1 Introduction:

The N-queens puzzle is a well-known challenge in computer science, where the task is to place N queens on an N x N chessboard such that no two queens can attack each other. Formally, the problem is defined as arranging N queens on the chessboard so that no two queens occupy the same row, column, or diagonal, thereby ensuring they are non-threatening. This puzzle is significant in the realm of computer science as it is often used to test and develop algorithms and methods, especially in combinatorial optimization and constraint satisfaction problems. Effectively solving the N-queens puzzle involves creating an algorithm capable of navigating a vast solution space while adhering to the given constraints.

Over the years, numerous solvers and strategies have been devised to address this problem. Here are some commonly used approaches:

**1-Backtracking**: This is a depth-first search algorithm that incrementally builds candidates to the solutions and abandons a candidate (i.e., backtracks) as soon as it determines that the candidate cannot possibly be completed to a valid solution. Backtracking is a common approach to solving constraint satisfaction problems like the N-queens.

2- **Heuristics**: Heuristic methods such as the min-conflicts algorithm aim to find a solution by iteratively improving an initial configuration. These methods often use domain-specific knowledge to make intelligent decisions about which paths to explore in the search space.

**3-Iterative** Algorithms: Iterative Algorithms offer an alternative to recursive methods. These approaches use repetition and loops to systematically generate and evaluate potential configurations of queens on the board. Iterative Algorithms often employ data structures such as stacks, queues, or arrays to maintain intermediate solutions and effectively search for valid placements. Unlike recursive algorithms, which rely on the call stack, iterative methods explicitly manage state information, which can lead to different performance characteristics and potential improvements in memory usage. Iterative Algorithms for the N-Queens problem typically exhibit similar time complexity to backtracking methods, which can range from exponential to factorial. The precise time complexity is influenced by the specific implementation details and the optimization strategies employed to reduce the search space, such as pruning invalid branches early, using heuristic methods to guide the search, or employing bitwise operations to efficiently handle board states. Furthermore, iterative approaches can be more amenable to parallelization, potentially offering significant speedups on multi-core or distributed systems.

**4-Simulated Annealing**: This probabilistic technique is used for approximating the global optimum of a given function. It is particularly useful for large search spaces where the goal is to find a sufficiently good solution rather than the absolute best solution.

**5-Genetic Algorithms**: Genetic algorithms, inspired by biological evolution, are effective for solving the N-Queens problem. These algorithms simulate natural selection by employing techniques like mutation, crossover, and selection to iteratively improve solutions. Through these processes, genetic algorithms evolve a population of candidate solutions towards optimal placements of queens on the board.

These methods vary in their efficiency and suitability depending on the size of N and the specific requirements of the problem.

The selected algorithm for solving the N-queen problem is the genetic algorithm. This approach involves using principles of natural selection and genetics to iteratively evolve solutions to the problem. In this algorithm, we start with a population of possible solutions, each representing a different arrangement of queens on the chessboard. These solutions are evaluated based on a fitness function that measures how many queens are non-threatening. The algorithm then selects the fittest solutions and applies genetic operators such as crossover and mutation to create a new generation of solutions. This process is repeated until a valid solution is found or a predefined number of generations is reached. The genetic algorithm is an effective choice for solving the N-queen problem because it can explore a large solution space and often finds good solutions in a reasonable amount of time. However, like other heuristic methods, it does not guarantee an optimal solution and may require tuning of parameters for best performance. This method works well with larger values of N, where traditional algorithms like backtracking become impractical due to their exponential time complexity.