Heuristic Analysis
for AIND planning project by A. Tkachenko

In this project we will evaluate and compare various search techniques to solve the air cargo planning problem. In particular, three air cargo problems of increasing complexity were used as test vehicles.

The examples of the optimal paths for each of these problems are shown below:

Problem1 (plan length = 6):
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Problem2 (plan length = 9):
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

Problem3 (plan length = 12):
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

We will start by investigating search strategies that do not utilize heuristics. Comparison of results for non-heuristic search is shown in Table 1.

Table 1. Non-heuristic search comparison

Prob. #	Search type	Expansions	Goal Tests	New Nodes	Plan Length	Time elapsed, s
1	breadth_first_search	43	56	180	6 (optimal)	0.03
	depth_first_graph_search	21	22	84	20 (non-opt)	0.014
	uniform_cost_search	55	57	224	6 (optimal)	0.036
2	breadth_first_search	3346	4612	30534	9 (optimal)	15.822
	depth_first_graph_search	107	108	959	105 (non-opt)	5.883
	uniform_cost_search	4853	4855	44041	9 (optimal)	13.516
3	breadth_first_search	14120	17673	124926	12 (optimal)	375.564
	depth_first_graph_search	292	293	2388	288 (non-opt)	4.43
	uniform_cost_search	18223	18225	159618	12 (optimal)	75.762

As we can see breadth-first and uniform cost searches yield optimal plan length, while depth-first search does not. Depth-first search however terminated faster than the other two, making it usable for a quick search for a sub-optimal solution.

For smaller problems the time difference between breadth-first and uniform search is negligible and either can be slightly faster than the other, but results for the last, larger, makes it clearer that uniform cost search finds optimal solution faster.

These search results can be improved by using informed search. A-star search with three different heuristics was used for this purpose. One is a simple heuristic that ignores preconditions for actions and just checks how many actions need to be taken to reach the goal state. The second, more complex, heuristic uses planning graph to determine the level cost of the actions and returns the sum of the level costs for actions needed to reach the goal state. Finally, a dummy heuristic (returning constant value of 1) was used as a baseline.

Table 2. Search using heuristics

Prob. #	A* heuristic	Expansions	Goal Tests	New Nodes	Plan Length	Time elapsed, s
1	h_1	55	57	224	6	0.0353
	h_ignore_preconditions	41	43	170	6	0.036
	h_pg_levelsum	11	13	50	6	3.376
2	h_1	4853	4855	44041	9	13.8
	h_ignore_preconditions	1450	1452	13303	9	5.995
	h_pg_levelsum	86	88	841	9	333.68
3	h_1	18223	18225	159618	12	61.914
	h_ignore_preconditions	5040	5042	44944	12	22.26
	h_pg_levelsum	325	327	3002	12	1214.518

The ”no preconditions” heuristic has found the optimal plan in the smallest amount of time; dummy heuristic came 2nd; while the complex level cost-based heuristic took substantially longer to solve the problem. Looking at the number of expansions and new nodes, we can see that more complex and accurate heuristics required substantially less exploration of the state space. The difference becomes more pronounced for complex problems with larger state space.

So, we can conclude that the best heuristic in both of these tables (“no preconditions”) can attribute its success to requiring substantially less space state exploration compared to a dummy heuristic (or non-heuristic search) while still being less computationally expensive compared to level cost-based one. Complexity of the problem can play a role in the selection of the best search strategy as well – for a very simple problem (such as problem 1) there is little difference between “no preconditions” heuristic, const heuristic, and some of the non-heuristic search approaches. However such simple problems are of a less practical interest compared to the more complex ones, so we can conclude that using the A* search with the “no preconditions” heuristic in general is a good strategy.