Heuristic Analysis
Masa Watanabe

In terms of optimality, among the non-heuristic functions, breadth first search performed the best. Despite an increase in the complexity, the search provided the shortest plan length for every problem. However, the search took the longest to run by a significant margin. In comparison, the depth first graph search provided efficiency. A plan was produced for each problem at a fraction of the time compared to the breadth first search. On the other hand, the length of the plan was significantly larger than that of the breadth first search. The two search functions are opposites of each other. One compromises efficiency for optimality while the other compromises optimality for efficiency. However, one distinction worth noting is the two search functions didn't order the problems in the same manner. It was most expensive to produce a plan for problem 3 in breadth first search whereas in depth first graph search, it was problem 2. The distinction is a potential indication that the two search functions prioritize factors that adds complexity different. For breadth first search, the lack of equality (4 cargos and airports vs 2 planes) caused the most complexity; for the depth first graph search, it was the ratio of positive to negative literals at the initial state (6:21 in problem 2 vs 6:27 in problem 3). The greedy best first graph search function found the happy medium between the first two functions. The function placed second in both efficiency and optimality when comparing the 3 functions.

Among the heuristic functions, although all searches provided the most optimal plan, in terms of efficiency, the ignore preconditions heuristic worked the fastest. However, unlike what we saw in the non-heuristic search functions, there isn't a correlation between the time elapsed and the number of expanded nodes and goal tests required to get to the plan. The ignore preconditions heuristic expanded more nodes and tested for the goal more than the planning level sum heuristic yet it was able to cut time in coming up with a plan. This is in line with what was discussed in Peter Norvig and Stuart Russell's Artificial Intelligence: A Modern Approach textbook. The level sum heuristic is inadmissible while the ignore preconditions heuristic has the potential to overestimate. The higher accuracy of the level sum heuristic may be the reason for the speedier solution.

On the other hand, the lack of speed in the level sum heuristic may potentially be caused by the inefficiencies of the execution of the heuristic function than the heuristic function themselves. If the level sum heuristic function was developed to run at the speed of the ignore preconditions heuristic, the level sum heuristic may be able to produce a plan even faster than that of the ignore preconditions heuristic function.

The heuristic search functions performed better overall in comparison to its non-heuristic counterparts. Despite a slightly longer run time, it consistently produced the optimal plan. Within the given data set, the heuristic search function ignore preconditions seems the obvious choice for each of these problems. However, considering the number of expansions the level sum heuristic produced, I think the level sum function can be engineered to reduce the time it takes to come up with a value and become the most optimal and efficient function for these problems.

Heuristic Analysis
Masa Watanabe

Optimal Solutions:

Problem 1:

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

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

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

Search Results:

Non-Heuristic

Problem	Search Function	Exp	Goal Tests	New Nodes	Time Elapse	Len
air_cargo_p1	breadth_first_search	43	56	180	0.029	6
air_cargo_p2	breadth_first_search	3343	4609	30509	14.685	9
air_cargo_p3	breadth_first_search	14663	18098	129631	95.897	12
air_cargo_p1	depth_first_graph_search	21	22	84	0.013	20
air_cargo_p2	depth_first_graph_search	624	625	5602	3.256	619
air_cargo_p3	depth_first_graph_search	408	409	3364	1.658	392
air_cargo_p1	greedy_best_first_graph_search	7	9	28	0.006	6
air_cargo_p2	greedy_best_first_graph_search	990	992	8910	2.696	21
air_cargo_p3	greedy_best_first_graph_search	5614	5616	49429	15.367	22

Heuristic Analysis
Masa Watanabe

Heuristic

Problem	Search Function	Exp	Goal Tests	New Nodes	Time Elapse	Len
air_cargo_p1	Astar_search h_1	55	57	224	0.037	6
air_cargo_p2	Astar_search h_1	4852	4854	44030	11.076	9
air_cargo_p3	Astar_search h_1	18235	18237	159716	50.102	12
air_cargo_p1	Astar_search h_ignore_preconditions	41	43	170	0.037	6
air_cargo_p2	Astar_search h_ignore_preconditions	1450	1452	13303	4.030	9
air_cargo_p3	Astar_search h_ignore_preconditions	5040	5042	44944	17.842	12
air_cargo_p1	Astar_search h_pg_levelsum	11	13	50	0.486	6
air_cargo_p2	Astar_search h_pg_levelsum	86	88	841	47.048	9
air_cargo_p3	Astar_search h_pg_levelsum	318	320	2934	201.667	12