Heuristic Analysis

Optimal Plans:

Problem 1
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 2
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
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Results

Problem 1						
Algorithms	**Expansions**	**Goal Tests**	**New Nodes**	**Time (S)**	**Length**	**Optimal**
breadth_first_search	43	56	180	0.0516	6	Yes
breath_first_tree_search	1458	1459	5960	1.63	6	Yes
depth_first_graph_search	21	22	84	0.0251	20	No
depth_limited_search	101	271	414	0.13	50	No
uniform_cost_search	55	57	224	0.0697	6	Yes
recusive_best_first_earch with h_1	4229	4230	17023	4.707	6	Yes
greedy_best_first_graph_search h_1	7	9	28	0.111	6	Yes (Lucky!)
astar_search h_1	55	57	224	0.073	6	Yes
astar_search h_ignore_preconditions	41	43	170	0.075	6	Yes
astar_search h_pg_levelsum	11	13	50	1.632	6	Yes
Problem 2						
Algorithms	**Expansions**	**Goal Tests**	**New Nodes**	**Time**	**Length**	**Optimal**
breadth_first_search	3343	4609	30509	19.26	9	Yes
breath_first_tree_search	Ran for 29 minutes and did not halt					
depth_first_graph_search	624	625	5602	4.01	619	No
depth_limited_search	222719	2053741	2054119	890.632	50	No
uniform_cost_search	4853	4855	44041	17.484	9	Yes
recusive_best_first_earch with h_1	Ran for 2000 minutes, did not halt					
greedy_best_first_graph_search h_1	998	1000	8982	4.546	17	No
astar_search h_1	4853	4855	44041	22.826	9	Yes
astar_search h_ignore_preconditions	1450	1452	13303	8.245	9	Yes
astar_search h_pg_levelsum	86	88	841	141.258	9	Yes
Problem 3						

Algorithms	Expansions	Goal Tests	New Nodes	Time	Length	Optimal
breadth_first_search	14663	18098	129631	95.60934187	12	Yes
breath_first_tree_search	Did not halt					
depth_first_graph_search	408	409	3364	1.699	392	No
depth_limited_search	Ran for 1.5 hours, did not halt					
uniform_cost_search	18223	18225	159618	49.842	12	Yes
recusive_best_first_earch with h_1	Did not even try					
greedy_best_first_graph_search h_1	5578	5580	49150	29.463	22	No
astar_search h_1	18223	18225	159618	95.695	12	Yes
astar_search h_ignore_preconditions	5040	5042	44944	30.881	12	Yes
astar_search h_pg_levelsum	325	327	3002	733.561	12	Yes

Non-heuristic Comparisons

Uniform cost search and breadth first search were the heros of the non-heuristic searches. BFS only performed poorer than UCS since it used a (poor data structure), otherwise they're essentially the same w/o the heuristic in UCS. Both are optimal searches and finished in a reasonable period of time.

DFS was non-optimal, but reached a (stupid!) solution quickly. The tree BFS search didn't even halt. For the non-heuristic searches, Expansions and goal tests were strongly correlated with search time.

Heuristic Comparisons

All of the heuristic searches halted, except for the resusive_best_first_search with h_1. Greedy first search is the fastest, but gives non-optimal solutions. The ignore_preconditions is a very basic heuristic that only had a range of 3 or 4 values (depending on the problem), but end a lot to guide the search and split the search time and expansion count in a third.

The pg_levelsum is the big surprise. Node expansions are 15 time lower than even than ignore_preconditions, but it's execution time is 24 times higher. That means it's heuristic guidance is excellent, but it's too computationally expensive. I think it'd be possible to modify the algorithm to not have to re-compute the planning tree at every heuristic execution, and get much better results out of it.

Comparison for Heuristic vs Non-heuristic search

For the trivial problems, non-heuristic search was the strongest for optimal searches. The effect size is small enough, it may not even be significant. But as the problem size grew, a heuristic was sorely needed. BFS still performed well throughout, but it’s clear that the heuristic is able to guide the search into fewer expansions and a stronger/shorter runtime.

If a more efficient execution of the graph algorithm were used that was even triple the cost of the iignore_preconditions heuristic, one could suppose a completion time of 6 seconds on the 3rd problem.

Ultimately though I found all of these results disappointing, since I think a human could complete any of these in about the time it’d take to read and express the answer. Something seems awfully wrong if computers take this much effort to solve the trivial air cargo problems we’ve looked at.