Air Cargo Planning Heuristic Analysis

by Sekaran, Harish Malan

For this project, we implemented a planning search agent to solve deterministic logistics planning problems for an Air Cargo transport system. We use a planning graph and automatic domain-independent heuristics with A* search and compare their results/performance against several uninformed non-heuristic search methods (breadth-first, depth-first, etc.).

Problem 1: Run A* planning searches using the heuristics you have implemented on air_cargo_p1, air_cargo_p2 and air_cargo_p3. Provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm and include the results in your report.

Air_Cargo_P1 **Plan length : 6**	**Air_Cargo_P2** **Plan length: 9**	**Air_Cargo_P3** **Plan length: 12**
At(C1, P1, SFO) At(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	At(C1, P1, SFO) At(C2, P2, JFK) At(C3, P3, ATL) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Unload(C2, P2, SFO) Unload(C1, P1, JFK)	At(C1, P1, SFO) At(C2, P2, JFK) Fly(P1, SFO, ATL) At(C3, P1, ATL) Fly(P2, JFK, ORD) At(C4, P2, ORD) Fly(P2, ORD, SFO) Fly(P1, ATL, JFK) Unload(C4, P2, SFO) Unload(C3, P1, JFK) Unload(C2, P2, SFO) Unload(C1, P1, JFK)

Problem 2:

Data Collection using run_search.py

Air Cargo Problem 1:

Search Type	Plan Length	Optimal	Expansions	Goal Tests	New Nodes	Execution Time
Breadth First Search	6	Yes	43	56	180	0.0452
breadth_first_tree_search	6	Yes	1458	1459	5960	1.245
depth_first_graph_search	20	No	21	22	84	0.0186
depth_limited_search	50	No	101	271	414	0.138
uniform_cost_search	6	Yes	55	57	224	0.051
recursive_best_first_search	6	Yes	4229	4230	17023	3.25
greedy_best_first_graph_search	6	Yes	7	9	28	0.006
astar_search	6	Yes	55	57	224	0.045
astar_search h_ignore_preconditions	6	Yes	41	43	170	0.049
astar_search h_pg_levelsum	6	Yes	11	13	50	0.919

Air Cargo Problem 2:

Search Type	Plan Length	Optimal	Expansions	Goal Tests	New Nodes	Execution Time
Breadth First Search	9	Yes	3343	4609	30509	17.677
uniform_cost_search	9	Yes	4852	4854	44030	14.72
greedy_best_first_graph_search	21	No	990	992	8910	3.22
astar_search	9	Yes	4852	4854	44030	14.74
astar_search h_ignore_preconditions	9	Yes	1450	1452	13303	5.94
astar_search h_pg_levelsum	9	Yes	86	88	841	80.2

Air Cargo Problem 3:

Search Type	Plan Length	Optimal	Expansions	Goal Tests	New Nodes	Execution Time
Breadth First Search	12	Yes	14663	18098	129631	134.844
uniform_cost_search	12	Yes	18264	13236	159707	13.449
greedy_best_first_graph_search	22	Yes	5605	5607	49360	22.529
astar_search	12	No	18234	18236	159707	73.65
astar_search h_ignore_preconditions	12	Yes	5040	5042	44944	24.49
astar_search h_pg_levelsum	12	Yes	325	327	3002	420.43

With this 3-problem set, **Breadth First Search** and **Uniform Cost Search** are the only two uninformed search strategies that **yield an optimal action plan** under the 10mn time limit. When it comes to execution speed and memory usage, **Depth First Graph Search** is the **fastest and uses the least memory**. However, it does not generate an optimal action plan

All three non-heuristic search strategies, that is; breadth first search, uniform cost search, and depth first graph search, find a solution to all air cargo problems. Breadth first search always considers the shortest path first [1] and a result of it it finds a solution to the problem in a reasonable amount of time and in an optimal way. Depth first graph search does find a quick solution and requires a small amount of memory, but it lacks optimality. It is not optimal because it does not consider if a node is better than another, it simply explores the nodes that take it as deep as possible in the graph even if the goal is to its right [1].
Non-heuristic based search did perform better in problem 1 and 2, which suggest that when working with simple problems using a more elaborated approach, such a A* search with heuristics, is not worth the increase in the solution complexity. Heuristic based search did perform better as the problem complexity increased. This is more evident in the air cargo problem 3, where the "A* Search with 'h_ignore_preconditions'" performance was optimal and the fastest amongst those that were optimal. It's also worth noting that the 'h_pg_levelsum' heuristic did in overall perform poorly, most likely due to the heuristic being too complex. According to the results obtained in this analysis, the breadth first search strategy can solve planning problems both fast and optimality, which makes it a good candidate to start off an analysis when dealing with search planning problems. As the complexity of the problems increase, it might be worth to consider if a heuristic based approach such as "A* Search with 'h_ignore_preconditions'" can outperform breadth first search and thus be used instead.

References

1. Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).