

Group Members:

Robert Riso (netid: rar267)

Yousuf khan (netid myk33)

## AI Project 1 Report

### **Task 1:**

First task is to formulate this problem as an AI problem.

Environment: The game board.

State Space: The set of paths to the goal node.

Actions: Any of the legal 2-directional movements from the current space.

Perception/Observations: The distance away from edge of board and the legal distance that can be moved from the space.

Transition Function: The successor state should be the state that allows the agent to have the shortest total path to the goal state.

Evaluation Metric: The evaluation metric is based on the number of total moves that an agent would take to get to the goal state with the shortest path. A larger evaluation function indicates a more difficult game and a smaller evaluation function indicates an easier game. A negative value of the evaluation function means that the game was not solvable and the absolute value of this negative number is the total number of spaces that the agent was unable to reach while trying to find the goal state.

### **Task 2:**

Examples of the GUI for  $n=5,7,9,11$ :

$n=5$

2	1	2	3	2
4	1	1	2	3
3	3	2	3	3
3	1	1	3	1
2	3	4	1	0

n=7

4	5	5	2	6	5	3
3	1	3	5	3	1	6
5	2	4	1	1	4	1
3	5	2	2	2	2	2
5	2	3	1	1	1	6
2	2	1	5	5	3	1
6	1	3	1	6	5	0

n=9

2	4	6	6	3	6	6	8	6
6	6	3	2	6	4	7	3	3
3	7	4	4	3	4	1	2	8
7	4	4	4	5	3	3	1	5
7	1	5	2	4	1	5	4	6
8	2	2	3	3	1	3	4	1
5	5	4	6	2	3	2	4	5
8	7	7	1	7	7	1	5	3
8	2	6	2	3	7	4	6	0

n=11

6	3	1	9	7	3	8	10	3	5	4
8	5	5	9	2	8	8	2	6	6	3
2	7	4	3	8	4	3	8	8	7	3
3	4	2	7	3	6	6	7	2	3	8
7	7	8	2	1	5	6	1	3	5	8
9	9	8	2	3	1	6	3	5	9	6
2	3	3	3	1	6	2	4	3	6	2
9	6	6	3	1	2	3	6	3	2	6
1	9	5	2	3	5	2	7	8	5	6
5	4	7	8	5	3	4	3	3	4	5
6	8	2	4	9	9	4	7	4	10	0

### Task 3:

Examples of 2 GUI representations after BFS for n=5,7,9,11 (one solvable, one not solvable):

n=5

Solvable:

3 Shortest Path=0	1 Shortest Path=5	1 Shortest Path=6	4 Shortest Path=1	3 Shortest Path=X
4 Shortest Path=5	3 Shortest Path=6	2 Shortest Path=7	3 Shortest Path=4	3 Shortest Path=6
2 Shortest Path=X	2 Shortest Path=4	1 Shortest Path=6	1 Shortest Path=5	3 Shortest Path=3
4 Shortest Path=1	1 Shortest Path=4	3 Shortest Path=5	2 Shortest Path=3	1 Shortest Path=2
1 Shortest Path=6	1 Shortest Path=5	3 Shortest Path=6	4 Shortest Path=2	0 Shortest Path=3 Eval Function=3

Not Solvable:

1 Shortest Path=0	3 Shortest Path=1	4 Shortest Path=4	1 Shortest Path=3	1 Shortest Path=2
3 Shortest Path=1	3 Shortest Path=X	3 Shortest Path=X	3 Shortest Path=2	4 Shortest Path=3
1 Shortest Path=5	2 Shortest Path=6	2 Shortest Path=X	2 Shortest Path=7	1 Shortest Path=X
1 Shortest Path=4	3 Shortest Path=2	3 Shortest Path=X	1 Shortest Path=X	4 Shortest Path=3
3 Shortest Path=2	4 Shortest Path=7	4 Shortest Path=5	3 Shortest Path=3	0 Shortest Path=X Eval Function=-7

n=7

Solvable:

2 Shortest Path=0	1 Shortest Path=2	1 Shortest Path=1	3 Shortest Path=2	4 Shortest Path=4	6 Shortest Path=5	2 Shortest Path=3
3 Shortest Path=X	4 Shortest Path=3	4 Shortest Path=2	3 Shortest Path=X	2 Shortest Path=5	1 Shortest Path=4	1 Shortest Path=3
2 Shortest Path=1	2 Shortest Path=6	4 Shortest Path=2	3 Shortest Path=5	4 Shortest Path=X	2 Shortest Path=5	6 Shortest Path=3
6 Shortest Path=4	4 Shortest Path=5	4 Shortest Path=4	3 Shortest Path=3	2 Shortest Path=6	4 Shortest Path=4	3 Shortest Path=4
5 Shortest Path=2	2 Shortest Path=5	4 Shortest Path=6	1 Shortest Path=5	3 Shortest Path=4	1 Shortest Path=3	3 Shortest Path=4
5 Shortest Path=5	1 Shortest Path=4	5 Shortest Path=3	5 Shortest Path=6	5 Shortest Path=7	4 Shortest Path=4	6 Shortest Path=X
1 Shortest Path=X	4 Shortest Path=5	3 Shortest Path=3	4 Shortest Path=4	3 Shortest Path=X	3 Shortest Path=4	0 Shortest Path=5 Eval Function=5

Not Solvable:

5 Shortest Path=0	5 Shortest Path=X	3 Shortest Path=X	5 Shortest Path=X	4 Shortest Path=X	6 Shortest Path=1	5 Shortest Path=X
2 Shortest Path=X	1 Shortest Path=X	5 Shortest Path=X	3 Shortest Path=X	2 Shortest Path=X	5 Shortest Path=X	1 Shortest Path=X
6 Shortest Path=X	5 Shortest Path=X	1 Shortest Path=X	3 Shortest Path=X	2 Shortest Path=X	2 Shortest Path=X	2 Shortest Path=X
5 Shortest Path=X	4 Shortest Path=X	4 Shortest Path=X	3 Shortest Path=X	4 Shortest Path=X	5 Shortest Path=X	2 Shortest Path=X
6 Shortest Path=X	2 Shortest Path=X	4 Shortest Path=X	3 Shortest Path=X	3 Shortest Path=X	1 Shortest Path=X	5 Shortest Path=X
6 Shortest Path=1	4 Shortest Path=X	3 Shortest Path=X	2 Shortest Path=X	3 Shortest Path=X	2 Shortest Path=X	6 Shortest Path=2
3 Shortest Path=X	2 Shortest Path=X	1 Shortest Path=X	3 Shortest Path=X	4 Shortest Path=X	6 Shortest Path=2	0 Shortest Path=X Eval Function=-44

n=9

Solvable:

8 Shortest Path=0	2 Shortest Path=5	8 Shortest Path=X	5 Shortest Path=2	6 Shortest Path=8	5 Shortest Path=6	5 Shortest Path=X	2 Shortest Path=5	5 Shortest Path=1
1 Shortest Path=8	7 Shortest Path=7	3 Shortest Path=X	6 Shortest Path=4	3 Shortest Path=8	7 Shortest Path=X	6 Shortest Path=7	4 Shortest Path=7	4 Shortest Path=8
3 Shortest Path=7	1 Shortest Path=6	4 Shortest Path=7	5 Shortest Path=8	2 Shortest Path=7	4 Shortest Path=8	6 Shortest Path=8	3 Shortest Path=6	4 Shortest Path=9
1 Shortest Path=6	3 Shortest Path=7	1 Shortest Path=X	3 Shortest Path=5	1 Shortest Path=7	4 Shortest Path=7	2 Shortest Path=6	2 Shortest Path=6	5 Shortest Path=4
5 Shortest Path=7	5 Shortest Path=X	6 Shortest Path=6	1 Shortest Path=5	2 Shortest Path=6	5 Shortest Path=8	2 Shortest Path=7	4 Shortest Path=4	1 Shortest Path=3
6 Shortest Path=4	5 Shortest Path=8	1 Shortest Path=7	4 Shortest Path=3	2 Shortest Path=6	3 Shortest Path=7	2 Shortest Path=5	7 Shortest Path=3	1 Shortest Path=2
3 Shortest Path=X	6 Shortest Path=4	1 Shortest Path=7	5 Shortest Path=6	1 Shortest Path=7	3 Shortest Path=8	2 Shortest Path=6	1 Shortest Path=5	7 Shortest Path=3
8 Shortest Path=9	7 Shortest Path=7	2 Shortest Path=8	7 Shortest Path=5	6 Shortest Path=7	1 Shortest Path=8	2 Shortest Path=6	6 Shortest Path=6	1 Shortest Path=7
8 Shortest Path=1	5 Shortest Path=8	2 Shortest Path=6	3 Shortest Path=X	2 Shortest Path=7	1 Shortest Path=8	7 Shortest Path=7	5 Shortest Path=5	0 Shortest Path=2 Eval Function=2

Not Solvable:

4 Shortest Path=0	3 Shortest Path=6	5 Shortest Path=4	8 Shortest Path=7	8 Shortest Path=1	5 Shortest Path=7	4 Shortest Path=3	8 Shortest Path=5	4 Shortest Path=X
8 Shortest Path=7	4 Shortest Path=X	4 Shortest Path=4	1 Shortest Path=6	7 Shortest Path=7	7 Shortest Path=6	4 Shortest Path=5	4 Shortest Path=X	8 Shortest Path=6
1 Shortest Path=6	4 Shortest Path=5	1 Shortest Path=6	4 Shortest Path=6	2 Shortest Path=5	1 Shortest Path=5	6 Shortest Path=5	7 Shortest Path=7	6 Shortest Path=X
7 Shortest Path=7	1 Shortest Path=7	2 Shortest Path=7	1 Shortest Path=6	4 Shortest Path=7	1 Shortest Path=6	5 Shortest Path=7	5 Shortest Path=6	8 Shortest Path=8
6 Shortest Path=1	1 Shortest Path=6	3 Shortest Path=3	3 Shortest Path=5	2 Shortest Path=4	2 Shortest Path=4	4 Shortest Path=2	6 Shortest Path=5	4 Shortest Path=X
2 Shortest Path=7	5 Shortest Path=6	4 Shortest Path=5	2 Shortest Path=5	4 Shortest Path=X	5 Shortest Path=6	3 Shortest Path=6	1 Shortest Path=X	7 Shortest Path=X
3 Shortest Path=6	2 Shortest Path=6	6 Shortest Path=4	3 Shortest Path=6	2 Shortest Path=3	3 Shortest Path=5	4 Shortest Path=4	1 Shortest Path=X	5 Shortest Path=5
3 Shortest Path=8	7 Shortest Path=5	1 Shortest Path=4	5 Shortest Path=5	3 Shortest Path=8	4 Shortest Path=X	2 Shortest Path=X	7 Shortest Path=9	6 Shortest Path=6
2 Shortest Path=5	6 Shortest Path=4	1 Shortest Path=3	3 Shortest Path=4	2 Shortest Path=2	4 Shortest Path=7	4 Shortest Path=3	5 Shortest Path=5	0 Shortest Path=X Eval Function=-12

n=11

Solvable:

3 Shortest Path=0	2 Shortest Path=6	1 Shortest Path=5	9 Shortest Path=1	3 Shortest Path=12	7 Shortest Path=3	8 Shortest Path=X	10 Shortest Path=13	7 Shortest Path=6	6 Shortest Path=7	5 Shortest Path=7
6 Shortest Path=10	2 Shortest Path=8	6 Shortest Path=6	7 Shortest Path=7	4 Shortest Path=9	6 Shortest Path=4	9 Shortest Path=11	5 Shortest Path=X	3 Shortest Path=4	8 Shortest Path=X	5 Shortest Path=5
6 Shortest Path=8	1 Shortest Path=7	7 Shortest Path=4	2 Shortest Path=X	5 Shortest Path=6	6 Shortest Path=X	5 Shortest Path=6	5 Shortest Path=X	2 Shortest Path=5	5 Shortest Path=5	9 Shortest Path=6
5 Shortest Path=1	7 Shortest Path=4	1 Shortest Path=3	7 Shortest Path=4	5 Shortest Path=7	3 Shortest Path=2	4 Shortest Path=4	7 Shortest Path=6	2 Shortest Path=3	8 Shortest Path=5	2 Shortest Path=4
10 Shortest Path=X	5 Shortest Path=6	4 Shortest Path=4	1 Shortest Path=9	2 Shortest Path=10	1 Shortest Path=X	5 Shortest Path=5	2 Shortest Path=X	2 Shortest Path=5	8 Shortest Path=7	4 Shortest Path=6
5 Shortest Path=4	3 Shortest Path=6	7 Shortest Path=7	4 Shortest Path=6	2 Shortest Path=6	4 Shortest Path=5	2 Shortest Path=5	7 Shortest Path=6	3 Shortest Path=4	5 Shortest Path=6	6 Shortest Path=5
3 Shortest Path=4	2 Shortest Path=5	7 Shortest Path=X	3 Shortest Path=3	6 Shortest Path=11	5 Shortest Path=3	1 Shortest Path=4	1 Shortest Path=5	8 Shortest Path=6	6 Shortest Path=8	9 Shortest Path=4
2 Shortest Path=3	3 Shortest Path=6	8 Shortest Path=4	2 Shortest Path=5	2 Shortest Path=7	2 Shortest Path=4	1 Shortest Path=5	2 Shortest Path=5	8 Shortest Path=7	3 Shortest Path=6	5 Shortest Path=5
1 Shortest Path=2	5 Shortest Path=3	6 Shortest Path=5	4 Shortest Path=8	7 Shortest Path=8	5 Shortest Path=5	1 Shortest Path=4	5 Shortest Path=5	2 Shortest Path=5	4 Shortest Path=8	9 Shortest Path=6
3 Shortest Path=3	2 Shortest Path=7	7 Shortest Path=5	3 Shortest Path=2	4 Shortest Path=8	6 Shortest Path=5	3 Shortest Path=3	6 Shortest Path=6	5 Shortest Path=9	6 Shortest Path=4	2 Shortest Path=X
4 Shortest Path=5	4 Shortest Path=5	8 Shortest Path=7	3 Shortest Path=5	8 Shortest Path=6	3 Shortest Path=6	4 Shortest Path=6	7 Shortest Path=7	9 Shortest Path=6	2 Shortest Path=7	0 Shortest Path=7 Eval Function=7

Not Solvable:

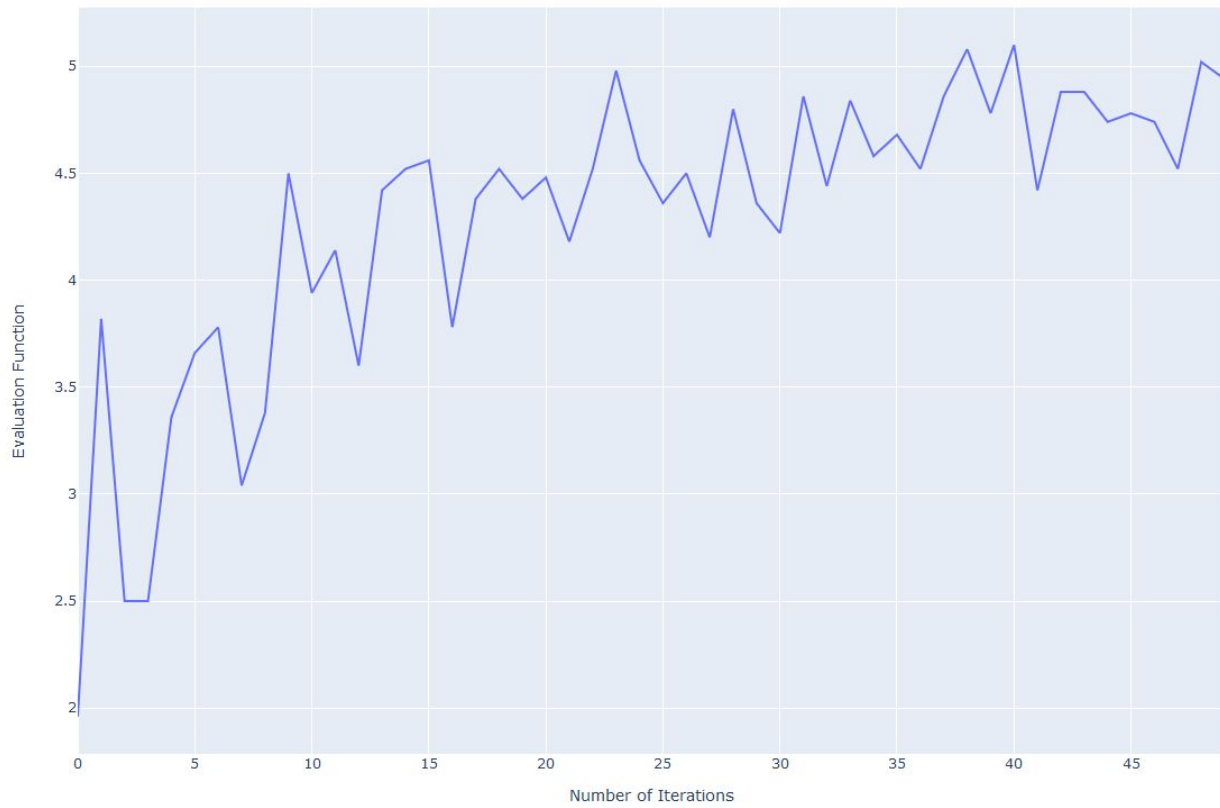
10 Shortest Path=0	5 Shortest Path=5	7 Shortest Path=2	9 Shortest Path=6	3 Shortest Path=4	8 Shortest Path=6	1 Shortest Path=6	7 Shortest Path=4	7 Shortest Path=5	2 Shortest Path=3	8 Shortest Path=1
6 Shortest Path=7	9 Shortest Path=5	4 Shortest Path=5	9 Shortest Path=7	5 Shortest Path=8	3 Shortest Path=X	5 Shortest Path=4	6 Shortest Path=5	7 Shortest Path=5	3 Shortest Path=9	2 Shortest Path=4
6 Shortest Path=7	9 Shortest Path=6	8 Shortest Path=9	8 Shortest Path=8	4 Shortest Path=7	7 Shortest Path=5	4 Shortest Path=8	3 Shortest Path=6	8 Shortest Path=8	4 Shortest Path=4	7 Shortest Path=7
1 Shortest Path=X	3 Shortest Path=6	8 Shortest Path=5	6 Shortest Path=6	7 Shortest Path=5	3 Shortest Path=5	4 Shortest Path=7	4 Shortest Path=7	3 Shortest Path=4	6 Shortest Path=5	9 Shortest Path=5
2 Shortest Path=6	2 Shortest Path=X	6 Shortest Path=6	3 Shortest Path=11	1 Shortest Path=X	1 Shortest Path=X	6 Shortest Path=12	1 Shortest Path=X	8 Shortest Path=5	6 Shortest Path=10	8 Shortest Path=X
4 Shortest Path=6	3 Shortest Path=5	2 Shortest Path=5	5 Shortest Path=5	5 Shortest Path=6	5 Shortest Path=X	1 Shortest Path=X	6 Shortest Path=7	7 Shortest Path=6	9 Shortest Path=7	9 Shortest Path=4
6 Shortest Path=2	9 Shortest Path=4	5 Shortest Path=4	5 Shortest Path=6	6 Shortest Path=3	3 Shortest Path=6	5 Shortest Path=3	4 Shortest Path=5	8 Shortest Path=5	9 Shortest Path=4	10 Shortest Path=4
6 Shortest Path=6	9 Shortest Path=4	1 Shortest Path=3	2 Shortest Path=4	6 Shortest Path=7	4 Shortest Path=5	6 Shortest Path=7	7 Shortest Path=5	3 Shortest Path=4	5 Shortest Path=6	2 Shortest Path=3
10 Shortest Path=5	9 Shortest Path=6	3 Shortest Path=4	3 Shortest Path=X	7 Shortest Path=7	6 Shortest Path=5	7 Shortest Path=14	7 Shortest Path=4	4 Shortest Path=6	2 Shortest Path=3	1 Shortest Path=2
8 Shortest Path=6	8 Shortest Path=X	6 Shortest Path=4	3 Shortest Path=5	7 Shortest Path=6	7 Shortest Path=6	6 Shortest Path=6	7 Shortest Path=7	4 Shortest Path=5	2 Shortest Path=6	8 Shortest Path=3
4 Shortest Path=1	10 Shortest Path=4	6 Shortest Path=5	5 Shortest Path=6	4 Shortest Path=2	6 Shortest Path=X	2 Shortest Path=13	7 Shortest Path=6	7 Shortest Path=3	7 Shortest Path=4	0 Shortest Path=X Eval Function=-13

## Task 4:

Hill Climbing Plot for n=5,7,9,11 (using up to 50 iterations and averaged over 50 runs):

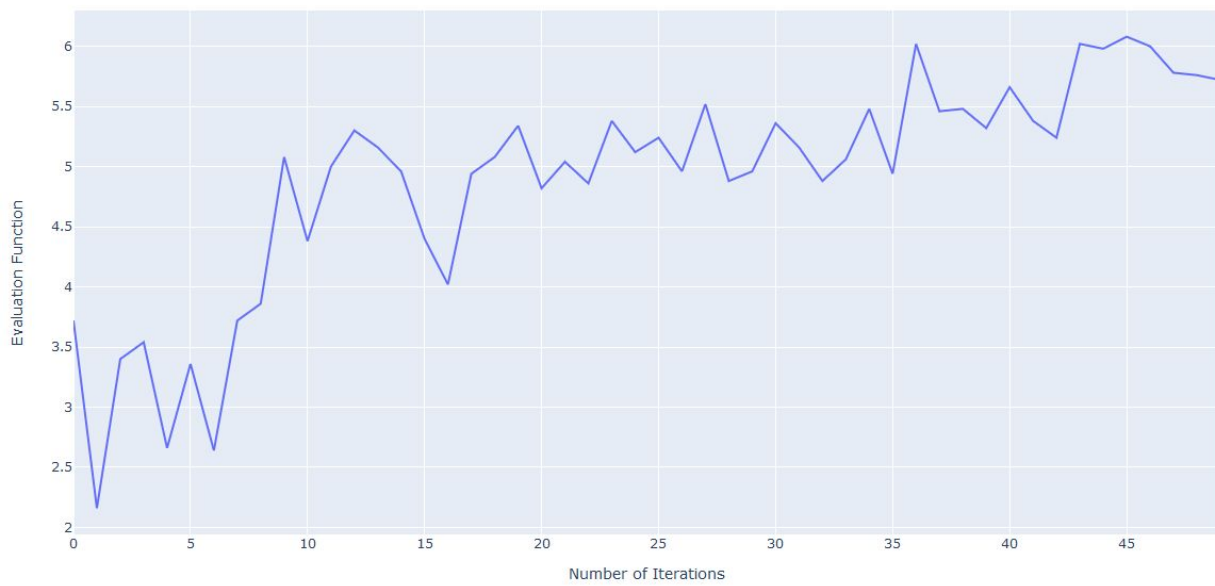
n=5

Hill Climbing



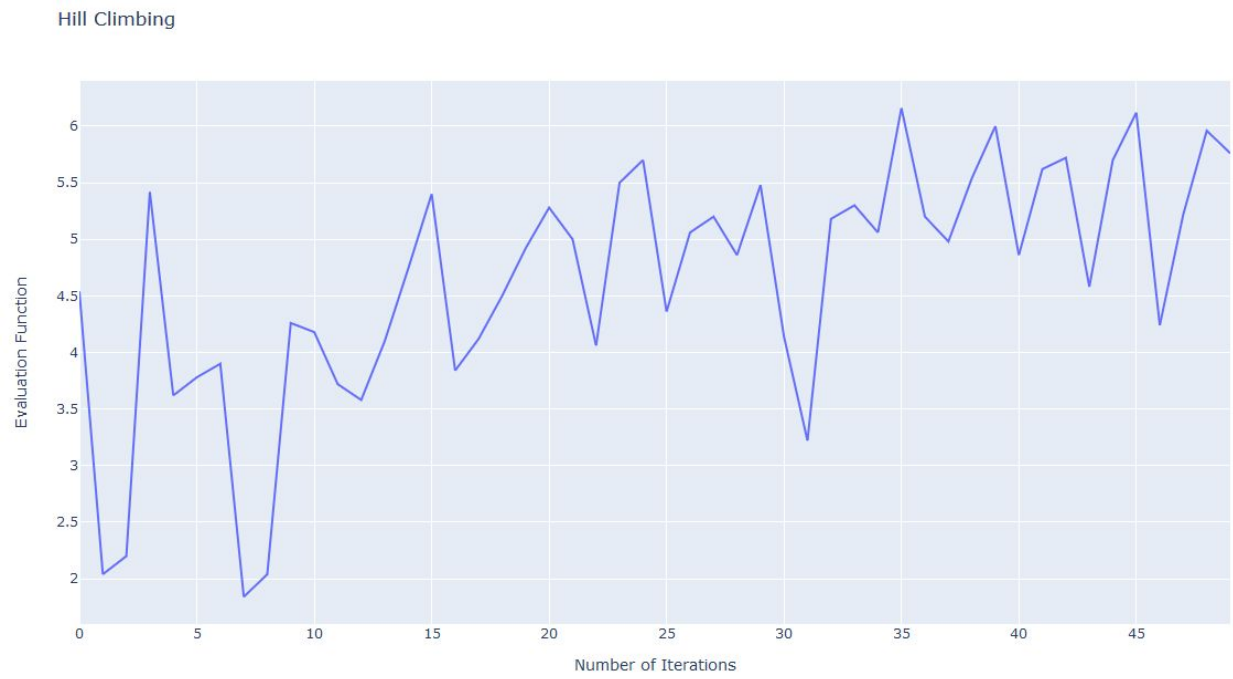
n=7

Hill Climbing

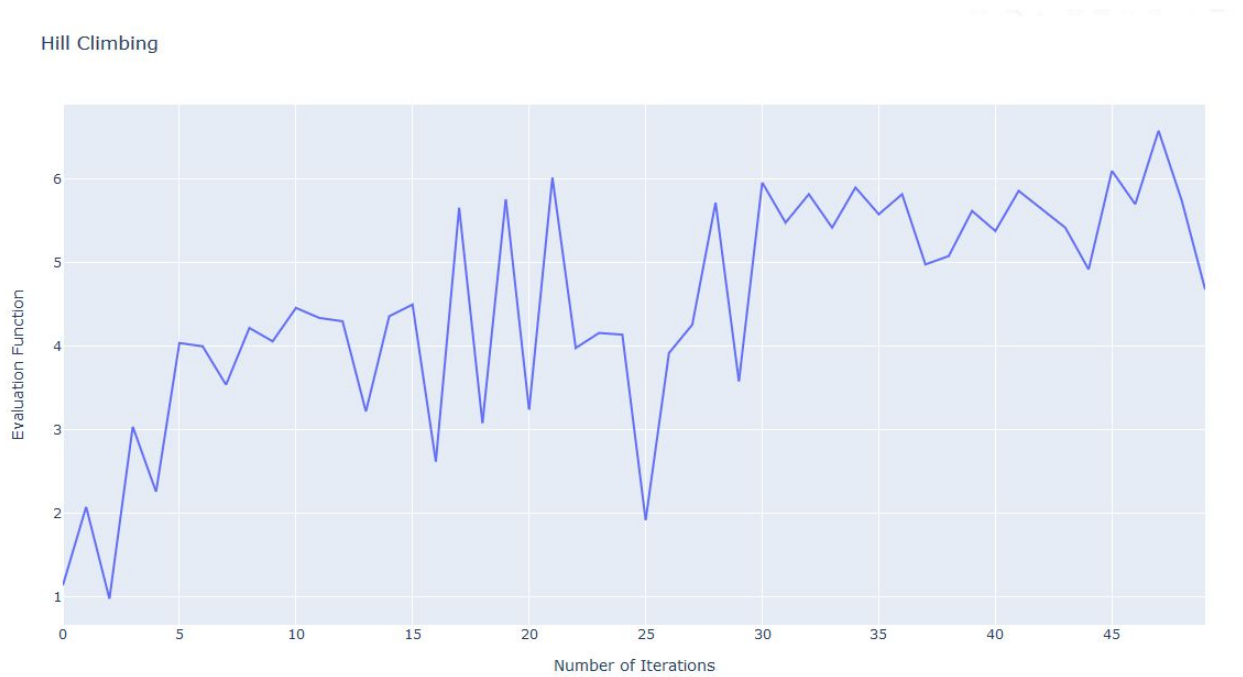




n=9



n=11



Visualize optimized puzzle configuration (50 iterations):

n=5

3 Shortest Path=0	4 Shortest Path=2	4 Shortest Path=3	2 Shortest Path=1	1 Shortest Path=10
1 Shortest Path=2	1 Shortest Path=3	2 Shortest Path=4	2 Shortest Path=3	4 Shortest Path=5
2 Shortest Path=3	2 Shortest Path=4	1 Shortest Path=3	1 Shortest Path=2	4 Shortest Path=3
2 Shortest Path=1	1 Shortest Path=4	3 Shortest Path=2	2 Shortest Path=3	1 Shortest Path=4
4 Shortest Path=4	2 Shortest Path=3	1 Shortest Path=4	2 Shortest Path=4	0 Shortest Path=5 Eval Function=5 Time=0.07045009999999996

n=7

1 Shortest Path=0	1 Shortest Path=1	1 Shortest Path=2	1 Shortest Path=3	5 Shortest Path=4	1 Shortest Path=4	3 Shortest Path=5
2 Shortest Path=1	4 Shortest Path=2	2 Shortest Path=2	5 Shortest Path=4	3 Shortest Path=3	1 Shortest Path=3	3 Shortest Path=4
3 Shortest Path=5	1 Shortest Path=6	4 Shortest Path=4	1 Shortest Path=6	1 Shortest Path=7	5 Shortest Path=4	5 Shortest Path=5
2 Shortest Path=2	2 Shortest Path=4	1 Shortest Path=3	2 Shortest Path=4	1 Shortest Path=8	4 Shortest Path=5	1 Shortest Path=6
3 Shortest Path=4	1 Shortest Path=4	4 Shortest Path=4	2 Shortest Path=5	4 Shortest Path=4	1 Shortest Path=6	2 Shortest Path=5
1 Shortest Path=3	1 Shortest Path=3	3 Shortest Path=4	2 Shortest Path=5	1 Shortest Path=5	4 Shortest Path=5	6 Shortest Path=X
3 Shortest Path=4	2 Shortest Path=4	3 Shortest Path=5	5 Shortest Path=5	1 Shortest Path=6	1 Shortest Path=6	0 Shortest Path=6 Eval Function=6 Time=0.15512499999999996

n=9

3 Shortest Path=0	3 Shortest Path=2	3 Shortest Path=7	2 Shortest Path=1	3 Shortest Path=3	5 Shortest Path=2	3 Shortest Path=8	6 Shortest Path=4	2 Shortest Path=7
1 Shortest Path=9	5 Shortest Path=1 0	6 Shortest Path=X	5 Shortest Path=6	4 Shortest Path=8	2 Shortest Path=5	7 Shortest Path=4	1 Shortest Path=5	1 Shortest Path=6
6 Shortest Path=3	3 Shortest Path=X	6 Shortest Path=7	3 Shortest Path=2	4 Shortest Path=5	1 Shortest Path=4	1 Shortest Path=3	1 Shortest Path=4	4 Shortest Path=5
7 Shortest Path=1	5 Shortest Path=3	6 Shortest Path=8	5 Shortest Path=3	3 Shortest Path=4	4 Shortest Path=5	1 Shortest Path=4	7 Shortest Path=2	1 Shortest Path=9
7 Shortest Path=5	7 Shortest Path=6	2 Shortest Path=6	2 Shortest Path=5	3 Shortest Path=7	1 Shortest Path=6	5 Shortest Path=5	1 Shortest Path=6	5 Shortest Path=7
1 Shortest Path=4	1 Shortest Path=5	5 Shortest Path=6	3 Shortest Path=3	1 Shortest Path=9	5 Shortest Path=3	1 Shortest Path=4	4 Shortest Path=5	2 Shortest Path=7
3 Shortest Path=5	2 Shortest Path=6	1 Shortest Path=6	2 Shortest Path=6	6 Shortest Path=5	4 Shortest Path=7	4 Shortest Path=5	4 Shortest Path=5	5 Shortest Path=6
4 Shortest Path=8	1 Shortest Path=7	5 Shortest Path=7	1 Shortest Path=9	1 Shortest Path=8	4 Shortest Path=6	4 Shortest Path=7	6 Shortest Path=8	2 Shortest Path=6
3 Shortest Path=4	8 Shortest Path=4	6 Shortest Path=6	4 Shortest Path=4	4 Shortest Path=6	7 Shortest Path=3	4 Shortest Path=5	3 Shortest Path=5	0 Shortest Path=7 Eval Function=7 Time=0.29112709

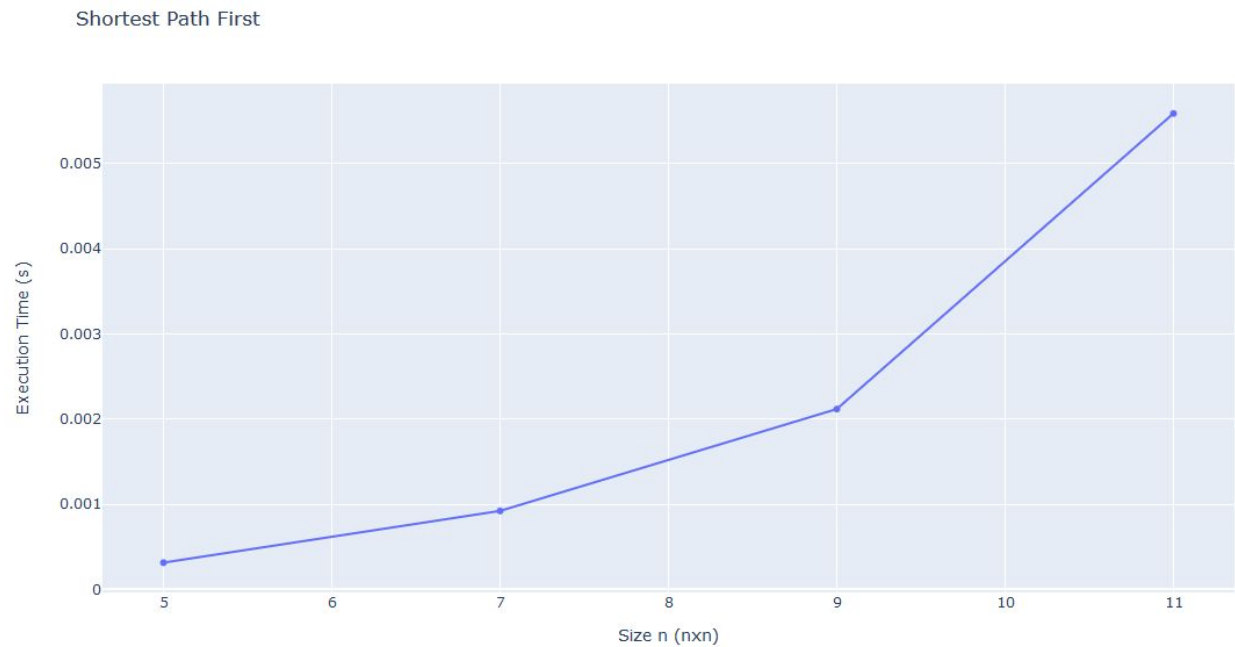
n=11

9 Shortest Path=0	10 Shortest Path=3	1 Shortest Path=2	7 Shortest Path=3	10 Shortest Path=5	1 Shortest Path=X	3 Shortest Path=5	10 Shortest Path=7	7 Shortest Path=5	7 Shortest Path=1	9 Shortest Path=4
6 Shortest Path=5	1 Shortest Path=4	1 Shortest Path=3	3 Shortest Path=4	7 Shortest Path=6	8 Shortest Path=5	6 Shortest Path=5	5 Shortest Path=6	1 Shortest Path=8	2 Shortest Path=9	4 Shortest Path=X
1 Shortest Path=7	3 Shortest Path=5	8 Shortest Path=4	4 Shortest Path=8	3 Shortest Path=6	6 Shortest Path=6	1 Shortest Path=8	2 Shortest Path=7	3 Shortest Path=9	9 Shortest Path=8	5 Shortest Path=5
2 Shortest Path=8	6 Shortest Path=5	2 Shortest Path=7	4 Shortest Path=7	3 Shortest Path=4	2 Shortest Path=6	4 Shortest Path=9	2 Shortest Path=5	1 Shortest Path=X	5 Shortest Path=3	4 Shortest Path=1 0
1 Shortest Path=8	1 Shortest Path=7	3 Shortest Path=5	4 Shortest Path=5	2 Shortest Path=5	3 Shortest Path=4	5 Shortest Path=6	5 Shortest Path=6	4 Shortest Path=5	3 Shortest Path=5	2 Shortest Path=X
6 Shortest Path=9	5 Shortest Path=6	4 Shortest Path=8	2 Shortest Path=6	6 Shortest Path=6	1 Shortest Path=7	3 Shortest Path=7	2 Shortest Path=6	8 Shortest Path=1 0	6 Shortest Path=7	9 Shortest Path=6
6 Shortest Path=7	6 Shortest Path=7	3 Shortest Path=6	1 Shortest Path=5	1 Shortest Path=5	2 Shortest Path=6	1 Shortest Path=7	3 Shortest Path=7	8 Shortest Path=6	1 Shortest Path=X	10 Shortest Path=6
5 Shortest Path=6	7 Shortest Path=5	7 Shortest Path=4	7 Shortest Path=4	5 Shortest Path=6	3 Shortest Path=3	1 Shortest Path=6	4 Shortest Path=7	7 Shortest Path=4	4 Shortest Path=2	2 Shortest Path=5
1 Shortest Path=8	5 Shortest Path=5	6 Shortest Path=5	1 Shortest Path=4	7 Shortest Path=5	4 Shortest Path=7	4 Shortest Path=6	8 Shortest Path=6	1 Shortest Path=6	8 Shortest Path=4	10 Shortest Path=7
9 Shortest Path=1	8 Shortest Path=5	1 Shortest Path=4	1 Shortest Path=3	5 Shortest Path=4	2 Shortest Path=5	8 Shortest Path=7	7 Shortest Path=6	8 Shortest Path=7	6 Shortest Path=2	3 Shortest Path=5
9 Shortest Path=6	9 Shortest Path=4	10 Shortest Path=5	4 Shortest Path=4	6 Shortest Path=5	1 Shortest Path=4	3 Shortest Path=5	2 Shortest Path=5	1 Shortest Path=X	5 Shortest Path=6	0 Shortest Path=5 Eval Function=5 Time=0.57503670

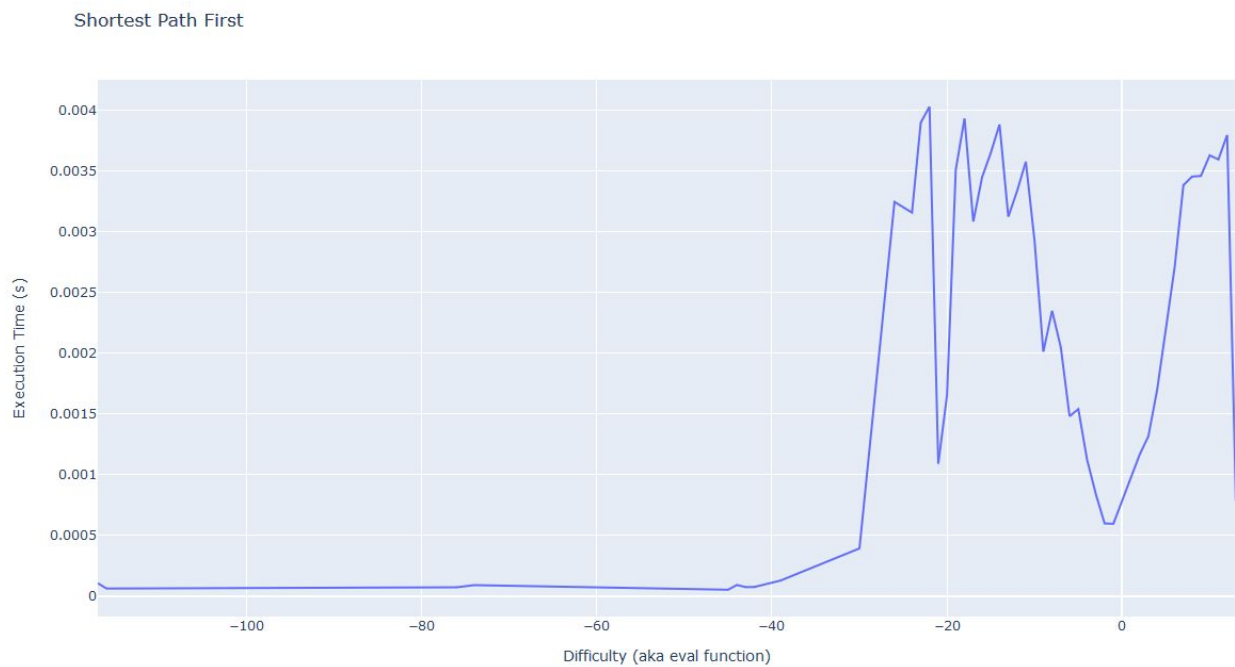
Task 5:

Both shortest path (aka BFS) plots explained:

Size vs Execution Time (averaged over 50 runs per size):



Difficulty vs Execution Time (averaged and used a total of 1000 games):

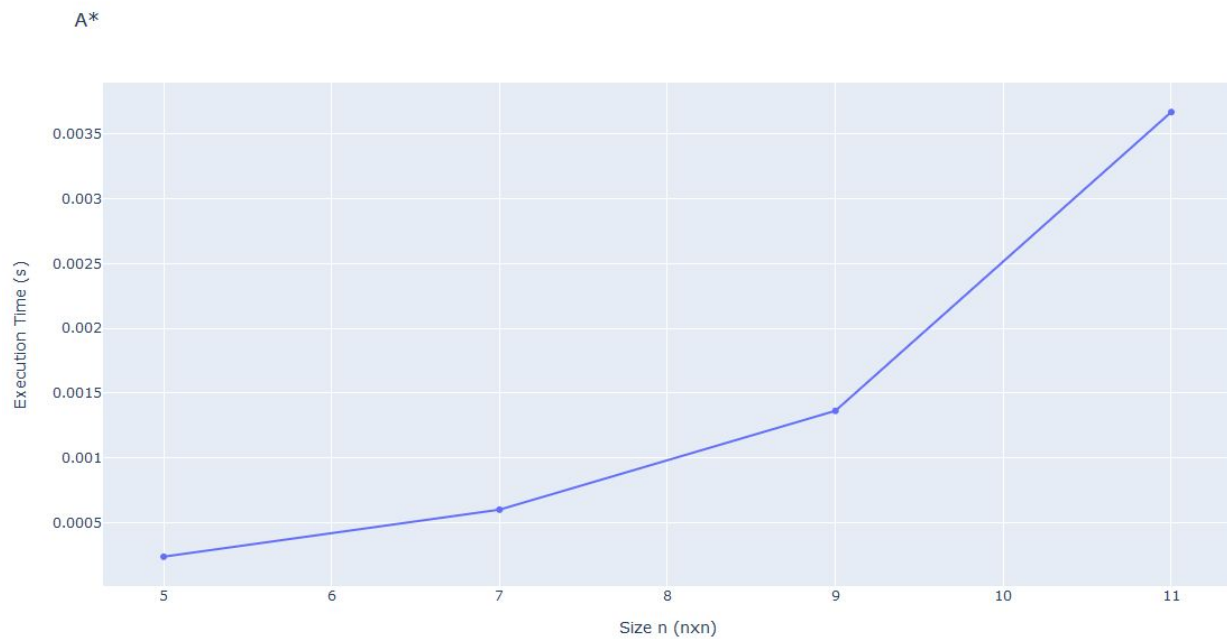


From the plots, it seems that execution time increases as the size of the problem increases. However, there isn't as clear of a relationship between the execution time and the difficulty of

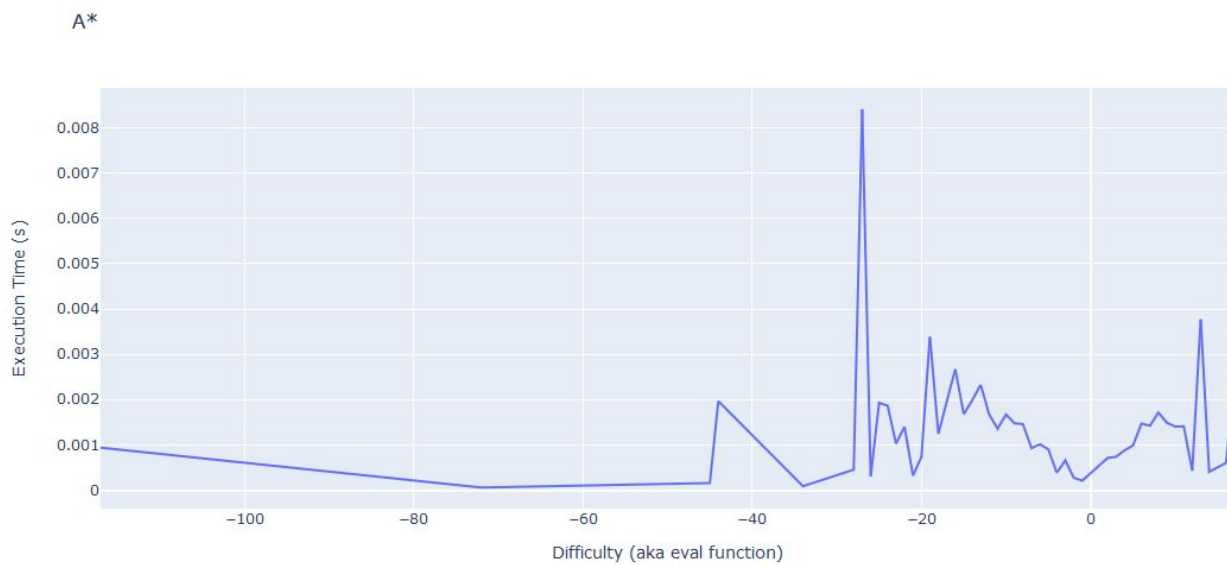
the problem. From a difficulty of 0 execution time seems to increase but then it decreases sharply at a certain point.

### Task 6:

Size vs Execution Time (averaged over 50 runs per size):



Difficulty vs Execution Time (averaged and used a total of 1000 games):



Explain Heuristic:

My heuristic for A\* works by looking at the children nodes of the current node in the path. If the children have a value that can reach another node that is located either in the last column (to the right) or the last row (on the bottom) then it is given a heuristic value of two. This is because after counting the movement of 1 to the child node from the parent, in the best case scenario there would be only one more move until the finish. Therefore, there is at least two more moves that can occur. The heuristic value is three if the child node cannot reach one of either the last column or the last row. This is because now there is at least three more moves in the best case scenario to get to the finish. If the parent node itself is on either the last column or the last row then the heuristic is 1 because there is at least 1 more move that can be made to get to the finish. And, lastly, if the parent node is the goal node the heuristic is 0 because there are at least 0 more moves until the goal. This heuristic is therefore also admissible because it is always less than or equal to the actual value of moves that still have to occur to get to the goal.

Compare computation time to SPF (aka BFS):

Compared to SPF, A\* is shown to take less time to compute the final for each of the sizes of the puzzles of 5,7,9, and 11. Additionally, it is more difficult to see any meaningful relationship for the difficulty vs execution time plots. However, A\* seems to have the majority of its values below the .003 time value, whereas SPF does not. So, it seems to be faster at finding solutions with higher difficulty as well.

Explain what problems A\* can solve that SPF cannot:

A\* would be able to solve problems of very large size potentially much better than SPF would be able to do. This is because A\* does not need to go through every node in the graph, since it constantly searches for the best route to go at any point in time, it can find the solution before traversing the entire graph.

### **Task 7:**

You are requested to describe in detail the population-based local search approach that you have developed in order to search this puzzle generation challenge. Any parameters that you need to define in order to execute the algorithm (e.g., size of population, probability of selection) need to be explicitly mentioned and evaluated in terms of their impacts. Justify the choices you made in terms of the algorithm you decided to implement.

The algorithm we implemented works by taking an array of games of n size which is defined by the user. The geneticAlgo then creates a Z amount of games when its finished. The games are then sorted by highest evaluation after being analyzed by the breadth first algorithm. The top two games with the highest eval values are chosen by the selection process and then forced to create an offspring if the offspring has a negative eval it is deleted and a new offspring is created. If the offspring has less or the same eval value then it is forced to have an offspring with the child before it and then the child is mutated. Once the mutated child is finally greater in

evaluation value of both of its parents it is considered to be evolved and the new game board is returned.

How we compare the hill climbing algorithm with the genetic algorithm is by using a compare function that I had created. The compare function we made calls the hill climbing function and the genetic algorithm function and has them optimize a game with the same n values for the dimensions of the board. We then calculate if both the algorithms were within 5% of each other if they are we then show both the optimized boards and compare their eval values. In my observation we saw that the genetic algorithm would consistently produce harder games if given the same time to optimize the game boards. However on average genetic algorithms are consistently slower than hill climbing at optimizing the boards.

Genetic algorithm

### Genetic algo on top Hill Climbing on bottom 5x5

4 SPath=0	1 SPath=3	1 SPath=4	3 SPath=2	1 SPath=1
2 SPath=5	3 SPath=4	3 SPath=3	3 SPath=4	2 SPath=2
2 SPath=7	1 SPath=X	2 SPath=8	3 SPath=6	2 SPath=9
3 SPath=2	2 SPath=4	2 SPath=5	2 SPath=3	3 SPath=3
1 SPath=1	4 SPath=2	1 SPath=4	2 SPath=5	0 SPath=10 Eval =10 #of iter=50
1 SPath=0	3 SPath=1	2 SPath=3	3 SPath=4	2 SPath=2
2 SPath=1	1 SPath=3	1 SPath=2	1 SPath=3	3 SPath=4
4 SPath=5	3 SPath=4	1 SPath=3	2 SPath=4	3 SPath=3
3 SPath=2	3 SPath=2	3 SPath=4	1 SPath=3	2 SPath=3
4 SPath=6	4 SPath=5	4 SPath=5	2 SPath=4	0 SPath=5 Eval=5 #of iter=0

```
0.11500009537
0.1051245
found equal time
genetic algo eval
10
hill climbing eval
5
for a 5x5 boards
```

### Genetic algo on top Hill Climbing on bottom 7x7



74 tk

3 SPath=0	6 SPath=8	2 SPath=4	4 SPath=1	2 SPath=3	2 SPath=6	3 SPath=4
1 SPath=5	4 SPath=6	3 SPath=4	5 SPath=5	1 SPath=4	2 SPath=5	5 SPath=5
4 SPath=6	5 SPath=5	1 SPath=5	2 SPath=4	1 SPath=4	5 SPath=5	6 SPath=6
4 SPath=1	5 SPath=3	3 SPath=4	1 SPath=3	3 SPath=2	2 SPath=5	6 SPath=4
3 SPath=4	1 SPath=X	3 SPath=3	1 SPath=2	4 SPath=3	3 SPath=4	1 SPath=9
4 SPath=6	5 SPath=7	5 SPath=4	1 SPath=3	1 SPath=4	5 SPath=5	1 SPath=8
1 SPath=7	3 SPath=8	6 SPath=5	4 SPath=4	5 SPath=3	6 SPath=X	0 SPath=9 Eval =9
3 SPath=0	6 SPath=6	2 SPath=2	1 SPath=1	2 SPath=2	3 SPath=4	6 SPath=3
4 SPath=3	4 SPath=4	4 SPath=7	5 SPath=2	1 SPath=4	1 SPath=5	4 SPath=6
4 SPath=5	4 SPath=5	3 SPath=3	1 SPath=5	2 SPath=3	2 SPath=4	4 SPath=4
3 SPath=1	2 SPath=3	4 SPath=7	2 SPath=2	4 SPath=6	5 SPath=3	5 SPath=8
6 SPath=6	3 SPath=5	2 SPath=5	1 SPath=4	2 SPath=4	5 SPath=5	2 SPath=5
4 SPath=4	1 SPath=4	3 SPath=4	1 SPath=3	5 SPath=4	4 SPath=5	2 SPath=7
5 SPath=2	6 SPath=5	3 SPath=6	3 SPath=3	3 SPath=5	6 SPath=3	0 SPath=4 Eval=4

## Genetic algo on top Hill Climbing on bottom 9x9

2 SPath=0	6 SPath=7	7 SPath=1	2 SPath=6	1 SPath=5	2 SPath=6	7 SPath=X	2 SPath=7	7 SPath=X
7 SPath=6	2 SPath=5	3 SPath=7	2 SPath=4	2 SPath=6	2 SPath=5	6 SPath=5	6 SPath=5	2 SPath=4
3 SPath=1	6 SPath=6	1 SPath=X	5 SPath=2	6 SPath=6	2 SPath=6	1 SPath=5	1 SPath=4	1 SPath=3
4 SPath=3	1 SPath=5	6 SPath=3	2 SPath=5	5 SPath=4	1 SPath=6	6 SPath=4	2 SPath=5	7 SPath=4
8 SPath=X	4 SPath=6	5 SPath=8	1 SPath=8	3 SPath=9	2 SPath=7	5 SPath=X	6 SPath=8	3 SPath=11
2 SPath=2	6 SPath=X	2 SPath=3	4 SPath=6	5 SPath=4	2 SPath=12	2 SPath=X	3 SPath=6	7 SPath=X
4 SPath=X	2 SPath=8	6 SPath=X	4 SPath=9	6 SPath=11	2 SPath=8	3 SPath=10	1 SPath=9	2 SPath=10
8 SPath=3	2 SPath=X	4 SPath=2	6 SPath=3	1 SPath=10	2 SPath=11	4 SPath=3	4 SPath=6	5 SPath=4
6 SPath=7	2 SPath=7	4 SPath=X	3 SPath=8	6 SPath=5	7 SPath=9	7 SPath=8	7 SPath=7	0 SPath=11
5 SPath=0	2 SPath=5	6 SPath=2	6 SPath=3	8 SPath=4	3 SPath=1	1 SPath=5	3 SPath=6	5 SPath=2
5 SPath=3	3 SPath=5	1 SPath=4	4 SPath=5	1 SPath=6	4 SPath=4	6 SPath=6	5 SPath=6	5 SPath=5
8 SPath=5	4 SPath=6	2 SPath=4	2 SPath=5	5 SPath=5	6 SPath=3	4 SPath=7	4 SPath=X	2 SPath=6



```
0.613999843597  
0.6154793  
found equal time  
genetic algo eval  
11  
hill climbing eval  
6  
for a 9x9 boards
```

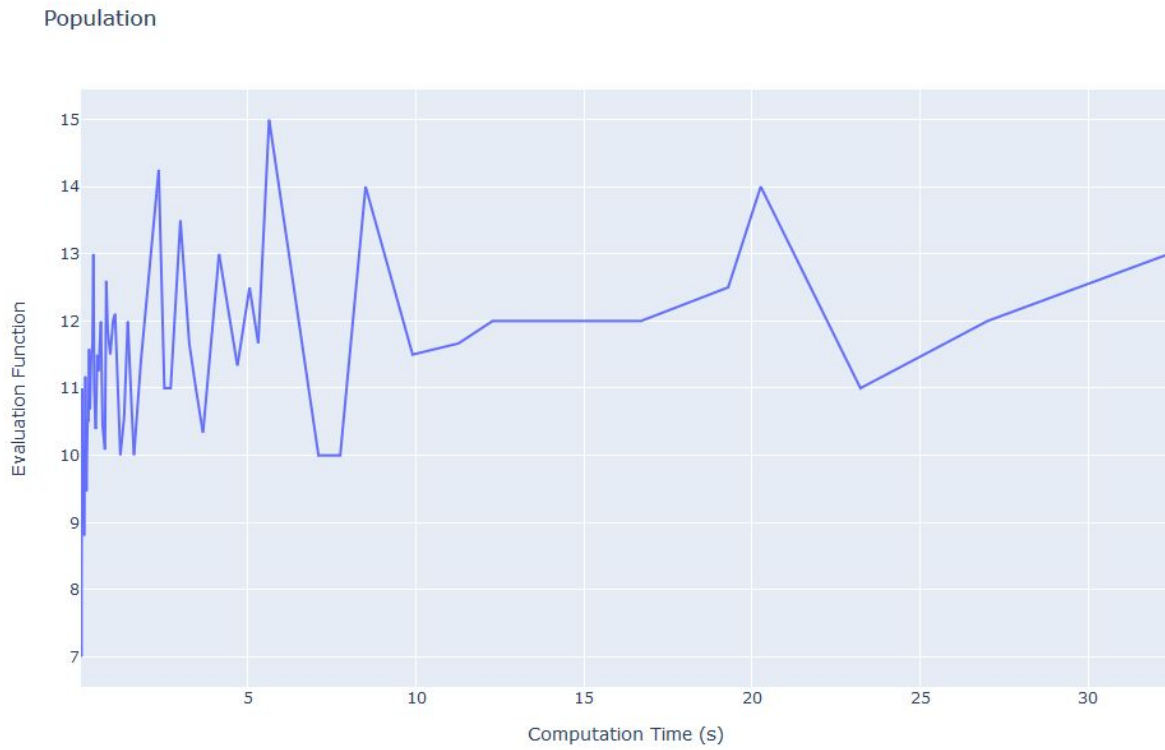
**Genetic algo on top Hill Climbing on bottom 11x11**

76 tk

5	10	3	2	9	6	8	4	1	4	7
SPath=0	SPath=8	SPath=X	SPath=7	SPath=7	SPath=1	SPath=9	SPath=6	SPath=9	SPath=10	SPath=X
3	6	6	9	3	5	6	3	6	6	10
SPath=4	SPath=9	SPath=6	SPath=5	SPath=11	SPath=3	SPath=10	SPath=10	SPath=7	SPath=10	SPath=4
7	1	6	1	5	6	6	4	2	4	5
SPath=10	SPath=X	SPath=7	SPath=6	SPath=7	SPath=9	SPath=9	SPath=5	SPath=8	SPath=8	SPath=9
7	5	2	1	3	5	3	5	2	2	8
SPath=6	SPath=7	SPath=5	SPath=7	SPath=6	SPath=11	SPath=8	SPath=7	SPath=8	SPath=9	SPath=9
10	4	1	7	6	2	4	2	7	1	9
SPath=5	SPath=7	SPath=6	SPath=7	SPath=8	SPath=8	SPath=X	SPath=7	SPath=9	SPath=8	SPath=6
7	9	6	2	2	5	2	7	2	4	7
SPath=1	SPath=10	SPath=6	SPath=9	SPath=9	SPath=10	SPath=8	SPath=2	SPath=7	SPath=9	SPath=8
7	6	3	4	2	5	4	4	2	1	8
SPath=3	SPath=X	SPath=4	SPath=5	SPath=7	SPath=2	SPath=8	SPath=4	SPath=10	SPath=9	SPath=3
2	8	1	6	5	3	6	1	8	9	9
SPath=9	SPath=8	SPath=7	SPath=8	SPath=8	SPath=X	SPath=9	SPath=X	SPath=8	SPath=9	SPath=10
4	7	4	5	8	5	2	4	6	5	3
SPath=11	SPath=8	SPath=8	SPath=9	SPath=8	SPath=10	SPath=9	SPath=8	SPath=9	SPath=X	SPath=11
2	5	5	6	4	4	8	5	1	6	9
SPath=9	SPath=X	SPath=5	SPath=11	SPath=8	SPath=X	SPath=X	SPath=6	SPath=9	SPath=10	SPath=X
6	7	10	10	8	5	8	10	10	10	0
SPath=7	SPath=9	SPath=X	SPath=6	SPath=9	SPath=11	SPath=8	SPath=5	SPath=10	SPath=X	SPath=12
4	1	3	5	6	9	8	1	5	4	2
SPath=0	SPath=4	SPath=9	SPath=4	SPath=1	SPath=5	SPath=6	SPath=5	SPath=3	SPath=4	SPath=2
3	4	6	4	6	1	9	1	1	2	5
SPath=4	SPath=8	SPath=5	SPath=5	SPath=7	SPath=6	SPath=5	SPath=4	SPath=4	SPath=5	SPath=5
7	8	1	3	8	4	1	2	5	2	8
SPath=6	SPath=5	SPath=4	SPath=5	SPath=3	SPath=5	SPath=6	SPath=4	SPath=5	SPath=3	SPath=3
3	9	4	4	6	5	1	4	7	1	5
SPath=6	SPath=7	SPath=5	SPath=7	SPath=7	SPath=5	SPath=6	SPath=7	SPath=6	SPath=6	SPath=6
9	5	4	2	4	1	4	3	5	2	3
SPath=1	SPath=7	SPath=8	SPath=6	SPath=4	SPath=7	SPath=7	SPath=3	SPath=5	SPath=2	SPath=4
9	6	3	1	4	2	3	2	5	2	4
SPath=5	SPath=9	SPath=6	SPath=5	SPath=6	SPath=6	SPath=8	SPath=7	SPath=4	SPath=6	SPath=X
5	1	5	4	3	3	4	4	5	7	3
SPath=3	SPath=9	SPath=4	SPath=4	SPath=2	SPath=4	SPath=X	SPath=5	SPath=3	SPath=3	SPath=6
4	1	2	6	2	2	3	7	6	4	8
SPath=5	SPath=10	SPath=6	SPath=7	SPath=6	SPath=7	SPath=7	SPath=4	SPath=6	SPath=7	SPath=5
3	6	2	1	3	4	4	5	5	2	9
SPath=8	SPath=6	SPath=7	SPath=6	SPath=5	SPath=6	SPath=7	SPath=6	SPath=5	SPath=7	SPath=7
3	7	7	5	9	4	9	3	3	8	8
SPath=7	SPath=6	SPath=7	SPath=7	SPath=7	SPath=5	SPath=6	SPath=8	SPath=6	SPath=6	SPath=7
5	1	9	10	4	1	1	8	2	6	0
SPath=4	SPath=6	SPath=7	SPath=5	SPath=3	SPath=5	SPath=5	SPath=6	SPath=4	SPath=8	SPath=4

```
1.42999982834
1.3624695
found equal time
genetic algo eval
12
hill climbing eval
4
for 11x11 boards
```

Show plot:

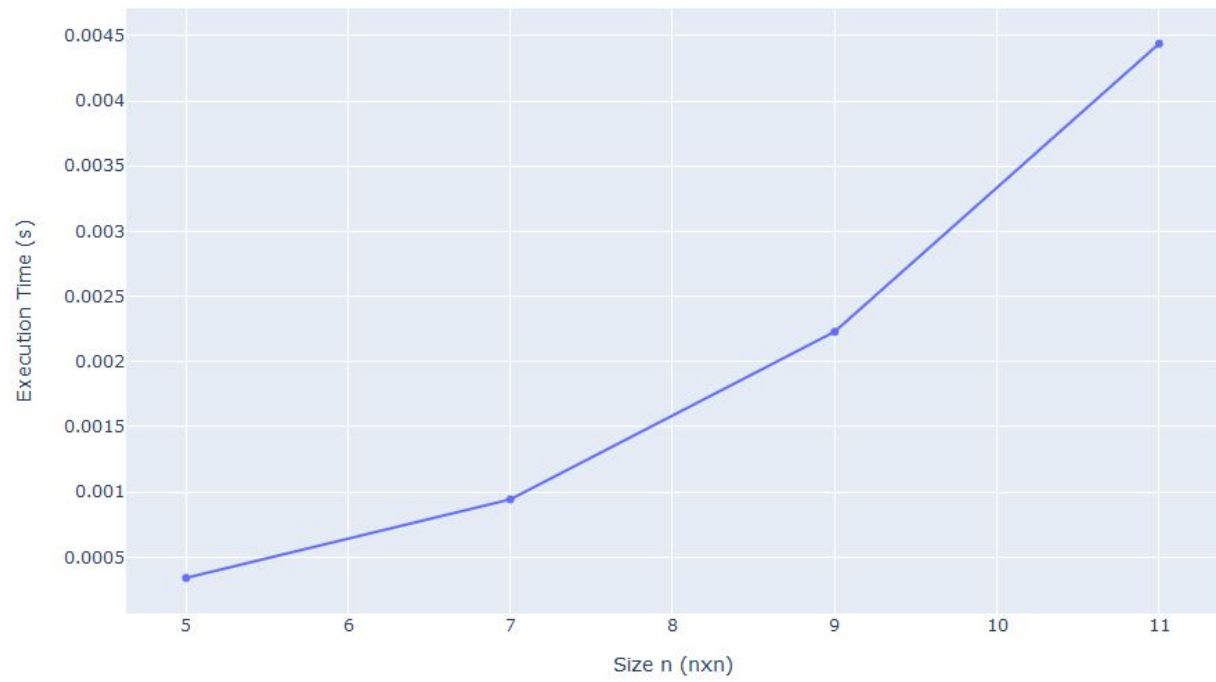


**Task 8:**

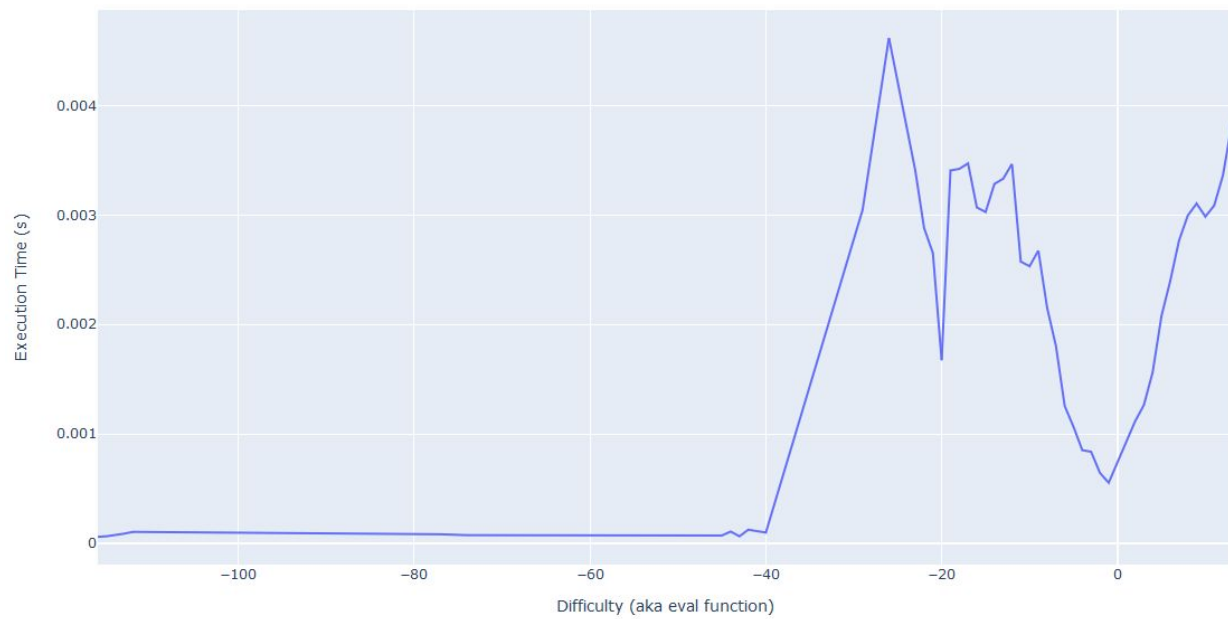
Show 2 plots for each of the methods:

SPF:

Shortest Path First

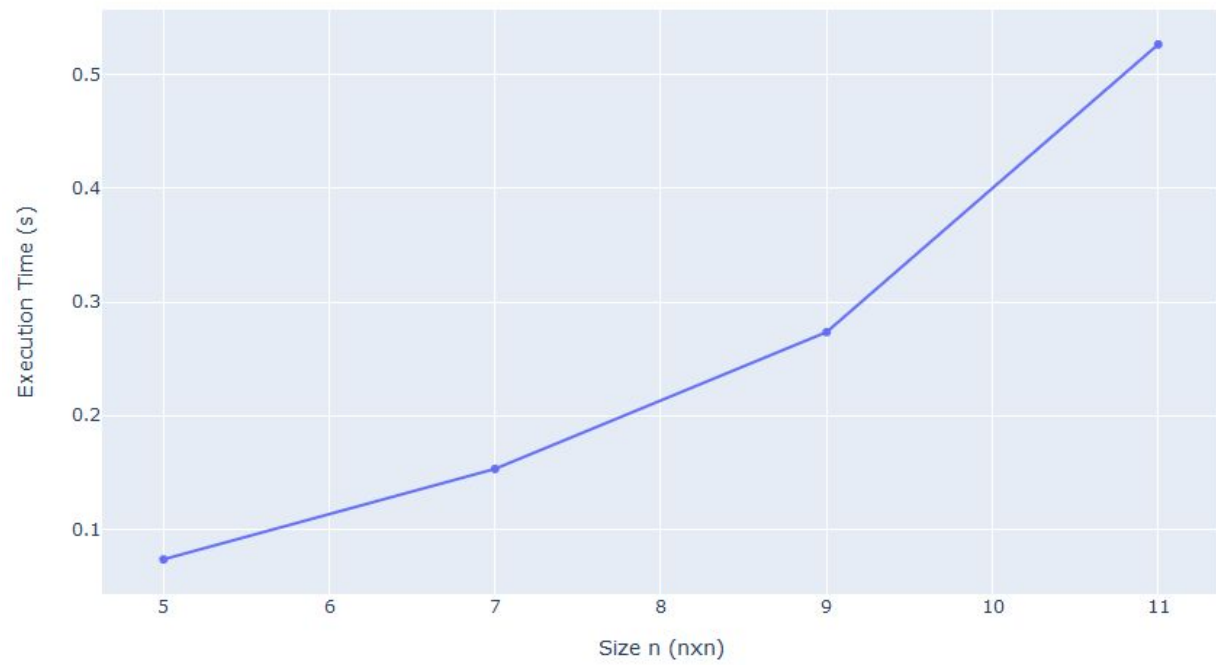


Shortest Path First

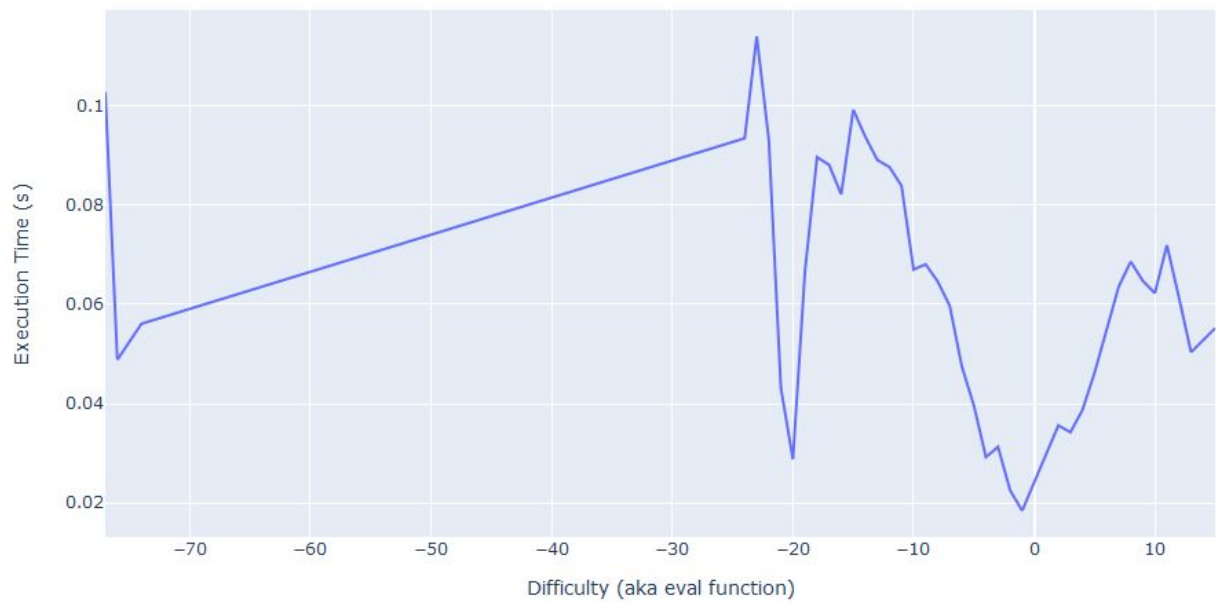


Hill Climbing:

## Hill Climbing

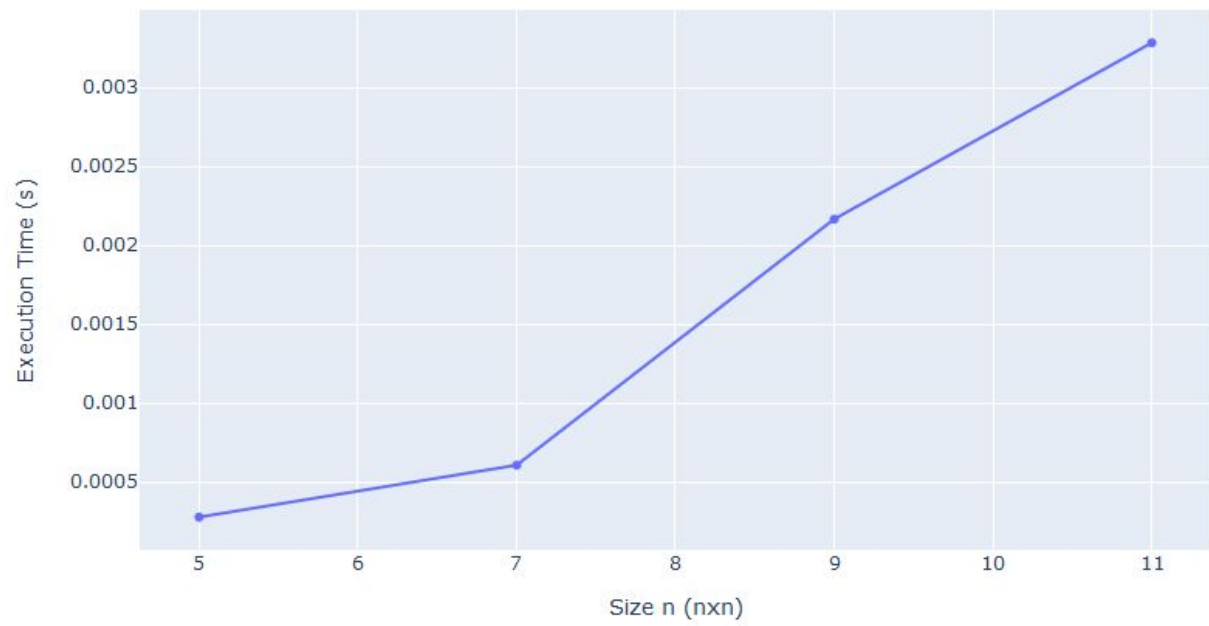


## Hill Climbing

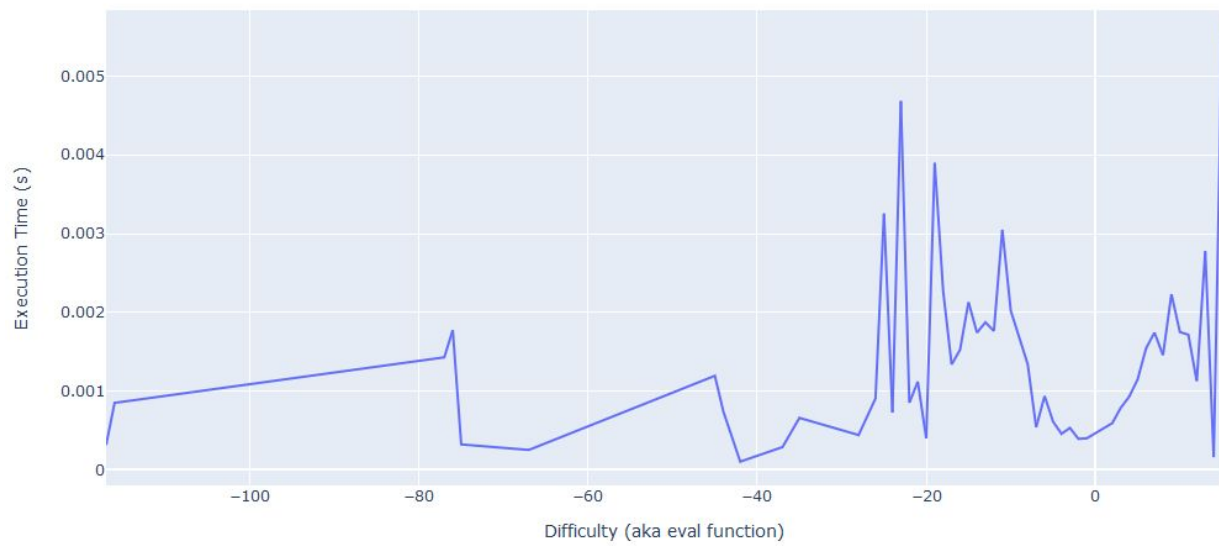


A\*:

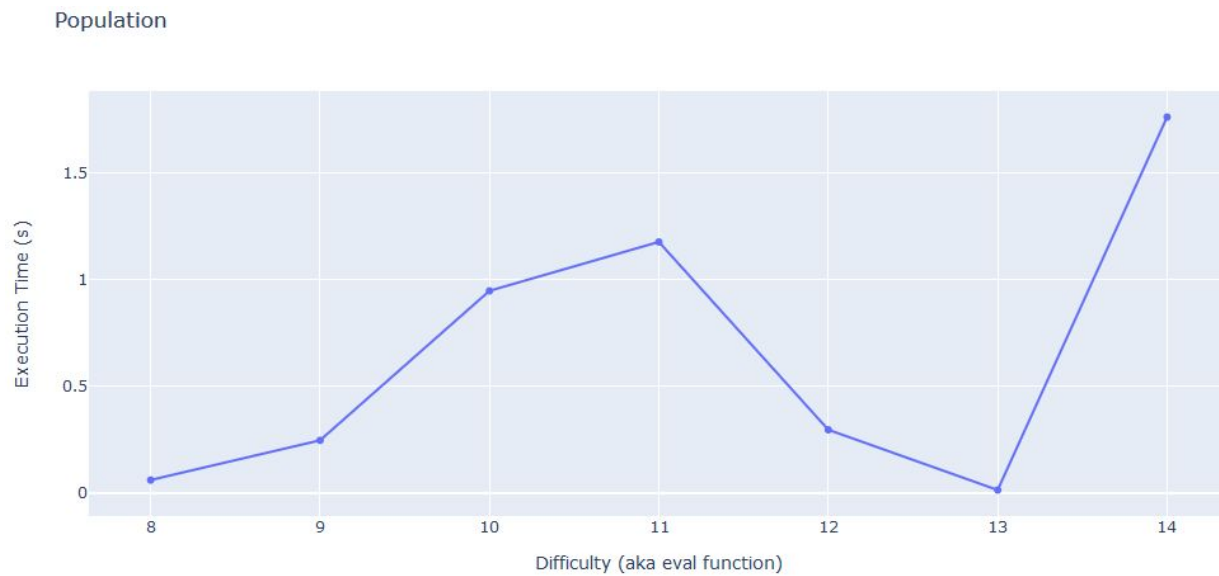
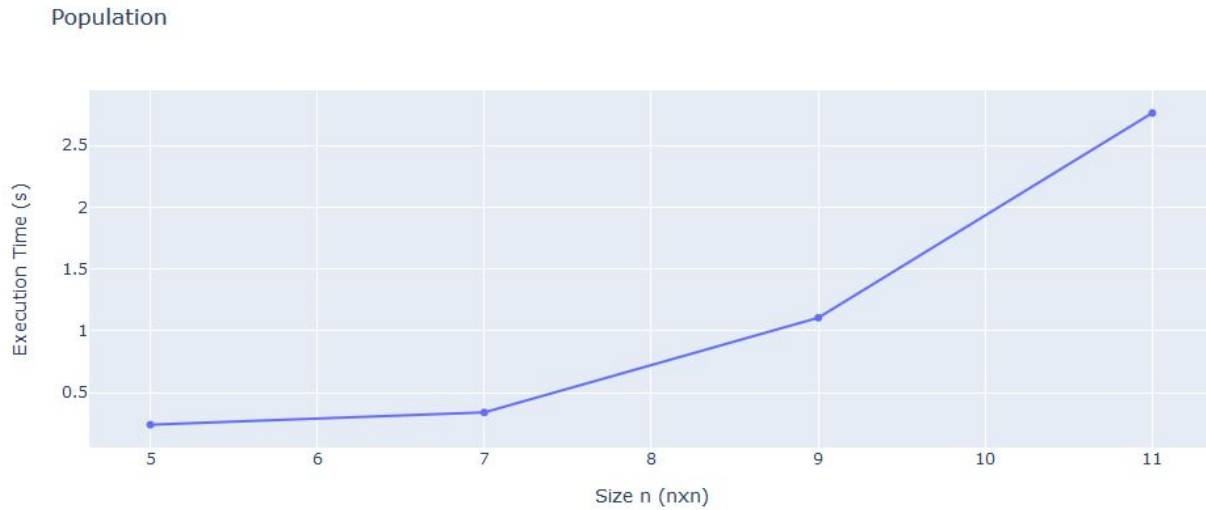
A\*



A\*



Population:



Based on plots provide a description of the advantages and disadvantages of each method:

SPF:

Advantages- Second quickest execution time based on the size of the game

Disadvantages- Big disadvantage is that it does a blind search and doesn't always choose the best path first, and visits every possible legal move of the maze before finding the best one.

Hill Climbing:

Advantages- Third quickest execution time based on the size of the game, is faster than population.

Disadvantages- It will give the best version by randomizing but might not get the hardest version of the game ,and needs more time to make a harder game. Where population makes a harder game given the same computation time

A\*:

Advantages- Quickest execution time based on the size of the game. Better suited for smaller games, predicates best possible move. So it doesn't waste time by checking every possible node.

Disadvantages- Slows down when given a big game to move through. so it will take longer and eat up more computation time if given a big game.

Population:

Advantages- Fourth quickest execution time based on the size of the game, usually gives the best output and game. Always creates the best game, longer computation time means more mutations which means harder game to navigate through.

Disadvantages- Its extremely slow compared to hill climbing, not good for making large computations, since evolution is random and it could possibly up runtime.