AM5600 TERM PROJECT REPORT

FIREFLY ALGORITHM

ROLL NO: AM20S019

1.Behavior of Fireflies:

The fireflies are a family of insects with more than two thousand species. They are soft-bodied beetles that are commonly called fireflies, glowworms or lightning bugs for their conspicuous use of bioluminescence during twilight to attract mates or prey. Most fireflies produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. Two fundamental uses of such flashes are either to attract partners to mate (communication) or to attract potential prey. Also, flashing may also indicate a protective warning mechanism against its predators of the bitter taste of fireflies.

The rhythm of flash, the rate of flashing and the amount of time a flash exists form the part of the communication that brings both sexes together. Within the species, male fireflies have a unique pattern of flashing and females get attracted to the flash. While in some species like Photuris, female fireflies eavesdrop on the flashing pattern of other species and mimic the behavior to attract the male fireflies to eat them.

We know that, the intensity of light is inversely proportional to the square of the distance(r)from the source. So as the distance between two fireflies increases, the light intensity decreases. Intensity of flashing light is directly proportional to attractiveness between two fireflies. That means a firefly can be attracted towards a brighter firefly. We can associate the flashing light with the objective function which needs to be optimized, thereby forming a new optimization algorithm known as Firefly algorithm.

2. Firefly Algorithm:

Firefly Algorithm is a bio-inspired metaheuristic algorithm for optimization problems. Firefly algorithm was first developed by **Xin-She Yang** at Cambridge University in **2007**. The algorithm is inspired by the flashing patterns and behavior of fireflies at night. The three important rules used to construct the algorithm are as follows:

1. All fireflies are unisex, which means any firefly can be attracted to any other brighter one regardless of their sex.
2. The brightness of a firefly is determined from the encoded objective function.
3. The attractiveness is directly proportional to brightness and they both decrease as the distance between two firefly increases. It means a firefly will move towards a brighter one and if there is no brighter one, it will move randomly.

For a maximization problem, the brightness is proportional to the value of the objective function. Brightness in firefly algorithm is similar to the fitness function in genetic algorithms. In firefly algorithm, the optimization of a function depends on the brightness of the fireflies and the movement of fireflies towards their brighter ones.

The pseudo code for the firefly algorithm can be described as follows:

Define an objective function f(x), x = (xi,….,xd)

Generating initial population of fireflies, xi, (i=1,2,3…..n)

Determine light intensity for xi by calculating f(xi)

Define light absorption coefficient y(gamma)

While (t<MaxGeneration)

For i=1:n all n fireflies

For j=1:n all n fireflies

If(I_j>I_i)

Move fireflies i and j according to attractiveness

Evaluating new solutions and updating light intensity for next iteration

End if

End for j

End for i

Sorting the fireflies to find the best fit

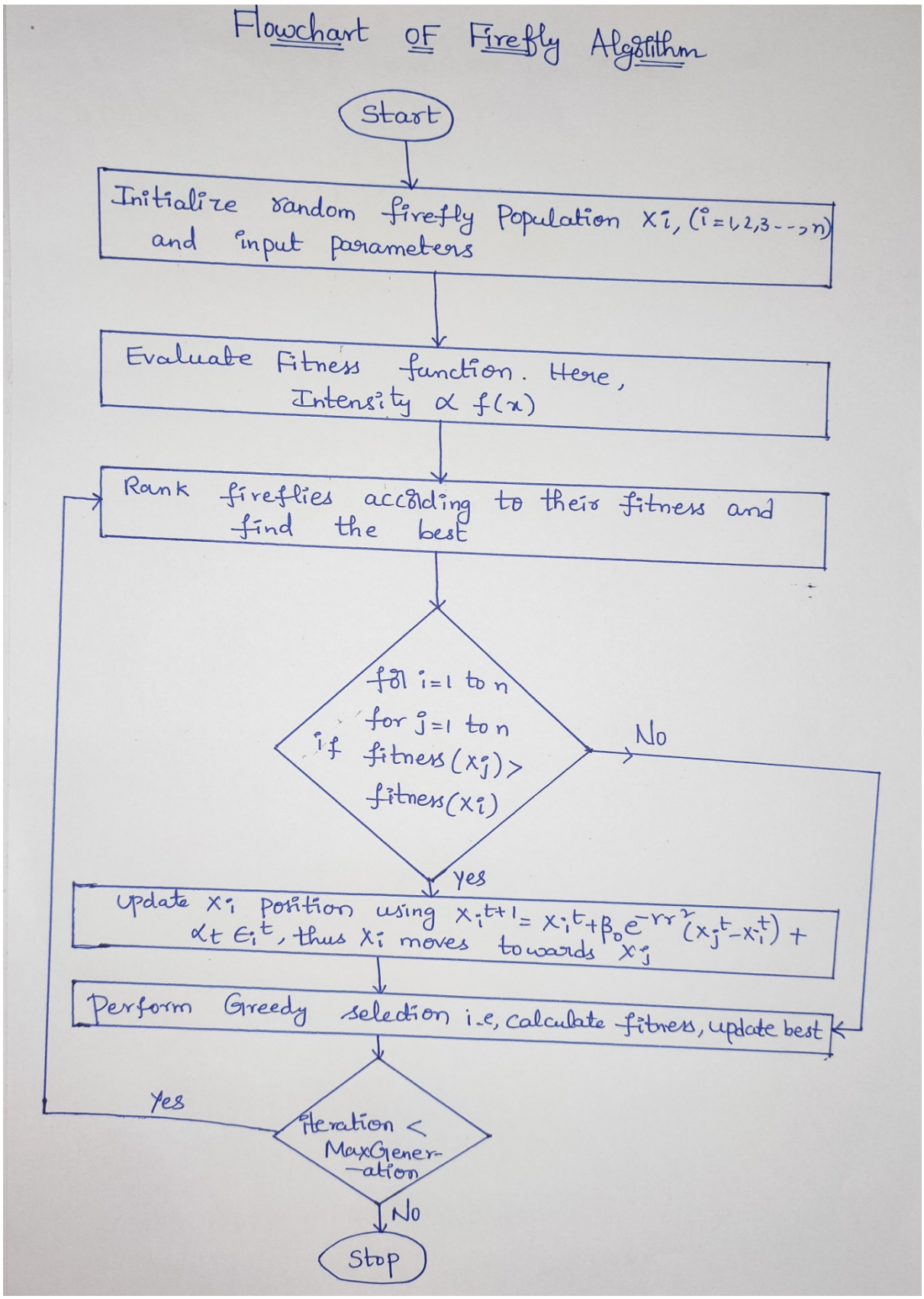
End While

Post processing and visualization of the results

Parameters of Firefly Algorithm:

Parameter	**Notation in Algorithm**
Brightness	Objective function
Beta	Attractiveness parameter
Alpha	Randomization parameter
Gamma	Absorption Coefficient
Number of Generations	Iterations
Number of fireflies	Population
Dimension	Problem dimension

FLOWCHART OF FIREFLY ALGORITHM:



Formulation of the algorithm:

Here movement of one firefly towards another one depends upon the attractiveness. This is the basic idea of formulation of the algorithm. Light intensity is defined as the amount of light energy transmitted and it varies with the distance. The attractiveness of fireflies is directly proportional to the brightness which is in turn related to objective function. The intensity decreases with increase in distance, hence a given firefly is attracted to a firefly which is close to it even though it is less bright than a farther but brighter firefly. The intensity of light(I) is inversely proportional to square of distance(r) i.e.,

I(r) proportional to (1/r^2)

Where I(r) is the light intensity as a function of distance and r is radius.

I(r) = I_s/r^2

Where I_s is intensity of source. Also,

I(r)=I0*exp(-gamma*r^2)

So, we can define the variation of attractiveness beta with distance r by:

Beta= Beta0*exp(-gamma*r^2)

where Beta0 is the attractiveness at r=0.

The position update of the firefly located at Xi is calculated as follows:

Xi(t+1) = Xi(t) + Beta0*exp(-gamma*r^2)*(Xj(t)-Xi(t)) + alpha_t*Ei(t)

If a firefly located at Xj is more attractive(brighter) than another firefly located at Xi, the firefly located at Xj will move towards Xj.

Here, the second term is due to the attraction towards Xj. The parameter Beta0 controls the attractiveness and for most applications, it is taken as 1.

The last term is a randomization term with alpha_t being a randomization parameter which lies between 0 to 1 and it controls the randomness.we can tune this parameter during iterations, so that it can vary with the iteration t. we can express alpha_t as alpha0*delta,where 0<delta<1. The algorithm will be more efficient if alpha0 is associated with scaling of design variables. Where alpha0 is initial randomness scaling factor and delta is a cooling factor. Ei(t) is a vector of random numbers drawn from Gaussian or uniform or other distribution at time t.

Special Cases of Firefly Algorithm:

The position update of the firefly located at Xi is calculated as follows:

Xi(t+1) = Xi(t) + Beta0*exp(-gamma*r^2)*(Xj(t)-Xi(t)) + alpha_t*Ei(t)

Here r=sqrt(summation((Xj-Xi)^2))

Firefly algorithm is indeed rich in many ways. In the above equation,

1.If Beta0 becomes zero ,it becomes a **simple random walk.**

2. If gamma tends to zero, the firefly algorithm reduces to **Standard particle swarm optimization** (PSO). Also, if the inner loop(for j) is removed and the brightness Xj is replaced by the current global best, then the firefly algorithm becomes the **standard PSO.**

3.If gamma is very large then exp(-gamma*r^2) tends to zero, thus second term becomes negligible and it reduces to **Simulated Annealing(SA).**

4.If gamma is very small then exp(-gamma*r^2) tends to one,then

a. if further we keep alpha_t = 0, then the above equation becomes a variant of **Dfferential evolution (DE).**

b. if we set Beta0 =0 and Ei(t) is related to Xi, then it becomes a variant of **Harmony Search (HS).**

3. IMPLEMENTATION OF THE ALGORITHM USING A BENCHMARK PROBLEM:

The algorithm can be summarized in the following steps:

Step 1: Generate a random solution set X.

Step 2: Compute Intensity (I) i.e., I proportional to f(x)

Step 3: update each firefly with position update equation.

Xi(t+1) = Xi(t) + Beta0*exp(-gamma*r^2)*(Xj(t)-Xi(t)) + alpha_t*Ei(t)

Step 4: perform Greedy selection. For example , if it is a maximization problem, if f(xnew)>f(old),then update the position of the firefly, otherwise no need to update.

Step 5: Terminate if maximum iterations are reached or convergence occurs otherwise go to step 2.

EXAMPLE PROBLEM: The benchmark problem used is a function called **De-Jong's function**.

De Jong function is a simple and a continuous benchmark problem. It is mathematically defined as:

f(x)= ∑ (xi)^2 where i= 1 to n,

where n is number of dimensions. De-Jong's function is unimodal, convex, multidimensional and is easier to solve and have one global optimum. This function can result in poor convergence to global optimum.

The example problem solved here is DE-jong's function with two dimensions and boundaries are from -5.12 to 5.12. The population size is 20 with maximum iterations being 100.

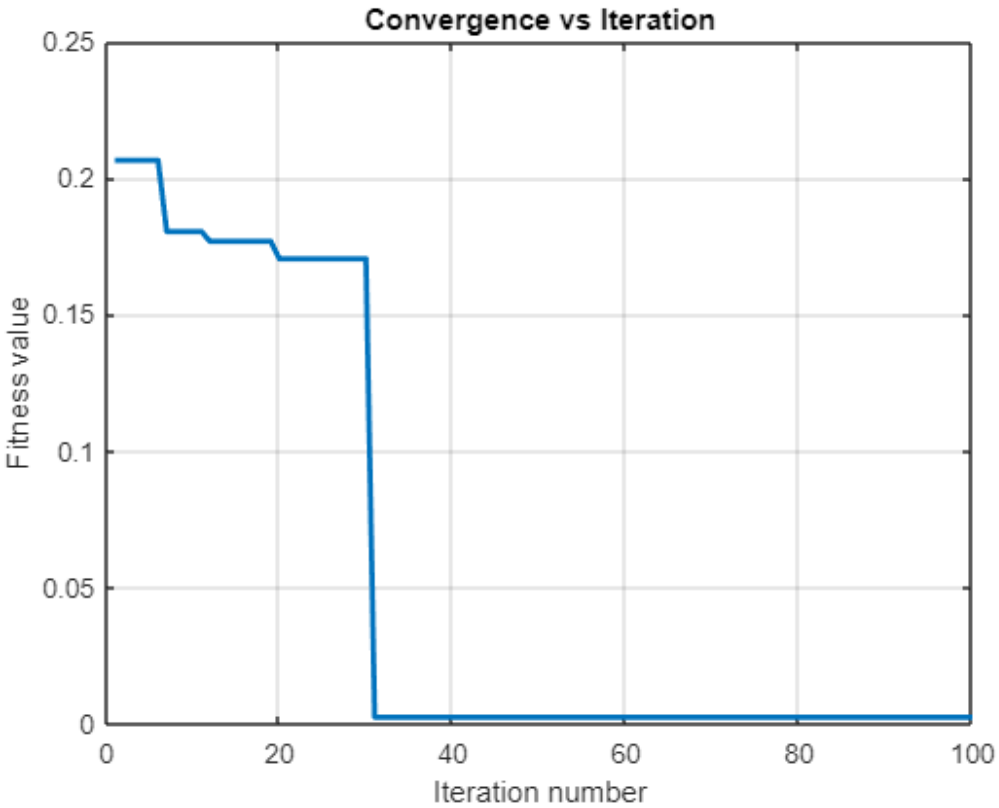
MATLAB CODE FOR FIREFLY ALGORITHM TO OPTIMIZE DE-JONG'S FUNCTION :

```
 %% Firefly Algorithm MATLAB code
format short
clear all
clc

%% step1: Input Parameters
D = 2;              % Dimension of the problem
lb = [-5.12 -5.12]; % Lower bound of the variables
ub = [5.12 5.12];   % Upper bound of the variables
N = 20;             % Population size
alpha =1.0;         % Randomness strength 0-1(highly random)
beta0 =1.0;         % Attractiveness constant
gamma =0.01;        % Absorption Coefficient
delta =0.97;        % Randomness Reduction Factor delta = 10^(-5/iter_max)
iter_max=100;       % Maximum number of iterations

%% Step 2: Defining objective function - Benchbark problem De-Jong's function
f(x)= sum(x_i)^2 for i=1:D where D= dimensions

function out = fun(x)
x1=x(:,1);
x2=x(:,2);
out = x1.^2+x2.^2;
end

%% step3: Generate initial population randomly

for i=1:N
    for j=1:D
        pop(i,j)=lb(:,j)+rand.*(ub(:,j)-lb(:,j)); % use x=L+rand.*(U-L)
    end
end

%% Evaluate Objective Function

fx = fun(pop);
alpha= alpha*delta;  % Reduce alpha by a factor theta
scale= abs(ub-lb);   % Scale of the problem
```

```
%% step 4:FIREFLY ALGORITHM MAIN LOOP STARTS

for iter = 1:iter_max
    %% for n fireflies,there are a total of two loops
    for i=1:N
        for j=1:N
            fx(i)=fun(pop(i,:));
            %% Brighter / more attractive firefly
            if fx(i) < fx(j)
                pop(i,:) = pop(i,:);
            elseif fx(i) > fx(j)
                xi=pop(i,:);
                xj=pop(j,:);
                r=sqrt(sum((xi-xj).^2));
                beta=beta0*exp(-gamma*r.^2);
                steps=alpha.*(rand(1,D)-0.5).*scale;
xnew=xi+beta*(xj-xi)+steps; % New position update equation

%% step 5:Check the bounds

for i=1:size(xnew,2)
if xnew(i)>ub(i)
xnew(i)=ub(i);
elseif xnew(i)<lb(i)
xnew(i)=lb(i);
end
end

%% step6: perform greedy selection
fnew = fun(xnew); % for minimisation, fnew<fold,then update the solution,otherwise
not
if fnew<fx(i)
fx(i)=fnew;
pop(i,:)=xnew;
end
end % end for "if fx(i)<fx(j)"
end % end for j
end %end for i

%% Memorize the best
[optval, optind] = min(fx);
Bestfx(iter) = optval;
Bestx(iter,:)=pop(optind,:);

%% Show iteration information
disp(['Iteration' num2str(iter)...
':Best Cost =' num2str(Bestfx(iter))]);

%% Plotting the result
plot(Bestfx, 'LineWidth', 2);
xlabel('Iteration number');
ylabel('Fitness value')
title('Convergence vs Iteration')
grid on
end %%%% end of the ITERATION LOOP
```

After the maximum number of iterations, the best cost is obtained as **`Best Cost =0.067655`**

The plot obtained for fitness value versus iteration number for the above problem is as follows:



4. Solution of a benchmark problem and comparison with the standard result:

Example : consider **De-Jong's function with d=256** dimensions

F(x) =sum(xi^2) where i=1 to d

For this function, if we use firefly algorithm to optimize the function, the function evaluations will be around 5657. Whereas the same function optimized using genetic algorithm, takes 25412 function evaluations and using Particle swarm optimization, it takes more than 17000 function evaluations. If I assume that one function evaluation takes one second, the firefly algorithm saves approximately 78% and 67% computational cost when compared to Genetic Algorithm and Particle swarm optimization.

5.Critical remarks of the firefly algorithm:

1. Algorithm Complexity: firefly algorithm has two inner loops when going through the population n and one outer loop for iteration t. So the complexity at the extreme case is of order n^2*t i.e., O(n^2t). It is linear in terms of t and non-linear in terms of n.
    a. If n is small (say n=50) and t is very large (say t=4000), then the computational cost is relatively inexpensive because the complexity of the algorithm is linear in terms of t.
    b. If n is relatively large, it is possible to use one inner loop by ranking the attractiveness of all fireflies using SORTING algorithms. In this case, the algorithm complexity of firefly algorithm will be O(ntlogn).
2. Firefly Algorithm uses nonlinear updating equation which can produce rich behavior and higher convergence than the linear updating equations used in standard PSO and DE.
3. Firefly Algorithm has two major advantages over other algorithms. They are:
    a. Automatical subdivision. That means Firefly algorithm is based on attraction and attractiveness decreases with distance. This leads to the fact that the whole population can automatically subdivided into subgroups and each group can swarm around each mode or local optimum. Among all these modes, the best global solution can be found. This subdivision allows the fireflies to be able to find all optima simultaneously if the population size is sufficiently higher than the number of modes.
    b. The ability of dealing with multimodality.
4. The parameters in the algorithm can be tuned to control the randomness as the iteration proceed, so that convergence can also be speed up by tuning these parameters.

6.Scope for improvement of the Firefly algorithm:

1. If the population is relatively large, it is possible to use one inner loop by ranking the attractiveness of all fireflies using SORTING algorithms. In this case, the algorithm complexity of firefly algorithm will be O(ntlogn). This is because the algorithm complexity is nonlinear(n^2) in n, and if n is very large it will result in the increase of computational cost making it less efficient compared to Particle Swarm Optimization(PSO) which has only one loop for population n.
2. The firefly algorithm is based on the attraction between fireflies and it uses the process of attraction to optimize an objective function. Given the capability that the algorithm can be useful in both continuous and discrete domain, it can be extended to solve real world optimization problems.
3. Various results obtained from previous research has shown that the algorithm is not as efficient as opposed to Particle Swarm optimization. In the future, we can go for a hybrid firefly algorithm by incorporating the functionality of other algorithms to improve the performance and also by changing various input parameters.

7. REFERENCES:

1. X.S.Yang , Nature-Inspired Metaheuristic Algorithms, Luniver Press(2008).

2. X.S.Yang, Firefly Algorithms for multimodal optimization, in:Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Science, 5792, 169-178(2009).