**ASSIGNMENT-9**

**Name: Pranathi**

**ID: AP21110011320**

**Subject: AI/ML – Lab**

**Professor: Anabik pal**

**AIM**: Implement the Genetic Algorithm (GA) for finding the solution to the 8-Queen problem. Consider the number of non-attacking pairs of queens as fitness value.

**Procedure:**

**Initialization:**

Constants like BOARD\_SIZE, POPULATION\_SIZE, MUTATION\_RATE, and MAX\_GENERATIONS are defined.

The generateBoardState function creates an initial population of board states. Each board state represents a placement of queens on the chessboard. The population size is specified by POPULATION\_SIZE.

**Fitness Calculation**:

Calculate the fitness of each board state in the population. The fitness function checks for conflicts between queens on the same board state. A lower number of conflicts indicates a higher fitness.

The fitness function is designed to return a value between 0 (no conflicts) and 28 (maximum possible conflicts).

**Main Loop (Generations):**

**For each generation:**

Identify the individual with the highest fitness in the current population.

Check if a solution has been found (fitness equals 28, meaning no conflicts between queens). If so, print the solution and break out of the loop.

Create a new population for the next generation. Add the best individual from the current generation to the new population.

Perform selection, crossover, and mutation to create new individuals until the new population size reaches the specified population size.

**Selection (Tournament Selection):**

Randomly select individuals from the current population.

Choose the individual with the highest fitness from the selected individuals as a parent.

**Crossover:**

Randomly select a crossover point.

Create a new child by combining the genetic information of two parents before and after the crossover point.

**Mutation:**

With a certain probability (MUTATION\_RATE), apply mutation to the child by swapping two random positions on the board.

Replace the old population with the new population.

**Replace Population:**

Replace the old population with the new population, which includes the best individual from the previous generation.

**Termination:**

If a solution with a fitness of 28 (no conflicts) is found, the algorithm terminates, and the generation in which the solution was found is printed.

**Final Output:**

After the algorithm terminates, the best solution found in the population is selected based on fitness, and the board state (queen positions) is printed.

**Additional Classes:**

The fit\_value class is used to pair board states with their fitness values for convenient storage in the population list.