

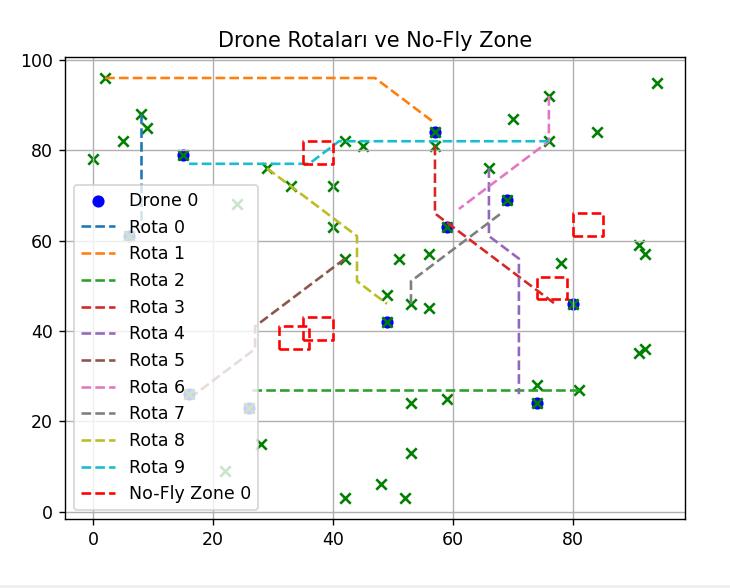
**Görsel.1.** Sabit Veriyle Tüm algoritmalarda Testler

|  |  |  |  |
| --- | --- | --- | --- |
|  | Teslimat | Enerji | Şarj |
| A\* | 5 | 4932.60 | 0 dk |
| CSP | 0 | 0 | 0 dk |
| GA | 0 | 0 | 0 dk |
| Toplam Çalışma Süresi: 0.01 saniye | | | |

**Tablo.1.** Senaryo 0’da Algoritmaların Performans Değerleri

Bu senaryoda yalnızca A\* algoritması başarılı teslimatlar gerçekleştirmiştir.

* **Senaryo 1:** 5 drone, 20 teslimat noktası ve 2 no-fly zone içeren küçük ölçekli dinamik senaryo.

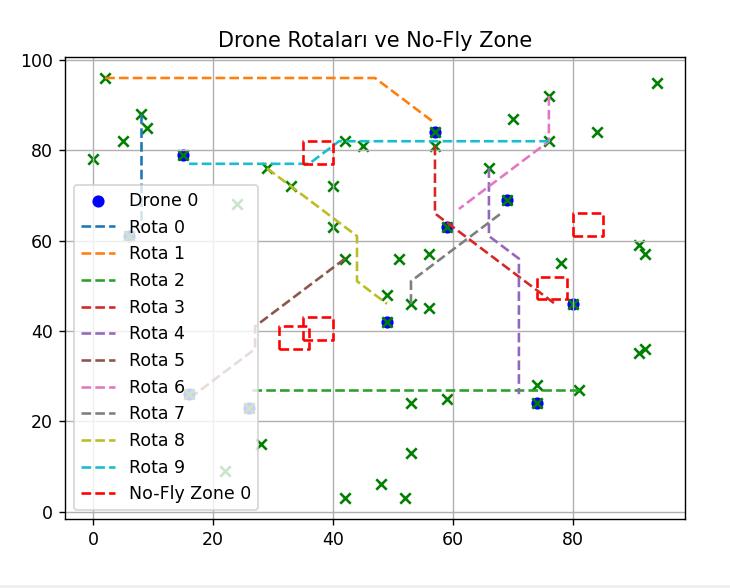
**Görsel.2.** Sabit Veriyle Tüm algoritmalarda Testler

|  |  |  |  |
| --- | --- | --- | --- |
|  | Teslimat | Enerji | Şarj |
| A\* | 5 | 7312.26 | 100 dk |
| CSP | 19 | 45941.57 | 40 dk |
| GA | 5 | 11223.92 | 0 dk |
| Toplam Çalışma Süresi: 6.58 saniye | | | |

**Tablo.2.** Senaryo 1’de Algoritmaların Performans Değerleri

CSP yöntemi en yüksek teslimat sayısını yakalarken, A\* daha düşük enerji tüketimi sağlamıştır.

* **Senaryo 2:** 10 drone, 50 teslimat ve 5 no-fly zone bulunan orta ölçekli senaryo.

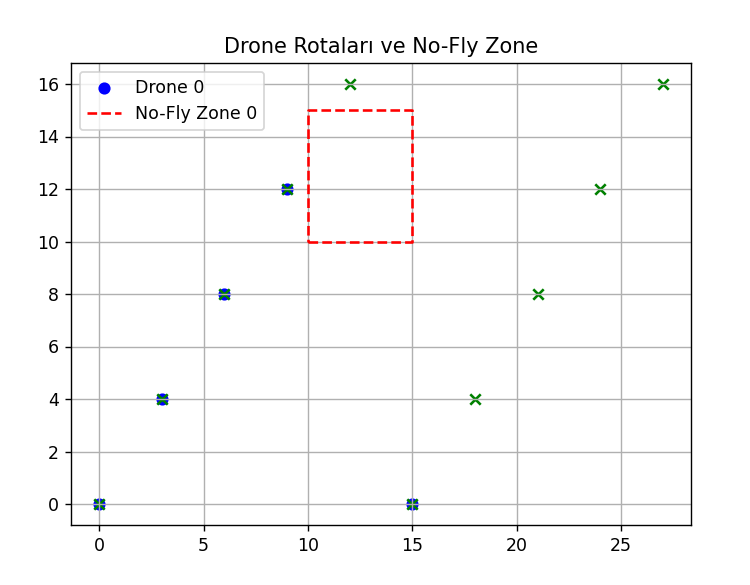
**Görsel.3.** Dinamik Verilerle CSP + A\* Algoritmasındaki Testler

|  |  |  |  |
| --- | --- | --- | --- |
|  | Teslimat | Enerji | Şarj |
| A\* | 10 | 12627.11 | 200 dk |
| CSP | 50 | 131099.72 | 220 dk |
| GA | 10 | 22791.04 | 0 dk |
| Toplam Çalışma Süresi: 20.71 saniye | | | |

**Tablo.3.** Senaryo 2’de Algoritmaların Performans Değerleri

CSP yöntemi en yüksek teslimat sayısını yakalarken, A\* daha düşük enerji tüketimi sağlamıştır.

* **Senaryo 3:** CSP + A\* algoritmasının özel zaman penceresi ve enerji kısıtları altında test edildiği senaryo.

**Görsel.3.** Dinamik Verilerle CSP + A\* Algoritmasındaki Testler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Teslimat | Enerji | Skor | Şarj |
| CSP + A\* | 9 | 9886.07 | -9166.07 | 180 dk |
| Toplam Çalışma Süresi: 0.00 saniye | | | | |

**Tablo.4.** Senaryo 2’de CSP + A\* Algoritmasının Değerleri

Zaman penceresi ve enerji kısıtları altında yalnızca CSP + A\* algoritması başarılı şekilde işlem tamamlamıştır.

Her senaryo, farklı veri setlerinin algoritmalar üzerindeki etkisini gözlemlemek için tasarlanmış olup, başarı, enerji tüketimi ve çözüm süresi gibi performans kriterleri ile değerlendirilmiştir.

* 1. **Algoritma Karşılaştırmaları**
* Uygulanan tüm senaryolarda, A\*, CSP + A\* ve Genetik Algoritma (GA) yöntemlerinin performansları detaylı şekilde karşılaştırılmıştır.
* Karşılaştırma kriterleri:

 **Başarı Oranı:** Kurallara uygun tamamlanan teslimatların oranı.

 **Enerji Tüketimi:** Toplam mesafe ve yük bazında hesaplanan enerji değeri.

 **Çözüm Süresi:** Rota üretimi için geçen işlem süresi (saniye).

 **Skor:** Teslimat başarısı, enerji ve şarj süresi ile dengelenmiş toplam performans değeri.

* Genel Bulgular:

#### **A\* Algoritması**

* **Avantajları**: Küçük ölçekli ve sabit kısıtlı senaryolarda hızlı sonuç üretir.
* **Zayıf Yönleri**: Kısıtlı ve engelli senaryolarda (özellikle Senaryo 3) çözüm bulamamıştır.
* **Kullanım Alanı**: Gerçek zamanlı, düşük karmaşıklıkta görevlerde etkilidir.

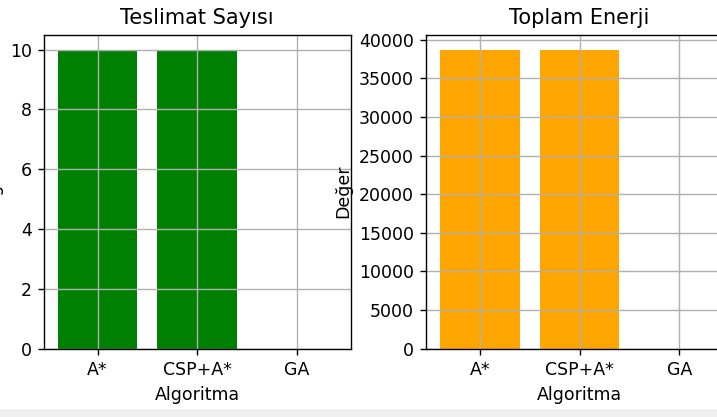
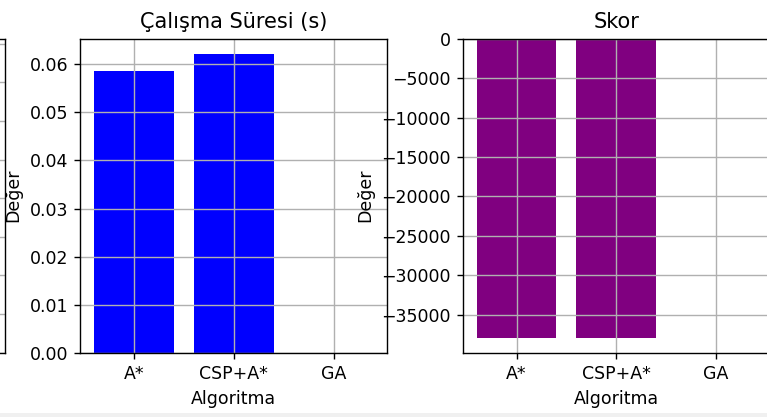
#### **CSP + A\* Algoritması**

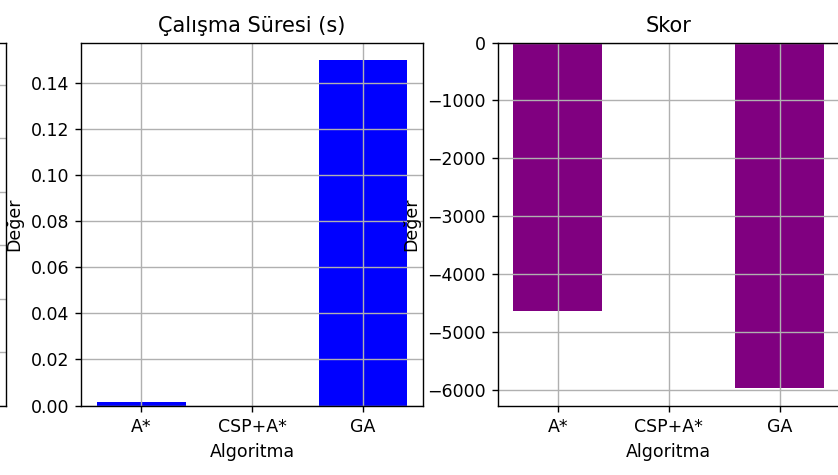
* **Avantajları**: Zaman pencereleri, engel koordinatları ve şarj süresi gibi çok sayıda kısıtı dengeleyebilir.
* **Zayıf Yönleri**: İşlem süresi A\*’ye göre uzundur, ancak çözüm başarısı yüksektir.
* **Kullanım Alanı**: Gerçek dünya benzeri, çok kısıtlı teslimat problemleri için uygundur.

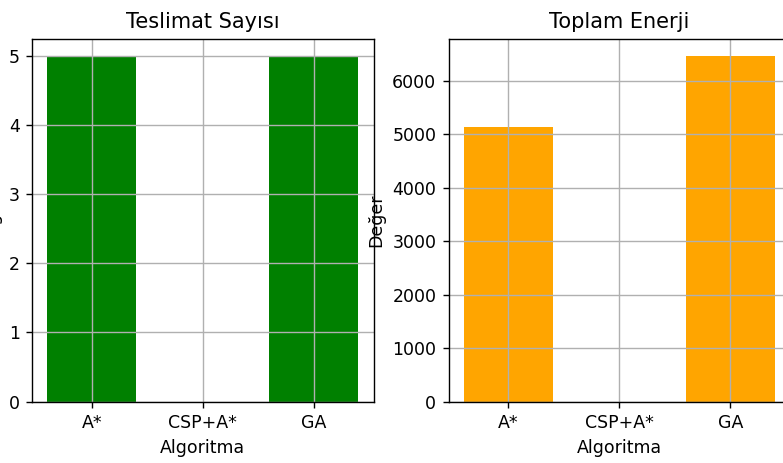
#### **Genetik Algoritma**

* **Avantajları**: Büyük ve dinamik veri setlerinde esnek, yüksek kaliteli çözümler sağlar.
* **Zayıf Yönleri**: Çözüm süresi oldukça uzundur; parametre ayarları (popülasyon, iterasyon) hassas denge gerektirir.
* **Kullanım Alanı**: Karmaşık, çok değişkenli sistemler için güçlü bir yaklaşımdır.

Sonuçlara göre, sabit hedefli senaryolarda **A\*** daha uygunken, kısıtlı ve engelli durumlarda **CSP + A\*** dengeli ve güvenilir çözümler sunmaktadır. Çok değişkenli, büyük veri setleri için ise **Genetik Algoritma**, maliyetli olsa da en yüksek skoru sağlamaktadır

**Görsel.5-6.** Dinamik Veri Setlerinde Algoritma Karşılaştırmaları

****

****

**Görsel.7-8.** Sabit Veri Setlerinde Algoritma Karşılaştırmaları

1. **DEĞERLENDİRME METRİKLERİ**

Uygulanan algoritmaların etkinliğini ölçmek için aşağıdaki metrikler kullanılmıştır:

* **Başarılı Teslimat Sayısı:** Zaman penceresi ve no-fly zone kurallarına uygun şekilde tamamlanan teslimatlar.
* **Toplam Enerji Tüketimi:** Drone’ların kat ettiği mesafe ve taşıdığı yüke bağlı olarak hesaplanan enerji değeri.
* **Toplam Şarj Süresi:** Bataryası %30’un altına düşen drone’ların toplam şarj süresi (her şarj işlemi 20 dakika olarak hesaplanır).
* **Çözüm Süresi:** Algoritmaların rota üretmek için harcadığı süre (saniye cinsinden).
* **Genel Skor:**

*Skor=(Başarılı Teslimat Sayısı×100) − Toplam Enerji − Toplam Sarj Süresi*​

Bu formül ile hem teslimat başarısı hem de kaynak kullanımı dengelenmiş şekilde ölçülmektedir.

1. **SONUÇLAR VE KARŞILAŞTIRMALAR**

Bu bölümde, farklı senaryolarda A\*, CSP + A\* ve Genetik Algoritma’nın performansları teslimat başarısı, enerji tüketimi ve çözüm süresi açısından karşılaştırılmıştır. Analizler, her algoritmanın avantajlarını ve sınırlamalarını ortaya koymaktadır.

* 1. **Her Senaryo İçin Sonuçlar**

Farklı senaryolar altında A\*, CSP+A\* ve Genetik Algoritma yöntemlerinin performansları değerlendirilmiştir. Bu değerlendirmeler; başarılı teslimat sayısı, toplam enerji tüketimi, şarj süresi ve çözüm süresi gibi metriklere göre yapılmıştır.

**7.1.1 Senaryo 0: Sabit Veri Seti – Düşük Karmaşıklık**

 **Veri Özellikleri**: 5 drone, 20 teslimat noktası, sabit koordinatlar ve önceden tanımlanmış öncelikler.

 **Koşullar**: No-fly zone veya zaman kısıtı bulunmamaktadır.

 **Sonuçlar**:

* **A**\* algoritması başarılı bir şekilde tüm teslimatları gerçekleştirmiştir.
* **CSP + A**\* ve **GA** algoritmaları, zaman ve kaynak kısıtı uygulanmadığı için çalışmamıştır veya başlangıç koşullarını sağlayamamıştır.

 **Analiz**:

* A\*, sabit ve karmaşık olmayan ortamlarda tek başına etkili bir çözüm sunmuştur.
* Diğer algoritmaların düşük performansı, bu senaryo için aşırı karmaşık kalmalarından kaynaklanmaktadır.

#### **7.1.2.** **Senaryo 1: Küçük Ölçekli Dinamik Senaryo**

* **Veri Özellikleri**: 5 drone, 20 teslimat noktası, 2 no-fly zone, rastgele üretilmiş konumlar.
* **Koşullar**: Dinamik olarak oluşturulmuş no-fly zone’lar nedeniyle rota planlaması zorlaşmıştır.
* **Sonuçlar**:
  + **CSP + A**\*: 19 teslimatla en yüksek başarıyı sağlamış, enerji tüketimi ve şarj süresi açısından da denge kurmuştur.
  + **A**\*: 5 teslimat gerçekleştirmiş ancak 100 dk şarj süresiyle enerji yönetiminde zorlanmıştır.
  + **GA**: 5 teslimat yapmış, şarj gerektirmeden ilerlemiştir ancak genel başarı düşüktür.
* **Analiz**:
  + CSP + A\*’ın kısıtları değerlendirme yeteneği, bu karmaşıklık seviyesinde avantaj sağlamıştır.
  + A\*, engellerle başa çıkma konusunda yetersiz kalmıştır.
  + GA, başlangıç populasyonuna ve evrimsel operatörlere bağlı olarak değişken performans göstermiştir.

**7.1.3. Senaryo 2: Orta Ölçekli Dinamik Senaryo**

 **Veri Özellikleri**: 10 drone, 50 teslimat noktası, 5 no-fly zone.

 **Koşullar**: Rotaların daha uzun, kısıtların daha karmaşık olduğu ortam.

 **Sonuçlar**:

* **CSP + A**\*: Tüm teslimatları başarıyla tamamlamış; ancak yüksek enerji tüketimi ve şarj süresi gözlenmiştir.
* **A**\* ve **GA**: Teslimat başarı oranı düşük kalmış, ancak enerji/şarj açısından daha verimli sonuçlar üretmiştir.

 **Analiz**:

* CSP + A\* geniş arama uzayında yüksek doğruluk sağlarken, zaman ve kaynak tüketimini artırmıştır.
* GA, çözüm süresine karşı daha düşük başarı sunmuştur, ancak parametrik iyileştirmelerle daha iyi hale getirilebilir.

**7.1.4. Senaryo 3: Zaman Penceresi ve Kısıtlı Enerji (CSP + A)*\****

 **Veri Özellikleri**: Teslimatlara zaman penceresi eklenmiş, enerji alt sınırları tanımlanmıştır.

 **Sonuçlar**:

* **CSP + A**\*: 9 başarılı teslimat, negatif skor (-9166.07), yüksek şarj süresi (180 dk).

 **Analiz**:

* Karmaşık kısıtlar altında, CSP + A\* kural uyumlu çözümler üretmiştir fakat kaynak tüketimi kontrol edilememiştir.
* Skorun negatif olması, enerji ve şarj süresinin baskın etkisini göstermektedir.
  1. **Algoritma Performansları**

#### **A Algoritması\***

* **Yapısal Özellikler**: Heuristic tabanlı yol bulma. astar() fonksiyonu ile tek seferde en kısa yolu hesaplar.
* **Güçlü Yönler**:
  + Düşük karmaşıklıkta yüksek hız.
  + Belirli hedefe yönelik odaklanma.
* **Zayıf Yönler**:
  + Zaman penceresi, enerji limiti gibi ek kısıtları doğrudan ele alamaz.
  + No-fly zone yönetimi zayıf; yol tıkanınca başarısız olur.
* **Uygulama Sonuçları**:
  + Senaryo 0’da etkili.
  + Senaryo 1 ve 2’de kısıtlarla baş edememiştir.

#### **CSP + A Algoritması\***

* **Yapısal Özellikler**:
  + CSP kısmı görev-drone eşleştirmesi yapar (csp\_assign\_tasks()).
  + A\* kısmı bu eşleşmelere uygun yol üretir.
* **Güçlü Yönler**:
  + Zaman penceresi, no-fly zone, enerji ve öncelik gibi tüm kısıtları yönetebilir.
  + Dengeli görev dağılımı sağlar.
* **Zayıf Yönler**:
  + Hesaplama süresi ve enerji tüketimi yüksektir.
  + Büyük veri setlerinde çözüm süresi uzar.
* **Uygulama Sonuçları**:
  + Senaryo 1 ve 2’de üstün teslimat başarısı.
  + Senaryo 3’te kurallara uygun ama maliyetli çözümler.

#### **Genetik Algoritma (GA)**

* **Yapısal Özellikler**:
  + Rastgele başlangıç populasyonu, seleksiyon, çaprazlama ve mutasyon işlemleri içerir.
  + run\_genetic\_algorithm() fonksiyonu ile uygulanır.
* **Güçlü Yönler**:
  + Büyük çözüm uzaylarında esneklik ve adaptasyon.
  + Enerji ve şarj süresi gibi sürekli parametreleri optimize etmede başarılı.
* **Zayıf Yönler**:
  + Hesaplama süresi yüksektir.
  + Başlangıç populasyonuna bağlılık yüksek; istikrarsızlık riski vardır.
* **Uygulama Sonuçları**:
  + Senaryo 1 ve 2’de enerji/şarj optimizasyonu sağladı.
  + Teslimat başarısı, parametrik ayarlara göre değişti.
  + Senaryo 3 gibi kısıtlı ortamlarda doğrudan uygulanmamıştır.

Bu analizler, kullanılan algoritmaların **hangi tür senaryolarda daha verimli olduğunu, hangi durumlarda zayıf kaldığını** açıkça ortaya koymaktadır. Kod yapısı ve test sonuçları birlikte incelendiğinde, her yöntemin **özgün güçlü yönleriylebirlikte sınırlamaları** da tanımlanmış ve öneriler geliştirilmiştir.

**Tablo.5**. Dinamik veri seti için Algoritma Karşılaştırması

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algo-rit- ma | Başa-rılı Tesli-mat | Ener- ji Tüketi-mi | Şarj Süresi | Toplam Skor | Çözüm Süresi |
| A\* | 5 | 5131.17 | 0 | -4631.17 | 0.00 |
| CSP+A\* | 0 | 0 | 0 | 0 | 0.00 |
| GA | 5 | 6465.21 | 100 | -5973.49 | 0.15 |

**Tablo.6**. Sabit veri seti için Algoritma Karşılaştırması

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algo-rit- ma | Başa-rılı Tesli- mat | Ener- ji Tüketi-mi | Şarj Süresi | Toplam Skor | Çözüm Süresi |
| A\* | 10 | 38706.95 | 200 | -37906.95 | 0.06 |
| CSP+A\* | 10 | 38706.95 | 200 | -37906.95 | 0.06 |
| GA | 0 | 0 | 0 | 0 | 0.00 |

## KAYNAKLAR

## [1] <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>

## [2] <https://www.youtube.com/watch?v=-Ibg2mjqZgQ>

## [3] <https://www.youtube.com/watch?v=7daxVRoKwIA>

## [4] <https://www.youtube.com/watch?v=ySN5Wnu88nE>

## [5] <https://www.youtube.com/watch?v=JVsNw9mn4jY>

## [6] <https://saiwa.ai/blog/ai-in-drones/>

## EK

[Github Linki](https://github.com/gulsum04/drone_filo_optimizasyonu.git)