# Table of Contents

## Introduction:

The N-Queens problem is a classical optimization problem where the goal is to place *N queens* on an *N x N chessboard* in such a way that no two queens attack each other. It is a constraint satisfaction problem that has many practical applications, such as *scheduling, puzzle-solving, and resource allocation*. The traditional method to solve this problem is backtracking; however, this report focuses on an alternative approach using the *Genetic Algorithm*. Genetic algorithms mimic the process of natural selection and have proven to be effective for optimization problems like N-Queens, where searching through large solution spaces is needed.

## Problem Description:

The N-Queens problem involves placing **N queens** on an **N x N chessboard** such that no two queens threaten each other. *A queen can attack another queen if they are in the same row, column, or diagonal*. The challenge is to find a configuration of queens where no two queens can attack each other. *The problem has multiple valid solutions*, and the goal is to either find one solution or all possible solutions.

## Genetic Algorithm Overview:

Genetic algorithms (GAs) are a family of optimization algorithms inspired by the process of natural selection. They are part of the class of evolutionary algorithms that use mechanisms such as **selection**, **crossover**, and **mutation** to evolve solutions to optimization problems. The general steps in a genetic algorithm are:

1. **Initial Population**: A set of candidate solutions (individuals) is randomly generated.
2. **Fitness Evaluation**: Each individual is evaluated based on a fitness function, which quantifies how good a solution is.
3. **Selection**: Individuals are selected to create offspring based on their fitness.
4. **Crossover**: Pairs of individuals are combined to produce offspring, inheriting features from both parents.
5. **Mutation**: Offspring undergo random mutations to introduce diversity and avoid local optima.
6. **Termination**: The algorithm terminates when a solution is found or a predefined number of generations have passed.

# Steps in Solving N-Queens using Genetic Algorithm

This section will dive deeper into how the Genetic Algorithm is applied to solve the N-Queens problem:

## Fitness Function:

- The fitness function determines *how good a solution (a board configuration)* is. In this case, it counts *the number of non-attacking pairs of queens*. The higher the fitness score, the better the solution.
- The fitness function is crucial to evaluating how well an individual solution satisfies the problem's constraints. In the case of the N-Queens problem, the fitness function counts the number of non-attacking pairs of queens. The more non-attacking pairs there are, the higher the fitness score. This helps guide the selection process toward better solutions. A perfect solution would have the maximum possible fitness score, meaning no queens are attacking each other.

## Parent Selection

- In this step, a selection mechanism is used to choose which individuals will be used to create the next generation. We used ***tournament selection***, where a small random subset of the population is chosen, and the individual with the highest fitness score is selected.
- Parent selection is done using tournament selection. In this method, a random subset of the population is selected, and the individual with the highest fitness score from this subset is chosen to be a parent. This process is repeated to select multiple parents, ensuring that fitter individuals are more likely to be chosen.

## Crossover

- Crossover is a genetic operation where two parent solutions combine their traits to produce offspring. In this case, we used ***uniform crossover***, where each column in the child board is randomly taken from one of the two parents.
- Crossover is performed to combine the genetic material of two parents. In our implementation, we used uniform crossover, which means that for each column in the child's solution, the queen's position is randomly selected from either parent.

This ensures that the offspring inherits traits from both parents, potentially combining good aspects from both.

### Mutation

- Mutation introduces randomness into the population, ensuring diversity and helping to avoid local optima. Here, mutation randomly changes the position of a queen on the board.
- Mutation is applied with a certain probability (mutation rate). In our implementation, mutation involves randomly changing the position of a queen in one of the columns. This helps to maintain genetic diversity in the population and prevents the algorithm from getting stuck in suboptimal solutions.

## Termination Condition

- The algorithm stops when it finds a solution or after a predefined number of generations.
- The algorithm terminates when one of two conditions is met: either a solution with no attacking queens is found, or the maximum number of generations has been reached. If a solution is found, it is returned; otherwise, the algorithm returns the best individual found.

## Results and Observations:

- Our genetic algorithm successfully solved the N-Queens problem for various board sizes. The algorithm was able to find a solution within X generations for an N x N board. We observed that the solution time increased with the size of the board, which is expected since the solution space grows exponentially with N. The algorithm demonstrated the ability to explore a large search space and find valid solutions within reasonable computational time.

# Conclusion:

- The genetic algorithm proved to be a useful technique for solving the N-Queens problem. By employing selection, crossover, and mutation, the algorithm was able to explore the solution space efficiently and find valid configurations of queens on the board. While the genetic algorithm did not always find the optimal solution immediately, it provided a powerful alternative to traditional methods like backtracking. Future work could focus on optimizing the mutation rate, population size, and crossover techniques to improve convergence speed and solution quality for larger boards.[i]

[i] https://chatgpt.com
https://youtu.be/Fgq3xDhRBv8?si=Z-HFwUp07CWrv9g4