

5 Temmuz 2024

Ondokuz Mayıs Üniversitesi

Dron Ağları İçin Yer İstasyonu Tasarımı

Danışman: Doç. Dr. Sercan DEMİRCİ

Zeynep Sıla Kaymak
Can Yılmaz
Cenkay Kucur

Genel Bakış

01 Proje Amacı

02 Problem

03 Yöntem

04 Algoritmalar

05 Senaryolar

06 Çıktı Verileri

07 Elde Edilen Grafik
Örneği

08 Sonuç

09 Grafikler

10 Teşekkürler

11 Sorular ve
Öneriler

Proje Amacı

- Dron ve İstasyon Eşleştirilmesi
- Optimizasyon Algoritmaları ile Elde Edilen Sonuçların Görselleştirilmesi
- Elde Edilen Verilerle Performans Karşılaştırılması

Problem

01

Dron istasyon ataması
hangi kısıtlar dikkate
alınarak oluşturulmalı?

Problem 1

- Drone'ların istasyonlara atanmasında kullanılan parametreler

Mesafe (Distance):

- Drone'ların konumları (x, y koordinatları) ile istasyonların konumları arasındaki mesafe hesaplanır.

Hız (Speed):

- Drone'ların maksimum hızları kullanılır.
- Mesafeye göre drone'un ulaşma süresi hesaplanır.

Batarya Seviyesi (Battery Level):

- Drone'ların mevcut batarya seviyeleri kullanılır.
- Batarya seviyesi, maksimum batarya seviyesine oranlanarak uygunluk (fitness) hesaplamasında kullanılır.

Problem

02

Dron istasyon ataması
yapılırken nasıl bir
yöntem izlenmeli?

Yöntem

Greedy Algoritma

- Probleme Uygunluk

Optimizasyon Algoritmaları

- Çözüm Kalitesini Artırma
- Çeşitli Problemler için Uygunluk
- Performans ve Verimlilik

Gri Kurt Optimizasyon Algoritması

Algorithm 4: Gri Kurt Optimizasyonu

```
1 [1] Kurt sayısı numwolves, Maksimum iterasyon sayısı maxiterations, Parametreler:  $a, A, C, b, l$ ,  
İstasyonlar, Drone'lar En iyi global konum, En iyi global uygunluk  
2 Kurtları rastgele başlat eniyiglobalkonum  $\leftarrow$  None, eniyiglobaluygunluk  $\leftarrow \infty$   
3 for  $iterasyon \leftarrow 1$  to  $maxiterations$  do  
  
4 end  
5 kurt in kurtlar Kurtun uygunluğunu değerlendir if  $uygunluk < eniyiglobaluygunluk$  then  
  
6 end  
7 eniyiglobaluygunluk  $\leftarrow$  uygunluk, eniyiglobalkonum  $\leftarrow$  kurt  
8 alfa, beta, delta pozisyonlarını kurtlardan al  
9 foreach  $drone\ i$  kurtta do  
  
10 end
```

```
11 istasyonindex veya drone  $i$  zaten atanmışsa istasyon veya drone  $i$  boşta işaretle else  
  
12 end  
13 drone  $i$  için yeni pozisyon ve batarya hesapla Sınırları kontrol et if yeni istasyonindex veya drone  $i$   
zaten atanmışsa then  
  
14 end  
15 istasyon veya drone  $i$  boşta işaretle else  
  
16 end  
17 Kurtu güncelle, istasyon ve drone  $i$  ata  
18 return eniyiglobalkonum, eniyiglobaluygunluk
```


Karınca Kolonisi Optimizasyonu

Algorithm 5: Karınca Kolonisi Optimizasyonu

1 [1] Parametreleri başlat:

$numants, numiterations, evaporationrate, \alpha, \beta, pheromonedeposit, initialpheromone$ Başlangıç feromon matrisi T ile başlangıç değerlerini ayarlayın Karıncalar için başlangıç çözümlerini oluşturun $bestsolution$ ve $bestfitness$ değerlerini başlangıç çözümünün değerlerine ayarlayın Uygunluk geçmişi listesi $fitnesshistory$ ve ortalama uygunluk geçmişi listesi $averagefitnesshistory$ başlatın

2 for $t = 1$ ila $numiterations$ do

3 end

4 Tüm karıncalar için çözümleri generate edin using $constructsolutions()$ Feromon matrisini T güncelleyin using $updatepheromones()$ En iyi çözümü $iterationbestsolution$ ve bu çözümün uygunluğunu $iterationbestfitness$ bulun $fitnesshistory$ ve $averagefitnesshistory$ güncelleyin

5 if $iterationbestfitness < bestfitness$ then

6 end

7 $bestfitness$ ve $bestsolution$ güncelleyin dönüş

$bestsolution, bestfitness, fitnesshistory, averagefitnesshistory$

8 $constructsolutionsstations, drones$ for her bir karınca için do

9 end

10 T baz alınarak olasılıkları kullanarak çözüm oluşturun Çözüm listesini dönün

11 $updatepheromonesolutions$ T içindeki feromonları buharlaştırın Karıncalar tarafından bulunan çözümlere göre feromonları yatırın

12 $calculateprobabilitiesdrone, stations, availablestations$ Sonraki istasyonu seçmek için olasılıkları hesaplayın

13 $calculatefitnesssolution, stations$ Toplam mesafeye dayalı bir çözümün uygunluğunu hesaplayın

Çekirge Optimizasyon Algoritması

Algorithm 6: Çekirge Optimizasyon Algoritması

```
1  [1] Popülasyon boyutu populationsize, Maksimum iterasyon sayısı maxiterations, Drone'lar,  
    İstasyonlar En iyi global konum, En iyi global uygunluk  
2  Çekirge popülasyonunu rastgele başlat eniyiglobalkonum  $\leftarrow$  None, eniyiglobaluygunluk  $\leftarrow \infty$   
3  for iterasyon  $\leftarrow$  1 to maxiterations do  
  
4  end  
  
5  çekirge Çekirgenin uygunluğunu değerlendir if uygunluk  $<$  eniyiglobaluygunluk then  
  
6  end  
  
7  eniyiglobaluygunluk  $\leftarrow$  uygunluk, eniyiglobalkonum  $\leftarrow$  çekirge  
8  Alfa, beta, delta pozisyonlarını çekirgelerden al  
9  foreach drone i için çekirgede do  
  
10 end
```

```
11 istasyonindex veya drone i zaten atanmışsa istasyon veya drone i boşta işaretle else  
  
12 end  
  
13 drone i için yeni pozisyon ve batarya hesapla Sınırları kontrol et if yeni istasyonindex veya drone i  
    zaten atanmışsa then  
  
14 end  
  
15 istasyon veya drone i boşta işaretle else  
  
16 end  
  
17 Çekirgeyi güncelle, istasyon ve drone i ata  
  
18 return eniyiglobalkonum, eniyiglobaluygunluk
```

Diferansiyel Evrim Algoritması

Algorithm 7: Diferansiyel Evrim Algoritması (DEA)

```
1 [1] Parametreleri başlat: numparticles, maxiterations, scalingfactor, crossoverrate Başlangıç
   partiküllerini oluştur bestglobalposition ve bestglobalfitness değerlerini başlangıç konumlarına ayarla
   Uygunluk geçmişi listesi fitnessvalues başlat Iterasyon = 0 olarak başlat

2 while Iterasyon < maxiterations do

3 end

4 her bir partikül i trialvector oluştur Tekil atanma veya atanmamış durumu sağla Çaprazlama yap Yeni
   partikülün uygunluğunu değerlendir if Yeni partikülün uygunluğu daha iyi ise then

5 end

6 Partikülü güncelle if Yeni uygunluk en iyi global uygunluktan daha iyi ise then

7 end

8 En iyi global uygunluğu ve konumu güncelle Ortalama uygunluğu hesapla ve fitnessvalues listesine
   ekle Iterasyonu bir artır

9 Bitiş zamanını hesapla Dönen değerler:
   bestglobalposition, bestglobalfitness, fitnessvalues, elapsedtime

10 generatetrialtorialvectorpartiküller, geçerliindex, ölçekfaktörü, istasyonlar, dronlar İndeksleri karıştır a, b, c
   seç Deneme vektörü oluştur Geçerli istasyon indeksi veya atanmamış durumu sağla Pil seviyesini
   sınırla Deneme vektörünü döndür

11 evaluatefitnesspartikül, dronlar, istasyonlar Toplam uygunluğu hesapla Döndür: toplam uygunluk
```

Genetik Algoritma

Algorithm 3: Genetik Algoritma

```
1 [1] Dronlar, İstasyonlar, Max Generasyon Sayısı En iyi çözüm ve uygunluk değeri
2 Rastgele popülasyonu başlat Başlangıç popülasyonunun uygunluk değerlerini hesapla
3 for generasyon = 1 to maxgenerasyon do

4 end

5 Popülasyondaki her bireyin uygunluk değerini hesapla
6 Fitness'e göre ebeveyn seçimi yap (rulet tekerleği seçimi)
7 Yeni popülasyonu başlat while yeni popülasyon doluncaya kadar do

8 end

9 İki ebeveynden rastgele seçim yap Çaprazlama işlemi yap (olasılıkla aprazlamaoranı) Mutasyon
   işlemi yap (olasılıkla mutasyonoranı) Oluşan çocukları yeni popülasyona ekle
10 Eski popülasyonu yeni popülasyonla değiştir
11 Mevcut popülasyondaki en iyi çözümü bul
12 En iyi uygunluk değerini geçmişi sakla
13 return En iyi çözüm, en iyi uygunluk değeri, uygunluk geçmişi
```

Arı Koloni Optimizasyon Algoritması

Algorithm 1: Arı Kolonisi Optimizasyonu

1 Başlangıçta, rastgele çözümler oluşturulur ve bunların fitness değerleri hesaplanır. **while** *sonlandırma kriteri sağlanana kadar* **do**

2 **end**

3 **İşçi Arı Aşaması:** **for** *her işçi arı için* **do**

4 **end**

5 Komşulukta yeni bir çözüm keşfet. **if** *yeni çözümün fitness değeri daha iyi ise* **then**

6 **end**

7 Çözümü güncelle ve deneme sayacını sıfırla. **else**

8 **end**

9 Deneme sayacını artır.

10 **Gözcü Arı Aşaması:** **for** *her gözcü arı için* **do**

11 **end**

12 Fitness değerine göre bir çözüm seç. Komşulukta yeni bir çözüm keşfet. **if** *yeni çözümün fitness değeri daha iyi ise* **then**

13 **end**

14 Çözümü güncelle ve deneme sayacını sıfırla. **else**

15 **end**

16 Deneme sayacını artır.

17 **Keşif Arı Aşaması:** **for** *her işçi arı için* **do**

18 **end**

19 deneme sayacı limiti aşılrısa Mevcut çözümü terk et ve yeni bir rastgele çözüm oluştur. Deneme sayacını sıfırla.

20 Şu ana kadar bulunan en iyi çözümü güncelle. **return** En iyi çözümü ve onun fitness değerini.

Parçacık Sürü Optimizasyon Algoritması

Algorithm 2: Parçacık Sürü Optimizasyonu

```
1  Girdi: Dronelar, İstasyonlar, İterasyon Sayısı Çıktı: En iyi drone-istasyon atamaları Başlat:  
    Parçacıkları rastgele konum ve hızlarla Global en iyi konum ve uygunluk değerini başlat (gBest) Her  
    parçacığın en iyi konumunu ve uygunluk değerini başlat (pBest)  
2  for iterasyon = 1 to iterasyon sayısı do  
  
3  end  
  
4  parçacık p Hesapla: p'nin uygunluk değeri if p'nin uygunluk değeri pBest'ten daha iyi ise then  
  
5  end  
  
6  Güncelle: pBest = p'nin mevcut konumu if p'nin uygunluk değeri gBest'ten daha iyi ise then  
  
7  end  
  
8  Güncelle: gBest = p'nin mevcut konumu  
  
9  Başlat: Atanmış istasyonlar kümesi (assignedStations) Başlat: Atanmış dronlar kümesi  
    (assignedDrones)  
10 foreach drone d ve istasyon s çifti do  
  
11 end
```

```
12 Hesapla: Atalet, bilişsel ve sosyal terimler Güncelle: Yeni istasyon indeksi  
13 if newstationindex atanmamış ise then  
  
14 end  
  
15 Güncelle: Parçacığın konumu Ekle: newstationindex'i assignedStations kümesine Ekle: drone'u  
    assignedDrones kümesine else  
  
16 end  
  
17 Seç: Rastgele bir istasyon Güncelle: Parçacığın konumu Ekle: newstationindex'i assignedStations  
    kümesine  
  
18 foreach atanmamış drone d do  
  
19 end  
  
20 Uygula: Ceza işlemi (isteğe bağlı)  
  
21 Hesapla: Ortalama uygunluk değeri Sakla: averagefitness değerini fitnessvalues listesine
```

Problem

03

Dron ve istasyon
konumlarının farklı
olması durumlarında ne
yapılmalı?

Senaryolar

Senaryo 1

- Sabit konumlu dron ve sabit konumlu istasyon eşleştirilmesi

Senaryo 2

- Hareketli dron ve sabit konumlu istasyon eşleştirilmesi

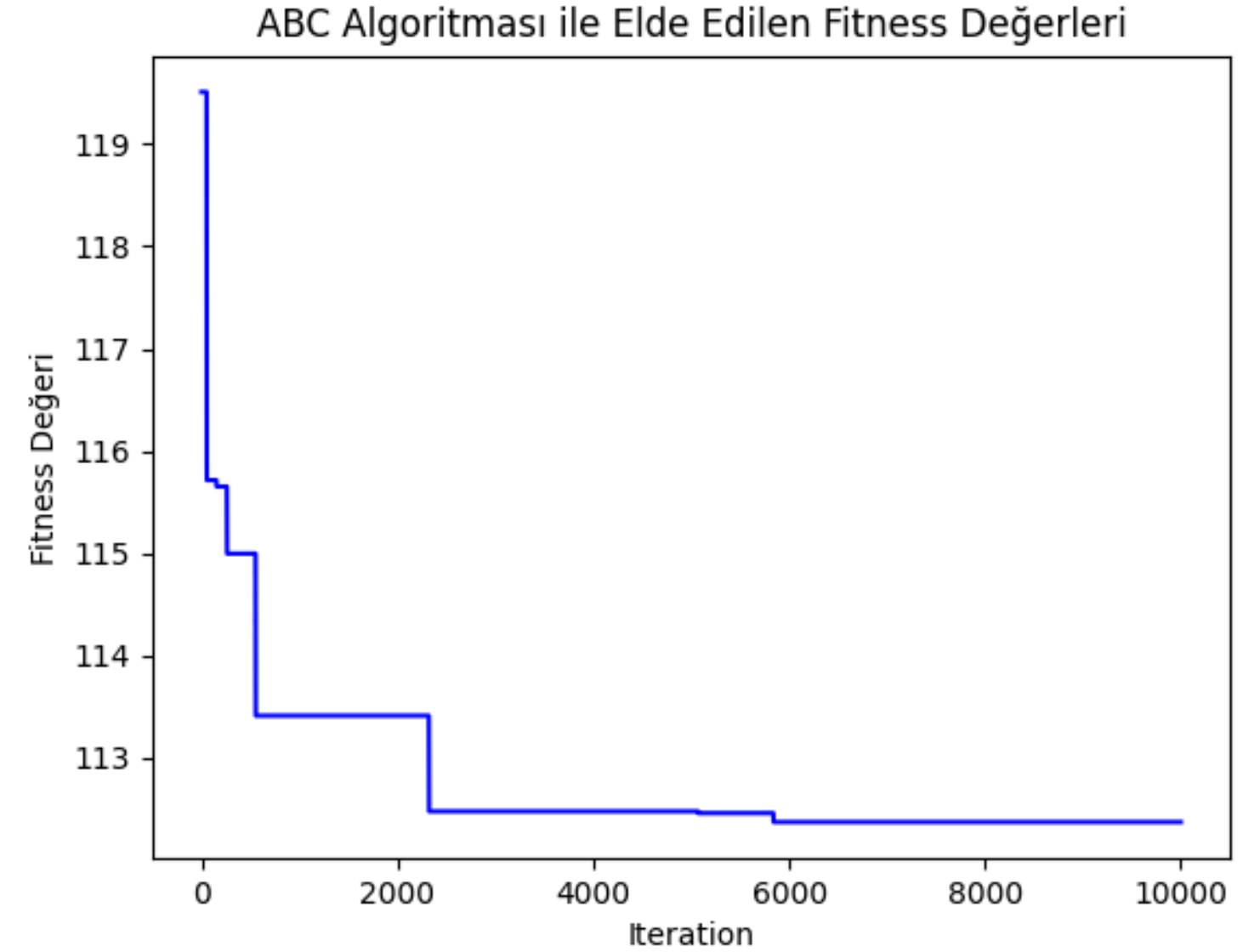
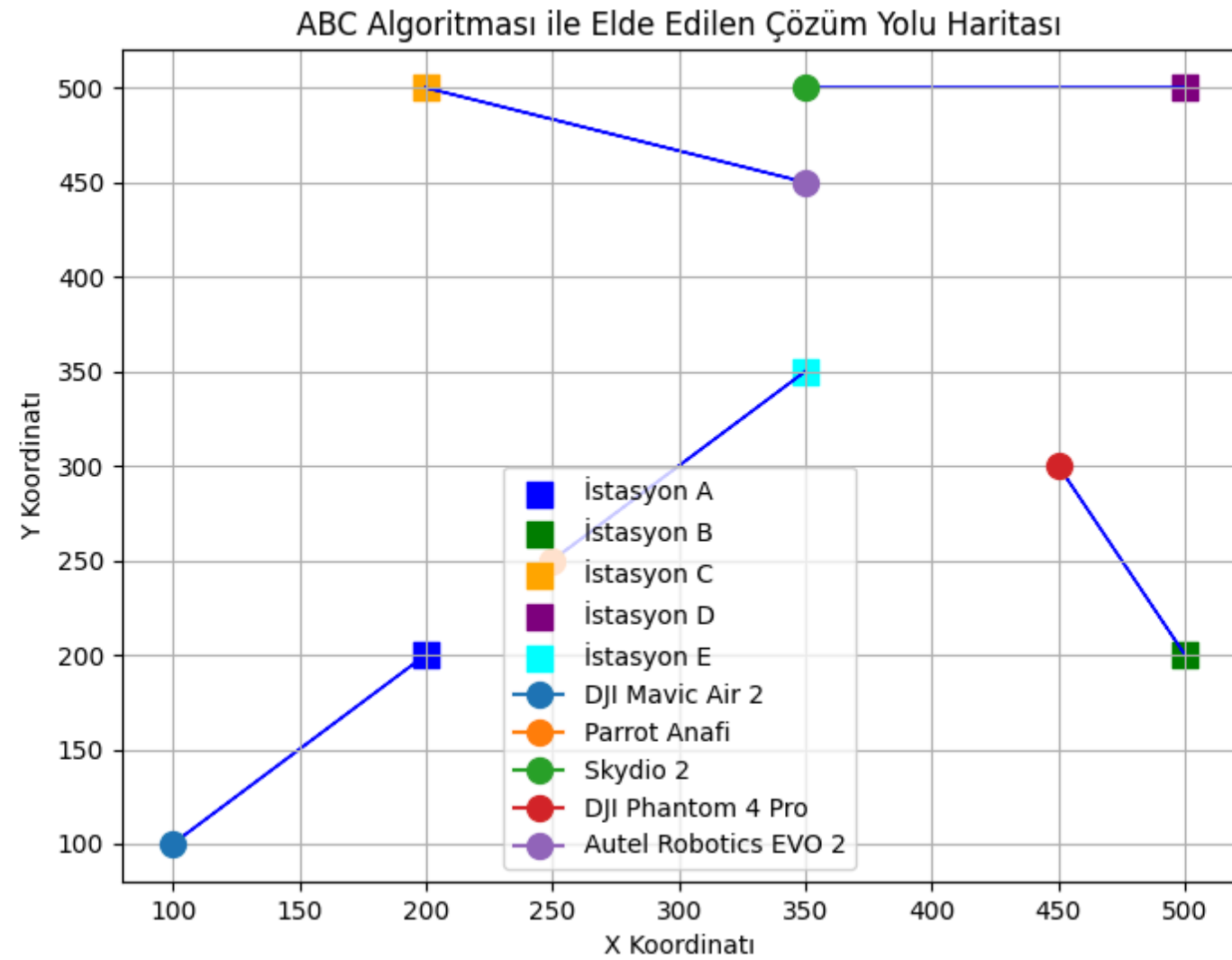
Senaryo 3

- Hareketli dron ve hareketli istasyon eşleştirilmesi

Çıktı Verileri

- Elde Edilen En İyi Uygunluk Degeri
- Ortalama Mesafe
- Ortalama Batarya Kullanımı
- Geçen Süre
- Kapanma Hızı

Elde Edilen Grafik Örneđi



Sonuç

10000 iterasyon ve 100 kez programın
çalıştırılması sonucunda
elde edilen değerlerin ortalamaları alınmış
ve bunların standart sapmaları
hesaplanmıştır

Senaryo 1

Algoritma	Ortalama	Standart Sapma
ABC	112.66	0.38
ACO	1112.13	2.27e(-13)
DEA	110.26	1.01
GA	38.69	6.10
GOA	22.34	2.84
GWO	1015.34	0.38
PSO	20.95	2.61

Senaryo 1

Optimizasyon Sonuçları:

PSO ile Elde Edilen En İyi Çözüm:

DJI Mavic Air 2 -> İstasyon A

Parrot Anafi -> İstasyon B

Skydio 2 -> İstasyon C

DJI Phantom 4 Pro -> İstasyon E

Bu dronlar hiçbir istasyona atanmadı: Autel Robotics EVO 2, Yuunec Typhoon H Pro

PSO ile Elde Edilen En İyi Uygunluk Değeri: 17.32467152366406

Ortalama Mesafe: 164.54393269798456

Ortalama Batarya Kullanımı: 0.4621307142649648

Geçen Süre: 0.4767 saniye

Kapanma Hızı: 20977.82 iterasyon/saniye

Senaryo 2

Algoritma	Ortalama	Standart Sapma
ABC	125.67	0.40
ACO	1112.13	2.27e(-13)
DEA	121.51	0.61
GA	43.48	5.89
GOA	30.18	3.57
GWO	1017.37	0.18
PSO	27.48	3.10

Senaryo 2

Optimizasyon Sonuçları:

GWO ile Elde Edilen En İyi Çözüm:
DJI Mavic Air 2 -> İstasyon B
Parrot Anafi -> İstasyon A
Skydio 2 -> İstasyon E
DJI Phantom 4 Pro -> İstasyon E
Autel Robotics EVO 2 -> İstasyon A
Yuunec Typhoon H Pro -> İstasyon C

GWO ile Elde Edilen En İyi Uygunluk Değeri: 1014.103179353905

Ortalama Mesafe: 282.31456305707394

Ortalama Batarya Kullanımı: 1.0

GWO Çalışma Süresi: 1.5248 saniye

Kapanma Hızı: 6558.42 iterasyon/saniye

Time 0:
Drone DJI Mavic Air 2: Konum: (34.00, 574.00)
Drone Parrot Anafi: Konum: (418.00, 309.00)
Drone Skydio 2: Konum: (64.00, 207.00)
Drone DJI Phantom 4 Pro: Konum: (651.00, 376.00)
Drone Autel Robotics EVO 2: Konum: (72.00, 72.00)
Drone Yuunec Typhoon H Pro: Konum: (412.00, 160.00)

Optimizasyon Sonuçları:

Time 1:
Drone DJI Mavic Air 2: Konum: (36.30, 575.03)
Drone Parrot Anafi: Konum: (414.09, 310.40)
Drone Skydio 2: Konum: (60.79, 211.64)
Drone DJI Phantom 4 Pro: Konum: (655.76, 373.81)
Drone Autel Robotics EVO 2: Konum: (68.89, 69.32)
Drone Yuunec Typhoon H Pro: Konum: (407.20, 156.50)

Time 2:
Drone DJI Mavic Air 2: Konum: (38.60, 576.07)
Drone Parrot Anafi: Konum: (410.18, 311.81)
Drone Skydio 2: Konum: (57.58, 216.29)
Drone DJI Phantom 4 Pro: Konum: (660.53, 371.62)
Drone Autel Robotics EVO 2: Konum: (65.78, 66.64)
Drone Yuunec Typhoon H Pro: Konum: (402.41, 153.00)

Time 3:
Drone DJI Mavic Air 2: Konum: (40.90, 577.10)
Drone Parrot Anafi: Konum: (406.27, 313.21)
Drone Skydio 2: Konum: (54.37, 220.93)
Drone DJI Phantom 4 Pro: Konum: (665.29, 369.43)
Drone Autel Robotics EVO 2: Konum: (62.67, 63.97)
Drone Yuunec Typhoon H Pro: Konum: (397.61, 149.50)

Senaryo 3

Algoritma	Ortalama	Standart Sapma
ABC	122.56	0.31
ACO	598.51	1.13e(-13)
DEA	119.28	0.94
GA	49.17	4.96
GOA	27.69	1.85
GWO	1018.32	0.51
PSO	25.15	1.59

Senaryo 3

Optimizasyon Sonuçları:

ACO ile Elde Edilen En İyi Çözüm:

DJI Mavic Air 2 -> İstasyon C
Parrot Anafi -> İstasyon D
Skydio 2 -> İstasyon A
DJI Phantom 4 Pro -> İstasyon B
Autel Robotics EVO 2 -> İstasyon E

Bu dronlar hiçbir istasyona atanmadı: Yuunec Typhoon H Pro

ACO ile Elde Edilen En İyi Uygunluk Değeri: 795.5228252806141

Ortalama Mesafe: 372.4540351465504

Ortalama Batarya Kullanımı: 0.8087238845716538

ACO Çalışma Süresi: 0.5836 saniye
Kapanma Hızı: 17134.26 iterasyon/saniye

Drone Pozisyon Geçmişi:

Time 0:
Drone DJI Mavic Air 2: Konum: (363.00, 20.00)
Drone Parrot Anafi: Konum: (17.00, 404.00)
Drone Skydio 2: Konum: (295.00, 215.00)
Drone DJI Phantom 4 Pro: Konum: (404.00, 555.00)

Optimizasyon Sonuçları:

Zaman 0:

İstasyon İstasyon A: Konum: (286.00, 211.00)
İstasyon İstasyon B: Konum: (367.00, 64.00)
İstasyon İstasyon C: Konum: (52.00, 464.00)
İstasyon İstasyon D: Konum: (243.00, 278.00)
İstasyon İstasyon E: Konum: (484.00, 320.00)

Zaman 1:

İstasyon İstasyon A: Konum: (296.00, 204.00)
İstasyon İstasyon B: Konum: (373.00, 72.00)
İstasyon İstasyon C: Konum: (45.00, 469.00)
İstasyon İstasyon D: Konum: (243.00, 273.00)
İstasyon İstasyon E: Konum: (479.00, 313.00)

Zaman 2:

İstasyon İstasyon A: Konum: (305.00, 195.00)
İstasyon İstasyon B: Konum: (371.00, 67.00)
İstasyon İstasyon C: Konum: (41.00, 469.00)
İstasyon İstasyon D: Konum: (251.00, 264.00)
İstasyon İstasyon E: Konum: (476.00, 321.00)

Zaman 3:

İstasyon İstasyon A: Konum: (299.00, 204.00)
İstasyon İstasyon B: Konum: (368.00, 73.00)
İstasyon İstasyon C: Konum: (33.00, 474.00)

Hedefimiz, drone ve yer istasyonu arasındaki iletişim verimliliğini artırmak ve enerji tüketimini minimize etmektir. Sonuç olarak, çeşitli optimizasyon algoritmalarını test ederek, en uygun çözüm yöntemlerini belirledik ve yer istasyonu tasarımımızın performansını optimize ettik.

Teşekkürler

Sorular ve Öneriler