AIR CARGO PLANNING ANALYSIS

Chiranjeeb Kataki

This review summarize the metrics for three uninformed searches, and three A* heuristic searches. We try to compare and contrast optimal solutions for three air cargo problems of increasing complexity, and look at the benefits of a heuristic-based solution for complex problems.

AIR CARGO PROBLEM 1:

TYPE	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	TIME
Breadth-first	43	56	180	6	0.1335s
Depth-first graph	12	13	48	12	0.0362s
Uniform cost	55	57	224	6	0.1619s
A* - h_1	55	57	224	6	0.1636s
A* - h_ignore_preconditions	41	43	170	6	0.1415s
A* - h_pg_levelsum	11	13	50	6	2.0528s

For the first problem, we have two airports, two planes, and two cargo units. For a simple problem like this, we see that all searches perform reasonably (with the exception of depth-first graph search). The A* search with planning graph performs efficiently in terms of expansions and goal tests, but takes 10x more time. The best solution for simpler problems is therefore just a breadth-first search.

AIR CARGO PROBLEM 2:

TYPE	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	TIME
Breadth-first	3343	4609	30509	9	55.82s
Depth-first graph	582	583	5211	575	11.00s
Uniform cost	4852	4854	44030	9	61.70s
A* - h_1	4852	4854	44030	9	61.98s
A* - h_ignore_preconditions	1450	1452	13303	9	19.76s
A* - h_pg_levelsum	86	88	841	9	341.5s

For the second problem, we have three airports, three planes, and three cargo units. Because of the added complexity, we now see the limitations of a breadth-first search. Depth-first search is still good in terms of time taken, but doesn't come anywhere near to an optimal solution. However, we see that a planning graph heuristic is still much slower (albeit 5x this time), though we start to see the benefits of a heuristic-based approach. The best solution here is the A* search with the ignore-preconditions heuristic.

AIR CARGO PROBLEM 3:

TYPE	EXPANSIONS	GOAL TESTS	NEW NODES	PLAN LENGTH	TIME
Breadth-first	14663	18098	129631	12	354.5s
Depth-first graph	627	628	5176	596	12.15s
Uniform cost	18235	18237	159716	12	270.9s
A* - h_1	18235	18237	159716	12	277.9s
A* - h_ignore_preconditions	5040	5042	44944	12	81.40s
A* - h_pg_levelsum	318	320	2934	12	~30min

For the third problem, we now have four airports, two planes, and four cargo units. This is a much more complex problem, and we see that breadth-first search is no longer optimal. Depth-first search still finds a solution very quickly, but is nowhere near optimal again. The planning graph heuristic ran for 30min before finding a solution; most likely because it is an overly complex heuristic. The best solution here is the A* search with the ignore-preconditions heuristic.

OPTIMAL SOLUTIONS:

The following table summarizes the optimal solution for each problem:

PROBLEM	SEARCH TYPE	TIME	ACTION SEQUENCE
Air cargo problem 1	Breadth-first	0.1335s	1. Load(C2, P2, JFK) 2. Load(C1, P1, SFO) 3. Fly(P2, JFK, SFO) 4. Unload(C2, P2, SFO) 5. Fly(P1, SFO, JFK) 6. Unload(C1, P1, JFK)
Air cargo problem 2	A* with ignore-preconditions	19.76s	1. Load(C3, P3, ATL) 2. Fly(P3, ATL, SFO) 3. Unload(C3, P3, SFO) 4. Load(C1, P1, SFO) 5. Fly(P1, SFO, JFK) 6. Unload(C1, P1, JFK) 7. Load(C2, P2, JFK) 8. Fly(P2, JFK, SFO) 9. Unload(C2, P2, SFO)
Air cargo problem 3	A* with ignore-preconditions	81.40s	1. Load(C2, P2, JFK) 2. Fly(P2, JFK, ORD) 3. Load(C4, P2, ORD) 4. Fly(P2, ORD, SFO)

			5. Unload(C4, P2, SFO) 6. Load(C1, P1, SFO) 7. Fly(P1, SFO, ATL) 8. Load(C3, P1, ATL) 9. Fly(P1, ATL, JFK) 10. Unload(C3, P1, JFK) 11. Unload(C1, P1, JFK) 12. Unload(C2, P2, SFO)

CONCLUSION:

Overall, we can see that breadth-first search is a good workhorse for simple to medium complexity problems; it finds a optimal solution, without an excessive running time. As the complexity ramps up, we see better performance from the heuristic-based searches. Extrapolating from the data, we may assume that the ignore-preconditions heuristic would perform adequately for medium to high complexity problems.