**Problem Description:**

The eight queen’s problem is the problem of placing eight queens on an 8×8 chessboard. None of them attacks one another (no two are in the same row, column, or diagonal). More generally, the n queen’s problem places n queens on an n×n chessboard.

There are different solutions for the problem. **Genetic Algorithm** is one of them.

**Description of the Algorithm:**

Algorithm is started with a **set of solutions** called**population**. Solutions from one population are taken and used to form a new population. Suppose that the new population will be better than the old one. Solutions, which are selected to form new solutions, are selected according to their fitness - the more suitable they are the more chances they have to reproduce.

Five phases are considered in a genetic algorithm.

* Initial population
* Fitness function
* Selection
* Crossover
* Mutation

**Initial population:** The process begins with a set of solutions that is called a **Initial** **Population**. Taken the set of solution randomly. In our project we consider 4 initial population.

**Fitness function:** The **fitness function** determines how fit an individual solution of initial population. In our project **fitness function** determines non-attacking queen. We are initialize max fitness 28. The fitness function calculate number of attack then subtract from max fitness.

**Selection:** Select two parent from a population according to their probability distribution. The better fitness, the bigger chance to be selected.

**Crossover:** After selection part, produce a child through selected two parents. Crossover means select a position then combined first portion from that position of one parent and second portion to that position of another parent. By the crossover create one or two child. In our project we create one child from two parents.

**Mutation:** With a small probability, mutate new child at each position.

Until the child number is equal to population number the loop is continued. Then the N number of child becomes new population and check their fitness value. When One parent of the population is equal the max fitness (means number of attack is zero) we get the solution.

**Implementation Details:**

**Array:** Using array for assigning population, new population, fitness value.

**Rand function:** Generate random number.

**Fitness function:** Calculate the fitness value of the population. Moreover, the value is returned able.

**Reproduce function:** This function is worked for crossover. A child is create from two parent.

**Population\_sort function:** In this function, Populations are shorted according to their fitness value.

**Array\_compare function:** This function ensure that, selected two parents are not same.

**Random\_Selection function:** The task of this function selected two parents by using probability distribution. The better probability, the bigger chance to be selected.

**GA() function:** Contain Genetic Algorithm.

**Solution function:** Print the exact or best solution.

**Mutation function:** A little change of child with some probability.

**Main function:** Create an object and run the GA() function.

**Conclusion and Future Work:**

The N-queen problem is used in many practical solutions like parallel memory storage schemes, VLSI testing, traffic control and deadlock prevention. This problem is also used to find out solutions to more practical problems which requires permutation like travelling salesman problem.