8 QUEENS PROBLEM

Pietro Cicciari,
Mario Choto,
Israel Sanchez,
Raj Kumar,
Armon Lee,
Venkat Sairam Ravala

COMP 469 - ARTIFICIAL INTELLIGENCE

CONTENT

VARIOUS ALGORITHMS CAN SOLVE THIS PROBLEM:

The challenge of the 8 queens problem is to place eight queens on an 8x8 chessboard such that no two queens can attack each other. In chess, a queen can move any number of squares horizontally, vertically, or diagonally.

- Hill Climbing Algorithm
- Simulated Annealing Algorithm
- Genetic Algorithm

FORMALIZING THE PROBLEM

THE 8 QUEEN PROBLEM CAN BE DEFINED AS:

- Initial State: Empty chess board.
- Goal State: Eight queens are present in the board and do not attack eachother.
- Actions: Place queen, move queen, check for solution.
- Transition Model: Defines the result of each queen's movement.
- Action Cost: Uniform cost for each move.

ALGORITHMS OVERVIEW

WE EXPLORED THE FOLLOWING ALGORITHMS:

1. HILL CLIMBING ALGORITHM
2. SIMULATED ANNEALLING ALGORITHM
3. GENETIC ALGORITHM

HILL CLIMBING ALGORITHM

1
Start with a random configuration of eight queens on the chessboard.

2
Compute the cost, defined as the number of attacking queen pairs.

3
Move one queen at a time to create neighboring configurations
.

4
Choose the neighbor with the lowest cost as the new configuration.

5
Continue evaluating and selecting until no improvement is possible or a solution is found.

- Completeness: Not complete
- Cost Optimality: Not optimal
- Time Complexity: O(n)

n: the number of evaluations or steps required

- Space Complexity: O(1)

SIMULATED ANNEALING ALGORITHM

1

Start with a random configuration of eight queens on the chessboard.

2

Set a high initial temperature and Compute the heuristic

3

Modify the configuration and calculate the difference from the current heuristic

4

Accept the new configuration based on a probability that decreases with increasing cost difference and decreasing temperature.

5

Gradually reduce the temperature according to a preset schedule,

6

Continue until the system cools to a final temperature or no improvements occur for a set period.

SIMULATED ANNEALING ALGORITHM (Cont.)

- Completeness: Not complete
- Cost Optimality: Not optimal
- Time Complexity: O(nm)
  - n: the number of iterations
  - m: the average time to evaluate and generate neighboring solutions
- Space Complexity: O(1)

GENETIC ALGORITHM

1

Generate random configurations of eight queens on the chessboard.

2

Assess fitness of each configuration by counting queen pairs that cannot attack

3

Choose configurations with higher fitness for reproduction.

4

Crossover: Mix parts of selected configurations to create new ones.

5

Mutation: Introduce small random changes to some new configurations.

6

Replacement: Replace less fit configuration s with new ones.

7

Repeat the process until a solution is found or a set number of generations is reached.

GENETIC ALGORITHM (Cont.)

- Completeness: Not complete

- Cost Optimality: Not optimal

- Time Complexity: O(gpe)
  – g: the number of generation
  – p: population size
  – e: evaluation time

- Space Complexity: O(pn)
  – p: population size
  – n: size of the solution representation

PERFORMANCE ANALYSIS

- Hill-Climb Algorithm: It's fast and uses little memory, but it's not always reliable.
- Genetic Algorithm: It's strong at finding solutions but needs more time and memory.
- Simulated Annealing: It's a good balance between finding solutions and using resources, but it can be slow if not set up correctly.

CONCLUSION

We conclude that each algorithm used to solve the 8 Queens problem has its pros and cons. The ideal algorithm to use depends on what's more important for the problem like speed, reliability, or resource use.

For general use, Simulated Annealing is the best choice because it balances everything without risking unwanted behavior.