C.B. Attidiya

ENG/15/021

Solving the 8 Queen problem using Simulated Annealing

Simulated Annealing gets its name from real life annealing in metallurgy. If we consider molten glass which is extremely hot, it cools down quickly. Room temperature is too cold for it and would crack the glass. Therefore, an annealing oven is used to slowly lower the temperature in a stable manner. The simulated annealing algorithm also has these temperature and annealing rate values which the user must configure.

The simulated annealing algorithm is unlike the hill climbing algorithm as it achieves the global maxima/minima. The algorithm starts by attempting risky moves as in molten glass could be deformed in higher temperatures but slowly try heuristic value optimizing moves as in forming the required shape in glass.

The simulated annealing algorithm also introduces randomness to the 8-queen problem, making it impossible to guess which of the 92 solutions it would produce or not. Any 8-queen state can be reached from any other 8-queen state.

The key feature of the simulated annealing algorithm is its flexibility. It gives the programmer freedom in selecting parameters, which can be used to optimize the output according to the nature of the problem. The temperature function, annealing rate, next state generator, evaluation function and number of iterations are the variable parameters.

The simulated annealing algorithm can deal with highly nonlinear models. When it comes to NP complete problems, it is one of the approaches which can handle large state spaces efficiently. It also performs heuristic evaluation in linear time.

Due to the flexible nature of the algorithm, quality of the results and performance of the algorithm gets influenced. Therefore, the algorithm must be customized for a given problem to obtain best performance and high-quality results.

In my implementation of the code the temperature is set to 64 while annealing rate is set at 0.95, There is also an iteration limit to stop the code in case the algorithm is unable to finds a solution as the temperature value reduces, which is also large enough that a solution can be found. From the implementation I have found out that there were several instances where the algorithm was unsuccessful in finding a solution. The algorithm is rendered useless due to the initial random placement of queens in the board. It also reduces the time required to obtain a solution occasionally, and it always provide a different solution compared to fixed starting states.

All in all, simulated annealing algorithm can quickly find good solutions to very difficult problems without even thinking about them.

Link to github : https://github.com/cha7ura/8-Queen-problem-simulated-annealing