Straight-forward steepest-ascent hill climbing

When designing this algorithm, I set it to find the local minimum of the number of queen pairs attacking (lower is better). If it cannot find one with less attacking pairs than the current state, the algorithm will return the current state regardless of whether it is solved.

I observed that the percentage of problems solved by this algorithm was typically around 14-15% as expected, and each solved problem took about 4 steps from the initial randomly generated board of queens. It is quite a fast algorithm, as it pursues the most beneficial moves.

However, it proves to not be reliable considering how low the percentage of solved cases is. Since it will only accept a new state if it has fewer attacking queens than the previous state, it will often get stuck with no possible moves to make.

Genetic algorithm

When designing this algorithm, I had it generate a random population of size 100 at each iteration. The fitness function returned the number of non-attacking queen pairs (higher is better), the selection and crossover points were randomized, and the mutation rate was set to 10%.

I observed that each iteration of the algorithm always found a solution. It typically took a few seconds each time to do so. For most iterations, hundreds of thousands of children were generated. It is a tedious but reliable algorithm, as it will always find a valid solution at the end.

Analysis and Comparison

The hill climbing algorithm tends to find solutions very fast given enough test cases, as it makes the most strategic moves it can to reach a valid n-queens board. However, because the algorithm is strict in not allowing “sideways” or “bad” moves, the percentage of problems successfully solved ends up being quite low.

The genetic algorithm, on the other hand, will always find a valid solution in the context of the n-queens problem since the random exploration is persistent. This algorithm is not the most efficient, but crossover and mutation allow for a very large number of possible board configurations to be checked.

The two algorithms are beneficial in their own ways; hill climbing is faster, while the genetic algorithm is more dependable.