Heuristic Analysis

By Martin Strandbygaard

September 10th, 2017

Search Result Metrics

The following sections provide results for solving each of the air cargo problems using both non-informed and heuristic methods.

A solution could be found for all problems with all algorithms in less than 11 minutes. Only three combinations took more than one minute to solve, and only one combination took more than approx. 2 minutes to find a solution (problem 3 with A* Search (h_pg_levelsum)).

Air Cargo Problem 1 Results

The optimal plan for air cargo problem 1 was found using the breadth first search algorithm.

Algorithm	Node Expansions	Goal Tests	New Nodes	Length	Time [s]	Optimal
Breadth-first	43	56	180	6	0.0215	Yes
Depth-first	12	13	48	12	0.0059	No
Uniform cost	55	57	224	6	0.0253	Yes
A* Search (h_1)	55	57	224	6	0.0260	Yes
A* Search (h_ignore_preconditions)	41	43	170	6	0.0255	Yes
A* Search (h_pg_levelsum)	11	13	50	6	0.7416	Yes

Air Cargo Problem 2 Results

The optimal plan for air cargo problem 2 was found using the uniform cost search algorithm. The performance of A* Search (h_1) was almost identical to that of uniform cost search, and it's difficult to determine, if the difference in measured execution time is significant or due to variance.

Algorithm	Node Expansions	Goal Tests	New Nodes	Length	Time [s]	Optimal
Breadth-first	3343	4609	30509	9	9.3092	Yes
Depth-first	582	583	5211	575	2.1052	No
Uniform cost	4853	4855	44041	9	8.0531	Yes
A* Search (h_1)	4853	4855	44041	9	8.0819	Yes
A* Search (h_ignore_preconditions)	1450	1452	13303	9	2.8961	Yes
A* Search (h_pg_levelsum)	86	88	841	9	122.76	Yes

Air Cargo Problem 3 Results

Algorithm	Node Expansions	Goal Tests	New Nodes	Length	Time [s]	Optimal

Algorithm	Node Expansions	Goal Tests	New Nodes	Length	Time [s]	Optimal
Breadth-first	14663	18098	129631	12	70.923	Yes
Depth-first	627	628	5176	596	2.228	No
Uniform cost	18164	18166	159147	12	35.469	Yes
A* Search (h_1)	18164	18166	159147	12	35.432	Yes
A* Search (h_ignore_preconditions)	5038	5040	44924	12	11.426	Yes
A* Search (h_pg_levelsum)	314	316	2894	12	632.66	Yes

Optimal Plan For Air Cargo Problems 1, 2, and 3

The following table lists the *optimal* sequence for solving each of the three air cargo problems.

Based on the results reported in the previous section, the fastest, optimal solution was chosen and the plan for this solution is listed in the table below.

Problem	Search Algorithm	Optimal Sequence
Air Cargo Problem 1	Breadth First	Load(C2, P2, JFK) Load(C1, P1, SFO) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)
Air Cargo Problem 2	A* Search (h_ignore_preconditions)	Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)
Air Cargo Problem 3	A* Search (h_ignore_preconditions)	Load(C1, P1, SFO) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C3, P1, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Unload(C4, P2, SFO) Unload(C2, P2, SFO)

Evaluation Of Search Strategies

Overall non-heuristic search algorithms performed the the best on air cargo problems 1 and 2. On problem 1 breadth-first performed the best, and uniform cost performed as well as A* Search (h_1) and A* Search (h_ignore_preconditions). This leads to the conclusion, that for the simple case in

problem 1 a non-informed algorithm seems to be the best choice, since it performs as well as a heuristic algorithm, while at the same time being a simpler solution, which is generally preferred.

For problems 2 and 3 A* Search (h_ignore_preconditions) outperformed all other algorithms, and this shows the advantages of a heuristic method, as the problem complexity increases.

For all problems, A* Search (h_pg_levelsum) performed significantly worse than the other two variants of A*. The most likely explanation for this, is that the heuristic is too complex for the given problems.

In all cases, the performance of A* Search (h_1) was almost identical to the non-informed uniform cost search algorithm.

Breadth first search is optimal and always finds the shortest path [1]. For simpler problem where the search space is small, the strategy of searching breadth first will reasonably quickly find an optimal solution.

Depth first search very quickly finds a solution, and depth first search also has the benefit of only requiring a limited amount memory because it only considers one path at a time [2]. However, depth first search is not guaranteed to find an optimal solution [1].

The intuition for this is, that depth first search, will search a path to the end. Suppose the search space is a circular graph with an odd number of nodes. Regardless of the direction in which depth first search starts, it will always find a solution, however, depending whether this solution is optimal depends on the direction the depth first search started in the graph.

Also, depth first search is not complete. The intuition being, that for a tree graph with a branch of infinite length, depth first search will continue to search in infinity, and will not find a solution [2].

Overall, for problem 1 breadth first seems to be the best approach, whereas for problem 2 and 3 A* Search (h_ignore_preconditions) is the best approach.

In all cases, if an optimal plan is not important, then depth first search is significantly faster for the three problems.

References

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

[2] https://www.youtube.com/watch?v=Fh5b8xVJhR8