Heuristic Analysis

Part 1

air_cargo_p1

	breadth_first_search	depth_first_graph_search	uniform_cost_search
number of node expansions required	43	12	55
number of goal tests	56	13	57
time elapsed (seconds)	0.06364409137670042	0.017920003267464985	0.07676354676030482
optimality	Plan length 6 Load(C2, P2, JFK) Load(C1, P1, SFO) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)	Plan length 12 Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Load(C1, P2, SFO) Fly(P2, SFO, JFK) Fly(P1, JFK, SFO) Unload(C1, P2, JFK) Fly(P2, JFK, SFO) Fly(P1, SFO, JFK) Load(C2, P1, JFK) Fly(P2, SFO, JFK) Fly(P1, JFK, SFO) Unload(C2, P1, SFO)	Plan length: 6 Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)

air_cargo_p2

	breadth_first_search	depth_first_graph_search	uniform_cost_search
number of node expansions required	3343	582	4853
number of goal tests	4609	583	4855
time elapsed (seconds)	14.82378517584687	4.133117471565579	21.932443338109326
optimality	Plan length: 9 Load(C2, P2, JFK) Load(C1, P1, SFO) Load(C3, P3, ATL) Fly(P2, JFK, SFO)	Plan length: 575 （too long to include）	Plan length: 9 Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P1, SFO, JFK)

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

air_cargo_p3

	breadth_first_search	depth_first_graph_search	uniform_cost_search
number of node expansions required	14663	627	18151
number of goal tests	18098	628	18153
time elapsed (seconds)	69.38180583599879	4.175201764139353	87.36087290910503
optimality	Plan length: 12 Load(C2, P2, JFK) Load(C1, P1, SFO) 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)	Plan length: 596 （too long to include）	Plan length: 12 Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Fly(P1, ATL, JFK) Unload(C1, P1, JFK) Unload(C2, P2, SFO) Unload(C3, P1, JFK) Unload(C4, P2, SFO)

Summary

From the above metrics, it's obvious that depth_first_graph_search always runs the fastest, but with much longer plan steps, which I think is not optimal in really life when it comes to execution (e.g. a lot of unnecessary fly, load and unload actions) compared to the other two algorithms.

Part 2

air_cargo_p1

	astar_search, h_ignore_preconditions	astar_search, h_pg_levelsum
number of node expansions required	41	11

number of goal tests	43	13
time elapsed (seconds)	0.06972565723017005	0.981207720836266
optimality	Plan length 6 Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)	Plan length 6 Load(C1, P1, SFO) Fly(P1, SFO, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)

air_cargo_p2

	astar_search, h_ignore_preconditions	astar_search, h_pg_levelsum
number of node expansions required	1450	86
number of goal tests	1452	88
time elapsed (seconds)	7.158673177278015	150.19270473599568
optimality	Plan length 9 Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO)	Plan length 9 Load(C1, P1, SFO) Fly(P1, SFO, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO) Unload(C3, P3, SFO)

air_cargo_p3

	astar_search, h_ignore_preconditions	astar_search, h_pg_levelsum
number of node expansions required	5038	314

number of goal tests	5040	316
time elapsed (seconds)	26.449234006396345	855.2740893227676
optimality	Plan length 12 Load(C1, P1, SFO) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Unload(C2, P2, SFO) Unload(C3, P1, JFK) Unload(C4, P2, SFO)	Plan length 12 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(C1, P1, JFK) Unload(C2, P2, SFO) Unload(C3, P1, JFK) Unload(C4, P2, SFO)

Summary

While both the heuristics with A* search achieve the right and succinct results, the A* search with the "ignore preconditions" (A1) heuristic runs almost 20 times faster than the A* search with the "level-sum" (A2) heuristic on these three problems. Without doubt the former is a better solution. However, if we compare A1 to the non-heuristic planning methods that we've explored in part 1 of this document, you see that for problem 1, the breadth first search and uniform cost search perform equally well. But on problem 2 and 3, A1 is the obvious winner in terms of speed. Hence, it's clear that non-heuristic search methods could work well in some simple problems such as problem 1, however, A* search with appropriate heuristic works much better in solving more complex problems.