

Hazırlayan: Ahmet Cevahir ÇINAR

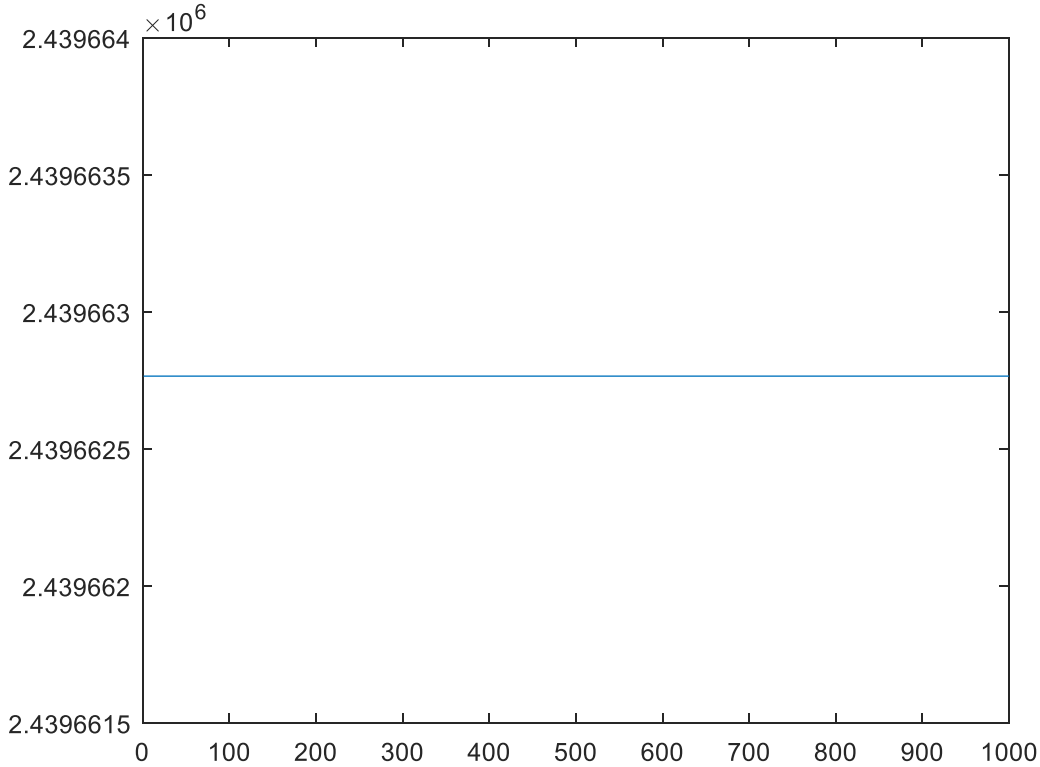
Her türlü soru, sorun, görüş ve öneri için www.ahmetcevahircinar.com.tr adresinden veya ahmetcevahircinar@gmail.com E-Posta adresinden iletişime geçebilirsiniz.

PSO'NUN PARAMETRE DEĞİŞİMLERİNİN BÜYÜK BOYUTLU OPTİMİZASYON PROBLEMİNİN ÇÖZÜMÜNE ETKİSİNİN ARAŞTIRILMASI

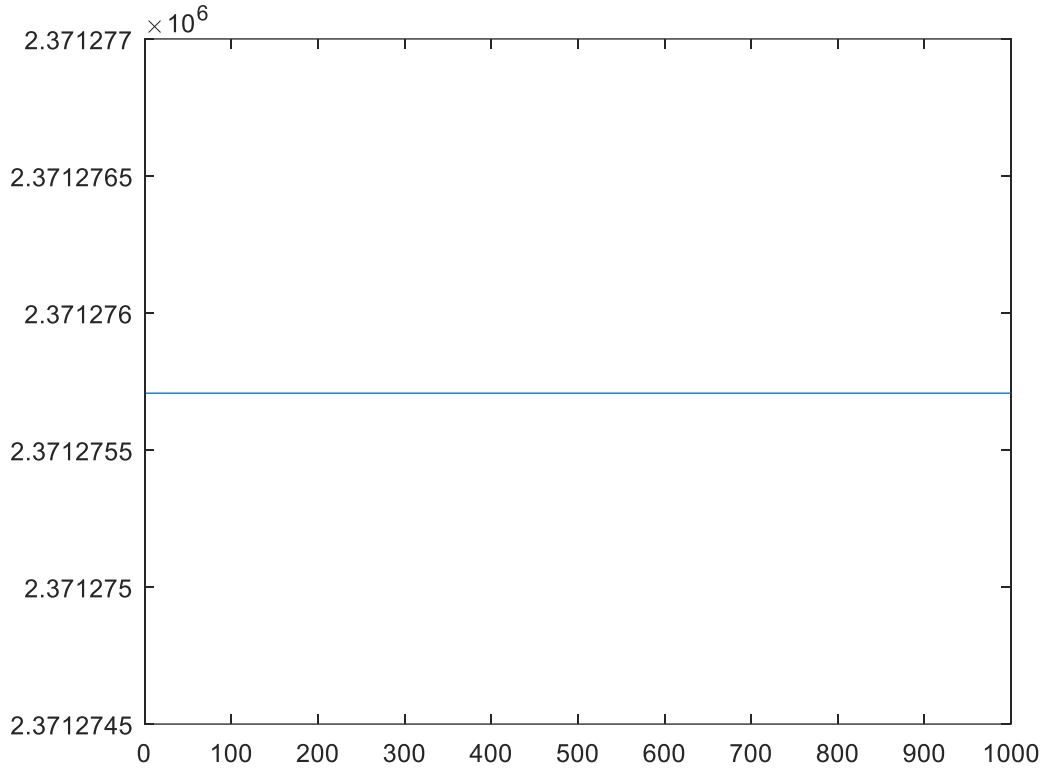
Yalın PSO'da Sphere fonksiyonunun 1000 boyutlu halini çözmek için yapılan kodlamanın hiçbir iyileştirme yapamadığı görülmüştür. Çalıştırma parametreleri:

```
N=20;  
D=1000;  
total_iter=1000;  
dmin=-100;  
dmax=100;  
c1=2.0;  
c2=2.0;
```

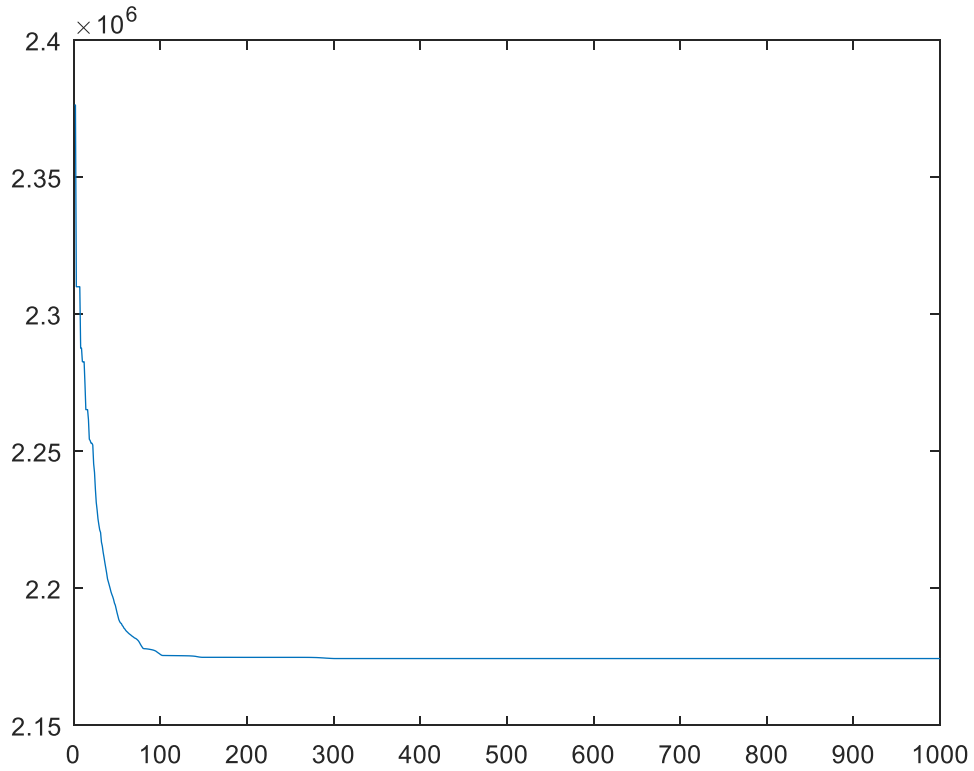
değiştirilerek bir çözüm üretilip, üretilemeyeceği tartışılmış, sonuçlar aşağıda açıklanmıştır.



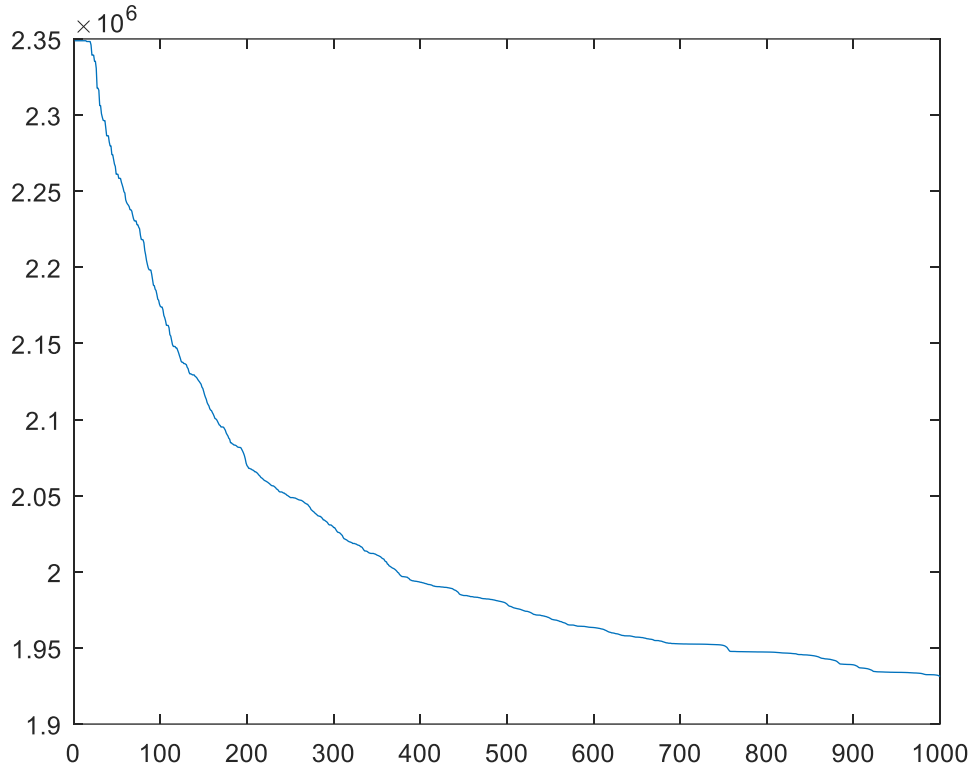
Aynı durumda iken inertia weight olarak 0.9 eklediğimizde yine bir değişim olmadığı görülmüştür.



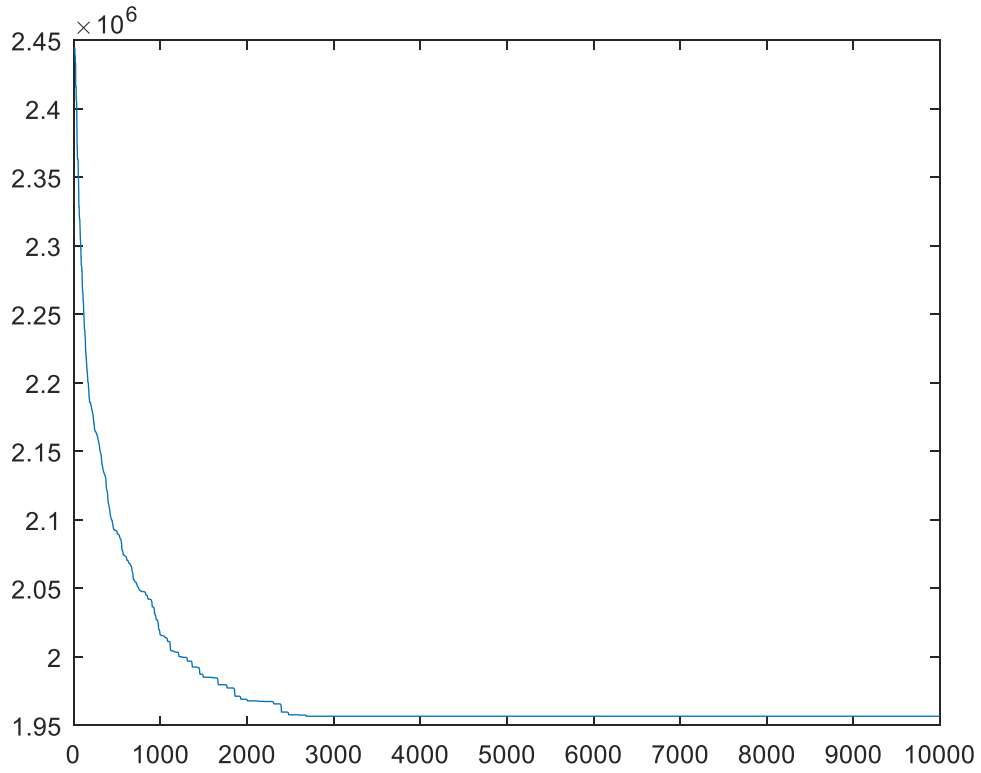
İnertia weight=0,4 olarak alındığı zaman ise aşağıdaki yakınsama grafiği elde edilmiştir.



İnertia weight yerine 0-1 arasında rastgele bir sayı koyulduğunda ise daha iyi bir yakınsama elde edilmiştir.

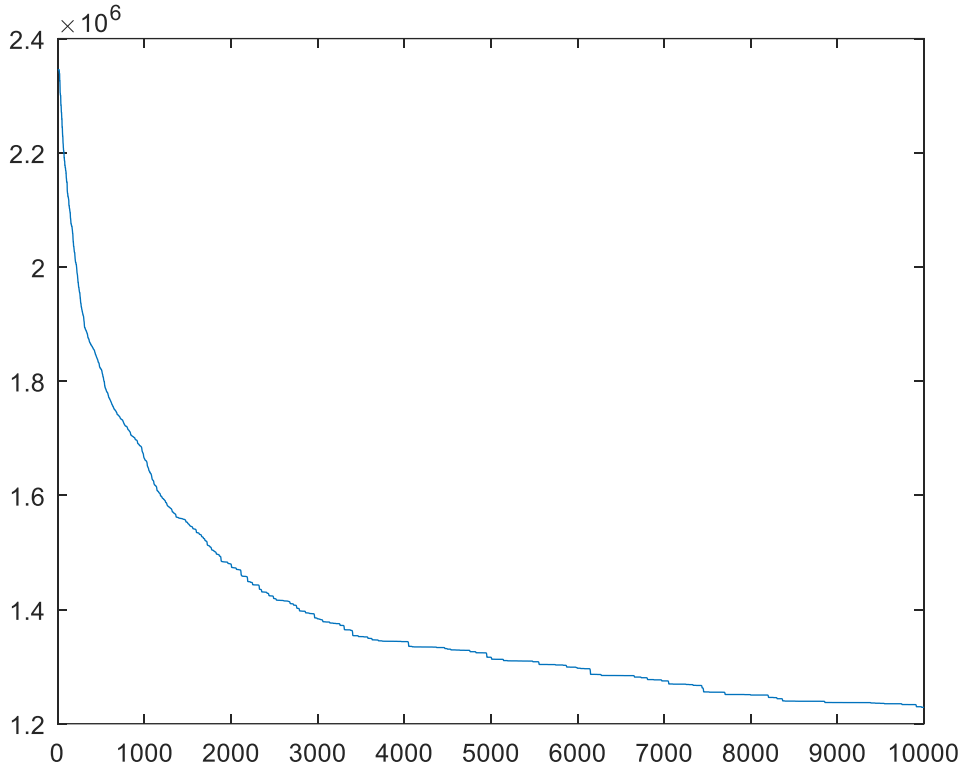


Bu durumun analizi için iterasyon sayısı 1000'den 10000'e çıkarılmıştır.



3000.iterasyondan sonra herhangi bir iyileştirme yapamadığı görülmüştür. Bu noktada 20 olan popülasyon sayısı 100 olarak değiştirilmiştir. Popülasyon sayısı 100 olduğu durumda 10000

iterasyonda sürekli bir iyileştirme ve yakınsama gerçekleştirmiştir. 20'de olmayıp 100'de olmasının sebebi 20 olduğu durumda bireylerin birbirine daha çabuk benzeşmesinden ötürüdür.

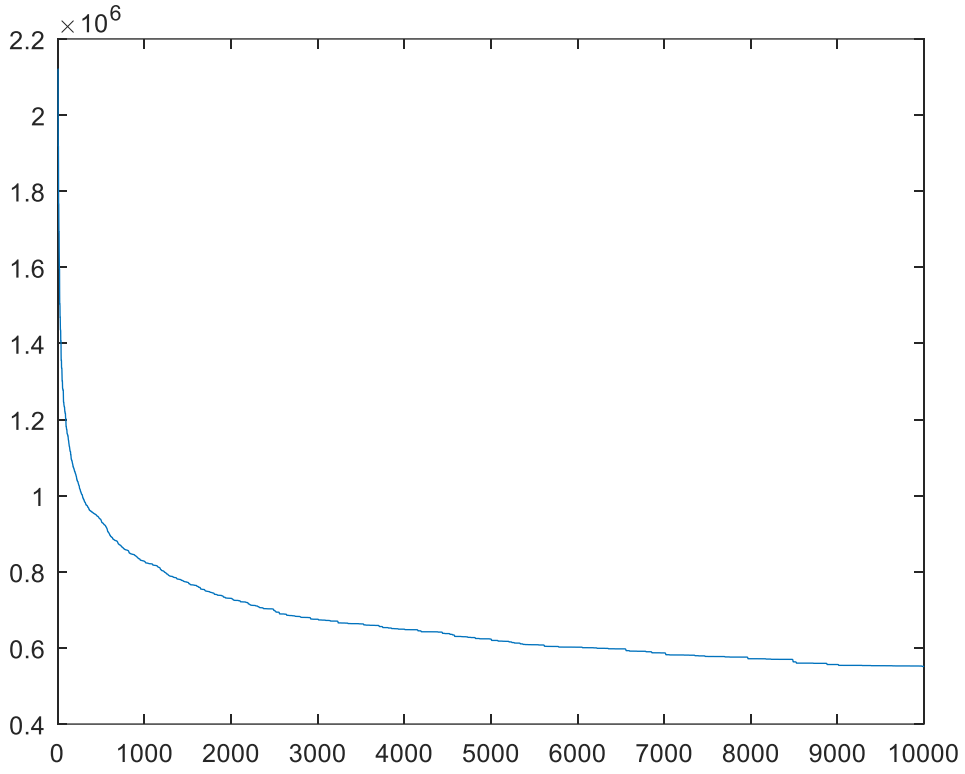


Parçacıkların problem sınırlarına geldiği durumda minimum ve maksimum değerlerini alması sağlanmıştır. İlgili kısımlar değiştirilerek sınırları aşan parçacıklar rastgele olarak yeniden üretildiği zaman daha iyi bir yakınsama grafiği elde edildiği görülmektedir. Bu işlemin nasıl yapıldığı aşağıda gösterilmiştir.

```
particles(i,j)=particles(i,j)+velocity(i,j);  
    if (particles(i,j)>dmax)  
        particles(i,j)=dmax;  
    end  
    if (particles(i,j)<dmin)  
        particles(i,j)=dmin;
```

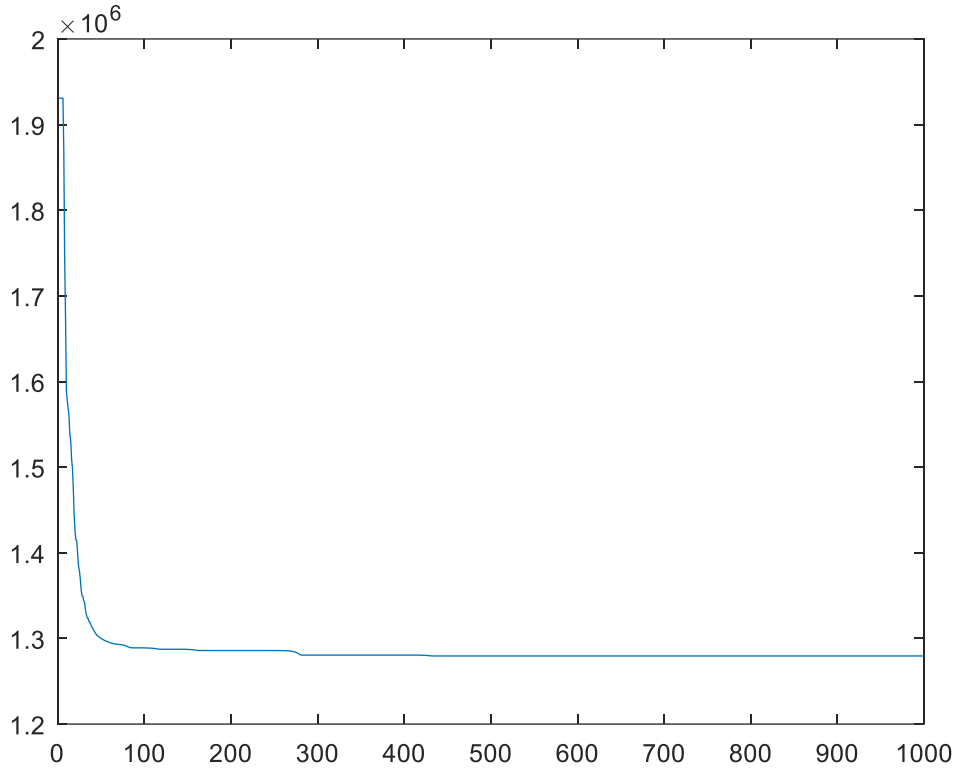
```
particles(i,j)=particles(i,j)+velocity(i,j);  
    if (particles(i,j)>dmax)  
        particles(i,j)=dmin+(dmax-dmin)*rand;  
    end  
    if (particles(i,j)<dmin)  
        particles(i,j)=dmin+(dmax-dmin)*rand;
```

Gelinen noktada her iterasyonda kısmi iyileşme olduğu görülsede de problemimizin optimum değerinin 0 olduğunu bildiğimizden bu çözüm uygun bir çözüm değildir.



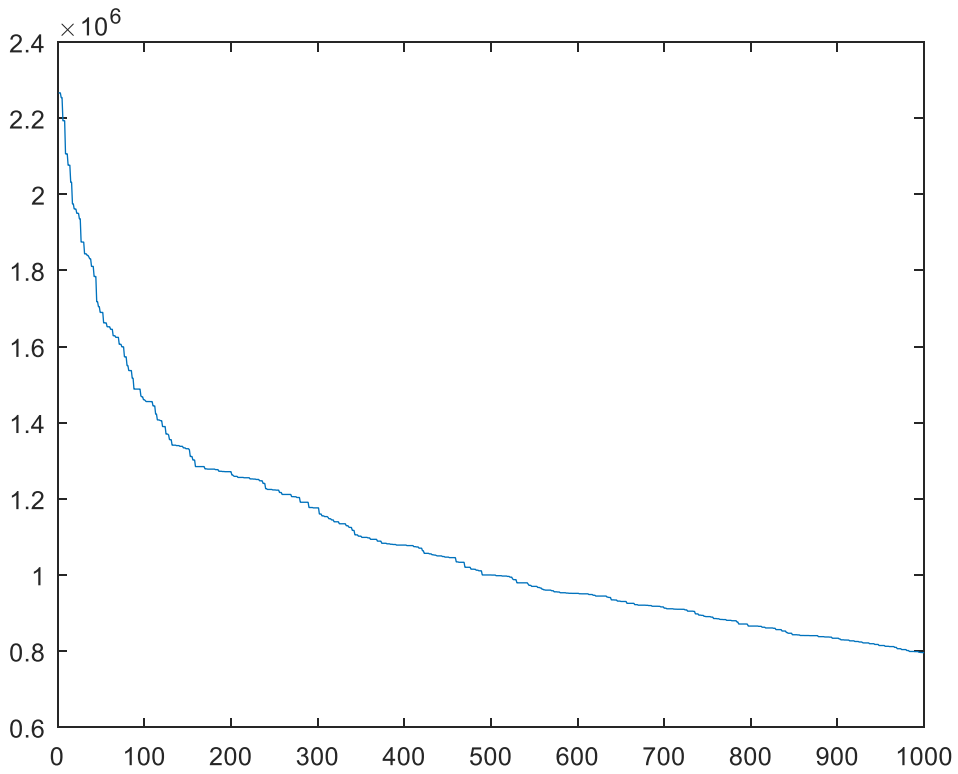
1000 iterasyonda yaklaşık 800000 gibi bir değeri bulabilmişken, 10000.iterasyonda 550000'e kadar inebilmiştir. Bu noktada iterasyon sayısını 1000 yaparak c1 ve c2 parametrelerini değiştirerek bir çözüme gitmeye çalışacağız.

C1=3 c2=1 olduğu durumda:



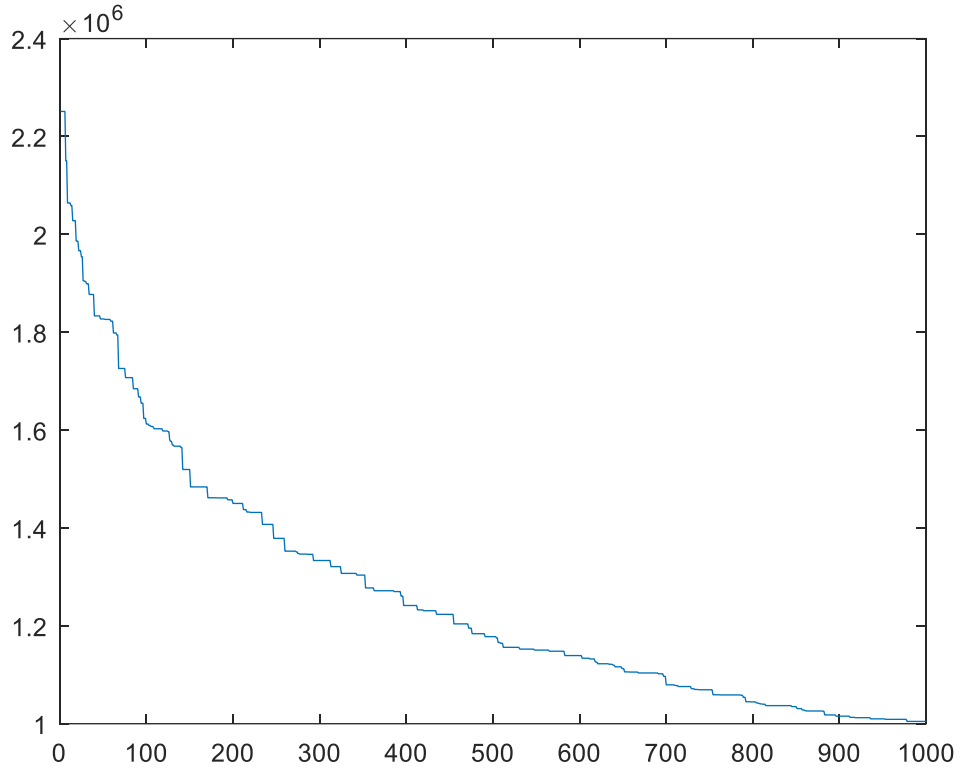
Daha kötü bir sonuç elde ettik.

C1=1 c2=3 olduğu durumda:

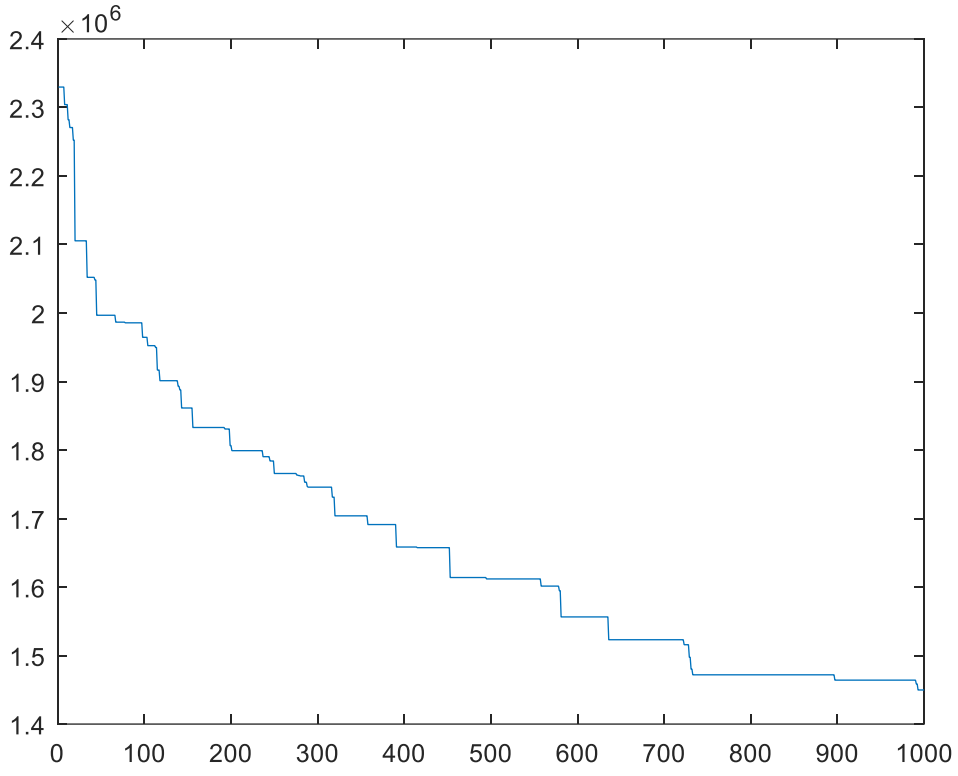


Eşit olduğu duruma yakın bir çözüm elde ettik. C2'nin büyütülmesi çözümün global best'i referans olarak ilerletilmesini sağlamaktadır.

C1=0,5 c2=3,5 alındığında:

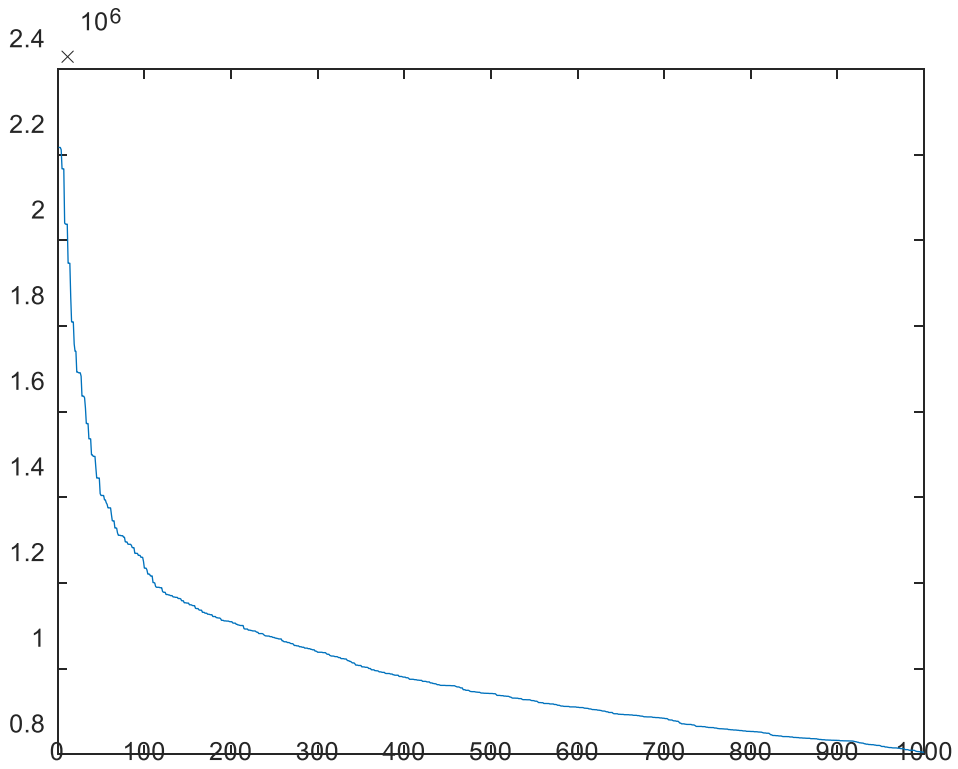


Çözüm biraz daha kötüleşti gibi c1=1 c2=4 olduğunda:



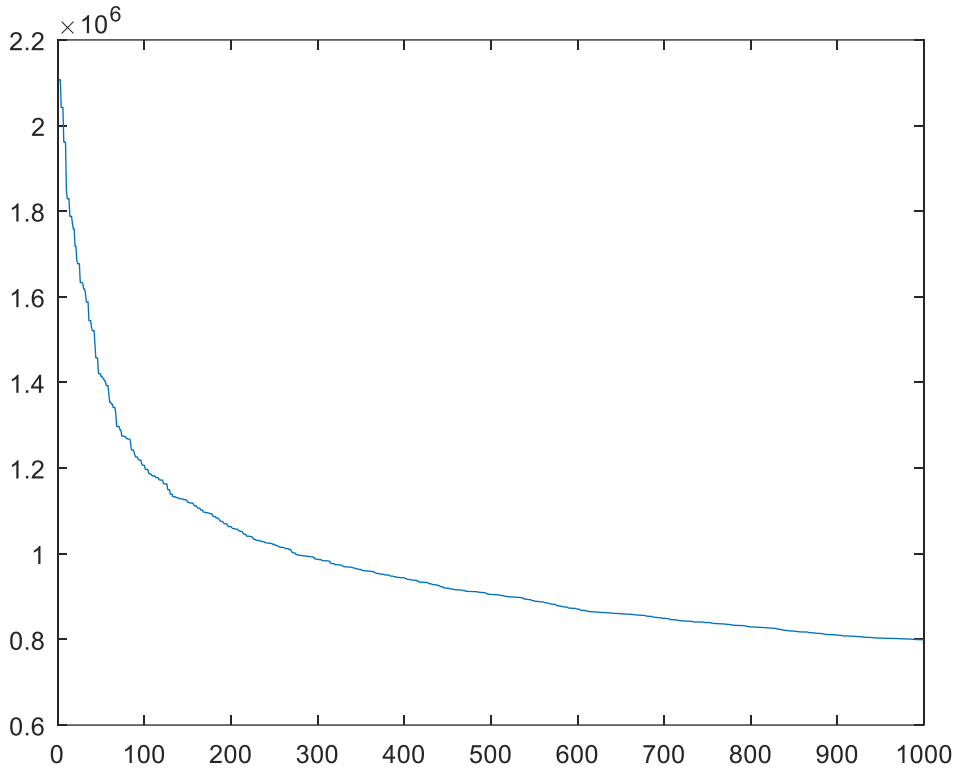
Çözüm 1,5 milyon civarında kalıyor.

C1=1,5 c2=2,5 olduğu durumda:



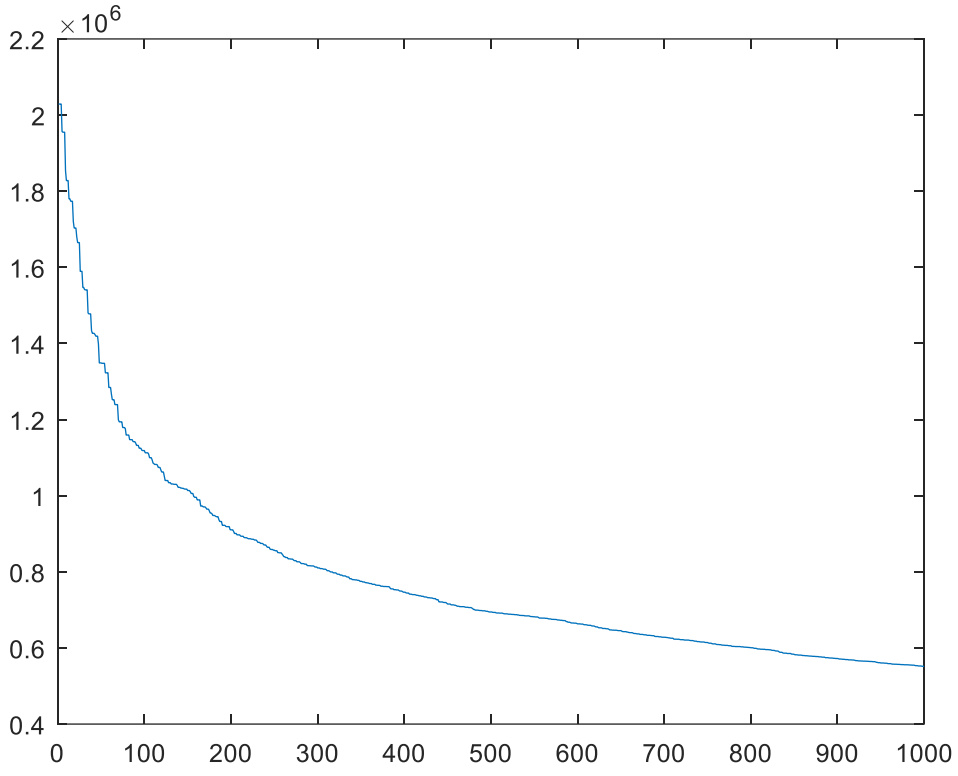
Yine 800000 civarına kadar çözüm indirgendi.

C1=1,6 c2=2,4 olduğu durumda:



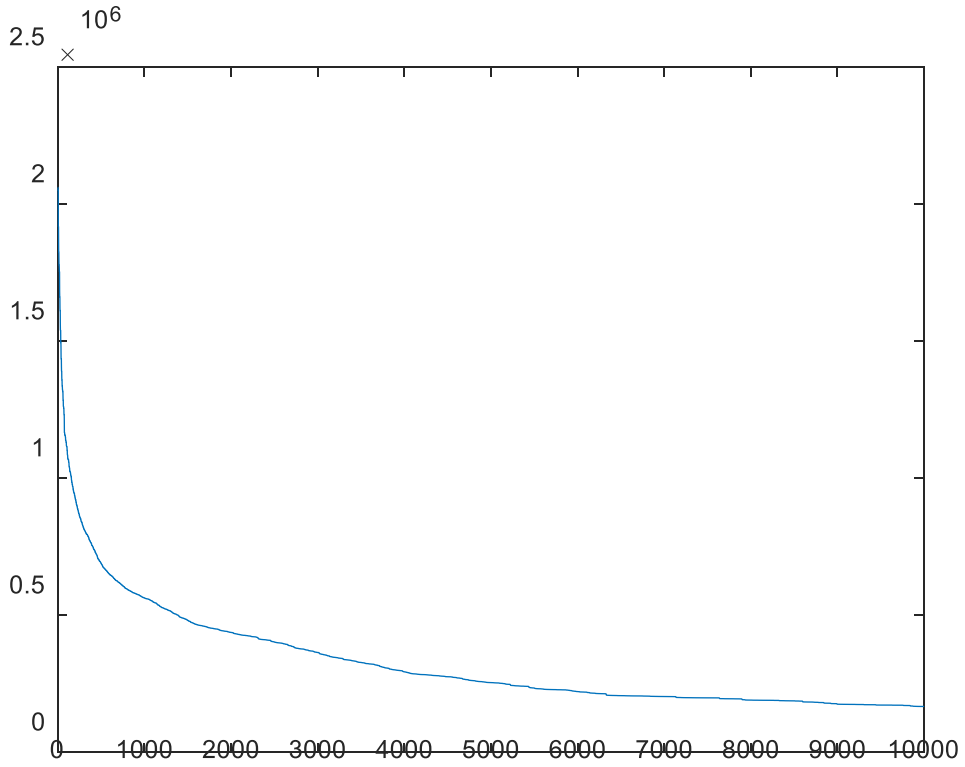
Yine 800000 e yakın bir noktada kaldı. Bu noktada c1 ve c2'nin önerilen değerleri olan 2'nin etrafında küçük değişiklikler belki çözüme katkı sağlayabilir fakat büyük değişimler olumsuz sonuçlara da sebep olabilir.

Son durumun üzerine popülasyon sayısını büyüttüğümüzde ne olur diye bakarsak örneğin N=1000 için:



Çözüm 550000 seviyelerine geldi ve her iterasyonda düşme eğilimi devam etmektedir.

10000 iterasyonda çözüme bir kademe daha yaklaşmıştır: 167021



İlgili kod aşağıdadır:

```
function [] =SeriPSO()

clear all;

F='Sphere';
N=1000;
D=1000;
total_iter=1000;

dmin=-100;
dmax=100;

c1=1.6;
c2=2.4;

particles=zeros(N,D);
obj_particles=zeros(1,N);
pbest=zeros(N,D);
iters=zeros(1,total_iter);

for i=1:N
    for j=1:D
        particles(i,j)=dmin+(dmax-dmin)*rand;
        pbest(i,j)=particles(i,j);
    end;
    obj_particles(i)=eval(strcat(F,'(particles(i,:))'));
end

pbests=obj_particles;
[gbest,index]=min(obj_particles);
gbestParams=particles(index,:);

velocity=zeros(N,D);

for iter=1:total_iter;

    for i=1:N
        for j=1:D
            velocity(i,j)=rand*velocity(i,j)+(c1*rand()*(pbest(i,j)-
particles(i,j)))+(c2*rand()*(gbestParams(j)-particles(i,j)));
            if (velocity(i,j)>dmax)
                velocity(i,j)=dmax;
            end
            if (velocity(i,j)<dmin)
                velocity(i,j)=dmin;
            end
            particles(i,j)=particles(i,j)+velocity(i,j);
            if (particles(i,j)>dmax)
                particles(i,j)=dmin+(dmax-dmin)*rand;
            end
            if (particles(i,j)<dmin)
                particles(i,j)=dmin+(dmax-dmin)*rand;
            end
        end
    end
    obj_particles=ones(1,N);
    for i=1:N
```

```
    obj_particles(i)=eval(strcat(F,'(particles(i,:))'));
end
for i=1:N
    if (obj_particles(i)<pbests(i))
        pbest(i,:)=particles(i,:);
    end
    if (eval(strcat(F,'(pbest(i,:))'))<gbest)
        gbest=eval(strcat(F,'(pbest(i,:))'));
        gbestParams=pbest(i,:);
    end
end;
fprintf('Iter=%d .... gbest=%g\n',iter,gbest);
iters(iter)=gbest;
end
plot(iters);
end
```