%% Initialization

clear

clc

n = 50; % Size of the swarm " no of birds "

bird\_setp = 50; % Maximum number of "birds steps"

dim = 2; % Dimension of the problem

c2 =1.1; % PSO parameter C1

c1 = 0.12; % PSO parameter C2

w =0.9; % pso momentum or inertia

fitness=0\*ones(n,bird\_setp);

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

% initialize the parameter %

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

R1 = rand(dim, n);

R2 = rand(dim, n);

current\_fitness =0\*ones(n,1);

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

% Initializing swarm and velocities and position %

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

current\_position = 10\*(rand(dim, n)-.5);

velocity = .3\*randn(dim, n) ;

local\_best\_position = current\_position ;

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

% Evaluate initial population %

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

for i = 1:n

current\_fitness(i) = Live\_fn(current\_position(:,i));

end

local\_best\_fitness = current\_fitness ;

[global\_best\_fitness,g] = min(local\_best\_fitness) ;

for i=1:n

globl\_best\_position(:,i) = local\_best\_position(:,g) ;

end

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

% VELOCITY UPDATE %

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

velocity = w \*velocity + c1\*(R1.\*(local\_best\_position-current\_position)) + c2\*(R2.\*(globl\_best\_position-current\_position));

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

% SWARMUPDATE %

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

current\_position = current\_position + velocity ;

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

% evaluate anew swarm %

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

%% Main Loop

iter = 0 ; % Iterations’counterwhile ( iter < bird\_setp )

iter = iter + 1;

for i = 1:n,

current\_fitness(i) = Live\_fn(current\_position(:,i)) ;

end

for i = 1 : n

if current\_fitness(i) < local\_best\_fitness(i)

local\_best\_fitness(i) = current\_fitness(i);

local\_best\_position(:,i) = current\_position(:,i) ;

end

end

[current\_global\_best\_fitness,g] = min(local\_best\_fitness);

if current\_global\_best\_fitness < global\_best\_fitness

global\_best\_fitness = current\_global\_best\_fitness;

for i=1:n

globl\_best\_position(:,i) = local\_best\_position(:,g);

end

end

velocity = w \*velocity + c1\*(R1.\*(local\_best\_position-current\_position)) + c2\*(R2.\*(globl\_best\_position-current\_position));

current\_position = current\_position + velocity;

x=current\_position(1,:);

y=current\_position(2,:);

clf

plot(x, y , 'h')

axis([-5 5 -5 5]);

pause(.2)

end % end of while loop its mean the end of all step that the birds move it

[Jbest\_min,I] = min(current\_fitness) % minimum fitness

current\_position(:,I) % best solution