|  |  |
| --- | --- |
| **Name:** | Bodhisatya Ghosh |
| **Branch:** | CSE – Data Science |
| **Batch:** | B |
| **Course:** | Soft Computing |
| **Experiment no:** | 9 |

**Aim:** To apply genetic algorithms for a given problem.

**Theory:** Genetic algorithms (GAs) are optimization and search algorithms inspired by the process of natural selection and the principles of genetics. They are often used to find approximate solutions to optimization and search problems, especially in complex spaces where other optimization methods may struggle.

**REAL TIME APPLICATION USED:**

**Problem Definition:**

We aim to find the maximum of the mathematical function f(x) =x^2−4x+4.

**Chromosome Representation:**

A potential solution (individual) is represented as a real number x within a certain range. In the context of GAs, this real number is the chromosome.

**Fitness Function:**

The fitness function evaluates how well a potential solution (chromosome) performs in the context of the problem. For this maximization problem, the fitness function is defined as the negative of the function we want to maximize:

Fitness (chromosome)=−(x^2−4x+4)

The negative sign is used because GAs seek to minimize, and in our case, we want to maximize f(x). Higher fitness values indicate better solutions.

**Genetic Operations:**

1. **Selection:**
   * Individuals are selected from the current population based on their fitness scores. In roulette wheel selection, individuals with higher fitness have a higher chance of being selected.
2. **Crossover (Recombination):**
   * Single-point crossover involves selecting a random point along the chromosome and exchanging the genetic material of two individuals. This mimics the crossover of genetic material in biological reproduction.
3. **Mutation:**
   * Small random changes are applied to the chromosome. This introduces diversity into the population and prevents premature convergence.

**Algorithm Workflow:**

1. **Initialization:**
   * Generate an initial population of chromosomes randomly within a specified range.
2. **Evaluation:**
   * Evaluate the fitness of each chromosome using the fitness function.
3. **Selection:**
   * Select individuals from the current population to form a mating pool based on their fitness.
4. **Crossover:**
   * Apply crossover to pairs of individuals in the mating pool to create new offspring.
5. **Mutation:**
   * Apply mutation to some individuals in the population to introduce variation.
6. **Replacement:**
   * Create a new generation by combining the offspring and the existing population.
7. **Termination:**
   * Decide when to stop the algorithm based on a termination condition (e.g., a maximum number of generations or achieving a satisfactory solution).

**Key Parameters:**

1. **Population Size:**
   * The number of individuals in each generation.
2. **Crossover Rate:**
   * The probability that crossover will be applied to a pair of individuals.
3. **Mutation Rate:**
   * The probability that mutation will be applied to an individual.

**Termination Condition:**

The algorithm can terminate after a certain number of generations or when a satisfactory solution (maximum fitness) is achieved.

**Objective:**

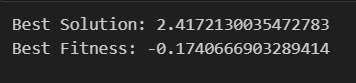
The objective is to evolve a population of chromosomes over multiple generations to find a chromosome that maximizes the fitness function, i.e., maximizes the given mathematical function f(x).

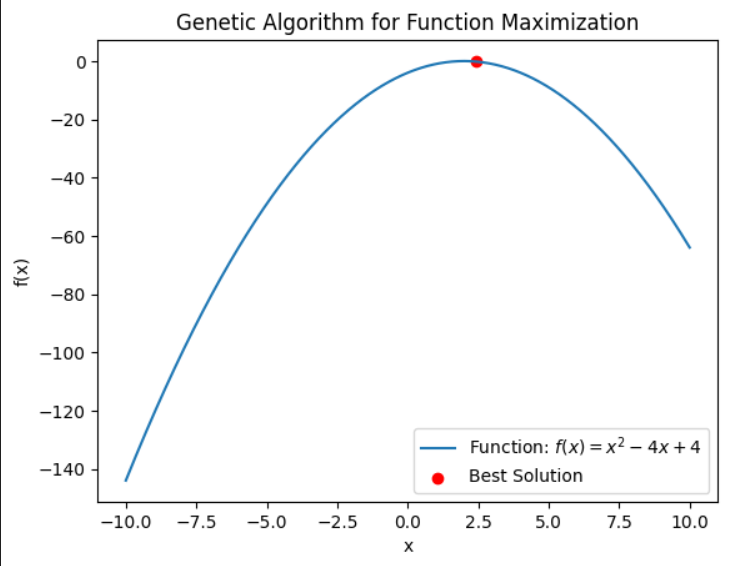
**Final Result:**

The final result of the algorithm is the chromosome that corresponds to the maximum of the function f(x).

**Program:**

**Results:**





**CONCLUSION: -** From this experiment I learnt about genetic algorithm and how to implement it in basic function optimization problem.