Experiment-2.2

Aim of the Experiment:

Implementation of Simple Genetic Application- Match word Problem.

Theory:

A genetic algorithm (GA) is a computational technique inspired by the process of natural selection and evolution. It is used to solve optimization and search problems by mimicking the process of natural selection and evolution in a population of candidate solutions. A genetic algorithm is an adaptive heuristic search algorithm inspired by "Darwin's theory of evolution in Nature."

Genetic Algorithm involves five phases to solve the complex optimization problems:

1) Initialization: Process of Genetic algorithm starts by generating the set of individuals, called population. Each individual is the solution for the given problem. An individual contains a set of parameters called Genes. Genes are combined into a string and generate chromosomes, which is the solution to the problem.

2) Fitness Assignment: Fitness function is used to determine how fit an individual is? In every iteration, individuals are evaluated based on their fitness function. The fitness function provides a fitness score to each individual. The high the fitness score, the more chances of getting selected for reproduction.

3) Selection: The selection phase involves the selection of individuals for the reproduction of offspring. All the selected individuals are then arranged in a pair of two to increase reproduction. Then these individuals transfer their genes to the next generation. (Roulette wheel selection, Tournament selection, Rank-based selection).

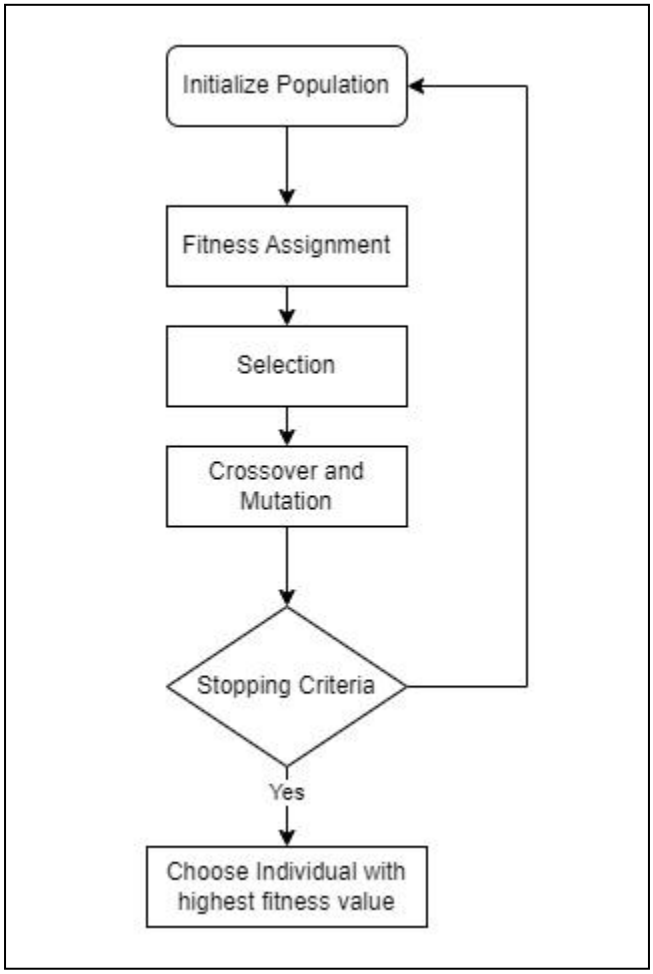
4) Reproduction: After the selection process, the creation of a child occurs in the reproduction step. In this step, the genetic algorithm uses two variation operators that are applied to the parent population.

a) Crossover: A crossover point is selected at random within the genes. Then the crossover operator swaps genetic information of two parents from the current generation to produce a new individual representing the offspring. The genes of parents are exchanged among themselves until the crossover point is met. (One point crossover, Two-point crossover).

b) Mutation: The mutation operator inserts random genes in the offspring (new child) to maintain the diversity in the population. It can be done by flipping some bits in the chromosomes. Mutation helps in solving the issue of premature convergence and enhances diversification. (Flip bit mutation, Gaussian mutation, Swap mutation).

5) Termination: After the reproduction phase, a stopping criterion is applied as a base for termination. The algorithm terminates after the threshold fitness solution is reached. It will identify the final solution as the best solution in the population.

FlowChart for Genetic Algorithm:



Code for Experiment :

```
% Parameters
TARGET_WORD = 'hello';
POPULATION_SIZE = 100;
MUTATION_RATE = 0.1;
MAX_GENERATIONS = 1000;

fprintf('Target word: %s\n', TARGET_WORD);
fprintf('Population Size: %d\n', POPULATION_SIZE);
fprintf('Mutation Rate: %f\n', MUTATION_RATE);
fprintf('Maximum Generations: %d\n\n', MAX_GENERATIONS);


% Initialize population
population = cell(POPULATION_SIZE, 1);
for i = 1:POPULATION_SIZE
      population{i} = generateRandomWord(length(TARGET_WORD));
end

% Main loop
for generation = 1:MAX_GENERATIONS

      % Calculate fitness for each individual
      fitness_scores = zeros(POPULATION_SIZE, 1);
      for i = 1:POPULATION_SIZE
            fitness_scores(i) = calculateFitness(population{i}, TARGET_WORD);
      End

      % Check if target word is found
      [best_fitness, idx] = max(fitness_scores);
      best_word = population{idx};

      if strcmp(best_word, TARGET_WORD)
            fprintf('Generation %d: Best Word = %s, Fitness = %d\n', generation, best_word,
             best_fitness);
            fprintf('Target word found: %s\n', best_word);
            break;
      End

      % Selection: Roulette wheel selection
      total_fitness = sum(fitness_scores);
      probabilities = fitness_scores / total_fitness;

      selected_indices = randsample(1:POPULATION_SIZE, POPULATION_SIZE, true,
                          probabilities);
      selected_population = population(selected_indices);
```

```
        % Crossover and Mutation
        new_population = cell(POPULATION_SIZE, 1);
        for i = 1:2:POPULATION_SIZE
                parent1 = selected_population{i};
                parent2 = selected_population{i+1};

                point = randi(length(TARGET_WORD) - 1) + 1;
                child1 = [parent1(1:point) parent2(point+1:end)];
                child2 = [parent2(1:point) parent1(point+1:end)];

                % Mutation
                if rand < MUTATION_RATE
                        mutate_index = randi(length(TARGET_WORD));
                        child1(mutate_index) = char(randi([97, 122])); % random lowercase letter
                end

                if rand < MUTATION_RATE
                        mutate_index = randi(length(TARGET_WORD));
                        child2(mutate_index) = char(randi([97, 122])); % random lowercase letter
                end

                new_population{i} = child1;
                new_population{i+1} = child2;
        end

        population = new_population;
        % Print best word in each generation
        fprintf('Generation %d: Best Word = %s, Fitness = %d\n', generation, best_word,
         best_fitness);
end

if generation == MAX_GENERATIONS
        fprintf('Maximum generations reached. Target word not found.\n');
end

% Function to generate a random word of given length
function word = generateRandomWord(length)
        letters = 'abcdefghijklmnopqrstuvwxyz';
        word = letters(randi(numel(letters), [1, length]));
end

% Function to calculate fitness of a word
function fitness = calculateFitness(word, target_word)
        fitness = sum(word == target_word);
end
```

Result/Output :

```
Command Window
>> SC_Experiment5_ASHISH_23MAI10008
Target word: hello
Population Size: 100
Mutation Rate: 0.100000
Maximum Generations: 1000

Generation 1: Best Word = httjo, Fitness = 2
Generation 2: Best Word = hlnlo, Fitness = 3
Generation 3: Best Word = hsnlo, Fitness = 3
Generation 4: Best Word = petlo, Fitness = 3
Generation 5: Best Word = hillo, Fitness = 4
Generation 6: Best Word = hlllo, Fitness = 4
Generation 7: Best Word = hlllo, Fitness = 4
Generation 8: Best Word = hillo, Fitness = 4
Generation 9: Best Word = hlllo, Fitness = 4
Generation 10: Best Word = hlllo, Fitness = 4
Generation 11: Best Word = hlllo, Fitness = 4
Generation 12: Best Word = hallo, Fitness = 4
Generation 13: Best Word = hlllo, Fitness = 4
Generation 14: Best Word = hlllo, Fitness = 4
Generation 15: Best Word = hello, Fitness = 5
Target word found: hello
fx >>
```

Learning outcomes :

1. Learnt about the concept of Genetic Algorithm.
2. Learnt about different phases of Genetic Algorithm.
3. Learnt about Crossover and Mutation methods in Genetic Algorithm.
4. Learnt about how to solve the Match Word problem.
5. Learnt about different methods of Selection in Genetic Algorithm.