Name: Shaurya Gupta

PRN: 21070122154

Class: Final Year CSE-C

Optimisation Techniques and Algorithm Lab 6

Aim:

1) Extend the Genetic Algorithm code to execute the following pseudocode:

```
D, LB, and UB as per problem
Npop = 50
Maxiter = 200
CR = 0.6
MR = 0.05

For i = 1:Npop
{
 For j = 1:D
 {
  X(i, j) = LBj + rand * (UBj - LBj)
 }
}

For it = 1:Maxiter
{
 For i = 1:Npop
 {
  r1 = randi(1, Npop) # r1 ≠ i
  if (rand(0, 1) < CR)
  {
   X(i)' = Single point crossover of the half of the vector between X(i) and X(r1)
```

}

r2 = randi(1, D)

if (rand(0, 1) < MR)

{

X(i)’ = Mutation of single dimension (replace the dimension by a random feasible point)

}

if f(X(i)’) < f(X(i))

{

X(i) = X(i)’

}

}

Sort the population on the descending order of the quality of f(x)

Print best f(X) and x

}

Plot the best f(X(i)) for all the iterations

2) Repeat the above algorithm for n = 20 runs and plot the average values of best f(X(i)) for each iteration

Codes:

1) Basic GA

```
clc;
clear;

% Hardcoded values
minMax = 'min'; % Optimization type: 'min' for minimization, 'max' for
maximization
MaxIter = 200; % Total iterations
str = 'x1^2 + x2^2 + x3^2'; % Fitness function (for testing, a simple function)
D = 3; % Number of dimensions
Npop = 50; % Population size
CR = 0.6; % Crossover rate
MR = 0.05; % Mutation rate

% Lower and Upper bounds for each dimension
LB = [-10, -10, -10]; % Lower bounds for each dimension
UB = [10, 10, 10]; % Upper bounds for each dimension
```

```
% Create symbolic variables and the fitness function
syms x [1 D];
fitnessFunction = str2sym(str);
f = matlabFunction(fitnessFunction, 'vars', {x});

% Initialize population
X = LB + rand(Npop, D) .* (UB - LB);

% Fitness evaluation
fitness = zeros(Npop, 1);
for i = 1:Npop
    fitness(i) = f(X(i, :));
end

bestFitness = zeros(MaxIter, 1);
bestX = zeros(MaxIter, D);

for it = 1:MaxIter
    X_new = X; % Copy population for new individuals

    for i = 1:Npop
        % Selection of r1
        r1 = randi(Npop);
        while r1 == i
            r1 = randi(Npop);
        end

        % Crossover
        if rand < CR
            crossoverPoint = floor(D / 2); % Single-point crossover at midpoint
            X_new(i, 1:crossoverPoint) = X(i, 1:crossoverPoint);
            X_new(i, crossoverPoint+1:end) = X(r1, crossoverPoint+1:end);
        end

        % Mutation
        if rand < MR
            mutationPoint = randi(D);
            X_new(i, mutationPoint) = LB(mutationPoint) + rand *
(UB(mutationPoint) - LB(mutationPoint));
        end

        % Fitness evaluation
        fitness_new = f(X_new(i, :));
        if fitness_new < fitness(i)
            X(i, :) = X_new(i, :);
            fitness(i) = fitness_new;
        end
    end

    % Sort population by fitness
    [sortedFitness, index] = sort(fitness, 'descend');
    if minMax == 'min'
        [sortedFitness, index] = sort(fitness, 'ascend');
    end

    % Track best fitness and corresponding solution
    bestFitness(it) = sortedFitness(1);
    bestX(it, :) = X(index(1), :);
```

```
    % Display best fitness
    fprintf('Iteration %d: Best Fitness = %f\n', it, bestFitness(it));
end

% Display best result
disp('The best values for the dimensions are:');
disp(bestX(end, :));

% Plot the best fitness over iterations
figure;
plot(1:MaxIter, bestFitness, 'LineWidth', 2, 'Color', 'r');
ylabel('Fitness', 'FontSize', 15);
xlabel('Iteration', 'FontSize', 15);
title('Best Fitness Over Iterations');
grid on;
```

2) GA with Average of 20 iterations:

```
clc;
clear;

% Hardcoded values
minMax = 'min'; % Optimization type: 'min' for minimization, 'max' for
maximization
MaxIter = 200; % Total iterations
str = 'x1^2 + x2^2 + x3^2'; % Fitness function (for testing, a simple function)
D = 3; % Number of dimensions
Npop = 50; % Population size
CR = 0.6; % Crossover rate
MR = 0.05; % Mutation rate
n = 20; % Number of runs

% Lower and Upper bounds for each dimension
LB = [-10, -10, -10]; % Lower bounds for each dimension
UB = [10, 10, 10]; % Upper bounds for each dimension

% Create symbolic variables and the fitness function
syms x [1 D];
fitnessFunction = str2sym(str);
f = matlabFunction(fitnessFunction, 'vars', {x});

% Initialize storage for average best fitness values and best values
averageBestFitness = zeros(MaxIter, 1);
finalBestValues = zeros(n, D);
finalBestFitness = inf; % Initialize with a large value

for run = 1:n
    % Initialize population
    X = LB + rand(Npop, D) .* (UB - LB);

    % Fitness evaluation
    fitness = zeros(Npop, 1);
    for i = 1:Npop
        fitness(i) = f(X(i, :));
    end

    bestFitness = zeros(MaxIter, 1);
```

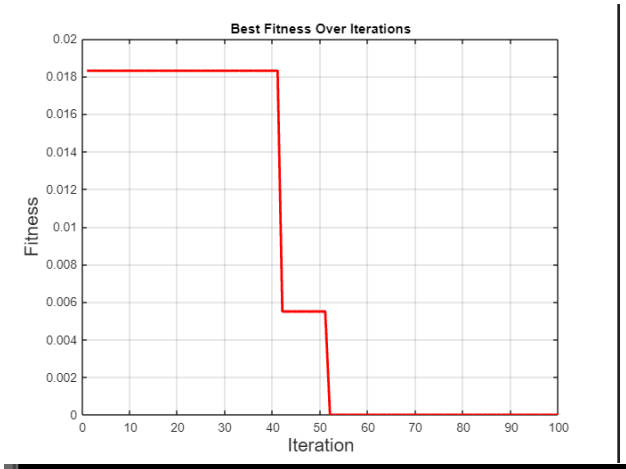
```
    bestX = zeros(MaxIter, D);

    for it = 1:MaxIter
        X_new = X; % Copy population for new individuals

        for i = 1:Npop
            % Selection of r1
            r1 = randi(Npop);
            while r1 == i
                r1 = randi(Npop);
            end

            % Crossover
            if rand < CR
                crossoverPoint = floor(D / 2); % Single-point crossover at
midpoint
                X_new(i, 1:crossoverPoint) = X(i, 1:crossoverPoint);
                X_new(i, crossoverPoint+1:end) = X(r1, crossoverPoint+1:end);
            end

            % Mutation
            if rand < MR
                mutationPoint = randi(D);
                X_new(i, mutationPoint) = LB(mutationPoint) + rand *
(UB(mutationPoint) - LB(mutationPoint));
            end

            % Fitness evaluation
            fitness_new = f(X_new(i, :));
            if fitness_new < fitness(i)
                X(i, :) = X_new(i, :);
                fitness(i) = fitness_new;
            end
        end

        % Sort population by fitness
        [sortedFitness, index] = sort(fitness, 'descend');
        if minMax == 'min'
            [sortedFitness, index] = sort(fitness, 'ascend');
        end

        % Track best fitness and corresponding solution
        bestFitness(it) = sortedFitness(1);
        bestX(it, :) = X(index(1), :);

        % Update average best fitness values
        averageBestFitness(it) = averageBestFitness(it) + bestFitness(it) / n;
    end

    % Store best values for dimensions from this run
    finalBestValues(run, :) = bestX(end, :);

    % Update the global best fitness and dimensions if current run has a better
fitness
    if minMax == 'min'
        if bestFitness(end) < finalBestFitness
            finalBestFitness = bestFitness(end);
            bestDimensionValues = bestX(end, :);
        end
```

```
    else
        if bestFitness(end) > finalBestFitness
            finalBestFitness = bestFitness(end);
            bestDimensionValues = bestX(end, :);
        end
    end
end

% Display the average best fitness values for each iteration
disp('Average best fitness values for each iteration:');
for it = 1:MaxIter
    fprintf('Iteration %d: Average Best Fitness = %f\n', it,
averageBestFitness(it));
end

% Display the best values for dimensions corresponding to the best fitness
disp('Best values for dimensions corresponding to the best fitness:');
disp(bestDimensionValues);

% Plot the average best fitness over iterations
figure;
plot(1:MaxIter, averageBestFitness, 'LineWidth', 2, 'Color', 'r');
ylabel('Average Best Fitness', 'FontSize', 15);
xlabel('Iteration', 'FontSize', 15);
title('Average Best Fitness Over Iterations');
grid on;
```

Output:

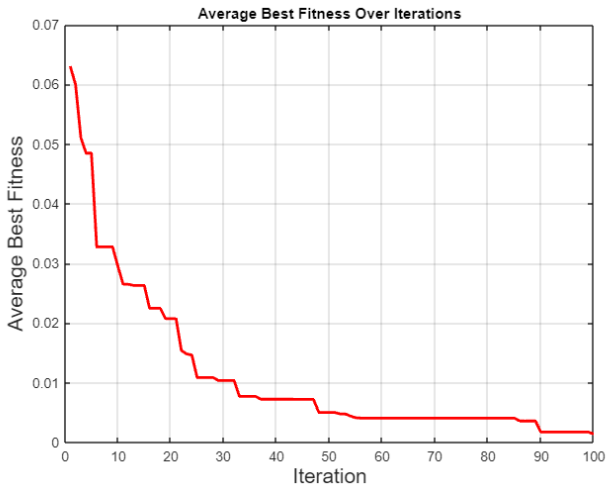
1) Function 1: Sphere function (1D)
f=x1^2

- Basic GA




```
Command Window
Iteration 93: Best Fitness = 0.000007
Iteration 94: Best Fitness = 0.000007
Iteration 95: Best Fitness = 0.000007
Iteration 96: Best Fitness = 0.000007
Iteration 97: Best Fitness = 0.000007
Iteration 98: Best Fitness = 0.000007
Iteration 99: Best Fitness = 0.000003
Iteration 100: Best Fitness = 0.000003
The best values for the dimensions are:
   -0.0018
```

- Average of 20 Iterations:

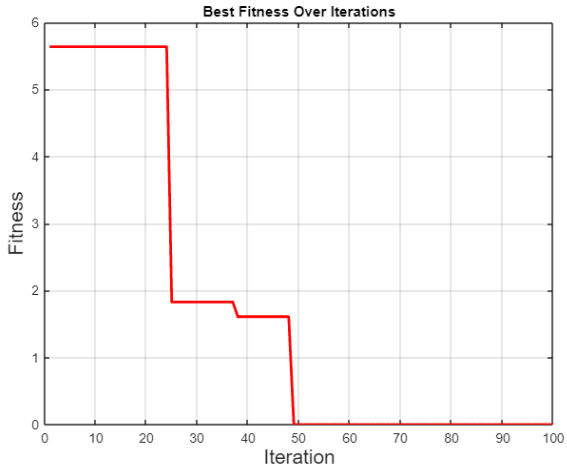



```
Iteration 94: Average Best Fitness = 0.001765
Iteration 95: Average Best Fitness = 0.001765
Iteration 96: Average Best Fitness = 0.001765
Iteration 97: Average Best Fitness = 0.001765
Iteration 98: Average Best Fitness = 0.001765
Iteration 99: Average Best Fitness = 0.001765
Iteration 100: Average Best Fitness = 0.001399
Best values for dimensions corresponding to the best fitness:
   -0.0023
```

2) Function 2: Rosenbrock function (2D)

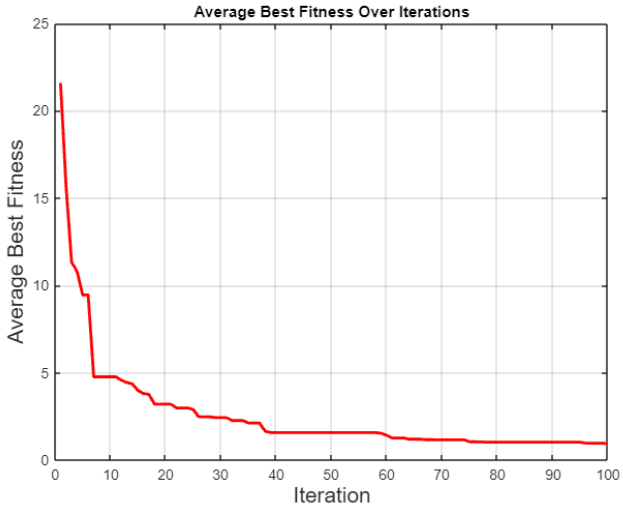
f= (a - x1) ^2 + b * (x2 - x1^2) ^2

- General GA




```
Command Window
Iteration 93: Best Fitness = 0.002122
Iteration 94: Best Fitness = 0.002122
Iteration 95: Best Fitness = 0.002122
Iteration 96: Best Fitness = 0.002122
Iteration 97: Best Fitness = 0.002122
Iteration 98: Best Fitness = 0.002122
Iteration 99: Best Fitness = 0.002122
Iteration 100: Best Fitness = 0.002122
The best values for the dimensions are:
    0.0550    0.0020
```

- Average Value of 20 iterations

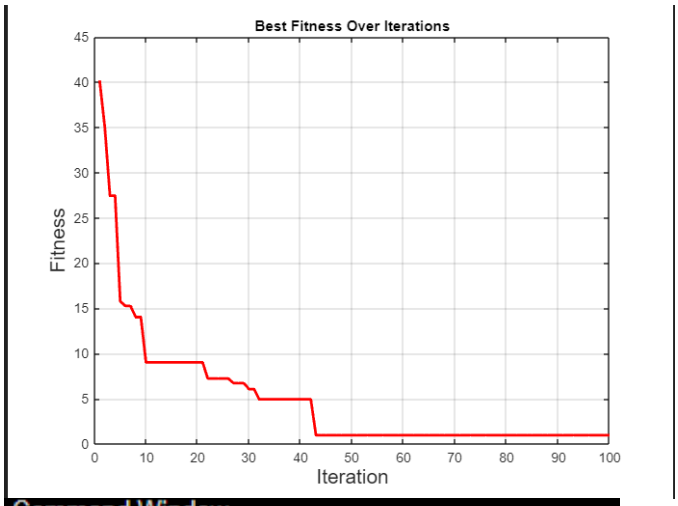



```
Command Window
Iteration 94: Average Best Fitness = 1.026007
Iteration 95: Average Best Fitness = 1.026007
Iteration 96: Average Best Fitness = 0.971734
Iteration 97: Average Best Fitness = 0.971734
Iteration 98: Average Best Fitness = 0.971734
Iteration 99: Average Best Fitness = 0.971734
Iteration 100: Average Best Fitness = 0.938350
Best values for dimensions corresponding to the best fitness:
    0.0686   -0.0069
```

3) Function 3: Rastrigin Function (3D)

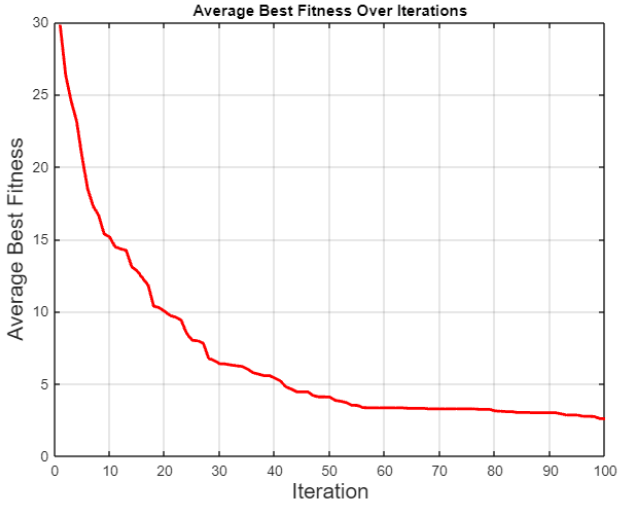
f=10 * 3 + x1^2 - 10 * cos(2 * pi * x1) + x2^2 - 10 * cos(2 * pi * x2) + x3^2 - 10 * cos(2 * pi * x3)

- General GA




```
Command Window
Iteration 94: Best Fitness = 0.980723
Iteration 95: Best Fitness = 0.980723
Iteration 96: Best Fitness = 0.980723
Iteration 97: Best Fitness = 0.980723
Iteration 98: Best Fitness = 0.980723
Iteration 99: Best Fitness = 0.980723
Iteration 100: Best Fitness = 0.980723
The best values for the dimensions are:
    0.0159    0.0316   -0.0612
```

- Average value of 20 iterations

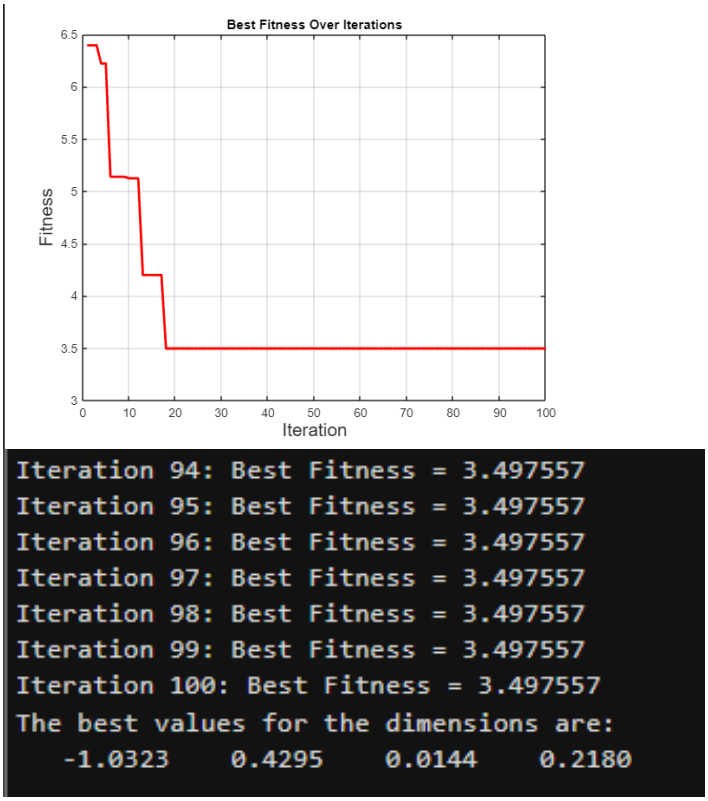



```
Command Window
Iteration 94: Average Best Fitness = 2.847051
Iteration 95: Average Best Fitness = 2.847051
Iteration 96: Average Best Fitness = 2.758981
Iteration 97: Average Best Fitness = 2.758981
Iteration 98: Average Best Fitness = 2.751517
Iteration 99: Average Best Fitness = 2.604244
Iteration 100: Average Best Fitness = 2.591492
Best values for dimensions corresponding to the best fitness:
   -0.0157    0.0277    0.0077
```

4) Function 4: Ackley Function (4D)

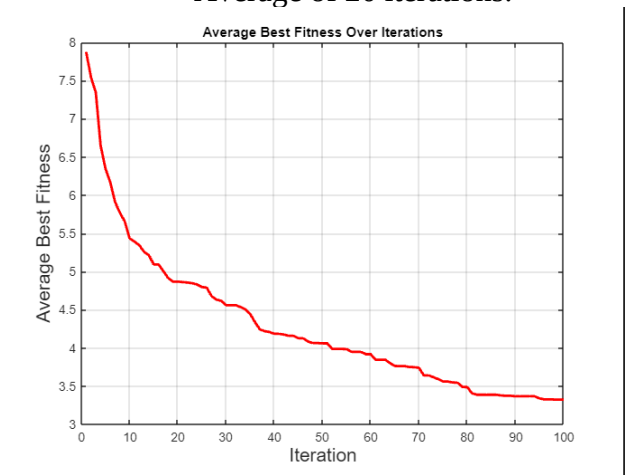
f=-20 * exp(-0.2 * sqrt((x1^2 + x2^2 + x3^2 + x4^2) / 4)) - exp((cos(2 * pi * x1) + cos(2 * pi * x2) + cos(2 * pi * x3) + cos(2 * pi * x4)) / 4) + 20 + exp(1)

- General GA




```
Iteration 94: Best Fitness = 3.497557
Iteration 95: Best Fitness = 3.497557
Iteration 96: Best Fitness = 3.497557
Iteration 97: Best Fitness = 3.497557
Iteration 98: Best Fitness = 3.497557
Iteration 99: Best Fitness = 3.497557
Iteration 100: Best Fitness = 3.497557
The best values for the dimensions are:
   -1.0323    0.4295    0.0144    0.2180
```

- Average of 20 iterations:

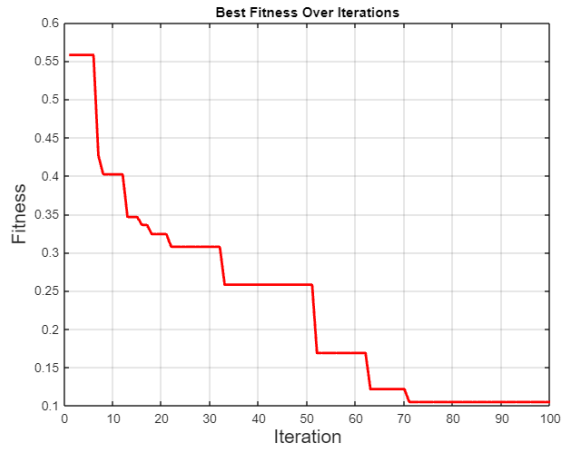



```
Command Window
Iteration 93: Average Best Fitness = 3.367860
Iteration 94: Average Best Fitness = 3.367860
Iteration 95: Average Best Fitness = 3.339488
Iteration 96: Average Best Fitness = 3.327533
Iteration 97: Average Best Fitness = 3.327533
Iteration 98: Average Best Fitness = 3.325504
Iteration 99: Average Best Fitness = 3.325504
Iteration 100: Average Best Fitness = 3.325504
Best values for dimensions corresponding to the best fitness:
   -0.0965    0.8854   -0.1251    0.0256
```

5) Function 5: Griewank Function (5D)

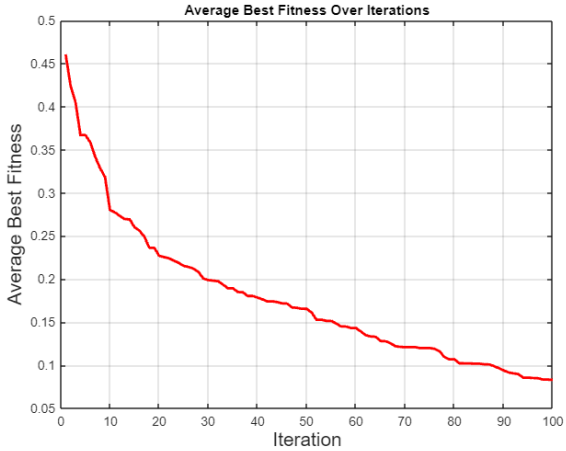
f=1 + (x1^2 + x2^2 + x3^2 + x4^2 + x5^2) / 4000 - cos(x1 / sqrt(1)) * cos(x2 / sqrt(2)) * cos(x3 / sqrt(3)) * cos(x4 / sqrt(4)) * cos(x5 / sqrt(5))

- General GA




```
Command Window
Iteration 94: Best Fitness = 0.104782
Iteration 95: Best Fitness = 0.104782
Iteration 96: Best Fitness = 0.104782
Iteration 97: Best Fitness = 0.104782
Iteration 98: Best Fitness = 0.104782
Iteration 99: Best Fitness = 0.104782
Iteration 100: Best Fitness = 0.104782
The best values for the dimensions are:
    3.1797    4.0125    0.2635    6.2763   -7.4482
```

- Average of 20 iterations:




```
Command Window
Iteration 94: Average Best Fitness = 0.085644
Iteration 95: Average Best Fitness = 0.085644
Iteration 96: Average Best Fitness = 0.085146
Iteration 97: Average Best Fitness = 0.084902
Iteration 98: Average Best Fitness = 0.083552
Iteration 99: Average Best Fitness = 0.083337
Iteration 100: Average Best Fitness = 0.083223
Best values for dimensions corresponding to the best fitness:
   -3.2356    4.2654   -0.2023    0.0843    0.1634
```