Kırıkkale Üniversitesi

Mühendislik ve Doğa Bilimleri Fakültesi,

Bilgisayar Mühendisliği Bölümü

Yapay Zekâ Dersi Vize Ödev Raporu

Ad Soyad : Nurunnisa KAPLAN

Numara / Bölüm: 200205073 / Bilgisayar Mühendisliği

**1. Genetik Algoritma (GA):**

**a. Problemi Çözerken Yaptıklarım ve Tasarımım**

Genetik algoritmalar, doğada gözlemlenen evrimsel mekanizmalara benzer mekanizmalar kullanarak çalışan en iyileştirme yöntemidir. Bu yöntem kullanılarak, verilen fonksiyona en iyi uyumu sağlayan polinom bulunmaya çalışılmıştır.

**1- İlk adım olarak verinin çağrılması ve görselleştirilmesi**

**2- Başlangıç Popülasyonunun Oluşturulması:** polinomun katsayılarını temsil eden rastgele bireylerden oluşan bir popülasyon oluşturulmuştur. Bu popülasyonun büyüklüğü 100 olarak belirlenmiştir.

**3- Uygunluk Fonksiyonunun (Fitness Function) belirlenmesi:** Her birey için uygunluk fonksiyonu, polinomun hata miktarını ölçen bir MSE (Mean Squared Error) fonksiyonu olarak seçilmiştir. Bu, bireylerin ne kadar iyi olduğunu değerlendirmek için kullanılmıştır.

**4- Seçim yapılması:** Her iterasyonda, popülasyondaki bireylerden en iyi uygunluğa sahip olanlar seçilmiştir. Bu seçim, turnuva seçimi olarak belirlenmiştir. Turnuvada belirli sayıda birey seçilip, bunlar arasından en iyi uygunluğa sahip olan seçilmiştir.

**5- Çaprazlama (Crossover) yapılması:** Seçilen iki bireyden yeni bir birey oluşturulmak için çaprazlama işlemi yapılmıştır. Çaprazlama noktası rastgele seçilmiş ve bireylerin genetik materyalleri değiştirilmiştir.

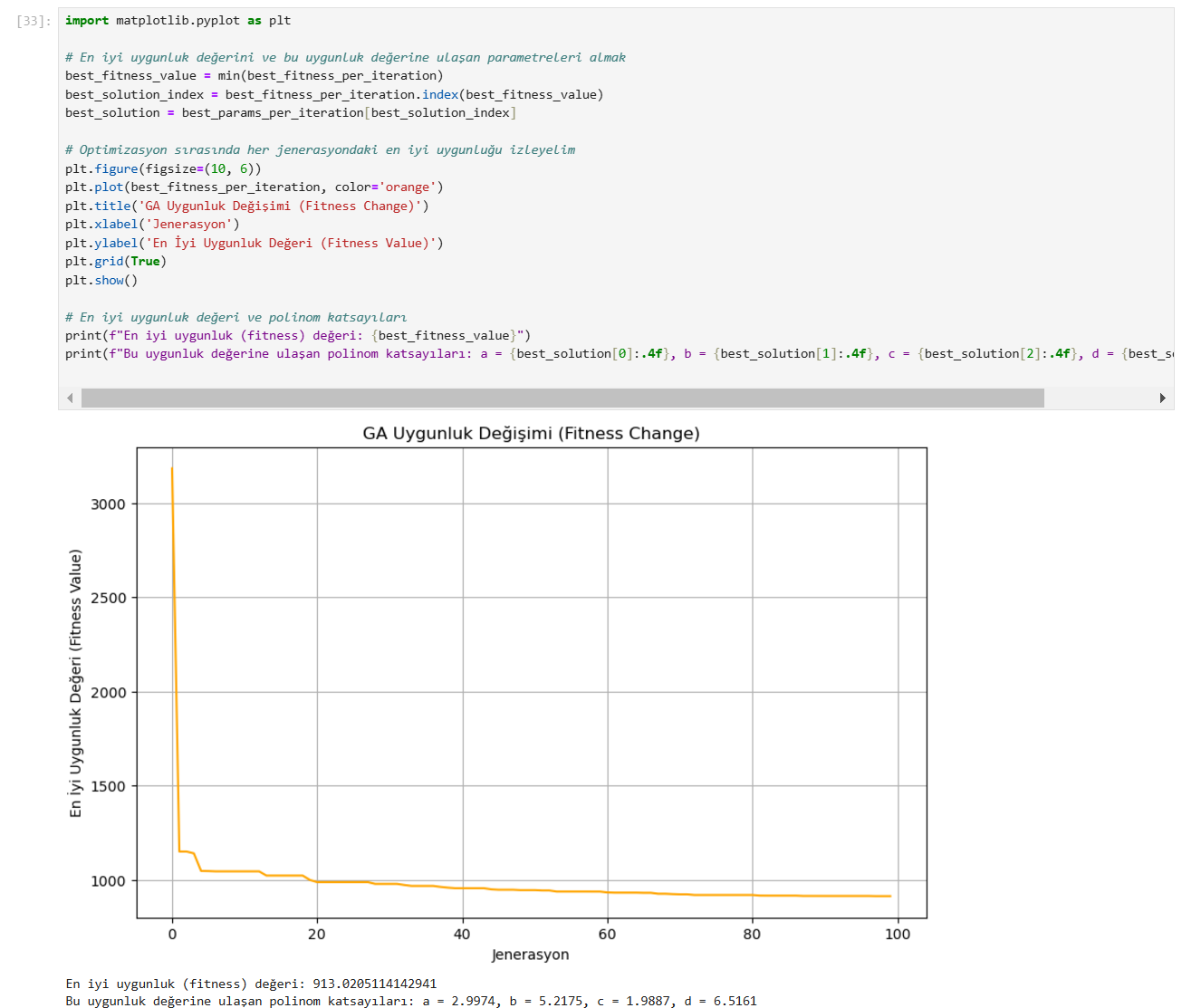
**6- Mutasyon (Mutation) yapılması:** Yeni oluşturulan bireylerde, rastgele bir katsayı küçük bir değişiklikle güncellenmiştir.

**7- Yeni Popülasyonun oluşturulması:** Yeni bireyler oluşturulduktan sonra, popülasyon büyüklüğü hedeflenerek eski ve yeni bireyler karıştırılmış ve yeni popülasyon oluşturulmuştur.

**8- Döngüye sokulması:** Bu işlemler, belirli sayıda nesil boyunca tekrarlanmıştır.

**b. Optimizasyon Yaparken Her İterasyonda Uygunluğun Değişiminin Grafikle Gösterilmesi**

Optimizasyon sırasında uygunluğun (fitness) nasıl değiştiğini izlemek için her neslin sonunda en iyi uygunluk değerleri kaydedilmiştir. Aşağıda, her yinelemede uygunluğun nasıl geliştiğini gösteren bir grafik çizilmiştir:



Bu grafik, genetik algoritmanın her nesilde en iyi uygunluk değerinin nasıl değiştiğini göstermektedir.

**2. Parçacık Sürüsü Optimizasyonu (PSO)**

**a. Problemi Çözerken Yaptıklarım ve Tasarımım**

Parçacık Sürüsü Optimizasyonu (PSO), bir grup parçacık kullanarak çözüm arayışını gerçekleştiren bir optimizasyon tekniğidir. Her parçacık, çözüm uzayında bir pozisyona sahiptir ve bu pozisyonlar zamanla güncellenmektedir. PSO'yu kullanarak benzer şekilde polinom katsayıları optimize edilmiştir.

**1- Veri setinin çağrılması ve görselleştirilmesi**

**2- Başlangıç parçacıklarının oluşturulması:** Parçacıklar rastgele başlangıç konumlarında ve hızlarında başlamıştır. Konum, polinom katsayılarını temsil etmektedir ve hız, bu katsayıların ne kadar hızlı değiştiğini göstermektedir.

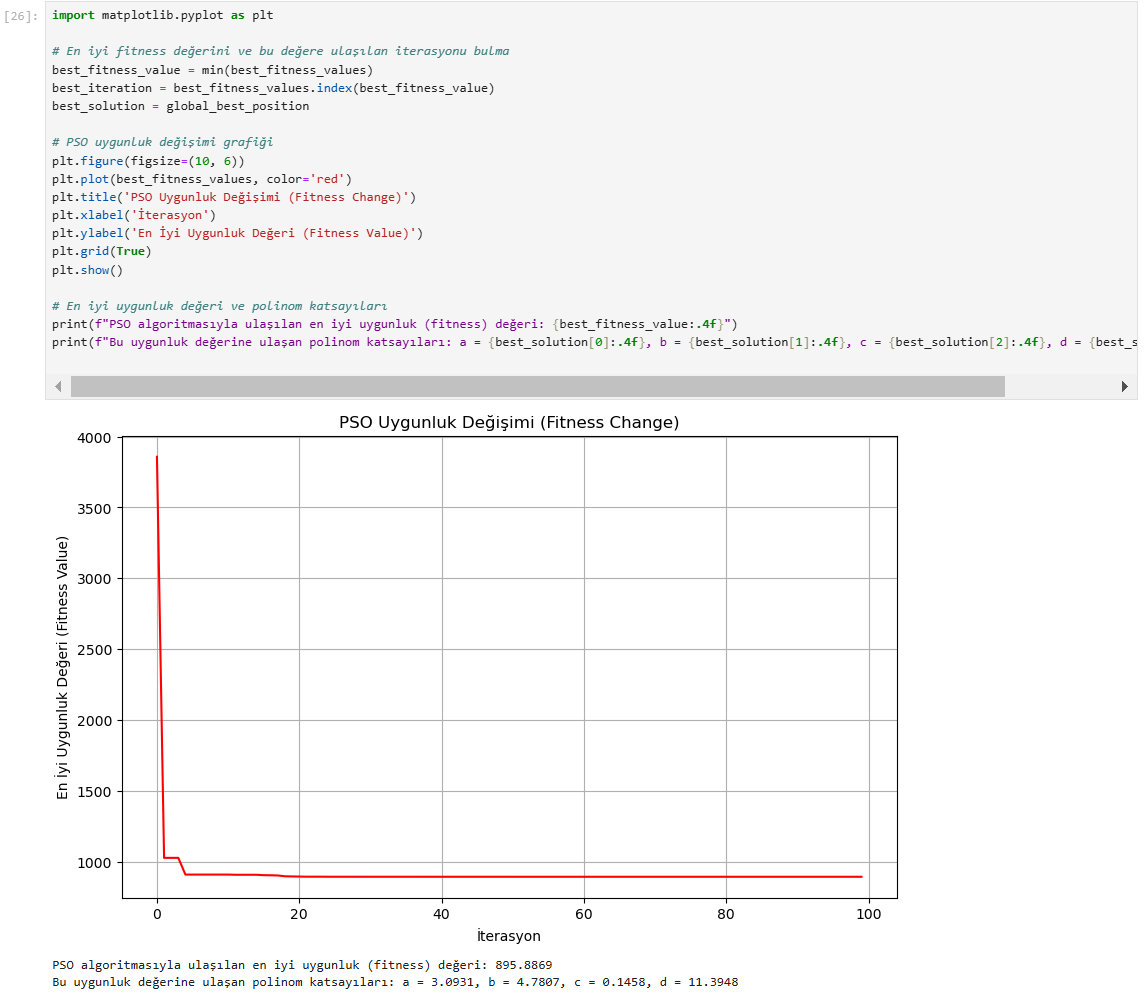
**3- Kişisel ve Küresel En iyi değerlerin bulunması:** Her parçacığın kişisel en iyi konumu (yani en iyi uygunluk değeri) vardır ve tüm parçacıklar arasında küresel en iyi konum (en düşük hata) takip edilir.

**4- Hız ve Konum Güncellemeleri:** Her yinelemede parçacıklar hızlarını ve konumlarını günceller ve en yüksek kişisel ve küresel konumlarına geçer.

**5- Döngüye sokulması:** Bu işlem belirli sayıda iterasyon için tekrarlanmıştır.

**b. Optimizasyon Yaparken Her İterasyonda Uygunluğun Değişimini Grafikle Gösterme**

PSO algoritmasının her iterasyonda uygunluğun (fitness) nasıl değiştiğini gösteren grafiği çizmek için, her iterasyondaki en iyi uygunluk değeri kaydedilmiştir. Aşağıda PSO algoritmasının optimizasyon sürecini izlemek için uygunluk değişimi grafiği çizilmiştir:



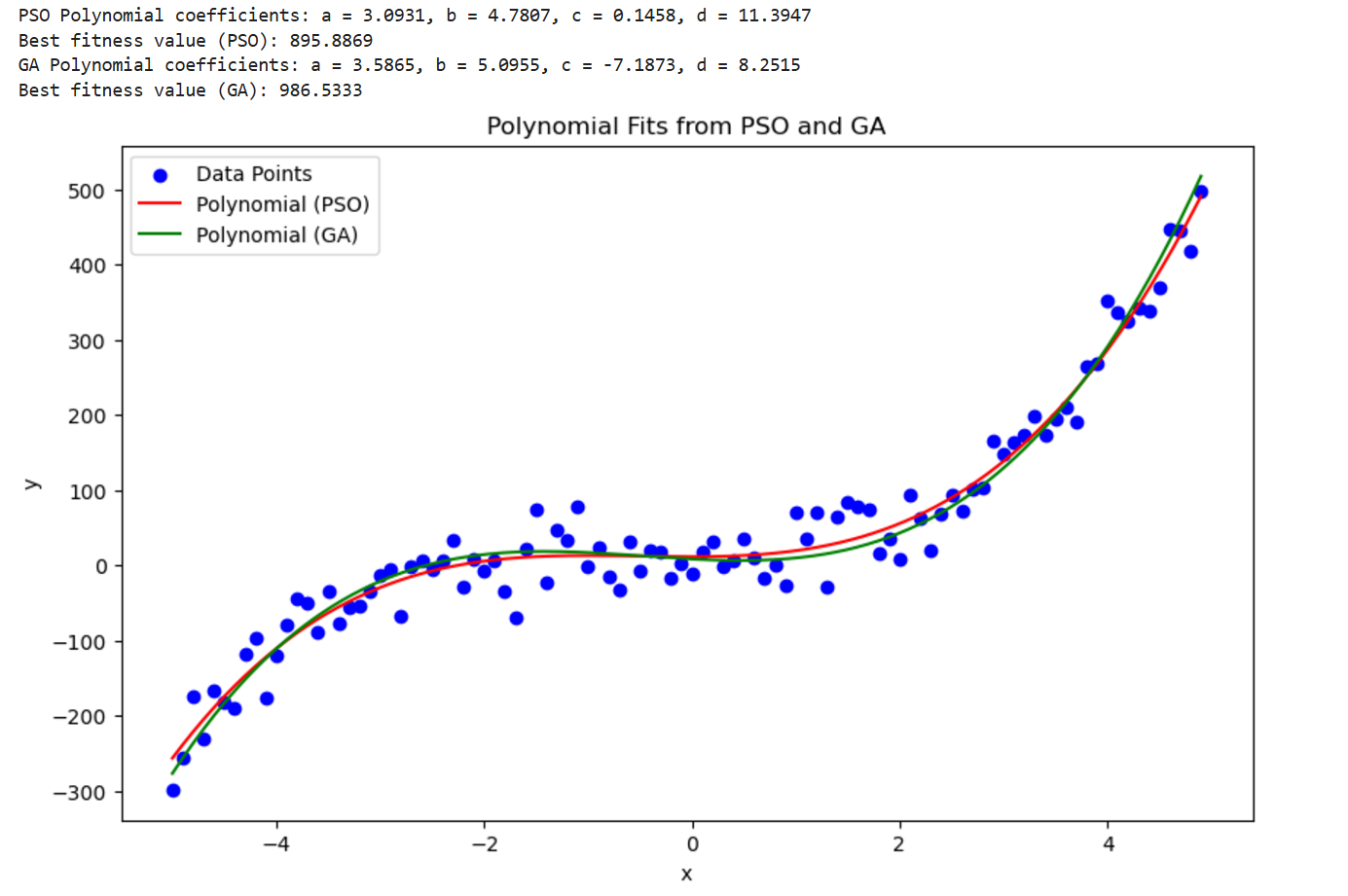
Bu grafik, PSO algoritmasının her iterasyonda en iyi uygunluk değerinin nasıl değiştiğini göstermektedir.

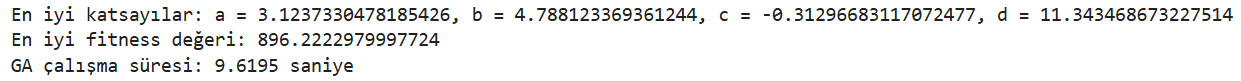
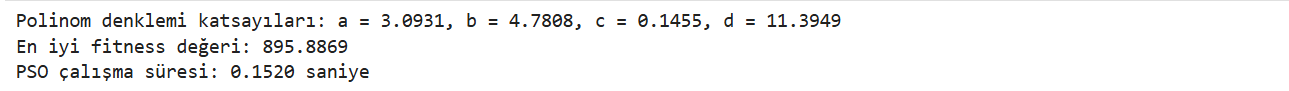
**3. Sonuç: İki Algoritmanın Karşılaştırılması**

Genetik Algoritma (GA) ve Parçacık Sürüsü Optimizasyonu (PSO) arasındaki farkları görmek için birkaç gözlem değerlendirilmiştir:

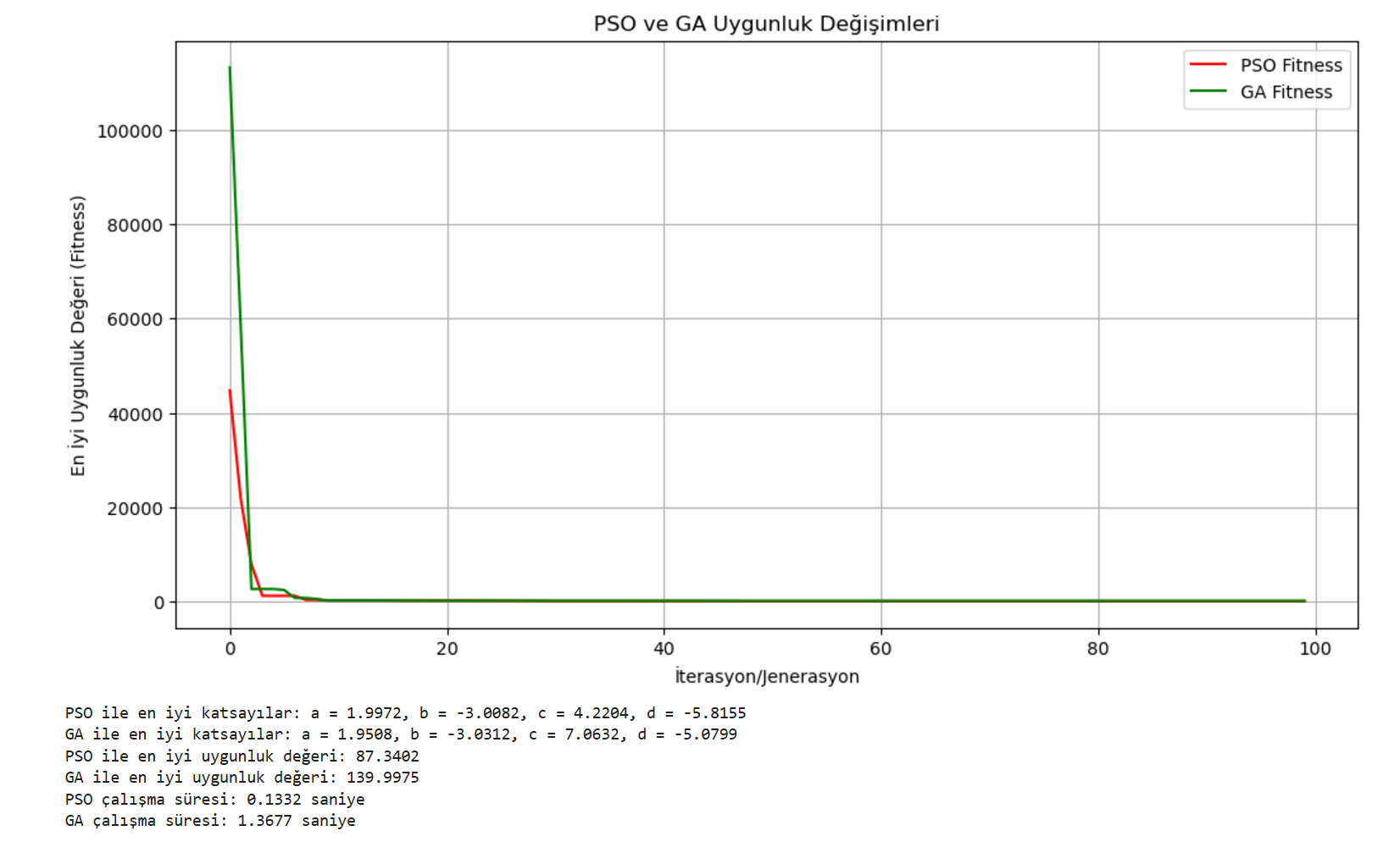
* **Optimizasyon Süresi:** Genetik algoritmalar, her nesilde yalnızca belirli bölümleri iyileştirdiği için daha fazla yineleme gerektirebilir. Buna karşılık, Parçacık Sürü Optimizasyonu (PSO), tüm parçacıkları aynı anda güncelleyerek yerel çözümleri daha hızlı bulabilir. Ancak, çok geniş çözüm alanlarında PSO'nun performansı yavaşlayabilir.
* **Sonuçların Kararlılığı:** GA, daha geniş bir popülasyonu göz önünde bulundurduğundan, çeşitli çözümler üretme olasılığı yüksektir. PSO, daha sınırlı sayıda çözüm üretir ancak global en iyi çözüme daha hızlı ulaşabilir.
* **Çözüm Kalitesi:** Her iki algoritma da en iyi çözümü bulmak için benzer şekilde çalıştırılabilir. Ancak, PSO genellikle yerel çözümlere daha hızlı ulaşırken, Genetik Algoritma (GA) daha geniş bir çözüm alanını keşfederek küresel optimal değerlere ulaşmada daha başarılı olabilir.

| **Özellik** | **GA** | **PSO** |
| --- | --- | --- |
| Optimizasyon Süresi | Daha Uzun Sürebilir | Daha Hızlı Olabilir |
| Çözümün Kararlılığı | Daha Kararlı ve Çeşitli | Daha hızlı ancak sınırlı çeşitlilik |
| Küresel Optimuma Ulaşma | Zorlayıcı, Geniş Arama Alanı | Hızlıca yerel çözüme ulaşabilir |
| Popülasyon Dinamiği | Ebeveynler ve çocuklar arasındaki evrimle gelişir | Parçacıklar arasındaki iletişimle gelişir |

Bu karşılaştırma sayesinde, her iki algoritmanın da avantajlarını ve dezavantajlarını ortaya koyarak, çözüm arayışındaki farklılıkları vurgulanmıştır.



Bunun yanında PSO ve GA algoritmalarının çalışma süreleri tabloda ifade edildiği şekilde karşılaştırılmıştır. Buna göre GA algoritmasının daha uzun sürdüğü görülmüştür.



Yapılan farklı bir denemede PSO ve GA uygunluk değerleri grafik üzerinde gösterilmiştir. Bu çalıştırmada GA algoritmasının uygunluk değerinin daha yüksek olduğu görülmektedir. Ancak yine aynı şekilde GA algoritmasının çalışma zamanının daha fazla olduğu gözlemlenmiştir.