Samanyu Satheesh

Dr. Tang

CS4200

1 August 2023

Project 2

For this project, we were asked to develop programs that could solve the 8 Queens problem. For part 1, I developed a steepest ascent hill-climbing algorithm, and for part 2, I developed a genetic algorithm. Both of these algorithms operate differently, so it was a learning experience to be able to implement them.

Firstly, here is an evaluation of the steepest ascent hill-climbing algorithm. I found that my algorithm seemed to solve around 14% of cases as can be seen in the table below.

Cases	Time (ms)	Nodes generated	Accuracy Percentage
1000	266	227304	14.7%
1000	291	222096	13.4%
1000	277	227696	13.9%
1000	292	226576	14.5%
1000	296	228424	15.1%

I believe the reason behind this performance is the nature of the hill-climbing algorithm. Steepest ascent hill-climbing stops as soon as it can't find an instant improvement. This results in the algorithm stopping at a local maximum instead of the global maximum, hindering its ability to solve the problem.

Next, I developed a genetic algorithm that utilized generations, selection of the fittest, creation of offspring, and mutations. Since Dr. Tang said that if we did the genetic algorithm then

the goal was to solve the problem, I chose to first develop a successful algorithm and THEN play around with the population size, generation size, and mutation rate to see where I could find more success. I felt that increasing the number of generations allowed for more successes and a better chance to evolve into a successful iteration of the 8 queen board.

I have attached my proof of solving 3 cases using the genetic algorithm in my example output file, as well as the hill-climbing algorithm tests.