Marlon Aquino

Professor Dominick Atanasio

CS 4200.01

Project 2

Approach

My approach to the Simulated Annealing and Genetic algorithms began with some guidelines which included the following:

1. Starting with an initial configuration
2. Repeatedly search the neighborhood and selecting a successor as a candidate
3. Evaluating a fitness function and accepting if the successor is better and if not select another successor
4. Stopping once no improvement can be found after some time.

I made different classes which are named Node, GeneticAlgorithm, SimulatedAnnealing, and Main. The Simulated Annealing algorithm had an initially set temperature, which was 1, and then was allowed to 'cool' as the algorithm ran. I also set the decayRate to be 0.999. The higher the temperature was, the higher the frequency to accept worse solutions than the current state. As the temperature cooled down, the algorithm was more careful in choosing a potential successor which led it closer to a viable solution.

The second approach involved the genetic algorithm which finds a solution by creating a scenario where only the fittest nodes survive. A mutation probability was calculated and a crossover between two parents "reproduce" which can lead the algorithm closer to a potential solution.

To determine the fitness score, I used the number of attacking queens as a way to calculate it. The lower the number of attacking queens, the better than fitness score. The fittest node would be one in which a node has no queens attacking it.

Problems Encountered

The Simulated Annealing algorithm overall I did not have problems with. On the other hand, I did have problems when implementing the Genetic Algorithm because for some reason, the chosen parents would be the same even though I initially had them to be chosen at random. Additionally, there is an infinite loop that I could not fix because the algorithm could not detect when the fitness score of a node was zero.

Analysis

Comparing both algorithms, even though the Genetic Algorithm did not work, Simulated Annealing is one that takes in only a single input while the Genetic Algorithm takes in a collection. With Simulated Annealing, if it has only one optimal solution, then it can be applied better than the Genetic Algorithm, but if the problem has more than one optimal solution, then Genetic Algorithm will find solutions much better than Simulated Annealing.