A Star Algorithm

Bijoy Singh (120050087)
Sai Kiran Mudulkar(120050068)
Manik Dhar(120050006)

LAB 01 Problem Statement

1. Code A*; keep it general enough to be able to adapt to any search problem.
2. Write modules for open and closed list management. Similarly for parent pointer redirection.
3. Verify experimentally the intuition, "better heuristic performs better".
4. Get the baseline figure of nodes expanded when h=0
5. Measure node expansion when displaced tiles and manhattan distance heuristics are applied
6. Come up with your own heuristic. Prove/disprove their admissibility (i.e., h(n) <= h*(n), for all n)
7. Verify that "if h(n)>h*(n), for all n, A* may find the goal faster, but may discover a suboptimal path".
8. Be liberal with numbers and graphs.Additional or repeated points for emphasis:
9. Arrange for non-reachability test that checks the start and goal node pair right at the start

A* Implementation

Made an abstract implementation in python using object oriented programming.

Implementation of A* Algorithm and storage data structure is independent of the problem at hand, making it abstract for the user.

OL and CL implementation

Made an abstract implementation of open and closed list using Heap and HashMap data structures respectively allowing for highly efficient operations at every step of the iterator.

State type is abstract and stores the value, parent pointer, g-score and h-score.

The Heuristic can be user defined without interfering with other code

8 Puzzle Problem

2	1	4
7	8	3
5	6	

1	2	3
4	5	6
7	8	

This is the problem we try to solve. We need to move the blank so that the state of the matrix reaches the final state

8-Puzzle Problem

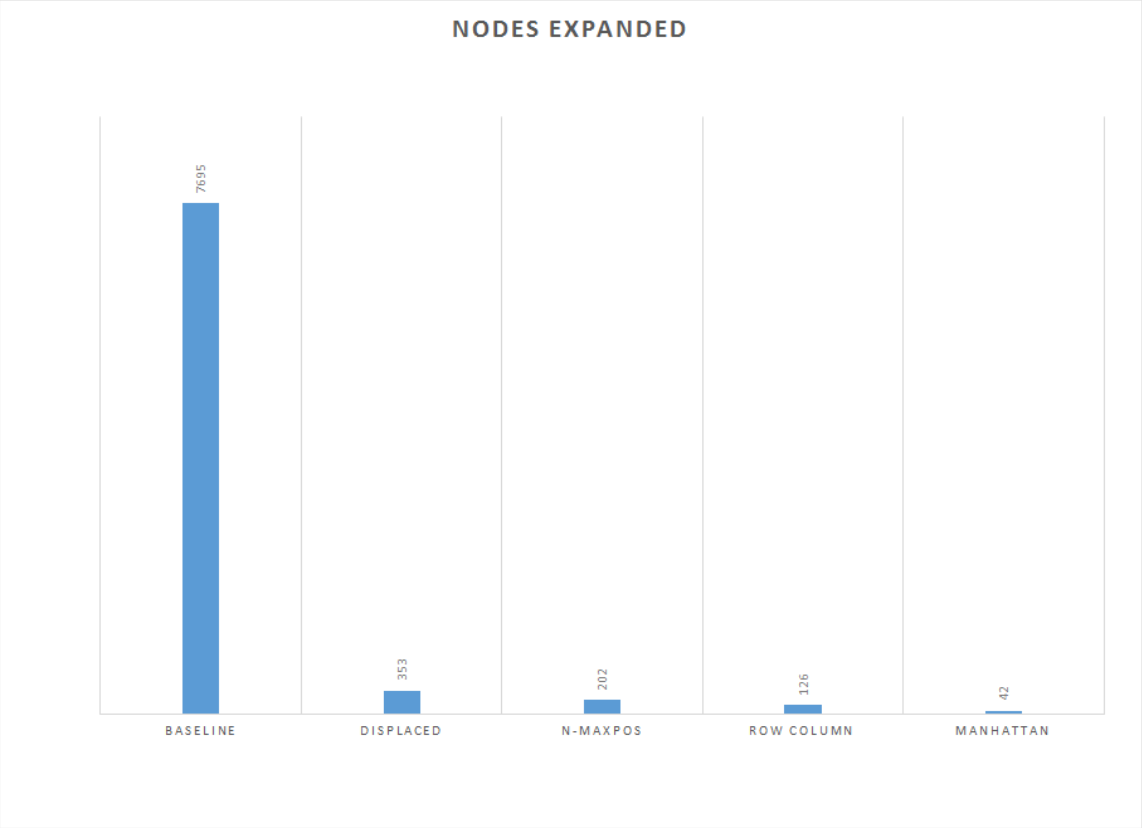
We implemented the 8-puzzle problem over the above API.

Here are the figures:

No Heuristic : 7695*
Displaced tiles : 353*
Manhattan distance : 42*

*The figures are subject to the definition of less than operator for f-score. The less than condition we have used take h-score as the next candidate if f-score is same.

Better Heuristics Perform Better



Our Own Heuristics

We implemented 2 heuristics for the lab

1. n-Max Swap : The number of moves to reach final states where in each move you can swap the blank tile with any tile

   Performance : 202 moves

2. RC Distance : The sum of the number of tiles not in correct column and number of tiles not in correct rows

   Performance : 126 moves

N-Max Swap

We can use the solution of a relaxed problem as a heuristic. The N-Max Swap heuristic is based on this principle.

N-Max Swap has the same goal as the 8-puzzle problem with a different set of allowed moves.

In N-Max Swap we are allowed to swap the blank space with any other tile.

Proof of Admissibility

The set of moves allowed in N-Max Swap is a superset of the moves allowed in the 8-puzzle problem. In the 8-puzzle the blank can be swapped with any of its ADJACENT tiles and in N-Max we can swap the blank with any tile on the board.

As we can copy any set of moves in the 8 puzzle problem in N-Max Swap,

Cost of N-Max Swap ≤ Cost of 8-puzzle problem

Therefore N-Max Swap is admissible

RC Distance Heuristic

This heuristic is better than the Displaced point Heuristic and worse than the Manhattan Heuristic.

The heuristic =

No of Nodes which are not in the correct column

+

No of Nodes which are not in the correct row

Proof Of Admissibility

We will need at least 1 move to get a node to its correct row (if it is not in the correct row) and at least 1 move to get to its correct column (if it is not in the correct column).

Therefore the heuristic is admissible.

Solvability of 8 puzzle

We will discuss the n-puzzle problem in general.

We first define the number of inversions in the problem.

Take the puzzle square and append all of its rows together to obtain an array A. Ignore the blank in the array.

An inversion is a pair (i,j) s.t $i < j$ and $A[i] > A[j]$

Solvability of 8 puzzle (Contd.)

Let the puzzle board be nXn.

If n is odd:

The parity of the no of inversions are an invariant on the swap operations. This means that it is necessary

for the goal state and initial state to have the same parity. (Parity is the remainder on division by 2)
It can also be shown that this is a sufficient condition.

Finding the number of inversions is an easy task and can be used as a reachability test.

Solvability of 8 puzzle (Contd.)

If n = even :

The parity for sum of the number of permutations and the row number of the blank space (with the top row as 0) is an invariant on the set of moves.

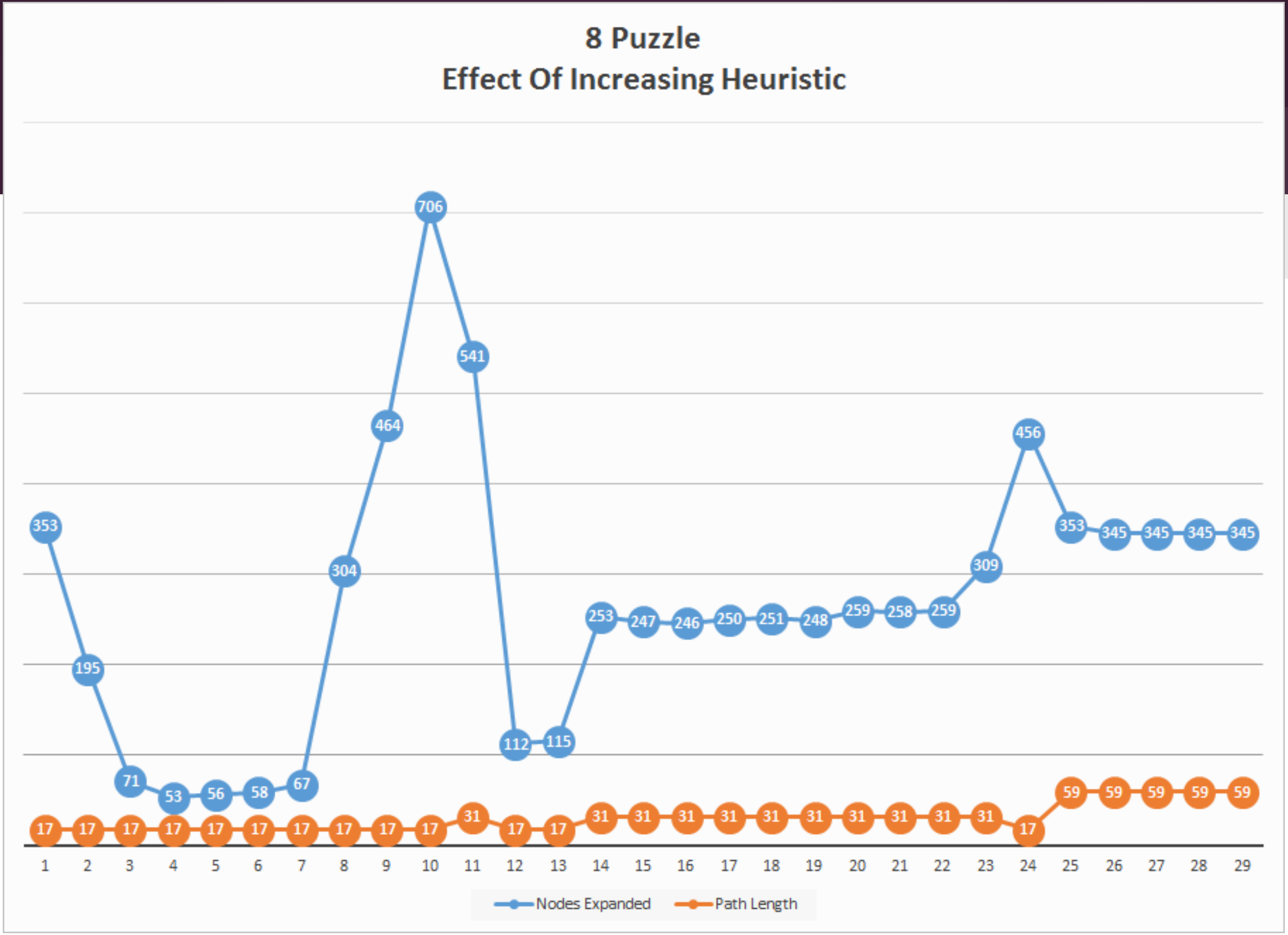
Therefore we can just compare the parity of the goal and initial state. This is also a sufficient condition for solvability.

Non Admissible Heuristic

We used a simple way of creating a non-admissible is to scale the Displaced Tiles Heuristic by a factor and added 32*

The effect of increasing the heuristic is evident in the following graph

*32 is the maximum number of moves needed to solve the problem.

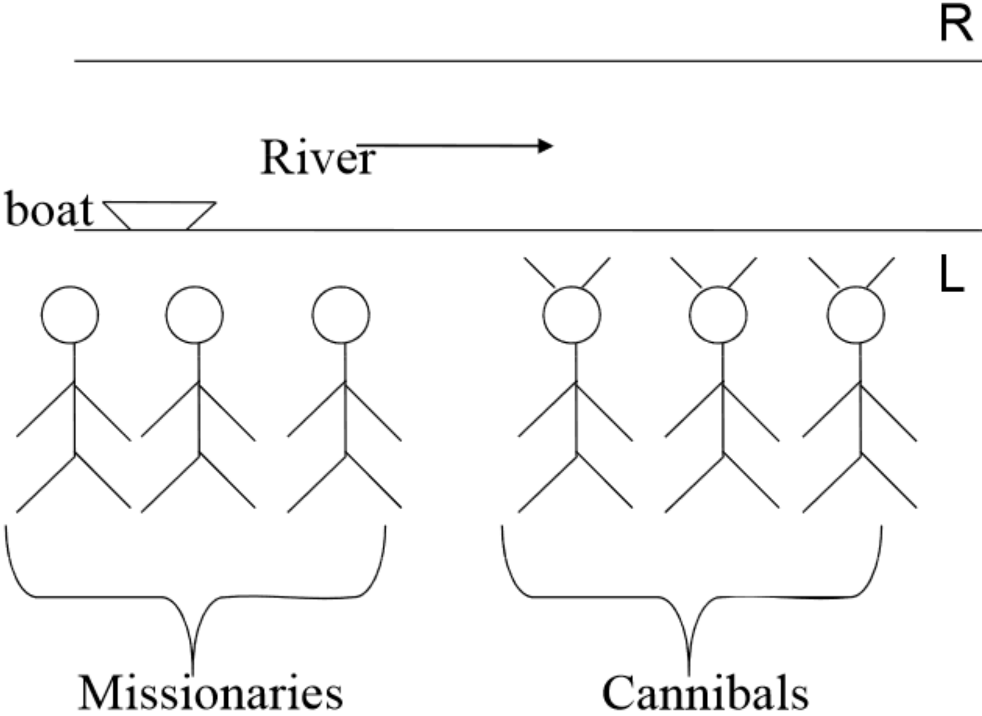


8 Puzzle
Effect Of Increasing Heuristic
353
195
71
53
56
58
67
304
464
706
541
112
115
253
247
246
250
251
248
259
258
259
309
456
353
345
345
345
345
17
17
17
17
17
17
17
17
17
17
31
17
17
31
31
31
31
31
31
31
31
31
31
17
59
59
59
59
59
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
Nodes Expanded
Path Length

ADD GRAPH

Add the plot for overall influence of heuristics across permutations

Cannibal Missionary Problem



Cannibal Missionary Problem

We solved the cannibal missionary problem using the same API we coded.

Results:

No Heuristic : 25 moves

Our Own Heuristics

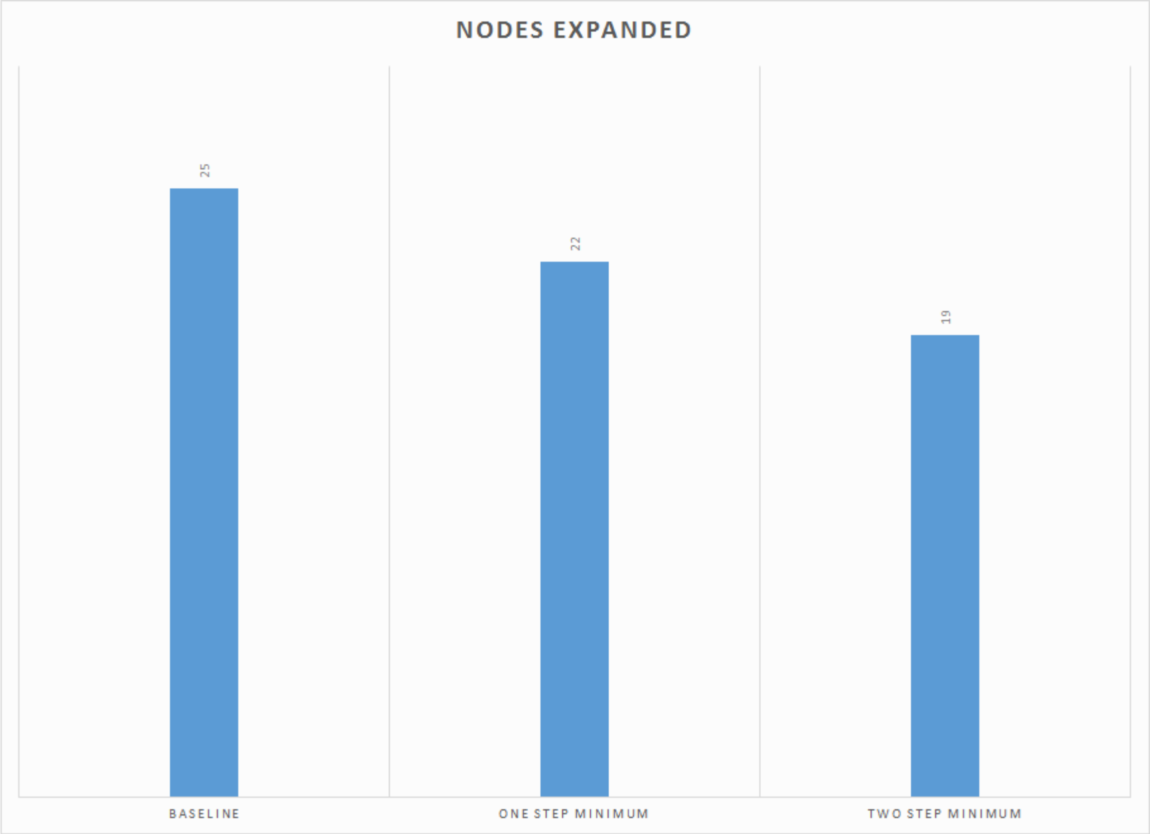
1 Basic One Step heuristic : The minimum times a boat must go back and forth for C people in the first side is (C - 1)

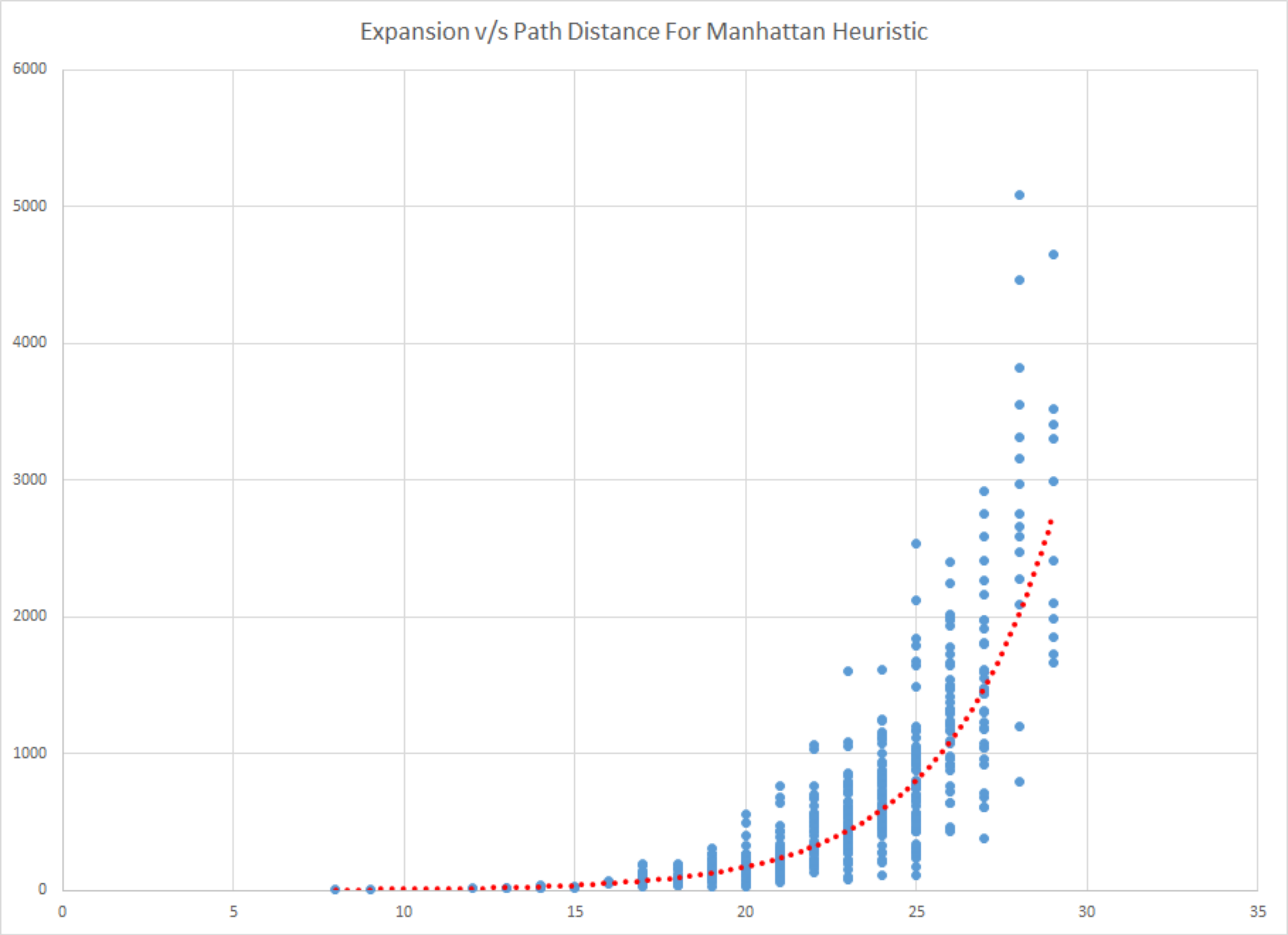
Performance 22 moves

2 Two Step heuristic : A more accurate heuristic, which takes care of which side the boat is and how many people are on first side

Performance 19 moves

Better Heuristic Perform Better




Expansion v/s Path Distance For Manhattan Heuristic
6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35

Conclusions

Heuristic improve the performance over BFS

Better heuristics perform better

Non admissible heuristics may exceed the optimal paths, at the expense of expansion time

THANK YOU