Solving the N-Queens Problem

Sumit (23N0456), Aryan Rajan (23N0451), Ditipriya (23N0465), Amarjeet (23N0461), Shivam Gupta (23N0460)

IE609

Indian Institute of Technology Bombay

November 24, 2024

Problem Description

Challenge

The N-Queens Problem asks: How can you place N chess queens on an N x N chessboard so that no two queens threaten each other?

Threat:

- Queens threaten each other if they share the same row, column, or diagonal.
- Finding a valid arrangement becomes more complex as N increases.

Mathematical Model

Solution Approach: Mixed Integer Programming

The problem can be modeled as a Mixed Integer Programming (MIP) problem with the following components:

- **Binary Variable Representation:** $x_{ij} = 1$ if a queen is placed in cell (i, j), otherwise $x_{ij} = 0$.
- **Objective Function:** Maximize the number of queens placed on the board.

Row and Column Constraints

Row Constraint

- Only one queen can be placed in each row, preventing horizontal threats.
- Mathematically: $\sum_{i=1}^{n} x_{ij} \leq 1, \text{ for all } j.$

Column Constraint

- Only one queen can be placed in each column, preventing vertical threats.
- Mathematically: $\sum_{j=1}^{n} x_{ij} \leq 1, \text{ for all } i.$

Diagonal Constraints (Top Right to Bottom Left)

Constraint Formulation: Diagonal

To prevent queens from threatening each other along diagonals sloping from top right to bottom left:

- Select values for k iteratively from the set $\{-(n-1), -(n-2), \ldots, 0, 1, 2, \ldots, (n-1)\}$.
- Add the constraint:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \leq 1 \quad \text{if} \quad i + j = k.$$

Diagonal Constraints (Top Left to Bottom Right)

Constraint Formulation: Diagonal

To prevent queens from threatening each other along diagonals sloping from top left to bottom right:

- Select values for k iteratively from the set $\{2, 3, 4, \ldots, 2n - 1\}$.
- Add the constraint:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \leq 1 \quad \text{if} \quad i + j = k.$$

Optimization Framework

Framework

Pyomo (For Optimization Modeling)

Solver

CBC (Open-source MILP solver)

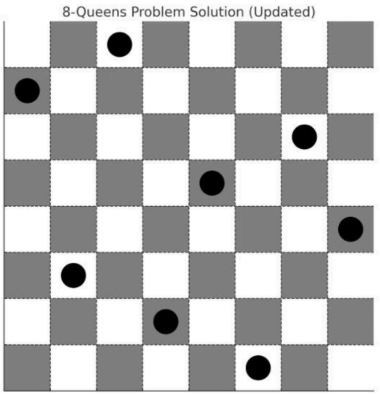
Approach

- Formulate the problem using Pyomo's constraint and objective modeling capabilities.
- Solve the problem using CBC to obtain the optimal solution.

Optimal Placement

Ideal Spot

The figure below shows the optimal placement of queens on the chessboard, where the black dots represent the positions of the queens that have been successfully placed.



What is a Genetic Algorithm?

Overview

Inspired by natural selection to search for optimal solutions.

Core Concepts

- **Population:** Group of candidate solutions.
- **Evolution:** Improvement over generations.
- **Key Operators:** Selection, Crossover, Mutation, Elitism.

Application

Useful in optimization problems like N-Queens.

Step 1: Initialization and Fitness Function

Initialization

- Create a random population of chromosomes.
- Each chromosome: A list of integers (row positions for queens in columns).

Fitness Function

Measures solution quality by counting conflicts:

$$\text{Fitness} = \text{maxFitness} - (\text{horizontal conflicts} + \text{diagonal conflicts})$$

Components of Fitness

- **MaxFitness:** $\frac{N \cdot (N-1)}{2}$ (maximum fitness with no conflicts).
- **Horizontal Conflicts:** Queens in the same row.
- **Diagonal Conflicts:** Queens on same diagonals:
  - Left: row − column, Right: row + column.

Step 2: Selection and Crossover

Selection

- Choose chromosomes for reproduction based on fitness.
- Method: **Roulette-Wheel Selection**
  - Chromosomes with higher fitness have a greater chance of selection.
  - Probability:

$$P(\text{chromosome}) = \frac{\text{fitness}}{\sum \text{fitness of population}}$$

Crossover

- Combines two parent chromosomes to produce offspring.
- In this problem, we use random gene inheritance (Uniform Crossover).
- Each gene (position) in the child is chosen randomly from one of the parents.
- For each position in the chromosome, randomly pick the value from either Parent 1 or Parent 2.

Step 3: Mutation and Elitism

Mutation

- Introduces random changes to maintain diversity in the population.
- **Key Benefits:**
  - **Improves Exploration:** Helps explore areas of the solution space that might not be reached through crossover alone.
  - **Escapes Local Optima:** Shakes up the population to move away from suboptimal solutions.

Elitism

- Ensures the best solutions are preserved across generations.
- **Process:**
  - Select the best and worst chromosomes from the current population.
  - Copy these chromosomes directly into the next generation without changes.
  - The rest of the next generation is created using crossover and mutation.

Step 4: Termination

Algorithm stops when:

- A solution with maximum fitness is found (no conflicts).
- A predefined number of generations is reached.

Output:

- The chromosome representing the solution.
- Visualization of the chessboard.

Result for $N = 8$

Solution

Chromosome: [0, 6, 4, 7, 1, 3, 5, 2].

Q	x	x	x	x	x	x	x
x	x	x	x	**Q**	x	x	x
x	x	x	x	x	x	x	**Q**
x	x	x	x	x	**Q**	x	x
x	x	**Q**	x	x	x	x	x
x	x	x	x	x	x	**Q**	x
x	**Q**	x	x	x	x	x	x
x	x	x	**Q**	x	x	x	x

Conclusion: CBC Solver vs. Genetic Algorithm

CBC Solver (Constraint-Based Approach)

The CBC (Coin-or branch and cut) solver is a robust mathematical programming solver used to solve the N-Queens problem by formulating it as a (MILP).

Genetic Algorithm (GA)

The Genetic Algorithm is a heuristic search method inspired by the process of natural selection. In the context of the N-Queens problem:

Key Differences

- **CBC Solver**: Guarantees optimality, but computationally expensive for large N
- **Genetic Algorithm**: Faster and scalable for large boards, but no guarantee of finding the optimal solution.

Thank You!

Your attention is greatly appreciated.

Have questions? Let's discuss!