Takuya Gerret Kubota

CS 420

Professor Tang

Project 2

N-Queens Problem

For this project, I have generated 100 cases for both steepest-ascent hill climbing algorithm and min-conflicts algorithm. This particular project was required to have the programmer to create an 18 by 18 board that has 18 queens on each column, and the objective is to keep each of the queens from attacking each other. As I have tested the 100 cases for the first algorithm of steepest-ascent hill climbing algorithm, I have done three outputs for it. Out of the three outputs, two of them, which had a 1.0% success rate of finding a solution, and the last output had a 0.0% success rate. I believe the success rate of finding the solution is very low to nothing, is because the algorithm only allows the board to be checked once, from left to right.

For the second algorithm, I have implemented the min-conflict algorithm. This algorithm chooses a random column of the board and finds the heuristic of how many queens are attacking in each of the coordinates of that random column. After you calculate the heuristic, choose the lowest heuristic and move the queen there and repeat the process of randomly choosing the column and calculating the heuristic until you find the solution of zero for the heuristic, which means that there are no more queens attacking each other. I believe this algorithm is more optimal than the steepest-ascent hill climbing algorithm because it goes through each column multiple times until the solution is found.

In conclusion, I believe that the min-conflict algorithm is more optimal because it actually finds the solution for the n-queens problem. The only con that I can identify is that the time of it finding the solution may be time consuming. Through the analysis of this algorithm, with the 100 test cases for this n-queens problem, the time it took to find the solution averaged around 30 milliseconds with three different outputs.