% clear all

% close all

% clc

%tg=Best\_pos

VarName=xlsread("ekim.xlsx")

VarName1=VarName(:,1);

VarName2=VarName(:,2);

VarName3=VarName(:,3);

VarName4=VarName(:,4);

VarName5=VarName(:,5);

VarName6=VarName(:,6);

nVar = 5;

ub = [1 1 1 1 50]\*2;

lb = [-1 -1 -1 -1 -50]\*2;

fobj = @energyPred;

function [Best\_fitness,Best\_position,Convergence\_curve]=DO(Popsize,Maxiteration,LB,UB,Dim,Fobj)

tic;

dandelions=initialization(Popsize,Dim,UB,LB);

dandelionsFitness = zeros(1,Popsize);

Convergence\_curve=zeros(1,Maxiteration);

for i=1:Popsize

dandelionsFitness(1,i)=Fobj(dandelions(i,:));

end

% Calculate the fitness values of initial dandelions.

[~,sorted\_indexes]=sort(dandelionsFitness);

Best\_position=dandelions(sorted\_indexes(1),:);

Best\_fitness = dandelionsFitness(sorted\_indexes(1));

Convergence\_curve(1)=Best\_fitness;

t=2;

while t<Maxiteration+1

%% Rising stage

beta=randn(Popsize,Dim);

rand\_=randperm(Popsize);

alpha=rand()\*((1/Maxiteration^2)\*t^2-2/Maxiteration\*t+1); % eq.(8) in this paper

a=-1/(Maxiteration^2-2\*Maxiteration+1);

b=-2\*a;

c=1-a-b;

k=1-rand()\*(c+a\*t^2+b\*t); % eq.(11) in this paper

if randn()<1.5

for i=1:Popsize

lamb=abs(randn(1,Dim));

theta=(2\*rand()-1)\*pi;

row=1/exp(theta);

vx=row\*cos(theta);

vy=row\*sin(theta);

NEW=rand(1,Dim).\*(UB-LB)+LB;

dandelions\_1(i,:)=dandelions(i,:)+alpha.\*vx.\*vy.\*lognpdf(lamb,0,1).\*(NEW(1,:)-dandelions(i,:)); % eq.(5) in this paper

end

else

for i=1:Popsize

dandelions\_1(i,:)=dandelions(i,:).\*k; % eq.(10) in this paper

end

end

dandelions=dandelions\_1;

% Check boundries

dandelions = max(dandelions,LB);

dandelions = min(dandelions,UB);

%% Decline stage

dandelions\_mean=sum(dandelions,1)/Popsize; % eq.(14) in this paper

for i=1:Popsize

for j=1:Dim

dandelions\_2(i,j)=dandelions(i,j)-beta(i,j)\*alpha\*(dandelions\_mean(1,j)-beta(i,j)\*alpha\*dandelions(i,j)); % eq.(13) in this paper

end

end

dandelions=dandelions\_2;

% Check boundries

dandelions = max(dandelions,LB);

dandelions = min(dandelions,UB);

%% Landing stage

Step\_length=levy(Popsize,Dim,1.5);

Elite=repmat(Best\_position,Popsize,1);

for i=1:Popsize

for j=1:Dim

dandelions\_3(i,j)=Elite(i,j)+Step\_length(i,j)\*alpha\*(Elite(i,j)-dandelions(i,j)\*(2\*t/Maxiteration)); % eq.(15) in this paper

end

end

dandelions=dandelions\_3;

% Check boundries

dandelions = max(dandelions,LB);

dandelions = min(dandelions,UB);

%%

% Calculated all dandelion seeds' fitness values

for i=1:Popsize

dandelionsFitness(1,i)=Fobj(dandelions(i,:));

end

% Arrange dandelion seeds from good to bad according to fitness values

[~,sorted\_indexes]=sort(dandelionsFitness);

dandelions=dandelions(sorted\_indexes(1:Popsize),:);

SortfitbestN = dandelionsFitness(sorted\_indexes(1:Popsize));

%Update the optimal dandelion seed

if SortfitbestN(1)<Best\_fitness

Best\_position=dandelions(1,:);

Best\_fitness=SortfitbestN(1);

end

Convergence\_curve(t)=Best\_fitness;

t = t + 1;

end

time = toc;

end

% \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

function [z] = levy(n,m,beta)

% beta is set to 1.5 in this paper

num = gamma(1+beta)\*sin(pi\*beta/2);

den = gamma((1+beta)/2)\*beta\*2^((beta-1)/2);

sigma\_u = (num/den)^(1/beta);

u = random('Normal',0,sigma\_u,n,m);

v = random('Normal',0,1,n,m);

z =u./(abs(v).^(1/beta));

end