

Project 3: Written Analysis

Author: Edgar Pino

The purpose of this paper is to analyze the results of non-heuristic search algorithms for problems 1, 2, and 3.

Problem 1

Algorithm	Expansions	Goal Test	New Nodes	Plan Length	Time Elapsed
breadth_first_search	43	56	180	6	0.04914
depth_first_graph_search	12	13	48	12	0.01577
uniform_cost_search	55	57	224	6	0.05811
astar_search with h_1	55	57	224	6	0.0602
astar_search with h_ignore_preconditions	41	43	170	6	0.0527
astar_search with h_pg_levelsum	11	13	50	6	0.5428

Analysis

Overall, this problem was very easy to solve. All of the algorithms I selected were able to find the solution fairly quickly, less than $\frac{1}{10}$ of a second. The most successful non-heuristic function I had was depth first graph search. It had the least amount of expansions, goal test, new nodes, and time elapse. I didn't have the best plan length due to it being a depth first search algorithm which goes deep into the graph. The heuristic function with best results was the A* search with h_pg_levelsum. It had the best expansion, goal test, new nodes, plan length, and time elapsed compared to others.

Best Plan:

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

Problem 2

Algorithm	Expansions	Goal Test	New Nodes	Plan Length	Time Elapsed
breadth_first_search	3343	4609	30509	9	17.0928
depth_first_graph_search	582	583	5211	575	3.61761
uniform_cost_search	4852	4854	44030	9	17.2327
astar_search with h_1	4852	4854	44030	9	17.5812
astar_search with h_ignore_preconditions	1450	1452	13303	9	5.87633
astar_search with h_pg_levelsum	86	88	841	9	44.9567

Analysis

This problem was a little more challenging. Some of the algorithms were able to find the solution within a few seconds but some took longer than 40 seconds. The most successful non-heuristic algorithm was the breadth first search which was very close to the uniform cost search. The depth first search had a low elapsed time but a very high plan length compared to others. On the other hand, the most successful heuristic algorithm was A* search with h_ignore_preconditions due to its low elapsed time.

Best Plan

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

Problem 3

Algorithm	Expansions	Goal Test	New Nodes	Plan Length	Time Elapsed
breadth_first_search	14663	18098	129631	12	114.288
depth_first_graph_search	627	628	5176	596	3.8456
uniform_cost_search	18235	18237	159716	12	69.8131
astar_search with h_1	18235	18237	159716	12	71.7233
astar_search with h_ignore_preconditions	5040	5042	44944	12	21.6268
astar_search with h_pg_levelsum	316	318	2912	12	223.428

Analysis

In this problem we see variant results like in problem 2. Depth first graph search has a low elapsed time but it has very high plan length, just like other problems. The top performant non-heuristic algorithm was breadth first search. It had a relatively low expansion, goal test, new nodes, and plan length. It does have a high elapsed time compared to other. The best heuristic algorithm was the A* with h_pg_levelsum with low expansions, goal test, new nodes, and plan length but not elapsed time.

Best Plan

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