clear all

close all

clc

Hasilmax=[];

fitnessvector =[];

XDueler=[];

convergemax = [];

convergeiter = [];

DFDAfit = [];

xmax = [];

minmax = 'max'; % 'max' Maximum or 'min' Minimum

Population = 200; % Total number of duelists in a population

MaxGeneration = 100; % Maximum Generation/Iteration

FightCapabilities = 50; % Fighting Capabilities

Champion = 0.1; % Champion Percentage

ProbLearning = 0.8; % Learning Probability

ProbInnovate = 0.1; % Innovate Probability

Luckcoeff = 0.01; % Luck Coefficient

LuckA = 0; % First Duelist Luck Coefficient

LuckB = 0; % Second Duelist Luck Coefficient

Duelist = [];

Duelisttemp1 = [];

Duelisttemp2 = [];

Duelisttemp3 = [];

DuelistInteger = [];

Datafit = [];

Data1fit = [];

DataSort = [];

ElitDuelist = [];

HMI = [];

DataFDAfit = [];

maxall = [];

Dimension = 3;

UB = [0.625 1300 40]; % Upper Bounds

LB = [0.2 1071 31]; % Lower Bounds

for rc = 1:Dimension

RangeB(rc) = UB(rc) - LB(rc);

end

if (strcmp(minmax,'max'))

mm = 1;

else

mm = -1;

end

%=====Registrasi Duelist=====

Duelist = floor(9\*rand(Population,(FightCapabilities\*Dimension))+rand());

%=====Array to Int=====

for i = 1:Dimension

for j = 1:Population

Duelisttemp1 = Duelist(j,((i\*FightCapabilities-FightCapabilities)+1):(i\*FightCapabilities));

Duelisttemp2 = num2str(Duelisttemp1);

Duelisttemp3 = Duelisttemp2(~isspace(Duelisttemp2));

DuelistInteger(j,i) = str2num(Duelisttemp3);

end

end

Datafit = [];

disp('DA Processing');

for Generasi = 1:MaxGeneration

%=====DA Processing=====

if (Generasi > 1)

clc

Generasi

%=====sortir=====

sort\_fit = sortrows(sort, (FightCapabilities\*Dimension) + 1);

Duelist1 = sort\_fit(randperm(size(sort\_fit,1)),:);

Remain = sort\_fit(round((1-Champion)\*Population) + 1:Population, :);

Winner = [];

X = Duelist1;

N = size(X,1);

if mod(N,2) == 0

M=N;

else

M=N-1;

end

for i=1:M

fitnessvector(i) = X(i,(FightCapabilities\*Dimension) + 1);

end

fitnessvector = fitnessvector';

%=====Setting Duelist=====

for i=1:M

XDueler = X;

end

%=====Setting Duel Arena=====

for i=1:2:M-1

LuckA = (fitnessvector(i)\*(Luckcoeff + rand\*2\*Luckcoeff));

LuckB = (fitnessvector(i+1)\*(Luckcoeff + rand\*2\*Luckcoeff));

if fitnessvector(i)+LuckA <= fitnessvector(i+1)+LuckB

Winner(i) = 0;

Winner(i+1) = 1;

elseif fitnessvector(i)+LuckA > fitnessvector(i+1)+LuckB

Winner(i) = 1;

Winner(i+1) = 0;

end

end

%=====Skill Transfer + Innovate=====

[M,d] = size(XDueler);

XAftermatch = XDueler;

for i=1:2:M-1

if (Winner(i)==1)

p = ceil(((d/2)-1)\*rand\*ProbLearning);

str = ceil(p+1+(((d/2)-2-p)\*rand\*ProbLearning));

XAftermatch(i,:) = [XDueler(i,1:p) XDueler(i+1,p+1:str) XDueler(i,str+1:d)];

for j=1:d

p = rand;

if (p<=ProbInnovate)

XAftermatch(i+1,j) = abs(floor(rand()\*9));

end

end

else

p = ceil(((d/2)-1)\*rand\*ProbLearning);

str = ceil(p+1+(((d/2)-2-p)\*rand\*ProbLearning));

XAftermatch(i+1,:) = [XDueler(i+1,1:p) XDueler(i,p+1:str) XDueler(i+1,str+1:d)];

XAftermatch(i,:) = XDueler(i,:);

for j=1:d

p = rand;

if (p<=ProbInnovate)

XAftermatch(i,j) = abs(floor(rand()\*9));

end

end

end

end

Xnew = XAftermatch;

sort\_fitnew = sortrows(Xnew, (FightCapabilities\*Dimension) + 1);

Duelistnew = sort\_fitnew(round((Champion)\*Population)+1:Population,:);

Duelist = [Duelistnew(:,1:(FightCapabilities\*Dimension));Remain(:,1:(FightCapabilities\*Dimension))];

end;

ElitDuelist = [ElitDuelist; Duelist];

for i = 1:Dimension

for j = 1:Population

Duelisttemp1 = Duelist(j,((i\*FightCapabilities-FightCapabilities)+1):(i\*FightCapabilities));

Duelisttemp2 = num2str(Duelisttemp1);

Duelisttemp3 = Duelisttemp2(~isspace(Duelisttemp2));

DuelistInteger(j,i) = str2num(Duelisttemp3);

end

end

Datafit = [];

for k = 1:Population

for ii=1:Dimension

X0(ii,k) = (((DuelistInteger(k,ii)+1)/(10^FightCapabilities))\*RangeB(ii))+LB(ii);

end

% cost = -((((X0(1,k).^2)+(X0(2,k).^2)).^0.5).\*cos((X0(1,k))-(X0(2,k)))).\*exp(cos(((X0(1,k)).\*(X0(2,k)+5))./7));

fitness = fobjco2egr( X0(:,k));

Datafit = [Datafit; mm\*fitness];

end

Data1fit = Datafit;

[fitnessmax, nmax] = max(Data1fit);

DataFDAfit = [DataFDAfit;fitnessmax];

DuelistMax = Duelist(nmax,:);

DuelistMaxLast = DuelistMax;

Hasilmax = DuelistMax;

sort = [Duelist Datafit];

maxall = [maxall; sort];

for i = 1:Dimension

HasilMaxtemp1 = Hasilmax(1,(((i\*FightCapabilities)-FightCapabilities)+1):(i\*FightCapabilities));

HasilMaxtemp2 = num2str(HasilMaxtemp1);

HasilMaxtemp3 = HasilMaxtemp2(~isspace(HasilMaxtemp2));

HasilMaxInt(1,i) = str2num(HasilMaxtemp3);

end

HMIt = [];

for ij=1:Dimension

HMIt = [HMIt, HasilMaxInt(1,ij)];

end

HMI = [HMI; HMIt];

end

plot(DataFDAfit);

hold on

[fitnessmaxf, nmaxf] = max(DataFDAfit);

for ik=1:Dimension

X0maxfix(ik) = (((HMI(nmaxf,ik)+1)/(10^FightCapabilities))\*RangeB(ik))+LB(ik);

end

X0maxfix

[fitnessmaxf, nmaxf] = max(DataFDAfit)

convergemax = [convergemax;fitnessmaxf];

convergeiter = [convergeiter;nmaxf];

xmax = [xmax;X0maxfix];

DFDAfit = [DFDAfit,DataFDAfit];