Planning Heuristic Analysis

By: Michael Hintz

The optimal plan length for problem 1 is 6, problem 2 is 9, and problem 3 is 12. We are able to evaluate the speed of the search algorithm very easily with the time it takes to return a solution, and we can use the number of expansions to indirectly determine how much memory is used in the search.

OPTIMAL PLANS

Problem	Optimal Plan
Air Cargo Problem 1	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 Problem 2	Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)
Air Cargo Problem 3	Load(C1, P1, SFO) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Unload(C4, P2, SFO) Unload(C3, P1, JFK) Unload(C2, P2, SFO) Unload(C1, P1, JFK)

UNINFORMED PLANNING SEARCHES

Search	Expansions	Goal Test	New Nodes	Time (ms)	Optimality	Plan Length
breadth_first_search	43	56	180	31.25	TRUE	6
depth_first_graph_search	12	13	48	10.09	FALSE	12
uniform_cost_search	55	57	224	41.39	TRUE	6

Air Cargo Problem 1

Search	Expansions	Goal Test	New Nodes	Time (s)	Optimality	Plan Length
breadth_first_search	3401	4672	31049	14.01	TRUE	9
depth_first_graph_search	350	351	3142	1.48	FALSE	346
uniform_cost_search	4761	4763	43206	11.22	TRUE	9

Air Cargo Problem 2

Search	Expansions	Goal Test	New Nodes	Time (s)	Optimality	Plan Length
breadth_first_search	14491	17947	128184	99.79	TRUE	12
depth_first_graph_search	1948	1949	16253	20.01	FALSE	1878
uniform_cost_search	17783	17785	155920	48.64	TRUE	12

Air Cargo Problem 3

The uninformed planning searches show a disputed winner in the **uniform_cost_search**.
This search performed the fastest in problem 2 and 3, however compared to the **breadth_first_search** it used slightly more memory and was actually slower for problem 1.
Since the complexity of problem 1 is lowest this makes sense.

While the **depth_first_graph_search** tended to return faster, due to the selection of a branch and evaluating it until a solution was found, it never returned an optimal solution.
I would regard speed and optimality as the most important metrics for a search of this nature, thus **uniform_cost_search** is the best algorithm of the uninformed searches.

DOMAIN-INDEPENDENT HEURISTICS (A* SEARCHES)

Search	Expansions	Goal Test	New Nodes	Time (s)	Optimality	Plan Length
astar_search h_1	55	57	224	0.04436	TRUE	6
astar_search h_ignore_preconditions	41	43	170	0.04159	TRUE	6
astar_search h_pg_levelsum	11	13	50	1.0212	TRUE	6

Air Cargo Problem 1

Search	Expansions	Goal Test	New Nodes	Time (s)	Optimality	Plan Length
astar_search h_1	4761	4763	43206	11.44	TRUE	9
astar_search h_ignore_preconditions	1450	1452	13303	4.22	TRUE	9
astar_search h_pg_levelsum	86	88	841	192.57	TRUE	9

Air Cargo Problem 2

Search	Expansions	Goal Test	New Nodes	Time (s)	Optimality	Plan Length
astar_search h_1	17783	17785	155920	50.34	TRUE	12
astar_search h_ignore_preconditions	5003	5005	44586	16.83	TRUE	12
astar_search h_pg_levelsum	311	313	2863	936.89*	TRUE	12

Air Cargo Problem 3

The domain-independent A* searches have clear winner in the **astar_search h_ignore_preconditions** algorithm. This search performed fastest on problems 1, 2, and 3 while also using less memory than the **astar_search h_1.** The **astar_search h_pg_levelsum** performed the best in terms of memory usage, but this comes at a significant cost to run time. Problem 3 actually results in a greater than 10 minute evaluation.

COMPARISON OF UNINFORMED AND INFORMED A* SEARCHES

With the exception of the **depth_first_graph_search** both sets of searches were able to solve the optimal plan for the air cargo problems. When comparing informed vs uninformed searches it becomes clear that knowing more about your problem and it's state space allows us to find efficiencies we would not normally gain with a "dumb" search algorithm.
Comparing the best uninformed search to the best informed A* search, we are able to use less memory, and get an optimal solution faster when we use a heuristic to guide our planning search.
Overall, the **astar_search h_ignore_preconditions** algorithm is the best search for the Air Cargo problems.

REFERENCES

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