Artificial Intelligence - **Assignment 1**

Name : N. J. Bulathsinghala

Reg. no. : ENG/15/074

Stream : Electrical and Electronic

Intake : 32

GitHub URL : https://github.com/NisalBulathsinghala/AI-Assignment-Nisal.git

---

The objective of this program is to arrange the eight queens in the chessboard such that queens are not attacking each other. This is a local search problem, and it can be defined as follow,

- The states of the problem are any arrangement of the eight queens is the chessboard, where only one queen will be placed per column.
- The initial state is that where all the eight queens are placed in a single row (topmost or bottommost).
- The successor function returns the next best move of a queen.
- The cost function is defined as the number of queens attacking each other for a given arrangement of the eight queens.

The algorithm used in this code is the Hill Climbing search algorithm. The program starts by creating the initial state of the chessboard and queen placements. Afterwards, the H cost for the initial state will be calculated. When calculating the H costs, the program will check whether two or more queens are in the same row or diagonally. For each violation, the H cost will increased by one. Since local search algorithms uses complete-state formulation, a list known as 'stateList' is maintained, which will contain all the possible states of the queens (8 x 7 = 56) and the corresponding H costs. The H costs of the possible states will vary.

Next, using the stateList, the successor function will return the next move of a queen. In this function, first the list will filtered to retrieve the moves with lowest H costs. Since there is a possibility of the stateList containing more than one state which will give the lowest H cost, another list known as 'bestList' will be created to hold the possible states with the lowest H costs. From the bestList the program will select the first entry and this will be executed. Again, the H costs for the new arrangement will be calculated and the process is repeated until the program reaches a solution.

The issue with this program is that, the successor selected is simply the first entry of all the possible states with lowest H costs. This does not guarantee a global maximum with a cost of H = 0. For this program the achieved solution is local maximum of a cost of H = 1. The chances of achieving a global maximum, in this case, depends on the initial state.

A slight modification can be done to this program, which will give the program better chances of achieving global maximums is to employ the Stochastic Hill Climbing algorithm. In this approach, the successor from the bestList will be selected at random, which will generate random solutions each time the program is run. Even this does not guarantee a global maximum 100% of the time, but it partly eliminates the error of the previous approach of being held at a local maximum completely.

The program can be further improved employing algorithms such as Random Restart Hill Climbing, which will guarantee a global maximum solution.