# CODING ASSIGNMENT 2 20410338

## Minimize x+y-2x^2-y^2+xy

Following functions used in matlab for genetic algorithm

**Population.m**

function Y = population(n)

% n = population size

Y=round(rand(n,40));

**Crossover.m**

function Y=crossover(P,n)

% P = population

% n = number of pairs of chromosomes to be crossovered

[x1 y1]=size(P);

Z=zeros(2\*n,y1);

for i = 1:n

r1=randi(x1,1,2);

while r1(1)==r1(2)

r1=randi(x1,1,2);

end

A1=P(r1(1),:); % parent 1

A2=P(r1(2),:); % parent 2

r2=1+randi(y1-1); % random cutting point

B1=A1(1,r2:y1);

A1(1,r2:y1)=A2(1,r2:40);

A2(1,r2:40)=B1;

Z(2\*i-1,:)=A1; % offspring 1

Z(2\*i,:)=A2; % offspring 2

end

Y=Z;

**Mutation.m**

function Y=mutation(P,n)

% P = population

% n = chromosomes to be mutated

[x1 y1]=size(P);

Z=zeros(n,y1);

for i = 1:n

r1=randi(x1);

A1=P(r1,:); % random parent

r2=randi(y1);

if A1(1,r2)== 1

A1(1,r2) = 0; % flick the bit

else

A1(1,r2) = 1;

end

Z(i,:)=A1;

end

Y=Z;

**Evaluation.m**

function Y=evaluation(P)

[x1 y1]=size(P);

H=zeros(1,x1);

for i = 1:x1

A=bi2de(P(i,1:y1/2));

x=-3+A\*(3-(-3))/(2^(y1/2)-1);

B=bi2de(P(i,y1/2+1:y1));

y=-3+B\*(3-(-3))/(2^(y1/2)-1);

H(1,i)= x+y-2\*x^2-y^2+x\*y;

end

Y=10^6-H;

**Selection**

function [YY1 YY2] = selection(P,F,p)

% P - population, F - fitness value, p - population size

[x y]=size(P);

Y1 = zeros(p,y);

F = F + 10; % adding 10 to ensure no chromosome has negative fitness

% elite selection

e=3;

for i = 1:e

[r1 c1]=find(F==max(F));

Y1(i,:)=P(max(c1),:);

P(max(c1),:)=[];

Fn(i)=F(max(c1));

F(:,max(c1))=[];

end

D=F/sum(F); % Determine selection probability

E= cumsum(D); % Determine cumulative probability

N=rand(1); % Generate a vector constaining normalised random numbers

d1=1;

d2=e;

while d2 <= p-e

if N <= E(d1)

Y1(d2+1,:)=P(d1,:);

Fn(d2+1)=F(d1);

N=rand(1);

d2 = d2 +1;

d1=1;

else

d1 = d1 + 1;

end

end

YY1 = Y1;

YY2 = Fn-10; % substract 10 to return the original fitness

end

**Main Program**

close all

clc

%--------------------------------------------------------------------------

p=100; % Population size

c=30; % number of pairs of chromosomes to be crossovered

m=30; % number chromosomes to be mutated

tg=80; % Total number of generations

%--------------------------------------------------------------------------

figure

title('Blue - Average Red - Minimum');

xlabel('Generation')

ylabel('Objective Function Value')

hold on

P=population(p);

K=0;

[x1 y1]=size(P);

P1 = 0;

for i=1:tg

Cr=crossover(P,c);

Mu=mutation(P,m);

P(p+1:p+2\*c,:)=Cr;

P(p+2\*c+1:p+2\*c+m,:)=Mu;

E=evaluation(P);

[P S]=selection(P,E,p);

K(i,1)=sum(10^6-S)/p;

K(i,2)=10^6-S(1); % best

plot(K(:,1),'b.'); drawnow

hold on

plot(K(:,2),'r.'); drawnow

end

Min\_fitness\_value=min(K(:,2))

P2 = P(1,:); % Best chromosome

% convert binary to real numbers

A=bi2de(P2(1,1:y1/2));

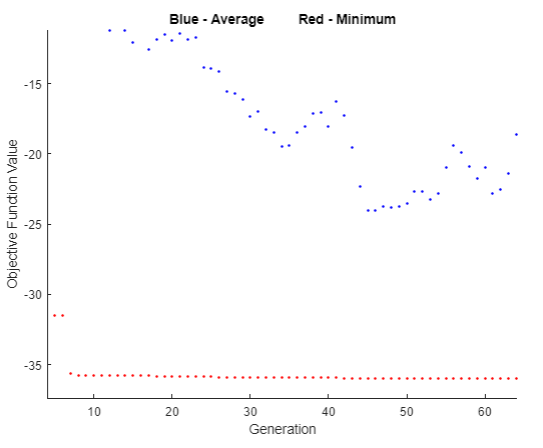
x=-3+A\*(3-(-3))/(2^(y1/2)-1);

B=bi2de(P2(1,y1/2+1:y1));

y=-3+B\*(3-(-3))/(2^(y1/2)-1);

Optimal\_solution=[x y]

# Objective function value VS Generations



# Optimal solution

Min\_fitness\_value =  
  
-35.9897  
  
  
Optimal\_solution =  
  
2.9993 -3.0000