Written Analysis

Optimal Plans

Problem 1

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

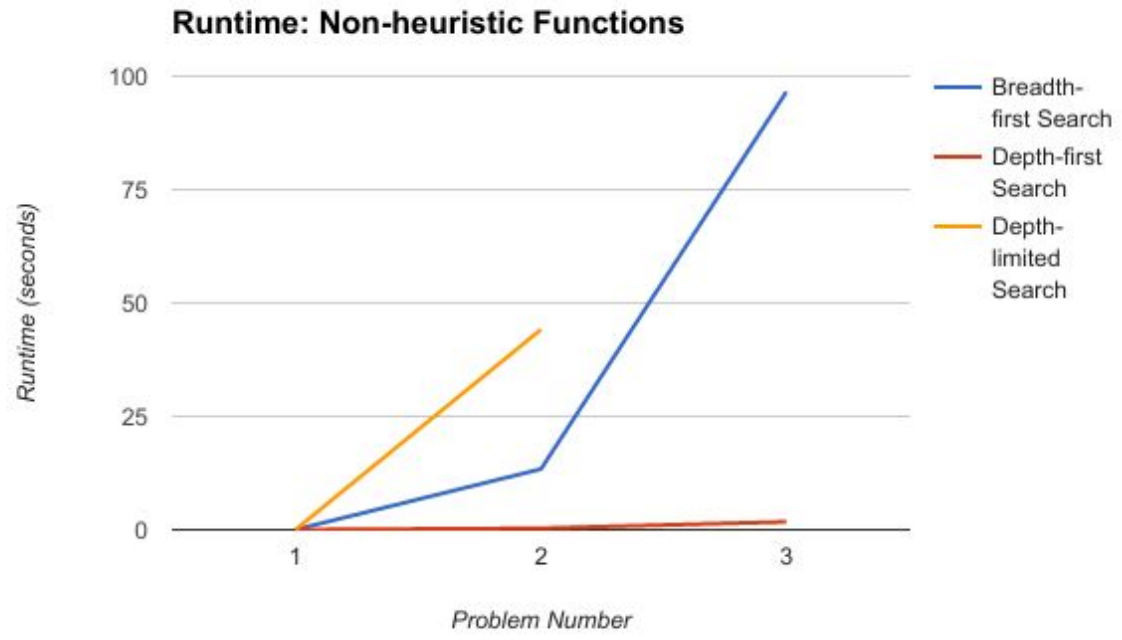
Problem 2

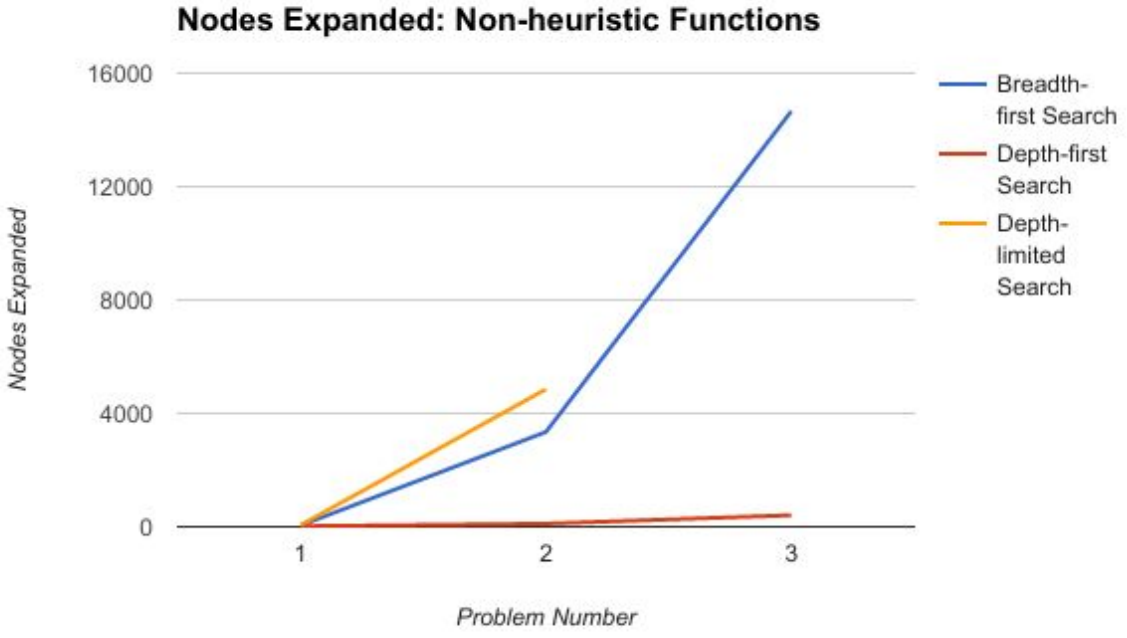
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 3

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

Compare and contrast non heuristic result metrics





Each algorithm's runtime is about proportional to the number of nodes expanded. This means that algorithms that can prune trees and expand fewer nodes will typically run more quickly.

Breadth-First Search

Breadth-First Search takes longer than Depth-First Search. By the nature of the algorithm, if it finds a solution, that solution is optimal. The optimal solutions are of length 6, 9 and 12 for problems 1, 2 and 3 respectively.

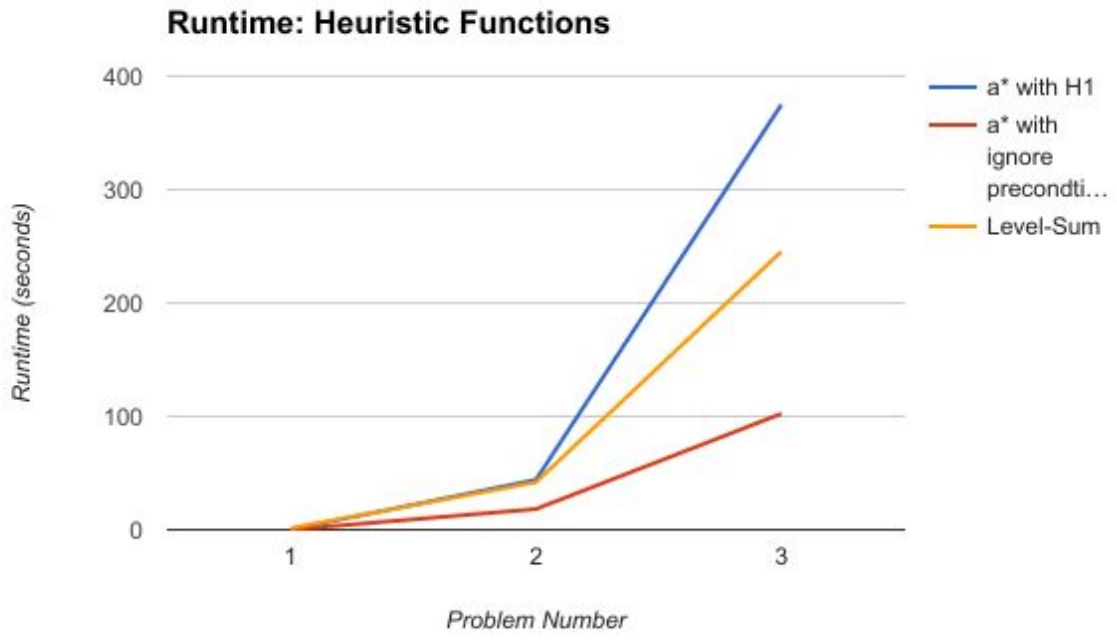
Depth-First Search

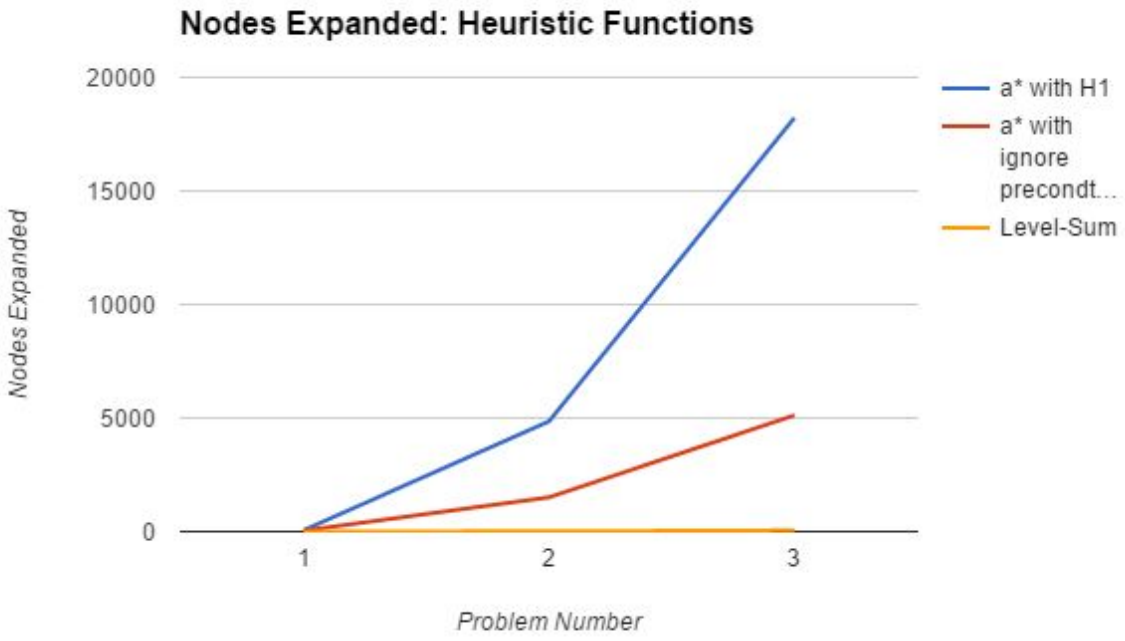
Depth-First Search takes less time than Breadth-First Search because it expands fewer nodes while searching down a single path of the tree. By the nature of this algorithm, if there is a solution, it will find it however it is not guaranteed to be optimal (and usually it is not). The solutions found by Depth-First Search are of length 20, 105 and 392 for problems 1, 2, and 3 respectively.

Uniform-Cost Search

Uniform-Cost search also found optimal solutions to each problem, but with slightly longer run-time than Breadth-first search due to the overhead of cost calculation and since in this case it expands more nodes.

Compare and contrast heuristic search result metrics using A*





Node expansions are proportional to runtime across problems using the same heuristic.

“H1 heuristic”

This heuristic uses a uniform cost for expanding each node, essentially performing a uniform cost search (in fact,, in the appendix, we can see that these algorithms are equivalent in nodes expanded and run-time). Its runtime increases proportionally to the number of nodes which could be expanded in the tree (proportional to the number of actions times the number of conditions in the worst case).

Ignore preconditions

Because the Ignore Precond heuristic can effectively prune its tree with its heuristic, it runs faster than the H1 heuristic. It expands about 30% of the nodes, and runs in about 30% of the runtime. It also finds an optimal solution in that time.

Level-sum

The level-sum heuristic unlike the other two heuristics does not find an optimal solution on problem 3, but performs close to optimally. The number of nodes expanded is significantly less than the other two algorithms, however the overhead in calculating the heuristic places its runtime above that of the Ignore Preconditions Heuristic.

Best heuristic

Given the results, the best heuristic on these problems was Breath-First-Search. It discovers an optimal solution in the shortest runtime. However, the ignore preconditions heuristic performed about as well, and expanded fewer nodes. Breadth-first Search may not scale as well as the level-sum heuristic or the ignore preconditions heuristics if the tree grows larger. It depends on how much time is spent calculating the heuristics for these two algorithms. For this reason, I would recommend the ignore preconditions heuristic overall.

Appendix

* Note: “N/A” indicates that the algorithm did not finish

Algorithm	Code	Problem	Expansions	Goal Tests	New Nodes	Time	Solution Len
Breadth-first Search	1	1	43	56	180	0.03282	6
Breadth-first Search	1	2	3346	4612	30534	13.37913	9
Breadth-first Search	1	3	14663	18098	129631	96.56095	12
breadth first tree search	2	1	1458	1459	5960	0.997574	6
breadth first tree search	2	2	N/A	N/A	N/A	N/A	N/A
breadth first tree search	2	3	N/A	N/A	N/A	N/A	N/A
Depth-first Search	3	1	21	22	84	0.015155	20
Depth-first Search	3	2	107	108	959	0.32484	105
Depth-first Search	3	3	408	409	3364	1.752834	392
Depth-limited Search	4	1	101	271	414	0.0980487	50
Depth-limited Search	4	2	213491	1967093	1967471	945.45581	50
Depth-limited Search	4	3	N/A	N/A	N/A	N/A	N/A

Uniform Cost Search	5	1	55	57	224	0.0411057	6
Uniform Cost Search	5	2	4853	4855	44041	44.1508	9
Uniform Cost Search	5	3	N/A	N/A	N/A	N/A	N/A
Recursive Best-First Search	6	1	4299	4230	17023	2.94161	6
Recursive Best-First Search	6	2	N/A	N/A	N/A	N/A	N/A
Recursive Best-First Search	6	3	N/A	N/A	N/A	N/A	N/A
Greedy Best-First Graph Search	7	1	7	9	28	0.00561	28
Greedy Best-First Graph Search	7	2	998	1000	8982	7.33194	21
Greedy Best-First Graph Search	7	3	5578	5580	49150	98.641086	22
a* with H1	8	1	55	57	224	0.04279	6
a* with H1	8	2	4853	4855	44041	44.05	9
a* with H1	8	3	18223	18225	159618	374.9345	12
a* with ignore precondtions	9	1	41	43	170	0.06873	6
a* with ignore precondtions	9	2	1506	1508	13820	18.250472	9
a* with ignore precondtions	9	3	5118	5120	45650	102.371076	12
Level-Sum	10	1	7	9	28	1.33518	6
Level-Sum	10	2	13	15	123	41.8990412	9

Level-Sum	10	3	59	61	523	245.365518	13