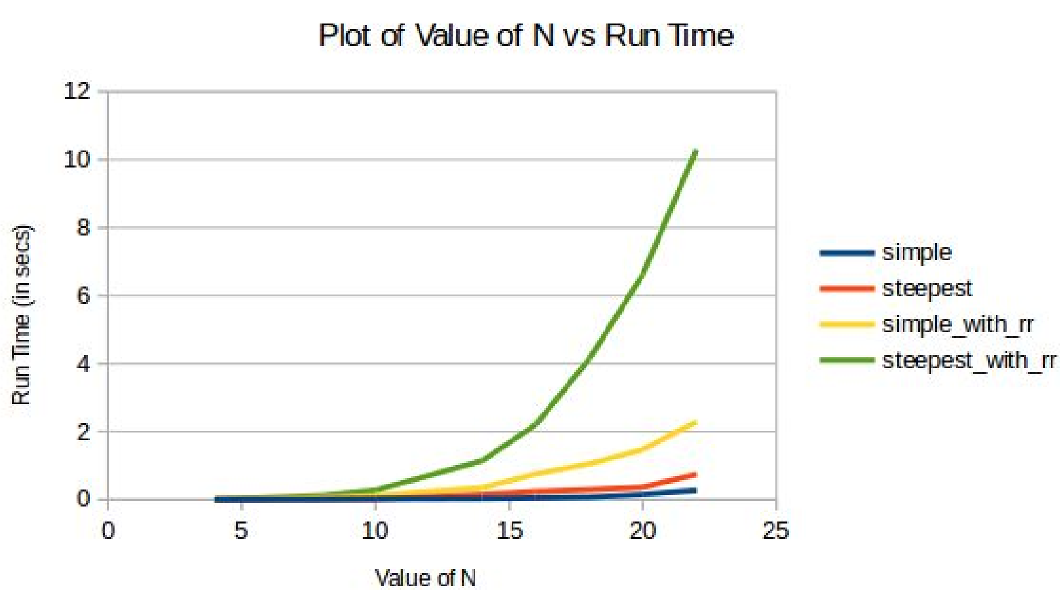
Artificial Intelligence
Assignment 1

Arushi Jain (2013023)

Introduction:
We have initially set up a square field of 64 cells with 8 queens (or agents) of allowed movements as given in the question. 2 different variants of the hill climbing technique have been used to arrive at the solution, namely Simple hill climbing and Steepest Ascent hill climbing.
The code has been designed to run any one of the following algorithms based on user input (please find code attached in the submission):

1. Simple hill climbing without random restart
2. Steepest Ascent hill climbing without random restart
3. Simple hill climbing with random restart
4. Steepest Ascent hill climbing with random restart

Extending the problem to N agents:
As value of N increases, the number of iterations and probable moves increases. Therefore the chance of solving the problem decreases as N increases.
Therefore, the time taken to solve the problem will increase with N for all the cases above. In the case of variants with random restarts, more restarts will be required for higher N to get a solution. To measure a suitable value of N, we set the maximum allowed restarts number to 10, and then measured the time taken to arrive to a solution. We concluded that the problem was solved in a reasonable time till around N = 20. We can clearly see in the graph below that there is a marginal increase in the runtime for N more than or around 20.

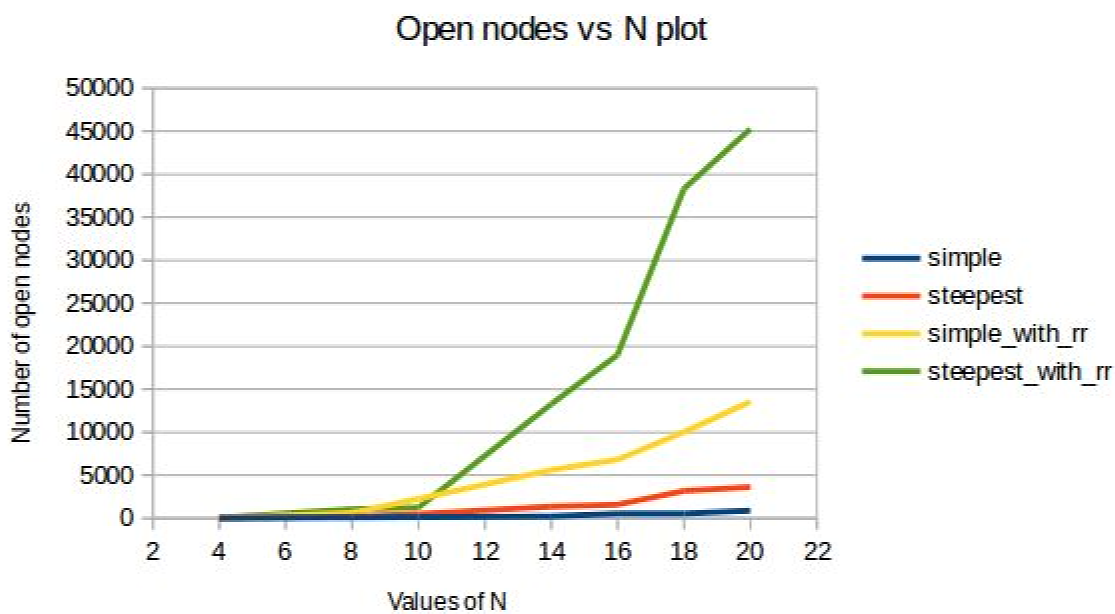


Measuring search cost and number of solved problems (with and without random restart):
(Plots also given below):
We have taken 100 runs **of each** of the above 4 algorithms/cases and noted the search cost (runtime and number of checked arrangements) and whether the problem was solved or not for each run. Following is the analysis from the values obtained. The files with the obtained values are attached in the submission. This is for the case when N=8.

1. **Simple hill climbing without random restart:**
   For N=8 :
   Average number of arrangements checked = 95
   Average runtime (search cost) = 0.0042 sec
   Number of problems solved = 19/100
2. **Steepest Ascent hill climbing without random restart:**
   For N=8 :
   Average number of arrangements checked = 222
   Average runtime (search cost) = 0.0076 sec
   Number of problems solved = 13/100

3. **Simple hill climbing with random restart:**
   For N=8 :
   Average number of arrangements checked = 489
   Average runtime (search cost) = 0.0188 sec
   Number of problems solved = 69/100
4. **Steepest Ascent hill climbing with random restart:**
   For N=8 :
   Average number of arrangements checked = 1150
   Average runtime (search cost) = 0.038 sec
   Number of problems solved = 77/100

We have also calculated the number of open nodes for each of the 4 cases against various values of N namely 4,8,10,14,16,18,20. The plots are given below. Here rr implies random restart.



Comparison of Simple Hill Climbing and Steepest Ascent Hill Climbing

We calculated the average number of arrangements (or search cost) checked and number of solved problems for both the variants. We observe that the number of arrangements checked (or search cost) for Steepest Ascent hill climbing is always much higher than Simple hill climbing. However, as we allow random restarts, the number of problems solved by Steepest Ascent hill climbing is much more than the case of Simple hill climbing. While one gives better results, the other is cost efficient. Therefore, it is difficult to conclude clearly which variant is better. This is probably why both the variants are popular. It depends on the need of the user, whether he wants to compromise on cost or results.

From the above plots we see that the number of open nodes mostly increases with the value of N. We can, however, notice that the jump in the number of open nodes with increasing N is higher in the case of Steepest Ascent hill climbing as compared to Simple hill climbing. Therefore, as N increases, the gap between the open nodes count for Steepest Ascent hill climbing and Simple hill climbing increases. This is because the number of unique arrangements opened by Steepest hill climbing is higher because it opens 56 arrangements each time while so is not the case for Simple hill climbing. Time complexity of both the variants algorithm is n^4.

Algorithm Used:

Initialization:

We have assumed that each column of the board has one queen only. Each queen is thus moved only in the column it belongs to. This reduces the number of maximum possible checks to get a solution considerably. Thus, the initial positions of the queens are random positions such that each column has one queen.

Heuristic Calculations:

For a given set of queen positions on the board, we calculate the heuristic value by calculating the number of attacking pairs of queens. For this, we store the position of each queen on the board and count the number of queens that lie on the same row or diagonal of this queen (columns not checked as each column has one queen). Since each attacking pair is counted twice by this method, the obtained value is divided by 2 in the end.

Random Restart:

In case we get stuck at a position such that the present heuristic value is non zero and a better heuristic value cannot be obtained, we repeat the process of assigning random intiial positions to the queens and then trying to find a solution. We have set the maximum number of allowed restarts to 8, which seems to be an optimal number to get a solution for 8 queens. We stop as soon as the obtained heuristic is 0 or if the number of restarts is 8.

Simple Hill Climbing:

We randomly move one queen to a new position and calculate the heuristic value of this new arrangement of queens.

If this heuristic value is more than or equal to the present heuristic value, we put the moved queen back to its previous position and make another new random movement and calculate its heuristic value. We keep on doing this till the heuristic value comes out to be lesser than the present heuristic value or if all the possible positions have been checked once.

If this heuristic value is lesser than the present heuristic value we return it as the obtained heuristic value.

If all positions have been checked once and none gave a heuristic value less than the present heuristic value, the value of the obtained heuristic value is set same as the present heuristic value and returned.

If the obtained heuristic value is lesser than the present heuristic value, we set this obtained heuristic value as the present heuristic value and the arrangement corresponding to it as the new position of the queens. We repeat the above process of getting a new heuristic value again.

If the obtained heuristic value is same as the present heuristic value (or zero), we stop. If it is zero, it means we have found a solution.

If it is not zero, we know that this was the best we could find with the present arrangement of queens. We either stop here (for without random restart) or start the entire procedure again with a new set of random initial positions of the queens (for with random restart).

Steepest Ascent Hill Climbing:
We calculate the heuristic value for each arrangement that is obtained by moving any one queen to a new position in the present arrangement. There will thus be 56 such arrangements. We then find the minimum heuristic obtained in these 56 arrangements. The obtained heuristic value is the minimum value among the 56 heuristic values obtained.
The obtained heuristic value will either be same as the present heuristic value or less than that.

If the obtained heuristic value is less than the present heuristic value, we set it as the present heuristic value and the new arrangement of the queens is set to the one which gave this new heuristic value. We then repeat the process to get another heuristic value.
If the obtained heuristic value is same as the present heuristic value (or zero), we stop. If it is zero, it means we have found a solution.
If it is not zero, we know that this was the best we could find with the present arrangement of queens. We either stop here (for without random restart) or start the entire procedure again with a new set of random initial positions of the queens (for with random restart).