Heuristic Analysis by Kenan Sooklall

This analysis presents a comparison of different search algorithms when used on 3 planning problems with search spaces of different size. The table below shows the performance of each algorithm and the resulting number of expanded nodes, goal tests, new nodes, execution time and performance.

Problem	Method	Expand	Goal Tests	New Nodes	Length	Opt	Time	P.L.T
1	BFS	43	56	180	6	1	0.027	222.222
1	DPS	12	13	48	12	1	0.009	1333.333
1	A*_H1	55	57	224	6	1	0.036	166.667
1	A*_LVS	11	13	50	6	1	0.694	8.646
2	BFS	3346	4612	30534	9	1	14.137	0.637
2	DPS	1124	1125	10017	1085	0	8.347	129.987
2	A*_H1	4696	4698	42642	9	1	11.578	0.777
2	A*_LVS	74	76	720	9	1	50.526	0.178
3	BFS	14120	17676	124926	12	1	89.78	0.133
3	DPS	677	678	5608	660	0	3.527	187.128
3	A*_H1	17962	17964	157470	12	0	52.483	0.227
3	A*_LVS	279	281	2552	12	0	260.509	0.046

Table 1: Analysis of search for all three problems

Depth first search is by far the fastest search method; however it comes at the cost of taking the longest path. For example, looking at problem DPS took about 9 seconds with a path length of 1085 while BFS took almost twice as long but output a path length of only 9, about 120 times shorter.

The last column in the table (PLT) is the ratio between the Path Length and the time the algorithm took to figure it out. A lower PLT will be defined as a more optimized path. According to that definition it is clear that A* with level sum heuristic is the best for the current problem.

Further analysis is need with other algorithms; however A* might still be on top, since it has shown promising results in other applications.