UNC CHARLOTTE
College of Computing and Informatics

PROJECT DOCUMENTATION REPORT

ON

Solving N-Queens Problem by Hill-Climbing and its variants

PROGRAMMING PROJECT 2

ITCS 6150 - Intelligent Systems

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED TO

Dewan T. Ahmed, Ph.D.

SUBMITTED BY:

Rishi Parikh **Raj Shah**

801217780 **801205036**

Table of Contents

PROBLEM FORMULATION

INTRODUCTION

What is N-Queen Problem?

The N-Queen problem originally introduced in 1850 by Carl Gauss. The goal of this problem is to place N queens can take each other. Queens can move horizontally, vertically, and diagonally, this means that there can be only one queen per row and one per column, and that no two queens can find themselves on the same diagonal.

It is a computationally expensive problem – NP complete, what makes it very famous problem in computer science.

Hill Climbing Search

In Hill-Climbing technique, starting at the base of a hill, we walk upwards until we reach the top of the hill. In other words, we start with initial state and we keep improving the solution until it's optimal.

It's a variation of a generate-and-test algorithm which discards all states which do not look promising or seem unlikely to lead us to the goal state. To take such decisions, it uses heuristics (an evaluation function) which indicates how close the current state is to the goal state.

Hill Climbing with Sideways move

The hill climbing search algorithms halts if it reaches plateau where the best successor has same value as the current state.

Then we allow algorithm to move sideways in the hope that the plateau is actually is a shoulder.

The algorithm will go in an infinite loop if the algorithm reaches to a flat local maximum and that is not a shoulder.

The only solution of it is to put limit on the number of consecutive sideways move allowed.

Random Restart Hill Climbing Without Sideways Move

Random restart hill climbing conducts series of hill climbing searches from randomly generated initial states, until a goal state is found.

The probability of this technique approaches 1, because sometimes it will eventually generate goal state as the initial state.

If each hill-climbing search has a probability p of success, then the expected number of restarts required is 1/p.

For 8-queens instances with no sideways moves allowed, $p \approx 0.14$, so we need roughly 7 iterations to find a goal (6 failures and 1 success). Roughly 22 steps in all.

Random Restart Hill Climbing With Sideways Move

Random restart hill climbing conducts series of hill climbing searches from randomly generated initial states, until a goal state is found.

The probability of this technique approaches 1, because sometimes it will eventually generate goal state as initial state.

If each hill climbing search has a probability p of success, then the expected number of restarts required is 1/p.

For 8-queens instances, when we allow sideways moves, $1/0.94 \sim 1.06$ iterations are needed on average and $(1 * 21) + (0.06/0.94) * 64 \sim 25$ steps.

ALGORITHM PSEUDOCODE

```
function HILL-CLIMBING(problem) returns a solution stateinputs: problem, a
    problem
    static: current, a nodenext, a node
    current <— MAKE-NODE(INITIAL-STATE[problem])loop do
        next— a highest-valued successor of current
        if VALUE[next] < VALUE[current] then return currentcurrent *—next
    end
```

EXPECTED OUTPUT

<u>Hill climbing</u>

Success rate is: 40.4 % and Failure rate is: 59.6 %

The average number of steps when the algorithm succeeds: 2.02

The average number of steps when the algorithm fails: 2.27

<u>Hill climbing with sideways</u>

Success rate is: 100.0 % and Failure rate is: 0.0 %

The average number of steps when the algorithm succeeds: 2.77

<u>Random restart without sideways</u>

The average number of random restarts required without sideways move 1.22

The average number of steps required without sideways move 4.75

<u>Random with sideways</u>

The average number of random restarts required with sideways move 0.0

The average number of steps required with sideways move 2.79

PROGRAM STRUCTURE

GLOBAL / LOCAL VARIABLES

We use various global/local variables in our program:

The local variables within function are:

- hits – calculates the number of row collisions as we as diagonal collisions for a particular cell
- i, j – for the range
- row – variable that stores the row index
- col – variable that stores the column index
- checkBetter – flag variable that stores true or false when a better solution is found
- best – duplicate array to handle different operations
- mins -- variable that stores the collisions of current board so that if finds better than this it will quit
- store – this is an array list that stores the columns from which a random column will be selected
- randomCount – counter variable that stores the number of random restarts
- movesTotal – variable that stores the total number of moves
- movesSolution – variable that stores the total number of moves in the solution set
- maxSteps – variable that stores the maximum steps of the iteration
- Iterations – variable that stores the number of iterations for sideways move hill climbing
- min_compare – variable to compare the position in column with the minimum conflicts
- randomValue -- this stores the current queue position in the randomly selected column
- currentValue – stores the value of current column
- Index – stores the index value of a particular column on the board

- The global variables within function are:
- choice – variable that stores the input value from the user for a particular operation that it chooses
- a – array that stores the initial configuration of the board
- b – duplicate array that stores the same value as variable a
- queen – object variable of the class NQueenProblem

FUNCTIONS / PROCEDURES

Our code implements n queens problem using Hill Climbing local search method and its variants – Hill Climbing, Hill Climbing with Sideways move and Random restart Hill Climbing using Python.

We use the following functions within the code:

- rowHits() - This method calculates all the row conflicts for a queen placed in a particular cell.
- diagonalHits() - This method calculates all the diagonal conflicts for a particular position of the queen
- totalHits() - This method returns total number of collisions for a particular queen position
- optimalSol() - This method calculates the conflicts for the current state of the board and quits whenever finds a better state.
- randomGen() – This function generates a random state of the board
- checkSol() – This method verifies whether the current state of the board is the solution(I.e. with zero conflicts)
- minHits() - This method finds the solution for the n-queens problem with Min-Conflicts algorithm with random restart
- minHitswithoutRestart() - This method finds the solution for the n-queens problem with Min-Conflicts algorithm without random restart
- collisionsMinHits() - This method returns the collisions of a queen in a particular column of the board
- fillList() - This function fills the Array List with numbers 0 to n-1

LOGIC

We are running the code in python 3 in Anaconda package Spyder idle. We first generate a random board with n queens in it. The functions row collisions and column collisions count the number of attacks and keep a track of it. The movements are made as per the hill climbing technique used (Hill Climbing or Sideways moves) and collisions are compared and accordingly the next step is taken. With random restart technique, on failure, again a random board gets generated and it starts looking for a solution, hence a solution is guaranteed. We print the steps, success failure rates at the end. For Sideways moves technique, weput a **limitation over sideways moves to stop it from entering an infinite loop**. The more queens, the more steps. The more iterations, the more steps.

Commenting the print of the sequences as there are minimum 20 moves on success and 64 moves on failure in Sideway moves, might let the system to crash.

Calculation of average steps for success – Total number of success moves/ Total number of success

Calculation of average steps for failure – Total number of failure moves/ Total number of failure

Success/Failure rate = (Success/Failure)/Total * 100

SAMPLE OUTPUT

```
Please select one from the below options:

1. Solve n queens with Hill Climbing without restart
2. exit

Please enter the choice: 1

Please enter the value of n: 4

Please enter the value of iterations: 1

**********Hill Climbing without Random Restart*********Q  - -  -

 -  Q  -  -

 -  -  Q  -

 -  -  -  Q

Queen Position [0, 1, 2, 3]

 -  Q  -  -

 -  Q  -  -

 -  -  Q  -

 -  -  -  Q

Queen Position [1, 1, 2, 3]

 -  -  Q  -

 -  Q  -  -
 - - Q -

 - - - Q

Queen Position [2, 1, 2, 3]

 - - - Q

 Q - - -

 - - Q -

 - - - Q

Queen Position [3, 0, 2, 3]
solution not found

Total number of moves taken: 4 Number of
moves in the solution set:  1

Time Taken in milli seconds:  0.0010073184967041016

 - - - Q

 Q - - -

 - - Q -

 - - - Q
```

Please select one from the below options:

1. Solve n queens with **Hill Climbing Ascent with Random Restart**
2.exit

```
Enter your choice: 1


Please enter the value of n: 4

**********Hill Climbing with Random Restart*********Reached
Local Maxima with 8 Regenerating randomly Number of Restarts:  1

Total number of moves taken: 2 Number of
moves in the solution set:  2Time Taken in
milli seconds:  0.0

 - - Q -

 Q - - -
 -  -  -  Q

 -  Q  -  -

 average steps 3.0



Please select one from the below options:

1. Min Conflict method with random restart2.exit


Please enter the choice: 1 Please


enter the value of n:4

**********Sideways with Random Restart*********

Reached Local Maxima with  4  Regenerating randomlyNumber
of Restarts:  1

Total number of moves taken:  20

Number of moves in the solution set:  20

Time Taken in milli seconds:  0.000982046127319336

 -  -  Q  -

 Q  -  -  -

 -  -  -  Q

 -  Q  -  -

 average steps 20.0



Please select one from the below options:

1. Min Conflict method without random restart2.exit

 Please enter the choice: 1

 Please enter the value of n:4


*******Min Conflict Sideways Without Random Restart*******
```

```
How many times do you want to run the code??

Please enter the value: 1
Please enter the maximum number of steps for iteration:

Please enter the value: 25
 -  Q  -  -
 -  Q  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 1, 0, 2] 1
 -  Q  -  -
 -  Q  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 1, 0, 2] 2
 -  Q  -  -
 -  Q  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 1, 0, 2] 3
 -  Q  -  -
 -  Q  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 1, 0, 2] 4
 -  Q  -  -
 Q  -  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 0, 0, 2] 1
 -  Q  -  -
 Q  -  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 0, 0, 2] 2
 -  Q  -  -
 Q  -  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 0, 0, 2] 3
 -  Q  -  -
 Q  -  -  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 0, 0, 2] 4
 -  Q  -  -
 -  -  Q  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 2, 0, 2] 1
 -  Q  -  -
 -  -  Q  -
 Q  -  -  -
 -  -  Q  -
```

```
Queen Position [1, 2, 0, 2] 2
 -  Q  -  -
 -  -  Q  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 2, 0, 2] 3
 -  Q  -  -
 -  -  Q  -
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 2, 0, 2] 4
 -  Q  -  -
 -  -  -  Q
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 3, 0, 2] 1
 -  Q  -  -
 -  -  -  Q
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 3, 0, 2] 2
 -  Q  -  -
 -  -  -  Q
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 3, 0, 2] 3
 -  Q  -  -
 -  -  -  Q
 Q  -  -  -
 -  -  Q  -
Queen Position [1, 3, 0, 2] 4

-  Q - -
- - -  Q
 Q - - -
- -  Q -
Total number of Random Restarts: 0Total
number of Moves:  1
Number of Moves in the solution set:  1Yes
Time Taken in milli seconds:  0.0079977512359619141
 average success steps 1
Success- 1
Failure- 0
Success moves- 1
Failure moves- 0
```

PROGRAM IMPLEMENTATION

Implement hill climbing

a. Run several times, say 100 to 500, and report success and failure rates
We did run the 8 queens problem 100 times, it was success for 15 times and failed 85 times

b. The average number of steps when it succeeds
It did succeed in 3 steps
c. The average number of steps when it fails
It did fail in 4 steps
d. The search sequences from three random initial configurations

Case 1:

Please enter the number of Queens on board: 8Please

enter the value of iterations: 100

```
Q - - - - - - -    - - Q - - - - -    - - - Q - - - -
- - - - - - - Q    - - - - - - - Q    - - - - - - - Q
Q - - - - - - -    Q - - - - - - -    Q - - - - - - -
- - - - - - - Q    - - - - - - - Q    - - - - - - - Q
- - - - - Q - -    - - - - - Q - -    - - - - - Q - -
- Q - - - - - -    - Q - - - - - -    - Q - - - - - -
- - Q - - - - -    - - Q - - - - -    - - Q - - - - -
- - - - - - - Q    - - - - - - - Q    - - - - - - - Q
```

Solution Not Found

Total number of moves taken: 3 Number of moves in the solution set: 1

Case 2:

Q - - - - - - - - Q - - - - - - - - - Q - - - - Q - - - - - - - - - - - - Q - - - - - - - - Q - - - Q - - - - - - - - - Q - - -	- Q - - - - - - - Q - - - - - - - - - Q - - - - Q - - - - - - - - - - - - Q - - - - - - - - Q - - - Q - - - - - - - - - Q - - -	- - - Q - - - - - Q - - - - - - - - - Q - - - - Q - - - - - - - - - - - - Q - - - - - - - - Q - - - Q - - - - - - - - - Q - - -	- - - - Q - - - - Q - - - - - - - - - Q - - - - Q - - - - - - - - - - - - Q - - - - - - - - Q - - - Q - - - - - - - - - Q - - -

Solution Not Found

Total number of moves taken: 4 Number of moves in the solution set: 1

Case 3:

Q - - - - - - - - - - - - - - Q Q - - - - - - - - - - - - - - Q - - - - - Q - -	- - Q - - - - - - - - - - - - Q Q - - - - - - - - - - - - - - Q - - - - - Q - -	- - - Q - - - - - - - - - - - Q Q - - - - - - - - - - - - - - Q - - - - - Q - -	

- Q - - - - - - - - Q - - - - - - - - - - - - Q	- Q - - - - - - - - Q - - - - - - - - - - - - Q	- Q - - - - - - - - Q - - - - - - - - - - - - Q	

Solution Not Found

Total number of moves taken: 3 Number of

moves in the solution set: 1

<u>Hill-climbing search with sideways move</u>

a. Run several times, say 100 to 500, and report success and failure rates

We did run the 8 queens problem 100 times, it was success for 93 times and failed for 7 times

b. The average number of steps when it succeeds

Average steps is 20 to success

c. The average number of steps when it fails

Average steps is 62 to failure

d. The search sequences from three random initial configurations

Case 1: (Failure – 63 steps)

Please enter the number of Queens on board: 8

How many times do you want run the code?
Please enter the value: 100

Please enter the maximum number of steps for iteration, limit on sideways moves to avoid infinite loop:

Please enter the value: 20

- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - Q - - - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - Q - - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - Q - - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -
- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - - - Q - - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - - - - Q - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - - - - - - Q

- - - - - Q - - - - - Q - - - - - - - - - - Q -	- - - - - Q - - - - - Q - - - - - - - - - - Q -	- - - - - Q - - - - - Q - - - - - - - - - - Q -	- - - - - Q - - - - - Q - - - - - - - - - - Q -
- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - Q - - - - - - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - Q - - - - - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - Q - - - -
- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - Q - - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - Q - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - Q - -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - - Q
- Q - - - - - - - - - - - - - Q - - - - - - - Q Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - - Q - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - - - Q - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -
- Q - - - - - - - - - - - - - Q - - - - - - - Q - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - - - - - Q - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - - - - - - Q - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q - - - - - - - Q - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -
- Q - - - - - - - - - - - - - Q - - - - - - - Q Q - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - - - Q -	- Q - - - - - - - - - - - - - Q - - - - - - - Q Q - - - - - - - - - Q - - - - - - Q - - - - - - - - - Q - - - - - - - - - - Q -	- - - - Q - - - - - - - - - - Q - - - - Q - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -	- - - - - Q - - - - - - - - - Q - - - - Q - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -

- - - - - - Q - - - - - - - - Q - - - - Q - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - -	- - - - - - - Q - - - - - - - Q - - - - Q - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - -	- Q - - - - - - - - - - - - - Q Q - - - - - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - -	- Q - - - - - - - - - - - - - Q - Q - - - - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - -

- - - - - - Q -	- - - - - - Q -	- - - - - - Q -	- - - - - - Q -
- Q - - - - - -	- Q - - - - - -	- Q - - - - - -	- Q - - - - - -
- - - - - - - Q	- - - - - - - Q	- - - - - - - Q	- - - - - - - Q
- - Q - - - - -	- - - Q - - - -	- - - Q - - - -	- - - - - Q - -
Q - - - - - - -	Q - - - - - - -	Q - - - - - - -	Q - - - - - - -
- - Q - - - - -	- - Q - - - - -	- - Q - - - - -	- - Q - - - - -
- - - - - Q - -	- - - - - Q - -	- - - - - Q - -	- - - - - Q - -
- - - Q - - - -	- - - Q - - - -	- - - Q - - - -	- - - Q - - - -
- - - - - - Q -	- - - - - - Q -	- - - - - - Q -	- - - - - - Q -
- Q - - - - - -	- Q - - - - - -	- Q - - - - - -	- Q - - - - - -
- - - - - - - Q	- - - - - - - Q	- - - - - - - Q	- - - - - - - Q
- - - - - - Q -	- - - - - - - Q	- - - - Q - - -	- - - - Q - - -
Q - - - - - - -	Q - - - - - - -	Q - - - - - - -	Q - - - - - - -
- - Q - - - - -	- - Q - - - - -	- - Q - - - - -	- - Q - - - - -
- - - - - Q - -	- - - - - Q - -	- - - - - Q - -	- - - - - Q - -
- - - Q - - - -	- - - Q - - - -	- - - Q - - - -	- - - Q - - - -
- - - - - - Q -	- - - - - - Q -	Q - - - - - - -	- Q - - - - - -
- Q - - - - - -	- Q - - - - - -	- Q - - - - - -	- Q - - - - - -
- - - - - - - Q	- - - - - - - Q	- - - - - - - Q	- - - - - - - Q
- - - - Q - - -	- - - - Q - - -	- - - - Q - - -	- - - - Q - - -
Q - - - - - - -	Q - - - - - - -	Q - - - - - - -	Q - - - - - - -
- - Q - - - - -	- - Q - - - - -	- - Q - - - - -	- - Q - - - - -
- - - - - Q - -	- - - - - Q - -	- - - - - Q - -	- - - - - Q - -
- - - Q - - - -	- - - Q - - - -	- - - Q - - - -	- - - Q - - - -
- - Q - - - - -	- - - Q - - - -	- - - - Q - - -	- - - - - Q - -
- Q - - - - - -	- Q - - - - - -	- Q - - - - - -	- Q - - - - - -
- - - - - - - Q	- - - - - - - Q	- - - - - - - Q	- - - - - - - Q
- - - - Q - - -	- - - - Q - - -	Q - - - - - - -	- Q - - - - - -
Q - - - - - - -	Q - - - - - - -	Q - - - - - - -	Q - - - - - - -
- - Q - - - - -	- - Q - - - - -	- - Q - - - - -	- - Q - - - - -
- - - - - Q - -	- - - - - Q - -	- - - - - Q - -	- - - - - Q - -
- - - Q - - - -	- - - Q - - - -	- - - Q - - - -	- - - Q - - - -
- - - - - Q - -	- - - - - - - Q	- - - - - - Q -	- - - - - - Q -
- Q - - - - - -	- Q - - - - - -	- Q - - - - - -	- Q - - - - - -
- - - - - - - Q	- - - - - - - Q	- - - - - - - Q	- - - - - - - Q
- - Q - - - - -	- - - Q - - - -	- - - Q - - - -	- - - - - Q - -
Q - - - - - - -	Q - - - - - - -	Q - - - - - - -	Q - - - - - - -
- - Q - - - - -	- - Q - - - - -	- - Q - - - - -	- - Q - - - - -
- - - - - Q - -	- - - - - Q - -	- - - - - Q - -	- - - - - Q - -
- - - Q - - - -	- - - Q - - - -	- - - Q - - - -	- - - Q - - - -
- - - - - - Q -	- - - - - - Q -	- - - - - - Q -	- - - - - - Q -
- Q - - - - - -	- Q - - - - - -	- Q - - - - - -	- Q - - - - - -
- - - - - - - Q	- - - - - - - Q	- - - - - - - Q	- - - - - - Q -
- - - - - Q - -	- - - - - - Q -	- - - - - - - Q	- - - - Q - - -
Q - - - - - - -	Q - - - - - - -	Q - - - - - - -	Q - - - - - - -
- - Q - - - - -	- - Q - - - - -	- - Q - - - - -	- - Q - - - - -
- - - - - Q - -	- - - - - Q - -	- - - - - Q - -	- - - - - Q - -
- - - Q - - - -	- - - Q - - - -	- - - Q - - - -	- - - Q - - - -
- - - - - - Q -	- - - - - - Q -	- - - - - - Q -	- - - - - - Q -

- Q - - - - - - - - - - - - Q - - - - - Q - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - - - - - Q -			

Case 2: (Success- 24 steps)

- - - - - Q - - - - - - - - - Q Q - - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q Q - - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q Q - - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q
- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q - - Q - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q - - - - Q - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q - - - - - Q - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - Q - - - - - - - - - - - - Q
- - - - - Q - - - - - - - - - Q - - - - Q - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - - - Q - - -	- - - - - Q - - - - - - - - - Q - - - - - - - Q - - - Q - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - - - Q - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - Q - - - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - - - Q - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - - - - - Q -
- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - Q - - - - - - - - - - - Q - - - - - Q - - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - - - - Q - - - - -
- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - -

Q - - - - - - - - - - - Q - - - - - - - Q - - - - - Q - - - - -	Q - - - - - - - - - - - - - Q - - - - - Q - - - - - Q - - - - -	Q - - - - - - - - - - - - - - Q - - - - Q - - - - - Q - - - - -	Q - - - - - - - - - - - - - Q - - - - - Q - - - - - Q - - - - -

Case 3: (Success – 24 steps)

Q - - - - - - - Q - - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- - - - Q - - - Q - - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- - - - - - - Q Q - - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	Q - - - - - - - Q - - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -
- - - Q - - - - Q - - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- - - - - - Q - Q - - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- Q - - - - - - - Q - - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -
- Q - - - - - - - - - - Q - - - - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- Q - - - - - - - - - - - - - Q - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - -	- Q - - - - - - - - - - - - - Q - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - Q - - - - - - - - - - - Q - - -	- Q - - - - - - - - - - - - - Q - - - - - Q - - - - - Q - - - - - - Q - - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - -
- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - Q - - - - - - - - - Q - - - - - - - - - - Q	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - Q - - - - - - - - - - - Q - - - - - Q - - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - Q - - - - - - - - - - Q - - - - - Q - - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - Q - - - - - - - - - - Q - - - - - - - - Q - - - - - Q - - - - -

- Q - - - - - - - - - - - - - - Q - - - - - - Q - - - - - Q - - - - - - Q - - - - -	- Q - - - - - - - - - - - - - - Q - - - - - - Q - - - - - - - - - - Q - - Q - - - - -	- Q - - - - - - - - - - Q - - - - - - - - - Q - - - - - - - - - - Q - - Q - - - - -	- - - - - Q - - - - - - - - - Q - Q - - - - - - - - - Q - - - - - - - - - Q - -

Q - - - - - - - - - Q - - - - - - - - - Q - - -	Q - - - - - - - - - - - - - Q - - - - - Q - - -	Q - - - - - - - - - - - - - Q - - - - - Q - - -	- - - - - - - Q - - Q - - - - - Q - - - - - - - - - - - - - Q - - - - - Q - - -

Random-restart hill-climbing search

a. The average number of random restarts used without sideways move

Average number of restarts is 7

b. The average number of steps required without sideways move

Average number of steps is 22

c. The average number of random restarts used with sideways move

Average number of restarts is 2

d. The average number of steps required with sideways move

Average number of steps is 20

PERFORMANCE MEASURE

Hill Climbing Search	Success rate	Failure Rate	Average number of steps when it succeeds	Average number of steps when it fails
	37.8	62.2	1.9	2.3

Hill Climbing Search with sideways move	Success rate	Failure Rate	Average number of steps when it succeeds	Average number of steps when it fails
	100	0	2.85	0

Random Restart hill climbing search	Average number of random restarts required without sideways move	Average number of steps required without sideways move	Average number of random restarts required with sideways move	Average number of steps required with sideways move
	1.44	5.03	0.0	2.04

CONCLUSION

- ➢ In this project we see that when n queens are implemented using hill climbing method, only 14% of the time it solves the problem where as 86% of the time it gets stuck at a local minimum. However, it takes only 4 steps on average when it succeeds and 3 on average when it gets stuck – (for a state spacewith 8^8 =~17 million states)

  Unfortunately, hill climbing often gets stuck for the following reasons:

- Local maxima: This figure is a local maximum (i.e., a local minimum forthe cost h =1); every move of a single queen makes the situation worse.

- Ridges: a ridge is shown in the Figure. Ridges result in a sequenceof local maxima that is very difficult for greedy algorithms to navigate.

- Plateau: a plateau is a flat area of the state-space landscape.

- ➢ It can be a flat local maximum, from which no uphill exit exists, or a shoulder, from which progress is possible. A hill-climbing search might get lost on the plateau.

- ➢ In order to escape the shoulders, we implement n queens using Sideways move. For 8-queens, we allow sideways moves with a limit of 100 which raises the percentage of problem instances solved from 14 to 94%. However, it takes 21 steps for every successful solution and 64 steps for each failure.

- ➢ Both the methods –Hill Climbing and Sideways move Hill climbing are incomplete -- they often fail to find a goal when one exists because they can get stuck on local maxima. Therefore, we implement the n queen problem using Random restart Hill climbing method. The algorithm conducts a series of hill- climbing searches from randomly generated initial states, until a goal is found. It iscomplete with probability approaching 1, because it will eventually generate a goal state as the initial state.