**Appendix A**

**GA** Code in **MATLAB**

clc;

clear all;

MaxIt=0000; % Maximum Number of Iterations

popsize=0000; % Population Size

dimension=0000; % Number of Decision Variables

stringlength=8; % Chromosomes

x\_bound=[0000 0000]; % Upper & Lower Bound of Variables

pm=0.001; % Probability of mutation [Set Accuracy (%)]

ObjFun=@(x) MyFun(x); % Objective Function Nested M-file

% encoding

pop=round(rand(popsize,dimension\*stringlength+1));

%decoding

popsize=size(pop,1);

temp=2.^(stringlength-1:-1:0)/(2^stringlength-1);

for i=1:dimension

bound(i)=x\_bound(i,2)-x\_bound(i,1);

end

for i=1:popsize

for j=1:dimension

m(:,j)=pop(i,stringlength\*(j-1)+1:stringlength\*j);

end

x=temp\*m;

x=x.\*bound+x\_bound(:,1)';

pop(i,dimension\*stringlength+1)=ObjFun;

end

[choice\_number,choice\_k]=max(pop(:,stringlength\*dimension+1));

choice=pop(choice\_k,:);

for i=1:MaxIt

% cross over

match=round(rand(1,popsize)\*(popsize-1))+1;

for i=1:popsize

cpoint=round((stringlength-1)\*rand(1,dimension))+1;

% cross running

for j=1:dimension

child1((j-1)\*stringlength+1:j\*stringlength)=...

[parent1((j-1)\*stringlength+1:(j-1)\*stringlength+cpoint(j)) ...

parent2((j-1)\*stringlength+cpoint(j)+1:j\*stringlength)];

child2((j-1)\*stringlength+1:j\*stringlength)=...

[parent2((j-1)\*stringlength+1:(j-1)\*stringlength+cpoint(j)) ...

parent1((j-1)\*stringlength+cpoint(j)+1:j\*stringlength)];

end

new\_pop(2\*i-1:2\*i,:)=[child1;child2];

end

% mutuation

new\_popsize=size(new\_pop,1);

for i=1:new\_popsize

if rand<pm

mpoint=round(rand(1,dimension)\*(stringlength-1))+1;

for j=1:dimension

new\_pop(i,(j-1)\*stringlength+mpoint(j))=1-...

new\_pop(i,(j-1)\*stringlength+mpoint(j));

end

end

end

%decoding

popsize=size(pop,1);

temp=2.^(stringlength-1:-1:0)/(2^stringlength-1);

[number,k]=max(pop(:,stringlength\*dimension+1));

if choice\_number<number

choice\_number=number;

choice\_k=k;

choice=pop(choice\_k,:);

end

% selection

popsize\_new=size(pop,1);

r=rand(1,popsize);

fitness=pop(:,dimension\*stringlength+1);

fitness=fitness/sum(fitness);

fitness=cumsum(fitness);

for i=1:popsize

for j=1:popsize\_new

if r(i)<=fitness(j)

selected(i,:)=pop(j,:);

break;

end

end

end

pop=selected;

[number,m]=min(pop(:,stringlength\*dimension+1));

pop(m,:)=choice;

end

[value,k]=max(pop(:,stringlength\*dimension+1));

% result

temp=2.^(stringlength-1:-1:0)/(2^stringlength-1);

for i=1:dimension

bound(i)=x\_bound(i,2)-x\_bound(i,1);

end

for j=1:dimension

m(:,j)=pop(k,stringlength\*(j-1)+1:stringlength\*j);

end

x=temp\*m;

x=x.\*bound+x\_bound(:,1)';

% End