ABHINAV SHARMA – 4 CREDITS
SHYAM RAJENDRAN - 4 CREDITS
SAIKAT ROYCHOWDHURY - 4 CREDITS

Summary

We have implemented the following enhancements in code for faster solution convergence.

1. Since a given board instance heuristic is constant, we pre compute the heuristic value for each instance of the board during its initialization (constructor) using a map and bring the score lookup time to O(1).
2. We have used a one dimensional representation of 8 puzzle. This helps us avoid any costly one-one matching with a goal board. (board_array[i] = i is the invariant for our goal board). Also such a layout allows an easy offset based access of the board tiles. This also means space efficiency, since we do not explicitly store any goal board for each board instance.

Gashnig's Heuristic

We compute Gashnig's heuristic score as follows

1. CASE I - Blank cell is at correct position.
   We maintain a list of misplaced tiles in a list. The list is sorted by tile number i.e a misplaced tile with lower id is placed before a higher misplaced tile. We peek at the head of the sorted list this gives the tile and its current position in the 8 puzzle. We swap the tile with the blank space and update the new position of the tile.
2. CASE II - Blank cell is not at correct position
   For this scenario, we determine which tile is the correct owner of the blank cell. Then that tile is located and replaced with this blank cell. Thus the swapped tile has now reached its correct position in the 8 puzzle. So we remove it off the list of misplaced tiles.

Every shuffle is counted as one step. The aggregate number of steps is the heuristic score.

Inference and observation from the different heuristics applied in solving 8 puzzle game.

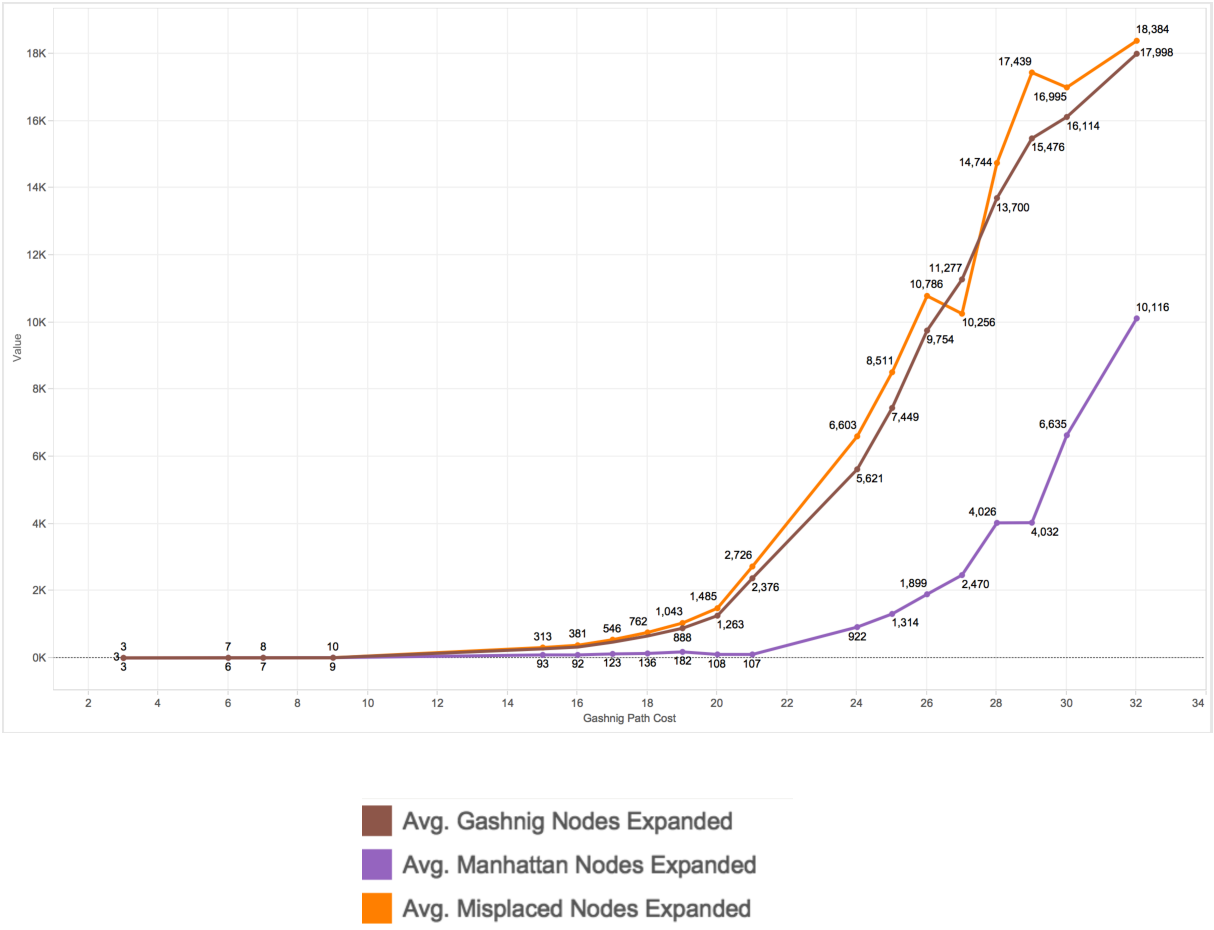
The plot of nodes expanded against the path cost proves our understanding that relaxed heuristics such as Gaschnig's actually expand far more nodes to converge to solution than heuristic based on Manhattan distance.

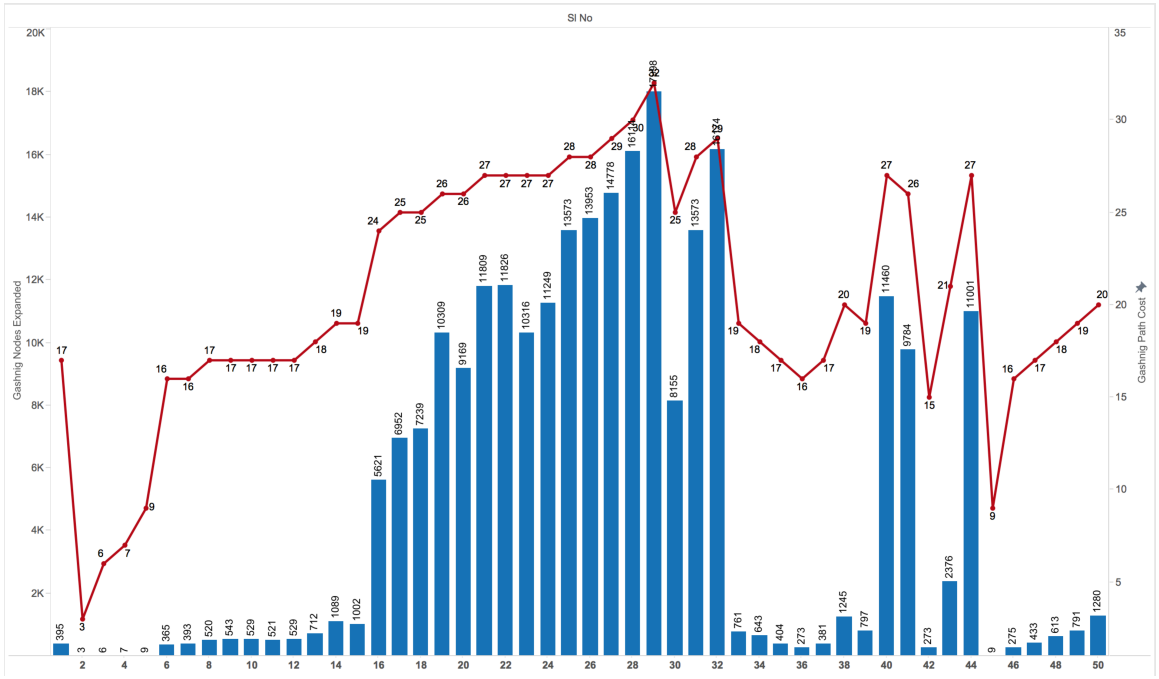
Sl No	PATH COST	Avg. Misplaced Nodes Expanded	Avg. Manhattan Nodes Expanded	Avg. Gashnig Nodes Expanded
1	17	456	89	395
2	3	3	3	3
3	6	7	7	6
4	7	8	8	7
5	9	10	10	9
6	16	426	109	365
7	16	448	146	393
8	17	599	169	520
9	17	616	170	543
10	17	607	162	529
11	17	604	117	521
12	17	607	162	529
13	18	825	194	712
14	19	1,248	355	1,089
15	19	1,181	280	1,002
16	24	6,603	922	5,621
17	25	8,002	1,093	6,952
18	25	8,322	1,432	7,239
19	26	11,389	2,147	10,309
20	26	10,196	1,595	9,169
21	27	12,778	2,765	11,809
22	27	12,878	2,460	11,826
23	27	11,220	1,979	10,316
24	27	12,302	2,402	11,249
25	28	14,693	3,882	13,573
26	28	14,847	4,315	13,953
27	29	17,862	4,971	14,778
28	30	16,995	6,635	16,114
29	32	18,384	10,116	17,998

30	25	9,210	1,418	8,155
31	28	14,693	3,882	13,573
32	29	17,016	3,092	16,174
33	19	931	121	761
34	18	754	108	643
35	17	479	93	404
36	16	320	60	273
37	17	450	77	381
38	20	1,440	77	1,245
39	19	924	41	797
40	27	12,357	2,537	11,460
41	26	10,772	1,956	9,784
42	15	313	93	273
43	21	2,726	107	2,376
44	27	-1	2,679	11,001
45	9	10	10	9
46	16	329	54	275
47	17	499	65	433
48	18	707	107	613
49	19	931	115	791
50	20	1,529	138	1,280

Graph 1:

Nodes Expanded vs Actual Path cost ***: Manhattan Vs Misplaced Tiles Vs Gaschnig'***

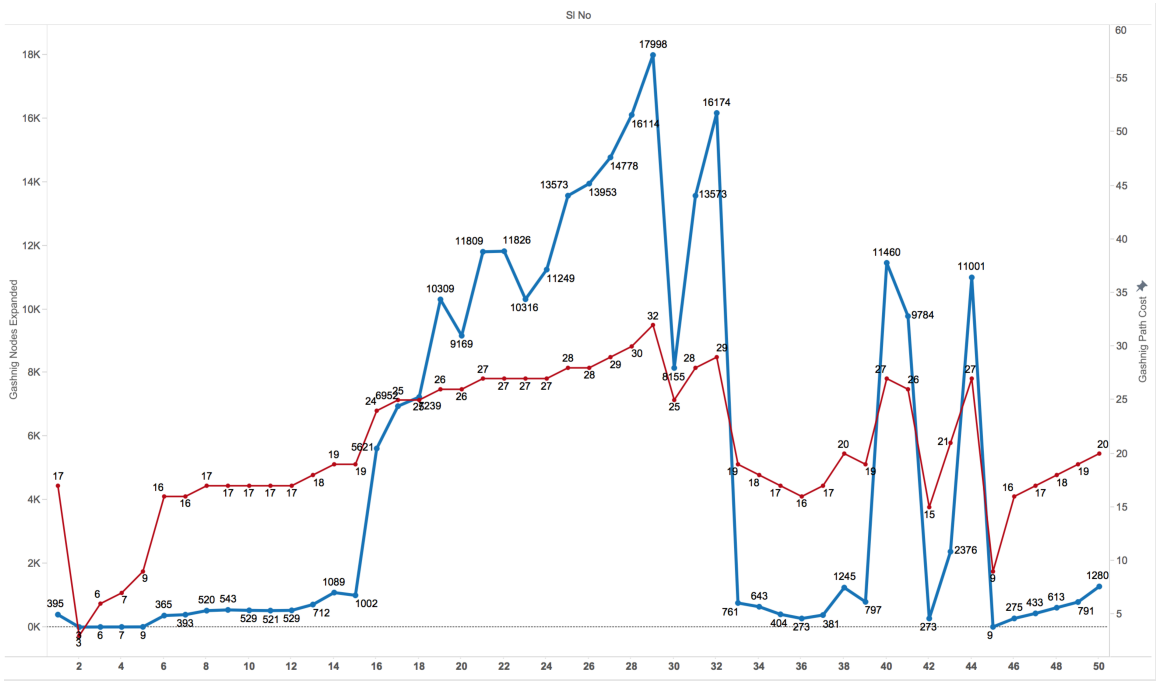




Graph 2

Path cost vs Nodes expanded for 50 puzzles solved using ***Manhattan heuristic***

Another interesting observation is when we plotted path cost vs nodes expanded across the 50 puzzle experiment for each heuristic. [Graph 2] . We can observe the relation of how complexity of the puzzle pulls the nodes expanded cost more. But its also relatively constant for easier puzzles (puzzles solved with path cost less than 10) where the nodes expanded is at an average around 50-100.

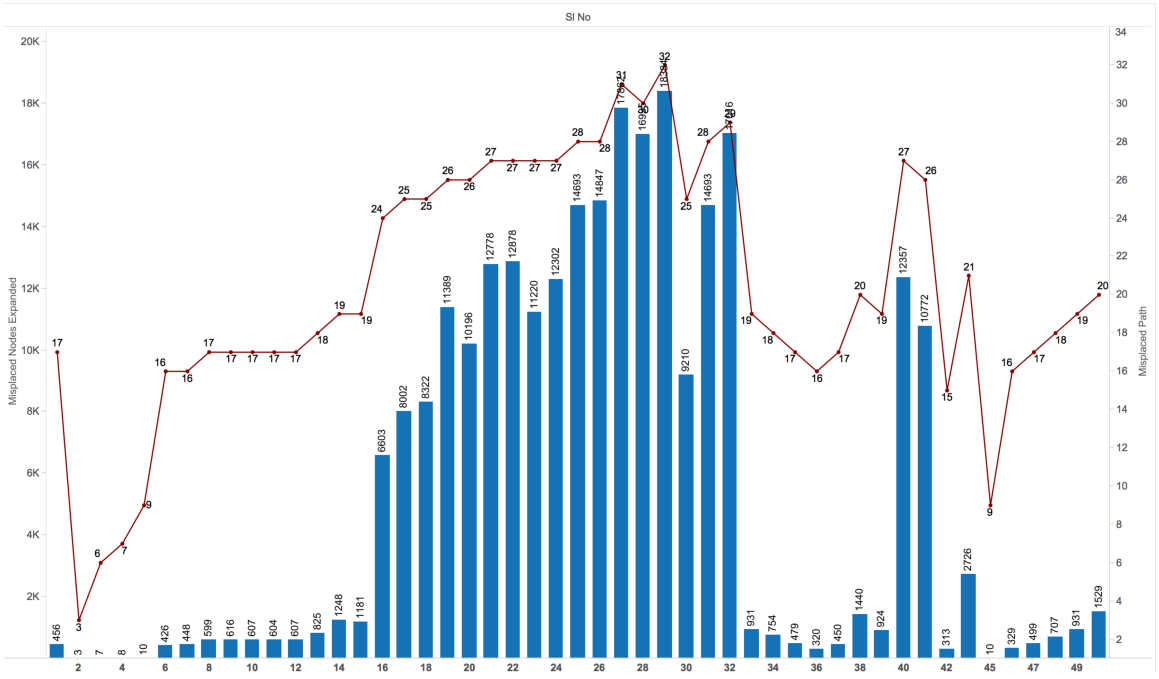


Graph 3

Path cost vs Nodes expanded for 50 puzzles solved using ***Gaschnig's heuristic***

For the same 50 puzzles, the average nodes expanded when solved using Gaschnig stands at around 200-300 for [**Significant increase** of expanded nodes from Manhattan heuristic solution].

We can observe the trend for Misplaced Tiles heuristics as well from the Graph 4. The expanded nodes are slightly higher compared to Gashnigs heuristic solution.



Graph 4

Path cost vs Nodes expanded for 50 puzzles solved using ***Misplaced Tile Heuristic***

Tabulated Summary of different Heuristic's Performance across 50 Puzzles

PATH COST	Gashnig Nodes Expanded	Manhattan Nodes Expanded	Misplaced Nodes Expanded
32	17998	10116	18384
30	16114	6635	16995
29	14778	4971	17862
	16174	3092	17016
28	13573	3882	14693
	13953	4315	14847
27	10316	1979	11220
	11249	2402	12302
	11460	2537	12357
	11809	2765	12778
	11826	2460	12878
26	9169	1595	10196
	9784	1956	10772
	10309	2147	11389
25	6952	1093	8002
	7239	1432	8322
	8155	1418	9210

24	5621	922	6603
21	2376	107	2726
20	1245	77	1440
	1280	138	1529
19	761	121	931
	791	115	931
	797	41	924
	1002	280	1181
	1089	355	1248
18	613	107	707
	643	108	754
	712	194	825
17	381	77	450
	395	89	456
	404	93	479
	433	65	499
	520	169	599
	521	117	604
	529	162	607
	543	170	616
16	273	60	320
	275	54	329
	365	109	426
	393	146	448
15	273	93	313
9	9	10	10
7	7	8	8
6	6	7	7
3	3	3	3

The above table shows the performance of each heuristic in terms of nodes expanded across the 50 8 puzzles games solved.