Evelyn Vu

CS 4200.E01

Dr. Daisy Tang

10 January 2024

CS 4200 - Artificial Intelligence Project 2 Report

This project is to solve n-Queen problems (with n = 8) using Local Search Algorithms. There are two algorithms implemented: (1) Straight-Forward Steepest-Ascent Hill Climbing and (2) Genetic Algorithm. This project also compares the two algorithms in terms of efficiency (using success rate, average search cost and runtime)

The n-Queen problem is the problem of putting n chess queens on an n x n chessboard such that none of them is able to capture any other using the standard chess queen's moves. The queens must be placed in such a way that no two queens would be able to attack each other.

Two approaches for this project:

(1) Straight-Forward Steepest-Ascent Hill Climbing works by starting with a random solution, then iteratively making small modifications to that solution that "climb the hill" towards higher quality solutions. At each step, it makes the modification that achieves the highest immediate gain, even if it leads to a local rather than global optimum.

(2) Genetic Algorithm works by generating a population of solutions, selecting the fittest ones, then applying crossover and mutation operators to produce new offspring solutions. This continues over many generations, allowing the solutions to evolve towards increasingly optimal solutions for the problem.

With 8x8 chess board, population size of 100 and 100 generations, after 100 instances, we captured the results of two algorithms:

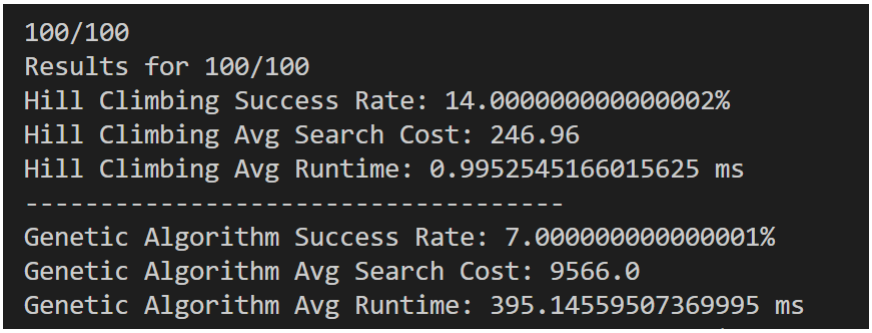

Figure 1: Results of 100 instances of 8-Queen problems

Based on the result of 100 instances, Straight-Forward Steepest-Ascent Hill Climbing gave better results than Genetic Algorithms in terms of success rate (14% > 7%), average search cost (246.96 < 9566) and average runtime (0.995 ms < 395.146 ms).

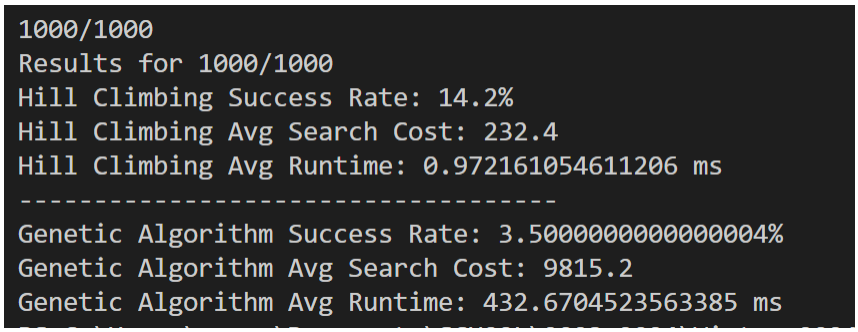

Figure 2: Results of 1000 instances of 8-Queen problems

With 1000 instances, Straight-Forward Steepest-Ascent Hill Climbing still out-performed Genetic Algorithms with success rate 14.2% > 3.5%, average search cost 232.4 < 9815.2, and average runtime 0.972 ms < 432.67 ms. Meanwhile, the result can be considered accurate as Hill Climbing algorithm keeps its success rate at around 14%.

The different results between these two algorithms might be because Hill climbing converges fast by making greedy choices but to suboptimal solutions, while Genetic algorithms take more computations through generations of evolution but can find better solutions. With only 100 generations, the Genetic Algorithm may not have enough time to fully explore the search space and converge on quality solutions.