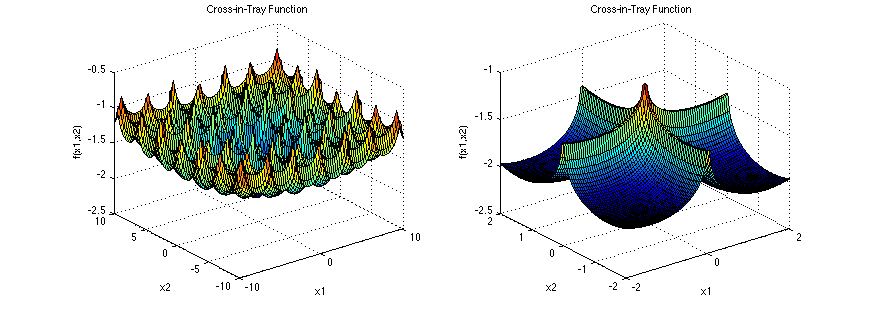
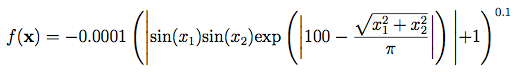
**CROSS-IN-TRAY FUNCTION**

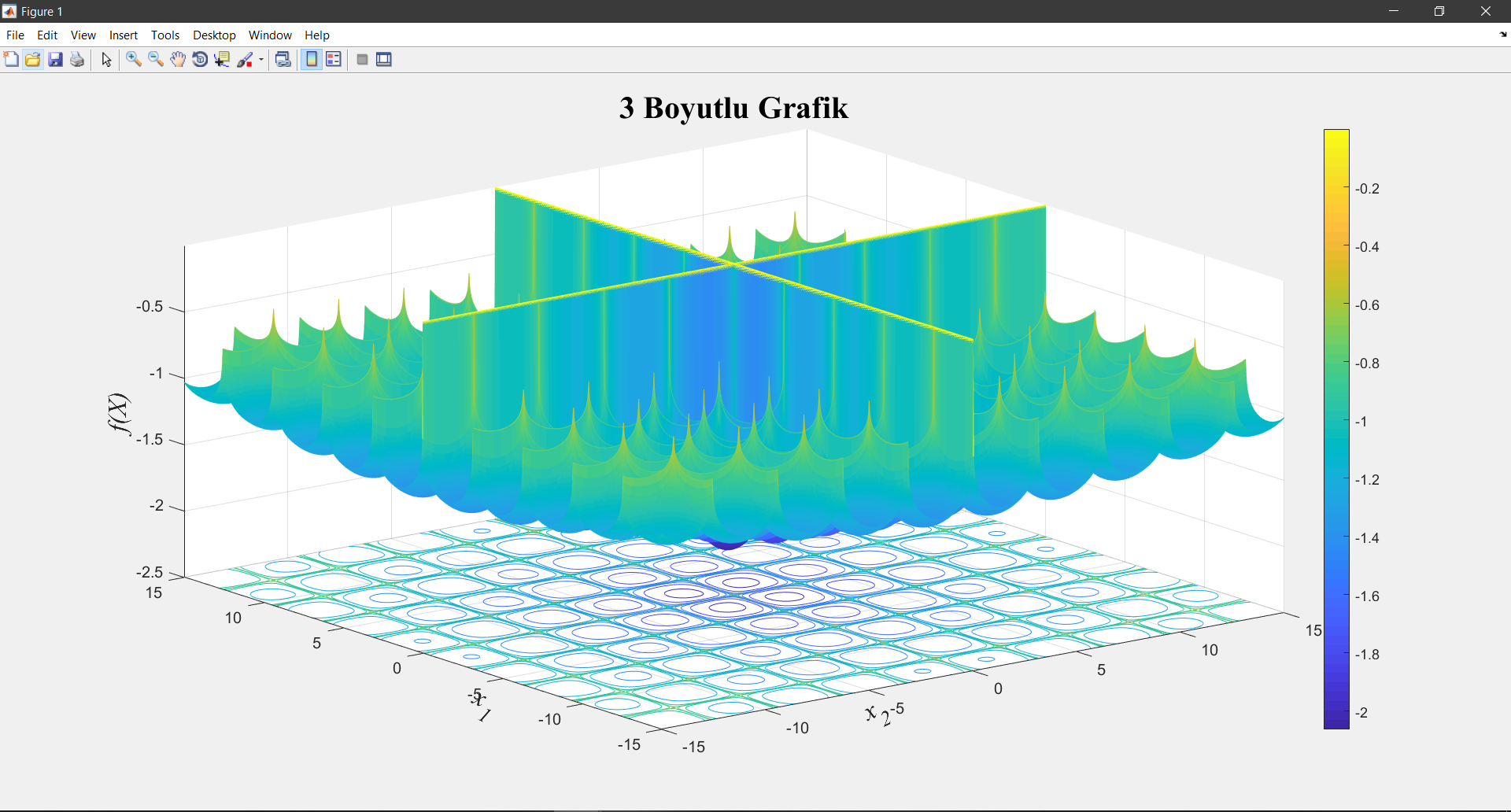






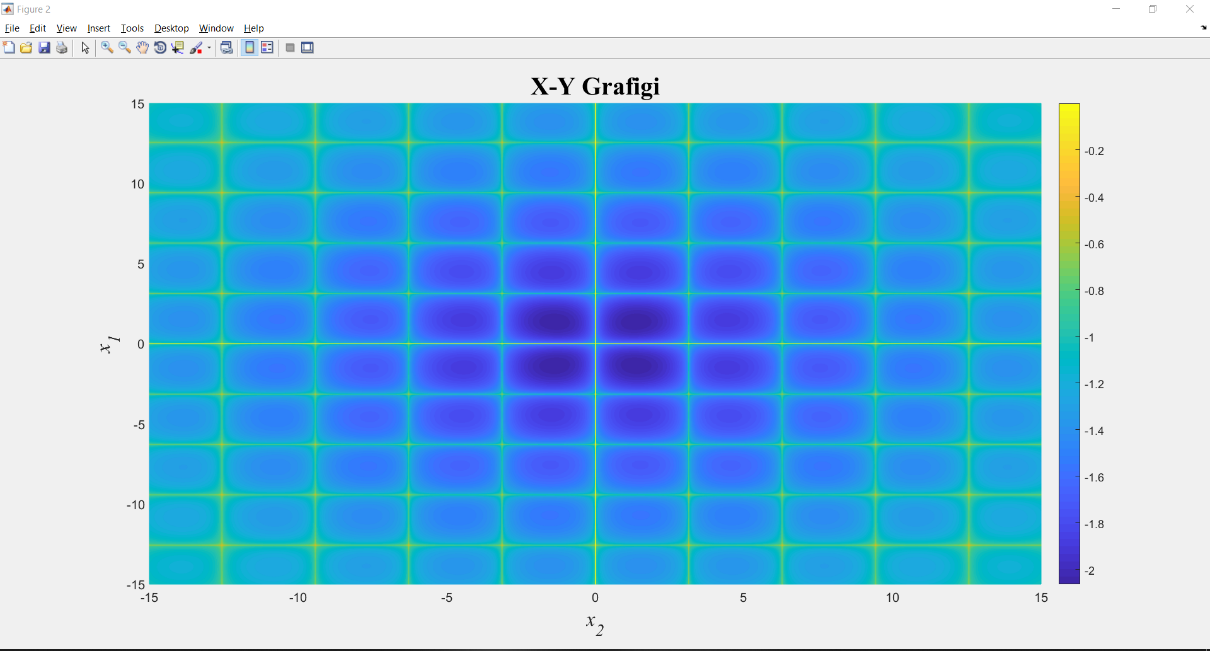
Cross in Tray fonksiyonunda x1,2 noktalarının ±1.3491 değerlerinde fmin(x1,2) = -2.06261 bulunmakta.

**MATLAB Çıktısı**

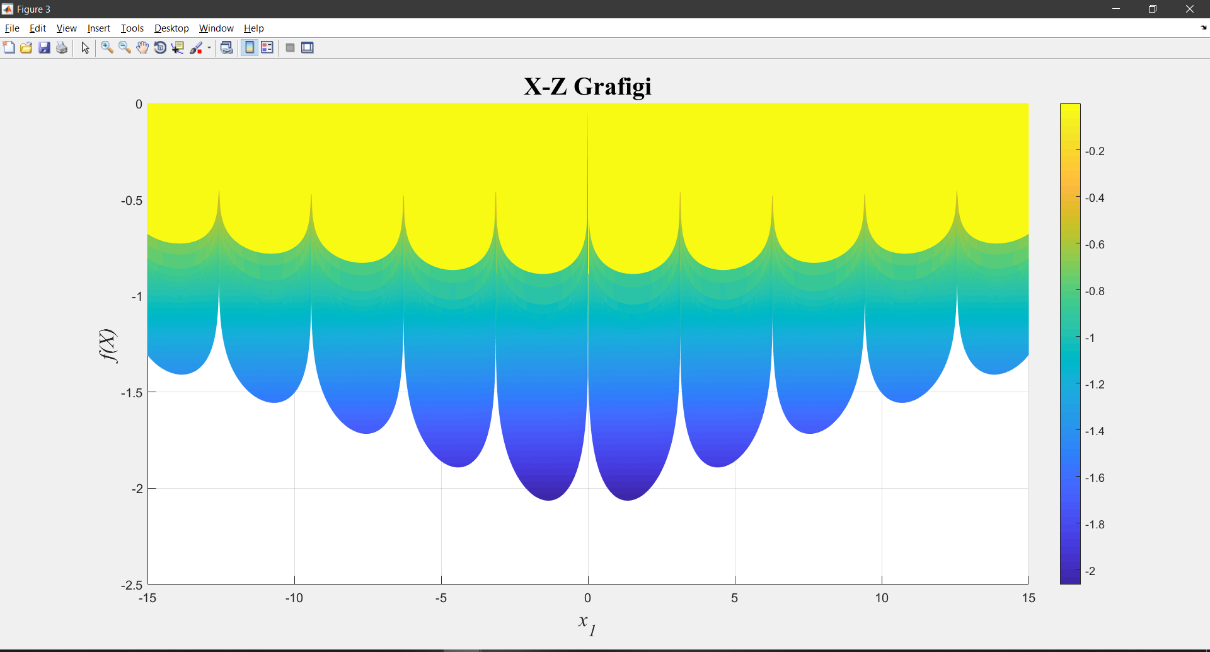


Figure

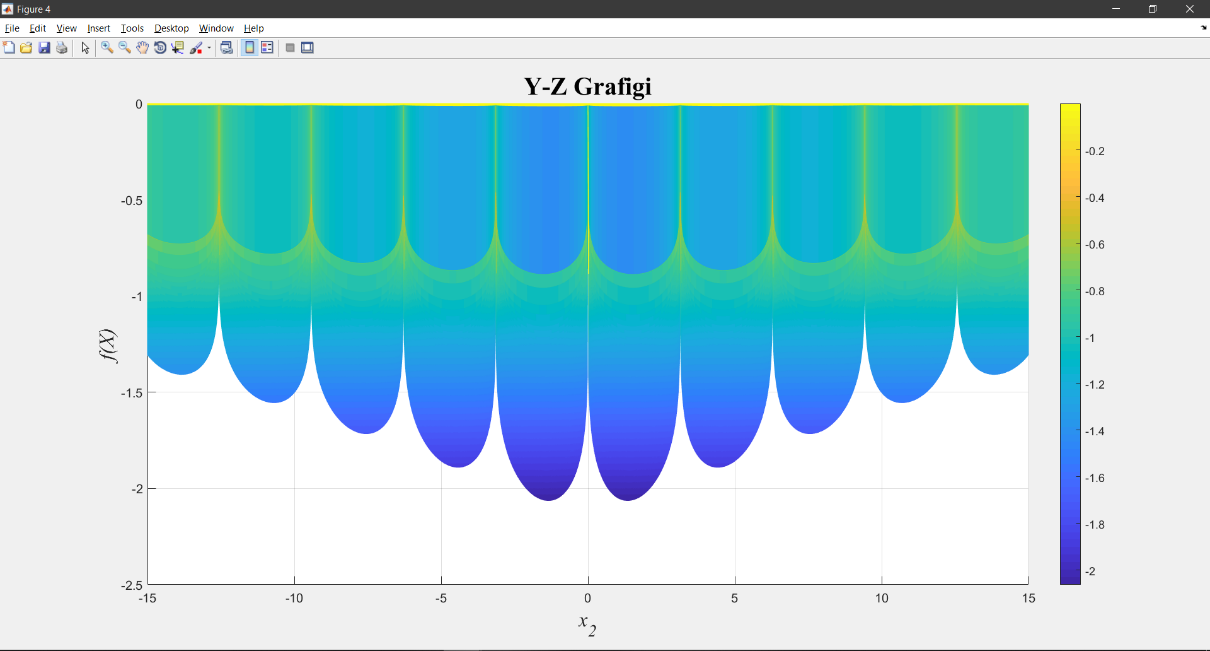
Dikkat edilmesi gerek bir husus herhangi x1,2 değerinin 0 olması halinde MATLAB 3 Boyutlu Grafik plotunda (Figure 1) görüldüğü gibi f(x)=0 yani Z değerimiz 0 olmuş olur.



Figure



Figure



Figure

Aşağıdaki metin kutusu MATLAB çıktısının (Figure 1, Figure 2, Figure 3, Figure 4) kodunu içermekte.

% Cross-In Tray Fonksiyonu

% Baslangic noktalar?n?n araliklari: -15 <= xj <= 15 , j=1,2

% 4 global minimum nokta var: (x1,x2)=+-1.349406608602084

% f(x1,x2)=-2.062611870822739

clear

clc

warning off

x1min=-15;

x1max=15;

x2min=-15;

x2max=15;

R=3000; % 3 boyutlu cizimin cözünürlügünü ayarliyorum

x1=x1min:(x1max-x1min)/R:x1max;

x2=x2min:(x2max-x2min)/R:x2max;

for j=1:length(x1)

for i=1:length(x2)

f(i)=-0.0001\*(abs(exp(abs(100-(sqrt(x1(j).^2+x2(i).^2)/pi)))\*sin(x1(j))\*sin(x2(i)))+1).^0.1; % cross-in tray fonksiyonu

end

f\_tot(j,:)=f;

end

figure(1)

meshc(x1,x2,f\_tot);colorbar;set(gca,'FontSize',12);

xlabel('x\_2','FontName','Times','FontSize',20,'FontAngle','italic');

set(get(gca,'xlabel'),'rotation',25,'VerticalAlignment','bottom');

ylabel('x\_1','FontName','Times','FontSize',20,'FontAngle','italic');

set(get(gca,'ylabel'),'rotation',-25,'VerticalAlignment','bottom');

zlabel('f(X)','FontName','Times','FontSize',20,'FontAngle','italic');

title('3 Boyutlu Grafik','FontName','Times','FontSize',24,'FontWeight','bold');

figure(2)

mesh(x1,x2,f\_tot);view(0,90);colorbar;set(gca,'FontSize',12);

xlabel('x\_2','FontName','Times','FontSize',20,'FontAngle','italic');

ylabel('x\_1','FontName','Times','FontSize',20,'FontAngle','italic');

zlabel('f(X)','FontName','Times','FontSize',20,'FontAngle','italic');

title('X-Y Grafigi','FontName','Times','FontSize',24,'FontWeight','bold');

figure(3)

mesh(x1,x2,f\_tot);view(90,0);colorbar;set(gca,'FontSize',12);

xlabel('x\_2','FontName','Times','FontSize',20,'FontAngle','italic');

ylabel('x\_1','FontName','Times','FontSize',20,'FontAngle','italic');

zlabel('f(X)','FontName','Times','FontSize',20,'FontAngle','italic');

title('X-Z Grafigi','FontName','Times','FontSize',24,'FontWeight','bold');

figure(4)

mesh(x1,x2,f\_tot);view(0,0);colorbar;set(gca,'FontSize',12);

xlabel('x\_2','FontName','Times','FontSize',20,'FontAngle','italic');

ylabel('x\_1','FontName','Times','FontSize',20,'FontAngle','italic');

zlabel('f(X)','FontName','Times','FontSize',20,'FontAngle','italic');

title('Y-Z Grafigi','FontName','Times','FontSize',24,'FontWeight','bold');

**Cross In Tray Fonksiyonunda Genetik Algoritması Kullanılarak Lokal Maksimum-Minimum Noktalarını Belirleme**

**Genetik Algoritma**

* Akış diyagramı

H

H

H

E

E

Bireyler mutasyon geçirecek mi?

Mutasyonu uygula

Bitiş kriteri sağlanıyor mu?

En iyi bireyi bul

Çaprazlamayı uygula

Çiftler çaprazlanacak mı?

Çaprazlanacak çiftleri belirle

Doğal seçilim uygula, hayatta kalan bireyleri belirle

Popilasyondaki kromozomların uygunluk fonksiyonu değerlerini hesapla

Başlangıç popilasyonunu genetic kodlama ile oluştur

Parametre değerlerini belirle

Akış diyagramını baz alarak MATLAB’ de 1 ana fonksiyon 3 alt fonksiyon oluşturdum;

* genetik\_algoritma.m
* dogal\_secilim.m
* crossover.m
* mutasyon.m

**Lokal Minimum Noktasının Bulunması**

* genetik\_algoritma.m

function [ Lokalminimum\_x1\_x2,Lokalminimum\_Z] = genetik\_algoritma ()

Altdegerler=-15; Ustdegerler=15; d=2; Populasyonkumesi=100;

%x1,x2 alt,ust sinirlari -+10 populasyondizi :100 bir matris

popilasyon = unifrnd(Altdegerler, Ustdegerler, [Populasyonkumesi,d]);

iterasyon=1;

Lokalminimum\_Z=1000000000;

%lokalminimum baslangic degerini 1000000000 yapmamin sebebi

%baslangic degerinin cok yuksek bir degerden baslatarak iterasyon

%yapa yapa en optimal sonucu bulmak icindir.

while (iterasyon<100)%100 kere iterasyon aliyor

Z= zeros(Populasyonkumesi,1);

for i=1:Populasyonkumesi

x(i)=popilasyon(i,1);

y(i)=popilasyon(i,end);

Z(i)=(-0.0001\*(abs(exp(abs(100-(sqrt(x(i).^2+y(i).^2)/pi)))\*sin(x(i))\*sin(y(i)))+1).^0.1)

end

if(min(Z)<Lokalminimum\_Z)

Lokalminimum\_Z=min(Z);

idx=find(Z==Lokalminimum\_Z);

Lokalminimum\_x1\_x2=popilasyon(idx,:)

end

x=x';

y=y';

[ arapopilasyon ] = dogal\_secilim( popilasyon, Z, Populasyonkumesi );

[ arapopilasyon ] = crossover( arapopilasyon,Populasyonkumesi,d );

popilasyon= mutasyon( arapopilasyon,Populasyonkumesi,d,Ustdegerler,Altdegerler );

iterasyon=iterasyon+1;

end

end

* dogal\_secilim.m

function [ arapopilasyon ] = dogal\_secilim( popilasyon, Z, Populasyonkumesi )

Z=1/Z; %Burada min degerinin popilasyonda en buyuk oranini olusturmak icin

toplam=sum(Z);

probs=Z/toplam;

cprobs=probs;

% Bu islem %yapiliyor

for i=2:Populasyonkumesi

cprobs(i)=cprobs(i-1)+probs(i);

end

rs=unifrnd(0,1,[Populasyonkumesi,1]);

arapopilasyon=popilasyon;

for i=1:Populasyonkumesi

idx=find(rs(i)<cprobs,1);

arapopilasyon(i,:)=popilasyon(idx,:);

%popilasyonda rastlan sayinin kucuk oldugu ilk index degerleri seciyor

end

* crossover.m

function [ arapopilasyon ] = crossover( arapopilasyon,Populasyonkumesi,d )

%Burada anne babadan gelen yani bir onceki degerlerden ara popilasyon

%Olusturuluyor ve yeni anne baba oluyor ( buble short algoritmasi)

pairs=randperm(Populasyonkumesi);

pcross=unifrnd(0.6,0.95);

for i=1:Populasyonkumesi/2

momidx=pairs(2\*i-1);

fatheridx=pairs(2\*i);

mom=arapopilasyon(momidx,:);

father=arapopilasyon(fatheridx,:);

rs=unifrnd(0,1);

if (rs<pcross)

cpoint=unidrnd(d-1);

dumy=mom(cpoint+1:end);

mom(cpoint+1:end)=father(cpoint+1:end);

father(cpoint+1:end)=dumy;

arapopilasyon(momidx,:)=mom;

arapopilasyon(fatheridx,:)=father;

end

end

end

* mutasyon.m

function [ arapopilasyon ] = mutasyon( arapopilasyon,Populasyonkumesi,d,Ustdegerler,Altdegerler )

%mutasyon orani %5 den buyuk olursa fonksiyonun bozulmasini engellemek icin

%yasanan deger kaybi kadar fonksiyona ekleme yapiyor

mutasyon\_orani=0.005;

delta=0.05;

rs=unifrnd(0,1,[Populasyonkumesi,d]);

for i =1:Populasyonkumesi

for j=1:d

if (rs(i,j)<mutasyon\_orani)

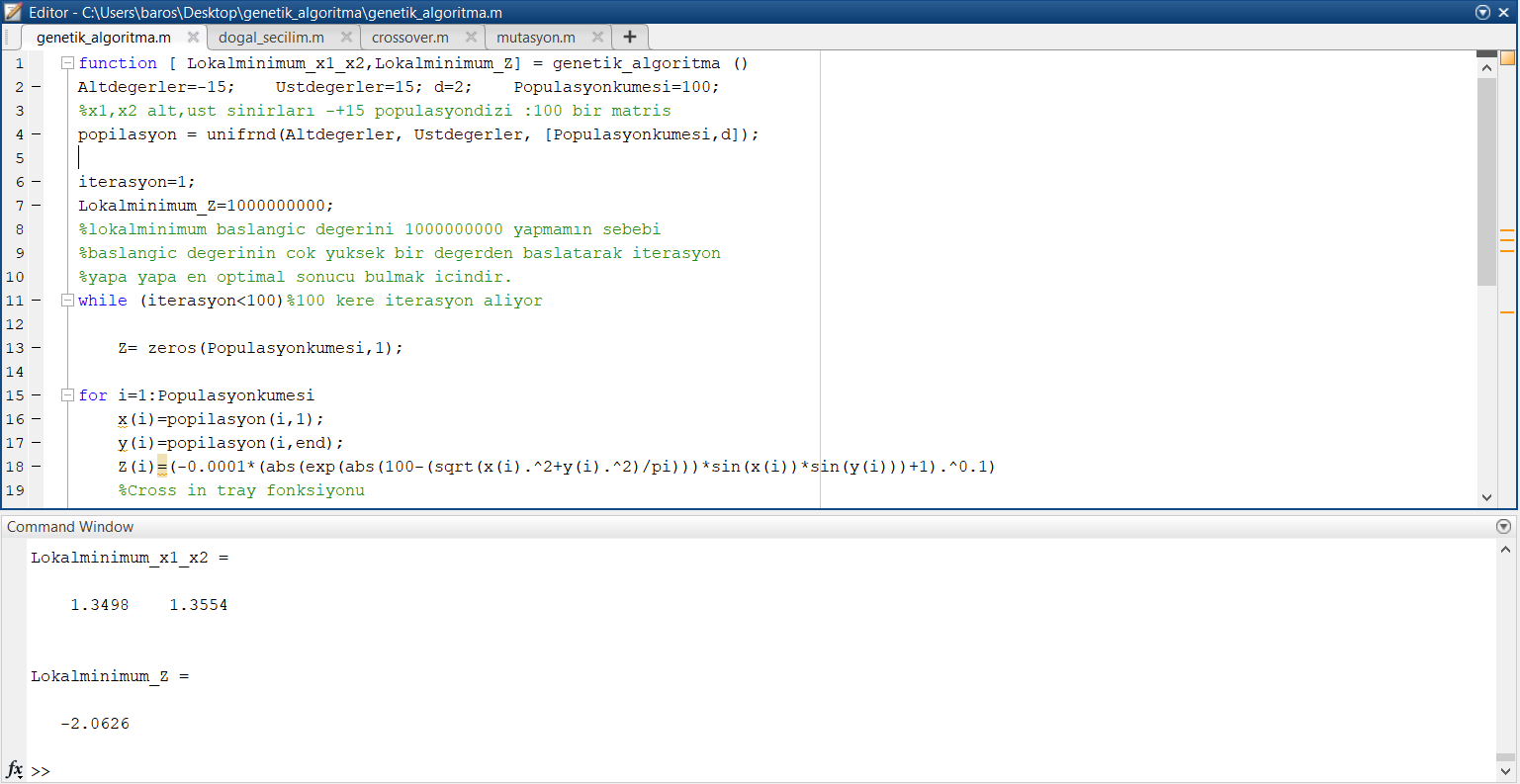
rs2=unifrnd(-1,1);

arapopilasyon(i,j)=arapopilasyon(i,j)\*rs2\*delta\*(Ustdegerler-Altdegerler);

end

end

end



Figure

Cross in Tray fonksiyonunda x1,2 noktalarının ±1.3491 değerlerinde fmin(x1,2) = -2.06261 bulunmaktaydı. Figure 5’ i dikkatli incelenirse hem x1,2 popülasyonunda hemde amaç fonksiyonunda tutarlılık söz konusu. Burada genetik algoritması virgülden sonra 4 hane alındığında %100 başarı sağlamış.

**Lokal Maksimum Noktasının Bulunması**

* genetik\_algoritma.m

function [ Lokalmaksimum\_x1\_x2,Lokmalaksimum\_Z] = genetik\_algoritma ()

Altdegerler=-15; Ustdegerler=15; d=2; Populasyonkumesi=100;

%x1,x2 alt,ust sinirlari -+15 populasyondizi :100 bir matris

popilasyon = unifrnd(Altdegerler, Ustdegerler, [Populasyonkumesi,d]);

iterasyon=1;

Lokmalaksimum\_Z=-1000000000;

%lokalmaksimum baslangic degerini -1000000000 yapmamin sebebi

%baslangic degerinin cok dusuk bir degerden baslatarak iterasyon

%yapa yapa en optimal sonucu bulmak icindir.

while (iterasyon<50)%100 kere iterasyon aliyor

Z= zeros(Populasyonkumesi,1);

for i=1:Populasyonkumesi

x(i)=popilasyon(i,1);

y(i)=popilasyon(i,end);

Z(i)=(-0.0001\*(abs(exp(abs(100-(sqrt(x(i).^2+y(i).^2)/pi)))\*sin(x(i))\*sin(y(i)))+1).^0.1)

%Cross in tray fonksiyonu

end

if(max(Z)>Lokmalaksimum\_Z)

Lokmalaksimum\_Z=max(Z);

idx=find(Z==Lokmalaksimum\_Z);

Lokalmaksimum\_x1\_x2=popilasyon(idx,:)

%burada maksimum degeri bir onceki degere gore kiyaslayip

%degistiriyor

end

x=x';

y=y';

[ arapopilasyon ] = dogal\_secilim( popilasyon, Z, Populasyonkumesi );

[ arapopilasyon ] = crossover( arapopilasyon,Populasyonkumesi,d );

popilasyon= mutasyon( arapopilasyon,Populasyonkumesi,d,Ustdegerler,Altdegerler );

iterasyon=iterasyon+1;

end

end

* dogal\_secilim.m

function [ arapopilasyon ] = dogal\_secilim( popilasyon, Z, Populasyonkumesi )

%Burada popilasyonda en buyuk orani olusturmak icin

toplam=sum(Z);%Bu islem yapiliyor

probs=Z/toplam;

cprobs=probs;

for i=2:Populasyonkumesi

cprobs(i)=cprobs(i-1)+probs(i);

end

rs=unifrnd(0,1,[Populasyonkumesi,1]);

arapopilasyon=popilasyon;

for i=1:Populasyonkumesi

idx=find(rs(i)<cprobs,1);

arapopilasyon(i,:)=popilasyon(idx,:);

%popilasyonda rastlan sayinin kucuk oldugu ilk index degerleri seciyor

end

end

* crossover.m

function [ arapopilasyon ] = crossover( arapopilasyon,Populasyonkumesi,d )

%Burada anne babadan gelen yani bir onceki degerlerden ara popilasyon

%Olusturuluyor ve yeni anne baba oluyor ( buble short algoritmasi)

pairs=randperm(Populasyonkumesi);

pcross=unifrnd(0.6,0.95);

for i=1:Populasyonkumesi/2

momidx=pairs(2\*i-1);

fatheridx=pairs(2\*i);

mom=arapopilasyon(momidx,:);

father=arapopilasyon(fatheridx,:);

rs=unifrnd(0,1);

if (rs<pcross)

cpoint=unidrnd(d-1);

dumy=mom(cpoint+1:end);

mom(cpoint+1:end)=father(cpoint+1:end);

father(cpoint+1:end)=dumy;

arapopilasyon(momidx,:)=mom;

arapopilasyon(fatheridx,:)=father;

end

end

end

* mutasyon.m

function [ arapopilasyon ] = mutasyon( arapopilasyon,Populasyonkumesi,d,Ustdegerler,Altdegerler )

%mutasyon orani %5 den buyuk olursa fonksiyonun bozulmasini engellemek icin

%yasanan deger kayb? kadar fonksiyona ekleme yapiyor

mutasyon\_orani=0.005;

delta=0.05;

rs=unifrnd(0,1,[Populasyonkumesi,d]);

for i =1:Populasyonkumesi

for j=1:d

if (rs(i,j)<mutasyon\_orani)

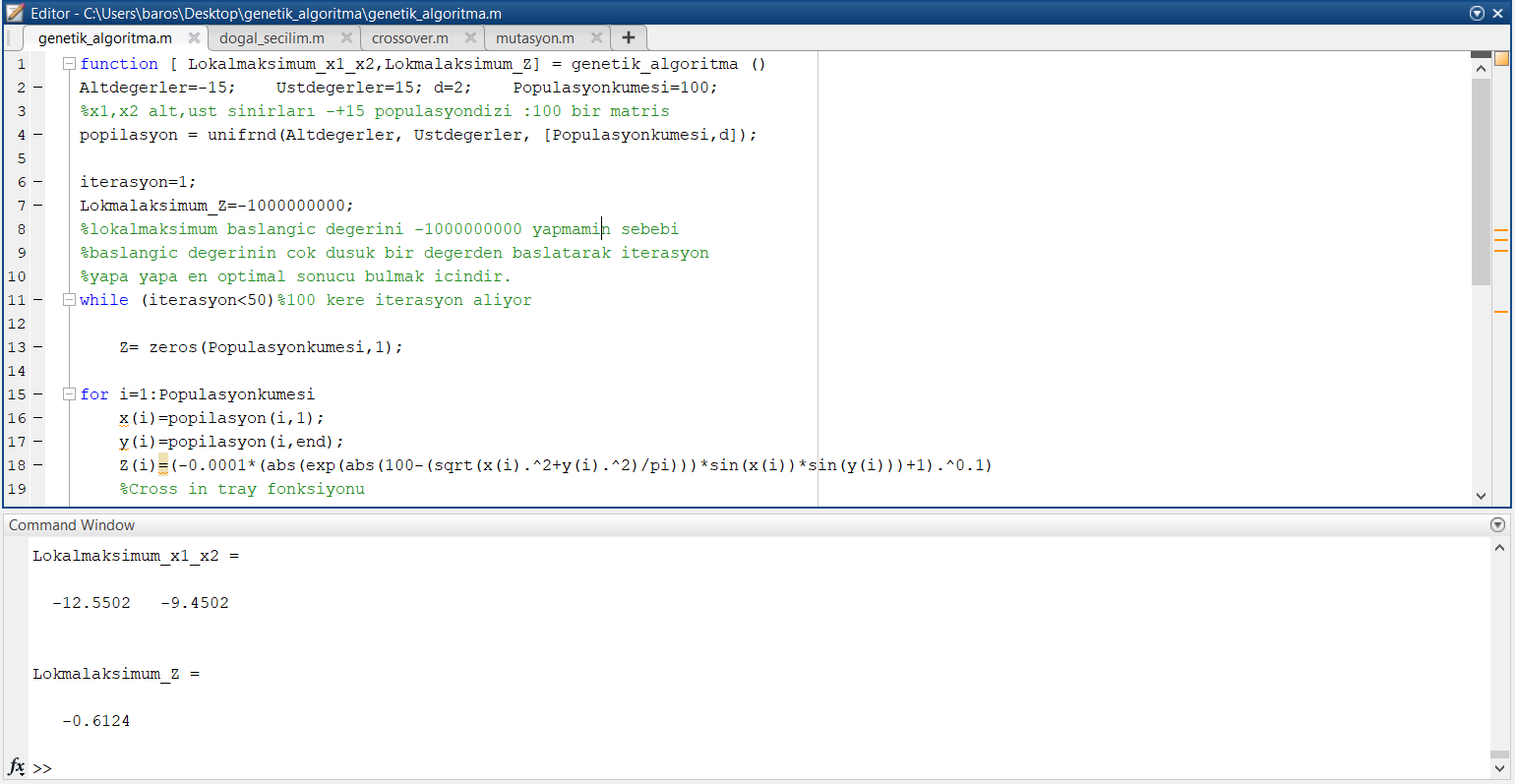
rs2=unifrnd(-1,1);

arapopilasyon(i,j)=arapopilasyon(i,j)\*rs2\*delta\*(Ustdegerler-Altdegerler);

end

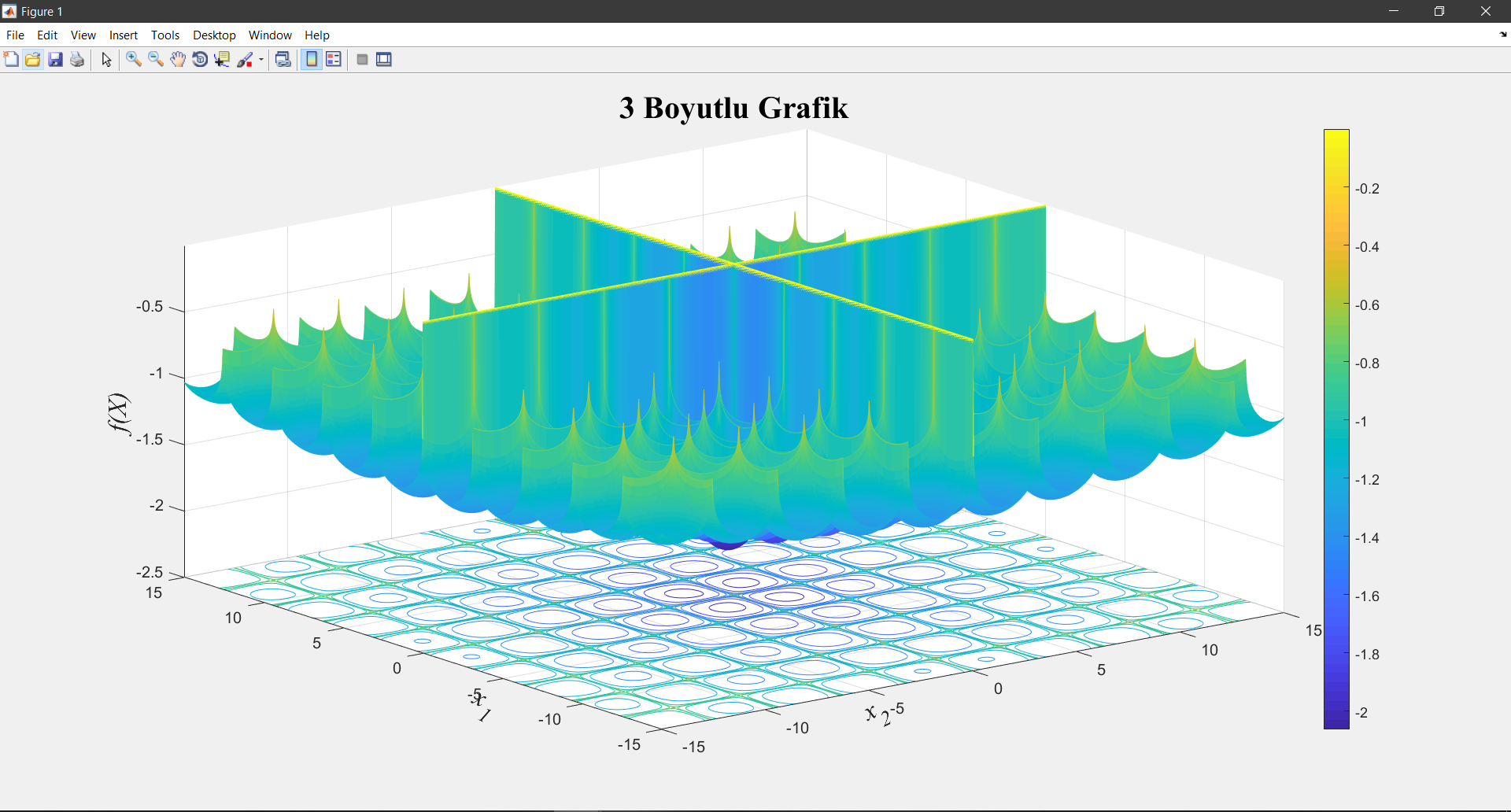
end

end



Figure

Figure 6 ve Figure 7’ e dikkatli bakarsanız Figure 7’ deki sivri uçların -0.7 < Z < -0.5 arasında olduğunu göreceksiniz. Figure 6 tam bu noktalardan bir tanesini genetik algoritmasıyla bulmuş.

Peki neden genetic algoritması Figure 7’ deki Z=0 noktasını bulamadı?

Figure

Bunun nedeni aslında genetic algoritmasının kusuruyla alakalı. Genetik algoritması optimal en yakın değeri bulduğundan yani tam nokta atışı yapmadığı için Cross In Tray fonksiyonu gibi özel fonksiyonlarda x1,2 noktalarında popilasyonunda 0 barındırması gerekir. Bunu barındırsa bile çaprazlama yaptığından mecburen yakınsayan değerleri almak zorunda. x1,2≠0 olduğu durumlarda fmin(x1,2)≡-0.6 değerini aldığı için Figure 6’ da görüldüğü gibi genetik algoritmamız aslında yine doğru sonucu vermiş oldu.

**NOT: Genetik algoritmasını çalıştırmak için her bir durum için oluşturulmuş .m dosyalarını sırayla bir klasöre isimleriyle kaydedin. Daha sonra Command Windows bölümüne;**

* **Minimum değer buldurmak için**
  + **[ Lokalminimum\_x1\_x2,Lokalminimum\_Z] = genetik\_algoritma ()**
* **Maksimum değer buldurmak için**
  + **[ Lokalmaksimum\_x1\_x2,Lokmalaksimum\_Z] = genetik\_algoritma ()**

**Yazın.**

**Kaynak**

* [**https://al-roomi.org/benchmarks/unconstrained/2-dimensions/44-cross-in-tray-function**](https://al-roomi.org/benchmarks/unconstrained/2-dimensions/44-cross-in-tray-function)
* [**http://benchmarkfcns.xyz/benchmarkfcns/crossintrayfcn.html**](http://benchmarkfcns.xyz/benchmarkfcns/crossintrayfcn.html)
* [**https://github.com/topics/cross-in-tray**](https://github.com/topics/cross-in-tray)
* [**https://www.sfu.ca/~ssurjano/crossit.html**](https://www.sfu.ca/~ssurjano/crossit.html)