



Assignment 2

Using Local Search and Constraint Satisfaction Algorithms to Solve 8-Queens

1 Overview

The eight queens puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other, thus, a solution requires that no two queens share the same row, column, or diagonal.

You will be given an initial configuration of 8×8 chessboard. The eight queens puzzle has 92 distinct solutions. You have to find any valid final configuration starting from the given initial configuration.

Your solution should read initial configuration from a file with the following format:

Q	Q	Q	Q	Q	Q	Q	Q
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#

Your output format must be as the input file.

You can test your implementation with a [Sample Input File](#) (attached in the resources files in piazza).

You should solve the 8-queens problem using the following algorithms:

- 1) Hill Climbing Algorithm
- 2) K-Beam Search Algorithm.
- 3) Genetic Algorithm.
- 4) Constraint Satisfaction Problem (CSP).

The cost function for all algorithms will be the total number of attacking queen pairs. For example, the cost function for the above configuration is 28.



1 Hill Climbing Algorithm

You will start from the initial board. You should move one queen horizontally, vertically or diagonally in one step.

Your algorithm should try sideways moves / random restarts if it was stuck in a local minimum.

You should print the following to the console:

1. The running time of your algorithm.
2. The final board in the same format used in the input.
3. The cost to the final board. (Number of steps to reach final board from the initial board).
4. Number of expanded nodes. (Number of unique boards visited to obtain the final board).

2 K-Beam Search Algorithm

You will start from the initial board. You should move one queen horizontally, vertically or diagonally in one step.

You should determine the best K for your algorithm.

You should show the behavior of your algorithm with $K = 1$ and with a very large value of K.

You should print the following to the console:

1. The running time of your algorithm.
2. The final board in the same format used in the input.
3. The cost to the final board. (Number of steps to reach final board from the initial board).
4. Number of expanded nodes. (Number of unique boards visited to obtain the final board).
5. The best K for your algorithm.



3 Genetic Algorithm

You will start from the initial board. You should apply the genetic algorithm to it to reach a valid configuration.

You should print the following to the console:

1. The running time of your algorithm.
2. The final board in the same format used in the input.
3. The cost to the final board. (Number of steps to reach final board from the initial board).
4. Number of expanded nodes. (Number of unique boards visited to obtain the final board).

4 Constraint Satisfaction Problem (CSP)

You will start from the initial board. You should move one queen horizontally, vertically or diagonally in one step.

You should solve the problem using backtracking or any other proper method and use the CSP to prune your solution and find a solution faster than ordinary backtracking.

You should print the following to the console:

1. The running time of your algorithm.
2. The final board in the same format used in the input.
3. The cost to the final board. (Number of steps to reach final board from the initial board).
4. Number of expanded nodes. (Number of unique boards visited to obtain the final board).



5 Deliverables

1. Your well commented code.
2. A report showing your work, including:
 - (a) Pseudo code for each algorithm mentioned.
 - (b) Data Structures used in each algorithm.
 - (c) Assumptions made for each algorithm.
 - (d) One Sample run showing the output to the sample input file given for each algorithm. You should show the requirements for each algorithm printed to the console.
 - (e) How to run your code from terminal and any required dependencies.

6 Further Notes

- You may use Java, Python or C++ for your implementation.
- GUI will be granted as a bonus.
- Copied assignments will be severely penalized.
- You **must** work in groups of at least 2 and at most 3.

Good Luck