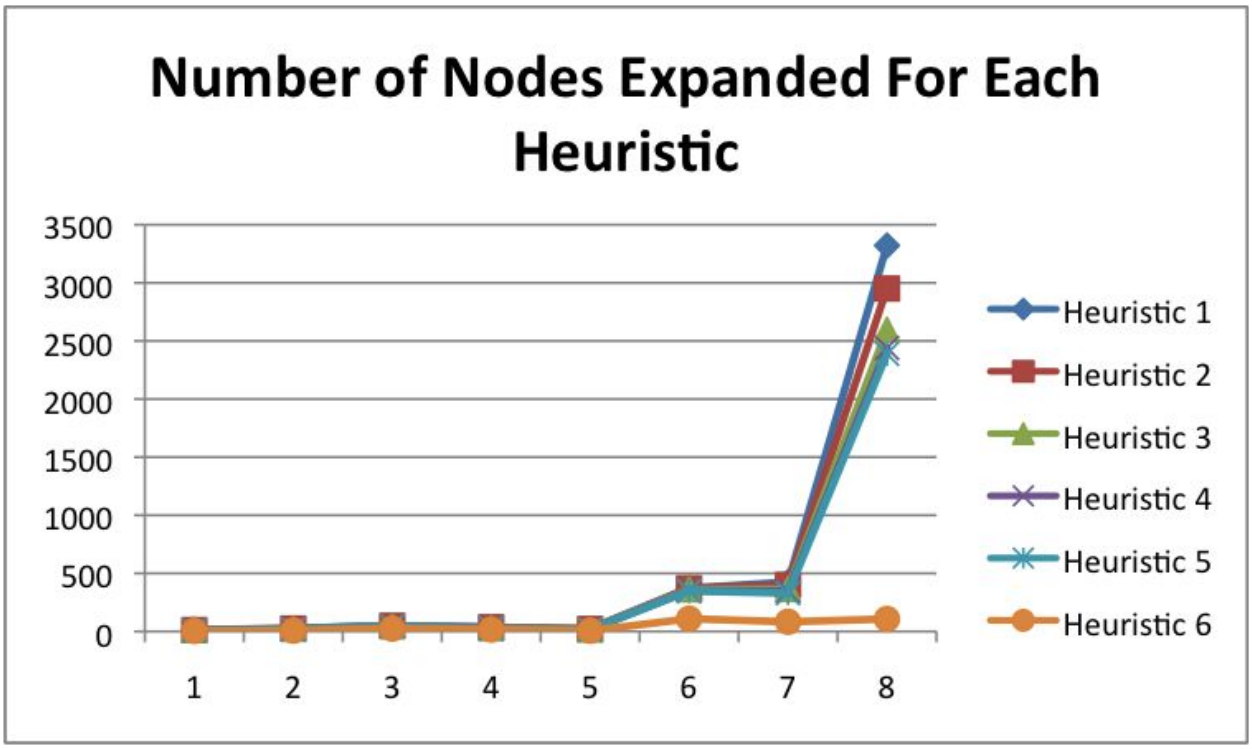
Josh Audibert
Nick Cebry
Breanne Happell
Morgan Hopeman

Assignment 1 Write-Up

Heuristic #	World #	Score	Number of Actions	Number of Nodes Expanded	Effective Branching Factor
1	1	90	5	15	1.718771928
2	1	90	5	12	1.64375183
3	1	90	5	12	1.64375183
4	1	90	5	12	1.64375183
5	1	90	5	9	1.551845574
6	1	90	5	4	1.319507911
1	2	84	7	26	1.592718592
2	2	84	7	25	1.583819609
3	2	84	7	22	1.555158537
4	2	84	7	20	1.534127405
5	2	84	7	18	1.511209391
6	2	84	7	7	1.320469248
1	3	81	10	50	1.478757637
2	3	81	10	49	1.475773162
3	3	81	10	48	1.472733358
4	3	81	10	47	1.469636013
5	3	81	10	45	1.46325916
6	3	81	10	24	1.37410881
1	4	85	9	36	1.489095325
2	4	85	9	36	1.489095325
3	4	85	9	30	1.459232803
4	4	85	9	28	1.448089274
5	4	85	9	25	1.429969148
6	4	83	7	16	1.485994289
1	5	86	4	23	2.189938703
2	5	86	4	21	2.140695143
3	5	86	4	19	2.08779763

4	5	86	4	15	1.967989671
5	5	86	4	15	1.967989671
6	5	86	4	6	1.56508458
1	6	50	24	372	1.279693476
2	6	50	24	371	1.279549956
3	6	50	24	362	1.278241334
4	6	50	24	353	1.276901154
5	6	50	24	348	1.27614239
6	6	50	24	109	1.215885756
1	7	51	14	425	1.540785087
2	7	51	14	400	1.534127405
3	7	51	14	364	1.523827522
4	7	51	14	347	1.518630456
5	7	51	14	329	1.512863368
6	7	51	14	83	1.371124865
1	8	-10	51	3321	1.172315448
2	8	-10	51	2951	1.169603373
3	8	-10	51	2597	1.166676447
4	8	-10	51	2440	1.165250796
5	8	-10	51	2389	1.164768273
6	8	-21	44	107	1.112044992



Q1: How do the 5 heuristics vary in effectiveness?

As the heuristic number increases so does the effectiveness. Heuristic 1 is the least effective and takes the most time and expands the most nodes whereas heuristic 5 expands the least nodes and runs the fastest.

Q2: How much gain is there to using *any* heuristic (#1 vs. #2)?

For all of the tests fewer nodes have to be explored when heuristic 2 is used rather than no heuristic (heuristic 1). The tests also run faster when heuristic 2 is used. So by using an admissible heuristic, even one that is not extremely effective, the number of nodes expanded can be significantly fewer than no heuristic (a difference of ~400 nodes in test 6). As a general rule, an admissible heuristic that is always greater than or equal to another admissible heuristics expands the same or smaller number of nodes. Therefore, 5<=4<=3<=2<=1 in terms of number of nodes expanded.

Q3: Is #5 noticeably more effective than the other heuristics?

Yes, heuristic 5 expands fewer nodes than any other admissible heuristic does. The difference between heuristics 4 and 5 is minimal for small search trees, but becomes more noticeable with larger search trees. Unlike heuristic 6, heuristic 5 always finds the optimal answer.

Q4: For heuristic 6 how does its solution quality compare with heuristic 5? Is it performing noticeably worse? How much more efficient is it?

Heuristic 6 is notably faster than any of the other heuristics. It is even significantly faster and expands fewer nodes than the very efficient heuristic 5. The number of nodes expanded is significantly smaller than the number of nodes expanded when heuristic 1 is used. Though heuristic 6 is notably faster and more effective it does not guarantee an optimal solution. This can be seen in the results when the score is lower when solved with heuristic 6 than the score of any of the other heuristics (such as in tests 4 and 6). These failures prove that heuristic 6 is not optimal but that it is still much more efficient given that it expands less nodes and runs much faster than any of the other heuristics.

Note: For this project, a graph search version of A* was used instead of a tree search version. The graph search version significantly improves the effective branching factor by not considering any new states that are duplicates of any already generated states unless the new states have a lower path cost than the previously generated states. State equality is determined by the robot having the same coordinates and heading in both states. When a node is expanded, each child state is tested for equality with each state that has already been expanded, and each state in the frontier. If the child node has a lower path cost, it replaces the old state. Otherwise, it is ignored and not added to the search tree.