ITCS 6150 – Intelligent Systems
Fall 2021
A Project report
On
"Solving N-queens problem using hill-climbing search and its variants"

Team details

Mohith Raju
mraju2@uncc.edu
801254279

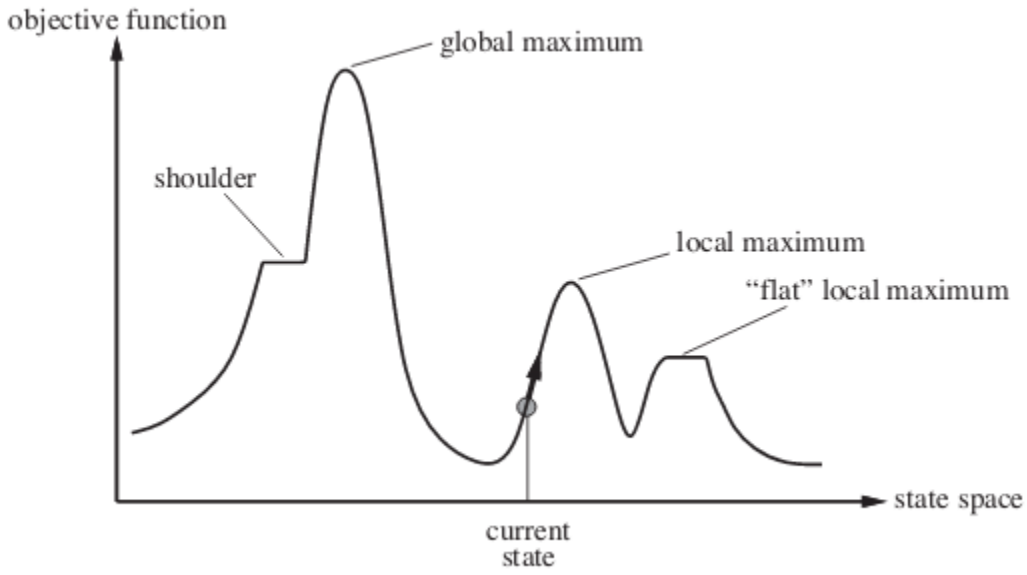
Nehal Kathale
nkathale@uncc.edu
801205316

Renu Karule
rkarule@uncc.edu
801261337

HILL-CLIMBING ALGORITHM:

Hill Climbing is a heuristic search technique that is used to optimize mathematical solutions in Artificial Intelligence. It is a local search algorithm that continuously moves in the direction of increasing values to find the best solution to a problem and terminates if no further improvements can be made to the solution. However, The Hill-climbing algorithm has certain limitations. It can not always find the best solution to a problem if it gets stuck at local maximum. i.e. It cannot find the global maximum.

State space diagram for Hill-Climbing:



Difference regions in the state space:

Local Maximum: Local Maximum is higher than the states adjacent to it but there are states higher than it.

Global Maximum: It is the best possible state of the state space. i.e. It is the best solution for a given state space.
Current state: It is the state in the state space that the algorithm is currently at.
Flat local maximum: It is flat space where all the neighbouring states have the same value.
shoulder: It is a flat-plateau region which has an uphill edge.

There are several ways to implement Hill-Climbing Algorithm,

Hill-Climbing Search:

This is the easiest way to implement Hill-Climbing search. It evaluates only the adjacent states at a time and moves to the first state that optimizes the current cost and sets it as the current state. However, This way of implementation is not always guaranteed to find the global maximum.

Hill-Climbing Search with Sideways moves:

It selects the state with heuristic value the same as the current state as the successor. This way of branching out can lead to a new set of iterations. This is usually done if the state space is stuck at the local maximum.

Random restart without sideways moves:

Random restart hill-climbing performs a hill-climbing algorithm by randomly generating initial states, running each iteration until it gets stuck at the local maximum or makes no progress. Then, each of these iterations are compared and the most optimal is chosen. The drawback to this is deciding on the number of iterations to perform.

Random restart with sideways moves:

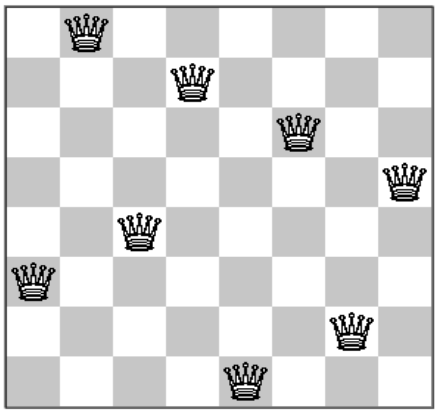
This variation of Random restart hill-climbing allows sideways moves at each iteration. This way of performing Hill-Climbing has a probability of reaching the goal of 1. This variation has the highest success rate compared to all the variations of Hill-Climbing search.

N-Queens Problem:

Problem Statement:

The N-queens Problem consists of a N x N grid and N queens. Each of the Queens can move any number of blocks either in vertical, horizontal or diagonal directions. Our goal is to arrange the Queens in such a way that no two Queens attack each other.

Example Goal state:

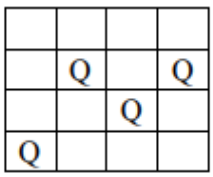


Actions:

The Queen can move any number of blocks in all directions, but only one direction per move. i.e. either vertical, horizontal or diagonal directions at a time.

Heuristic Function:

The Heuristic function for the N-Queens Problem consists of the number of Queens attacking each other. For example,



H=3

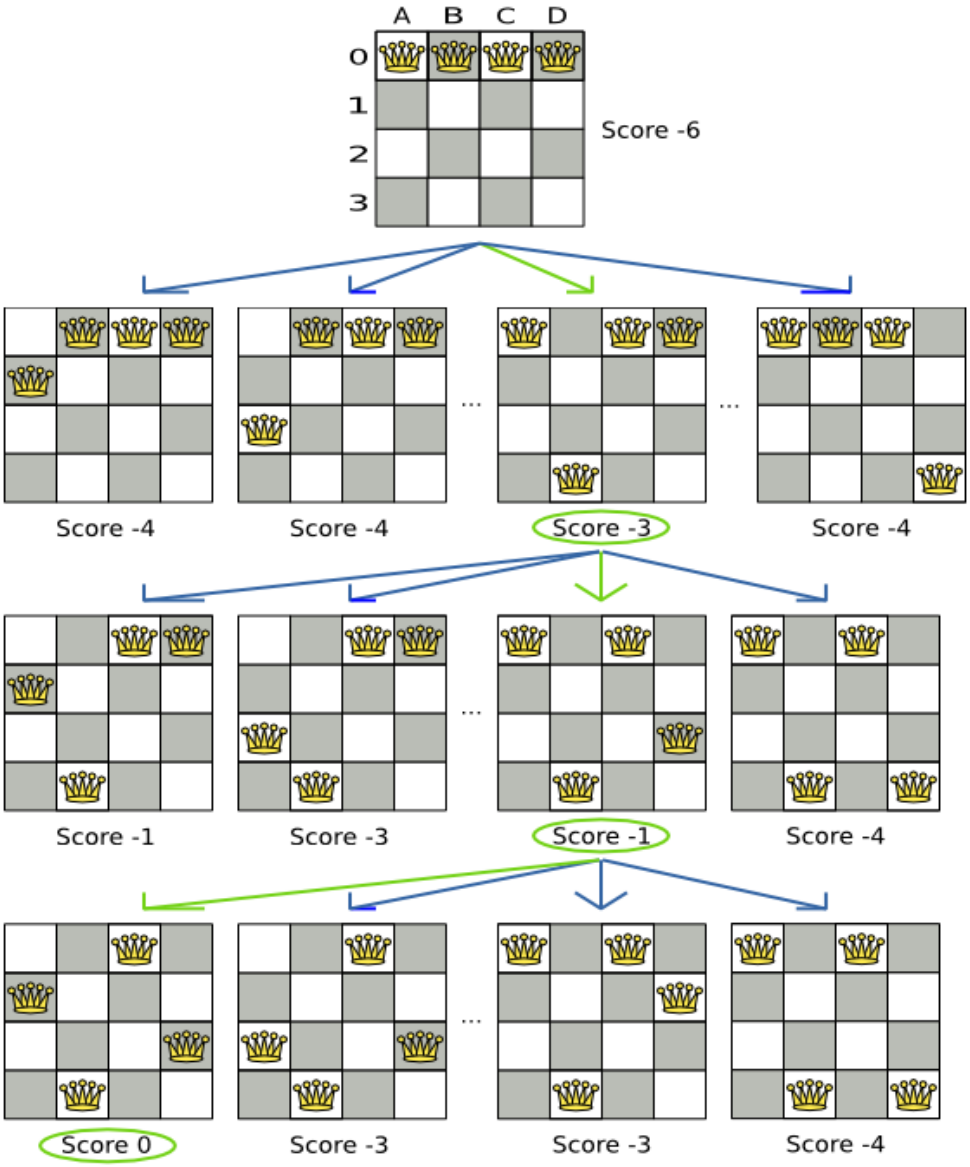
For the above 4 x 4 grid with 4 Queens, The Heuristic value is 3 as it has three pairs of Queens attacking each other.

1. Queen(2,2) -> Queen(3,3)
2. Queen(3, 3) -> Queen(2, 4)
3. Queen(2, 2) -> Queen(2, 4)

Constraints:

1. There can be only one Queen per block.
2. Since we use the Hill-climbing Algorithm, We might run into scenarios where we might have to stop with the local maximum.

Example:



Citation :
https://docs.optaplanner.org/6.3.0.Beta1/optaplanner-docs/html/ch10.html

Program Design and Implementation:

Functions Used:

HillClimbingUtility():

The functions in this class are common to all the four methods of Hill Climbing. It contains all the utility functions required to solve the N-Queens problem.

discoverNextSteps():

This method is used to determine all the successors from the current state. For the N-Queens problem, if we have 4 queens, the number of possible successors will be 12 and for 8 queens we have 56 successors. This method clones the current state while moving to a new state and moves each queen in a time column wise manner and generates successors.

determineHeuristicCostValue():

This method is used to determine the Heuristic value. This value will help us to decide the successor on which we perform the next iteration.

discoverNextBestStep():

This method helps us to find the best successor from the current state. It does this by selecting the successor with least heuristic value. The heuristic values are sorted using priority queues.

passNodeStateData():

This method is a utility function that copies one state to another. i.e. The parent node to all the possible child nodes.

formArbitraryArrangement():

Thai method is a utility function that takes all n queens from the input and places them randomly on the board such that there is only one queen per column. It is used to form the initial state.

displayResult(): This method is used to visualize the solution. It displays a state and the successor that was branched on to each iteration which makes it easier to understand the working of our algorithm.

Hill Climbing Search:

Method:

HillClimbingAlgorithm() - This method with reference to HillClimbingUtility class displays the result of processed nodes with all statistics.

Variables:

numberOfQueens
steps
noOfCorrectSteps
noOfIncorrectSteps
numberOfExecutionSuccess
numberOfExecutionFailure

Function Implementation:

This method implements the simple Hill climbing algorithm. The Utility functions formArbitraryArrangement, determineHeuristicCostValue, discoverNextSteps and discoverNextBestStep are used to come up with the solution. If there are no possible successors to the current state and the heuristic value is not equal to '0', this results in a failure. This method returns the numberOfCorrectSteps, numberOfExecutionSuccess, numberOfIncorrectSteps and the numberOfExecutionFailure function values.

Hill Climbing Search with Sideways Moves:

Method:

HillClimbingWithSidewaysMove() - This method with reference to HillClimbingUtility class displays the result of processed nodes with all statistics.

Variables:

numberOfQueens
steps
noOfCorrectSteps
noOfIncorrectSteps
numberOfExecutionSuccess
numberOfExecutionFailure
maximumSideWalkCount

numberOfSideWalkSteps

Function Implementation:

This method implements the Hill climbing search with sideways moves. The Utility functions formArbitraryArrangement, determineHeuristicCostValue, discoverNextSteps and discoverNextBestStep are used to come up with the solution. If there are no possible successors to the current state and the heuristic value is not equal to '0', then the sideways moves are considered. The best successor is randomly selected until we reach the maxSideWalkCount. This method returns the numberOfCorrectSteps, numberOfExecutionSuccess, numberOfIncorrectSteps and the numberOfExecutionFailure function values

Random Restart Hill Climbing without Sideways moves:

Method:

RandomRestartHillClimbingWithoutSidewaysMove() - This method with reference to HillClimbingUtility class displays the result of processed nodes with all statistics.

Variables:

numberOfQueens
steps
noOfCorrectSteps
noOfIncorrectSteps
numberOfExecutionSuccess
numberOfExecutionFailure
noOfAttemptedRestarts
restartCounter

Function Implementation:

This method implements Hill Climbing with sideway moves using random restart. The Utility functions formArbitraryArrangement, determineHeuristicCostValue, discoverNextSteps and discoverNextBestStep are used to come up with the solution. If there are no possible successors to the current state and the heuristic value is not equal to '0', then the board is randomly regenerated until the solution is found. This method returns the numberOfCorrectSteps, numberOfExecutionSuccess, numberOfIncorrectSteps, numberOfExecutionFailure, numberOfAttemptedRestarts and restartCounter function values.

Random Restart Hill Climbing with Sideways moves:

Method:

RandomRestartHillClimbingWithSidewaysMove() - This method with reference to HillClimbingUtility class displays the result of processed nodes with all statistics.

Variables:

numberOfQueens
steps
numberOfSideWalkSteps
noOfCorrectSteps
noOfIncorrectSteps
numberOfExecutionSuccess
numberOfExecutionFailure
noOfAttemptedRestarts
restartCounter
maximumSideWalkCount

Function Implementation:

This method implements Hill Climbing with sideway moves using random restart. The Utility functions formArbitraryArrangement, determineHeuristicCostValue, discoverNextSteps and discoverNextBestStep are used to come up with the solution. If there are no possible successors to the current state and the heuristic value is not equal to '0', then the sideways moves are considered. The best successor is randomly selected until we reach the maxSideWalkCount. If the solution is not found and the maxSideWalkCount is reached, then the board is randomly regenerated until the solution is found. This method returns the numberOfCorrectSteps, numberOfExecutionSuccess, numberOfIncorrectSteps, numberOfExecutionFailure, numberOfAttemptedRestarts and restartCounter function values.

ExecuteHillClimbing: main() function: This is the main function of our program. It takes as input the number queens to place on the board and validates them. It also calculates the success and failure ratio of each Hill Climbing variant.

Source Code:

https://drive.google.com/drive/folders/1rDIvu5hwVeGlxDqm-Qs5A_XbVYy0Va6d?usp=sharing

Result (Screenshots)

A. Hill-Climbing Search

```
Please provide the number of Queens :
8
Please select Hill Climbing variant:
1. N - Queens Hill Climbing Search
2. N - Queens Hill Climbing search with Sideway Moves
3. N - Queens Random Restart Hill Climbing search without Sideway Moves
4. N - Queens Random Restart Hill Climbing search with Sideway Moves
1
### BEGIN STATE ###
        _       _       _       _       _       _       _       _
        _       _       _       _       _       _       Q       _
        _       _       _       _       _       Q       _       Q
        _       _       _       _       _       _       _       _
        _       _       _       _       _       _       _       _
        Q       _       Q       _       _       _       _       _
        _       Q       _       Q       Q       _       _       _
        _       _       _       _       _       _       _       _
------------------
```

B. Hill-Climbing Search with Sideway moves

```
Please provide the number of Queens :

8
Please select Hill Climbing variant:
1. N - Queens Hill Climbing Search
2. N - Queens Hill Climbing search with Sideway Moves
3. N - Queens Random Restart Hill Climbing search without Sideway Moves
4. N - Queens Random Restart Hill Climbing search with Sideway Moves
2
### BEGIN STATE ###
        _       _       _       _       _       _       _       _
        _       _       _       _       Q       _       _       _
        _       Q       _       Q       _       _       _       _
        _       _       _       _       _       _       _       _
        _       _       Q       _       _       _       _       _
        Q       _       _       _       _       _       _       _
        _       _       _       _       _       Q       Q       Q
        _       _       _       _       _       _       _       _
------------------
```

```
### FINAL STATE ###
        _       _       Q       _       _       _       _       _
        _       _       _       _       _       _       Q       _
        _       Q       _       _       _       _       _       _
        _       _       _       _       _       _       _       Q
        _       _       _       _       _       Q       _       _
        _       _       _       Q       _       _       _       _
        Q       _       _       _       _       _       _       _
        _       _       _       _       Q       _       _       _
------------------

_____________________________________
N - Queens Problem with Hill Climbing
Number of Queens : 8
Algorithm will execute 110 times
Execution Statistics are :
Percentage Success : 94.55%
Percentage Failure : 5.45%
Average Steps for Success : 19.47
Average Steps for Failure : 54.50
Do you want to continue execution?
Y : Yes
N : No
```

C. Random Restart Hill-Climbing search without Sideway Moves

```
Please provide the number of Queens :
8
Please select Hill Climbing variant:
1. N - Queens Hill Climbing Search
2. N - Queens Hill Climbing search with Sideway Moves
3. N - Queens Random Restart Hill Climbing search without Sideway Moves
4. N - Queens Random Restart Hill Climbing search with Sideway Moves

3
### BEGIN STATE ###
        _       _       _       _       Q       _       _       _
        Q       _       _       _       _       _       _       _
        _       _       _       _       _       _       Q       _
        _       _       _       _       _       _       _       Q
        _       _       Q       Q       _       Q       _       _
        _       Q       _       _       _       _       _       _
        _       _       _       _       _       _       _       _
        _       _       _       _       _       _       _       _
------------------
```

```
### FINAL STATE ###
        _       _       _       Q       _       _       _       _
        _       _       _       _       _       Q       _       _
        Q       _       _       _       _       _       _       _
        _       _       _       _       Q       _       _       _
        _       Q       _       _       _       _       _       _
        _       _       _       _       _       _       _       Q
        _       _       Q       _       _       _       _       _
        _       _       _       _       _       _       Q       _
------------------


_____________________________________________
N - Queens Problem with Hill Climbing
Number of Queens : 8
Algorithm will execute 138 times
Execution Statistics are :
Percentage Success : 100.00%
Percentage Failure : 0.00%
Average Steps for Success : 22.36
Average Steps for Failure : 0.00
Do you want to continue execution?
Y : Yes
N : No
```

D. Random Restart Hill-Climbing search with Sideway Moves

```
Please provide the number of Queens :
8
Please select Hill Climbing variant:
1. N - Queens Hill Climbing Search
2. N - Queens Hill Climbing search with Sideway Moves
3. N - Queens Random Restart Hill Climbing search without Sideway Moves
4. N - Queens Random Restart Hill Climbing search with Sideway Moves
4
### BEGIN STATE ###
        _       _       _       Q       Q       _       _       _
        _       Q       _       _       _       _       _       _
        Q       _       _       _       _       Q       Q       _
        _       _       _       _       _       _       _       Q
        _       _       _       _       _       _       _       _
        _       _       _       _       _       _       _       _
        _       _       Q       _       _       _       _       _
        _       _       _       _       _       _       _       _
------------------
```

```
### GOAL STATE ###
        _       _       Q       _       _       _       _       _
        _       _       _       _       Q       _       _       _
        _       Q       _       _       _       _       _       _
        _       _       _       _       _       _       _       Q
        _       _       _       _       _       Q       _       _
        _       _       _       Q       _       _       _       _
        _       _       _       _       _       _       Q       _
        Q       _       _       _       _       _       _       _
------------------
```

```
N - Queens Problem with Hill Climbing
Number of Queens : 8
Algorithm will execute 158 times
Execution Statistics are :
Percentage Success : 100.00%
Percentage Failure : 0.00%
Average Steps for Success : 20.34
Average Steps for Failure : 0.00
Do you want to continue execution?
Y : Yes
N : No
```

```
*** Local Minima State ***
Heuristic Value entered at: 2
        _       _       Q       _       _       Q       _       _
        Q       _       _       _       _       _       _       _
        _       _       _       _       _       _       _       _
        _       _       _       _       Q       _       _       _
        _       _       _       _       _       _       _       Q
        _       _       _       _       _       _       _       _
        _       _       _       Q       _       _       Q       _
        _       Q       _       _       _       _       _       _
------------------
```

```
N - Queens Problem with Hill Climbing
Number of Queens : 8
Algorithm will execute 191 times
Execution Statistics are :
Percentage Success : 12.57%
Percentage Failure : 87.43%
Average Steps for Success : 4.25
Average Steps for Failure : 3.21
Do you want to continue execution?
Y : Yes
N : No
```

```
Please provide the number of Queens :
5
Please select Hill Climbing variant:
1. N - Queens Hill Climbing Search
2. N - Queens Hill Climbing search with Sideway Moves
3. N - Queens Random Restart Hill Climbing search without Sideway Moves
4. N - Queens Random Restart Hill Climbing search with Sideway Moves
1
### BEGIN STATE ###
        Q       _       _       _       _
        _       _       Q       _       _
        _       _       _       _       Q
        _       Q       _       Q       _
        _       _       _       _       _
------------------
```

```
### FINAL STATE ###
        Q       _       _       _       _
        _       _       _       Q       _
        _       Q       _       _       _
        _       _       _       _       Q
        _       _       Q       _       _
------------------
```

```
N - Queens Problem with Hill Climbing
Number of Queens : 5
Algorithm will execute 102 times
Execution Statistics are :
Percentage Success : 74.51%
Percentage Failure : 25.49%
Average Steps for Success : 2.45
Average Steps for Failure : 1.69
Do you want to continue execution?
Y : Yes
N : No
```