Air Cargo Planning Heuristic Analysis

Here are the results of running uninformed and heuristic based planning searches for the air cargo planning problem. The goal is to review the results of each of the planning searches and find the optimal solution for each of the problems i.e. lowest path to transport a cargo from one airport to the other. The optimal solution for each problem is highlighted in Green.

Results of the tests

Problem 1

Optimal Path

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

Metrics

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed (sec)	Optimal
breadth_first_search	43	56	180	6	0.44	T
depth_first_graph_search	21	22	84	20	0.02	F
uniform_cost_search	55	57	224	6	0.05	T
h_ignore_preconditions	41	43	170	6	0.6	T
h_pg_levelsum	11	13	50	6	1.3	T

Problem 2

Optimal Path

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)

Metrics

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed (sec)	Optimal
breadth_first_search	3343	4609	30509	9	18.69	T
depth_first_graph_search	624	625	5602	619	4.52	F
uniform_cost_search	4852	4854	44030	9	16.1	T
h_ignore_preconditions	1450	1452	13303	9	5.75	T
h_pg_levelsum	86	88	841	9	249.26	T

Problem 3

Optimal Path

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)

Metrics

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed (sec)	Optimal
breadth_first_search	14663	18098	129631	12	137.34	T

depth_first_graph_search	408	409	3364	392	2.26	F
uniform_cost_search	18235	18237	159716	12	71.55	T
h_ignore_preconditions	4951	4953	44051	12	20.82	T
h_pg_levelsum					> 10 mins	F

Analysis of the Results

It was seen that all the non-heuristic based searches namely Breadth First Search(BFS), Depth First Graph Search(DFS) and Uniform Cost Search(UCS) were able to find solutions to all the problems in a reasonable amount of time. BFS and UCS were able to find optimal solutions as well with minimum plan lengths for each of the problems as well.

DFS was able to find solutions but they were very non optimal because it keeps digging deeper in the graph to look for a solution without evaluating its optimality. However, the number of expansions, goal tests and new nodes were least for DFS. This means that it must have left the least memory footprint.

For Problem 1, the best solution was found in minimum time by UCS.

Once the complexity of the problem started increasing, A-star based *h_ignore_preconditions* outperformed the non-heuristic searches. So, the best solution for Problem 2 and Problem 3 were found by h_ignore_preconditions in least time.

Level-sum heuristic was able to find an optimal solution for Problem 1 and 2 within 10 minutes but was very slow because of the complexity of the heuristic.

It looks like non-heuristic based simple searches are a good starting point to do a planning search as they lead to reasonable results. But once the complexity increases, we should consider heuristic based searches like h_ignore_preconditions to find optimal results in a fast manner.