Udacity AIND Project III – Heuristic Analysis

Rongyu Lin

Part 1. Non-heuristic Planning Methods Metrics

Problem	Algorithm	Expansions	Goal Tests	New Nodes	Time	Plan Length	Optimal
1	breadth_first_search	43	56	180	0.0245s	6	Yes
1	breadth_first_tree_search	1458	1459	5960	0.7553s	6	Yes
1	depth_first_graph_search	21	22	84	0.0116s	20	No
1	depth_limited_search	101	271	414	0.0739s	50	No
1	uniform_cost_search	55	57	224	0.0335s	6	Yes
2	breadth_first_search	3343	4609	30509	11.659s	9	Yes
2	breadth_first_tree_search	N/A	N/A	N/A	>20min	N/A	N/A
2	depth_first_graph_search	624	625	5602	3.086s	619	No
2	depth_limited_search	222719	2053741	2054119	984.85s	50	No
2	uniform_cost_search	4853	4855	44041	15.502s	9	Yes
3	breadth_first_search	14663	18098	129631	150.98s	12	Yes
3	breadth_first_tree_search	N/A	N/A	N/A	>20min	N/A	N/A
3	depth_first_graph_search	408	409	3364	2.3543s	392	No
3	depth_limited_search	N/A	N/A	N/A	>20min	N/A	N/A
3	uniform_cost_search	17797	17799	156081	47.227s	12	Yes

Part 2. Heuristic Planning Methods Metrics

Problem	Algorithm	Expansions	Goal Tests	New Nodes	Time	Plan Length	Optimal
1	astar_search h_1	55	57	224	0.036s	6	Yes
1	astar_search h_ignore_preconditions	41	43	170	0.0344s	6	Yes
1	astar_search h_pg_levelsum	11	13	50	0.8269s	6	Yes
2	astar_search h_1	4853	4855	44041	10.459s	9	Yes
2	astar_search h_ignore_preconditions	1450	1452	13303	3.792s	9	Yes
2	astar_search h_pg_levelsum	86	88	841	180.16s	9	Yes
3	astar_search h_1	17797	17799	156081	51.433s	12	Yes
3	astar_search h_ignore_preconditions	5034	5036	44886	18.648s	12	Yes
3	astar_search h_pg_levelsum	313	315	2885	775.172s	12	Yes

Part 3. Written Analysis

Optimal Plans:

Problem 1.	load(C1, P1, SFO) load(C2, P2, JFK) Fly(P2, JFK, SFO) load(C2, P2, SFO) Fly(P1, SFO, JFK) load(C1, P1, JFK)	*Problem 2.*	load(C1, P1, SFO) load(C2, P2, JFK) load(C3, P3, ATL) Fly(P2, JFK, SFO) load(C2, P2, SFO) Fly(P1, SFO, JFK) load(C1, P1, JFK) Fly(P3, ATL, SFO) load(C3, P3, SFO)	*Problem 3.*	load(C1, P1, SFO) Fly(P1, SFO, ATL) load(C3, P1, ATL) Fly(P1, ATL, JFK) load(C1, P1, JFK) load(C2, P2, JFK) Fly(P2, JFK, ORD) load(C4, P2, ORD) Fly(P2, ORD, SFO) load(C2, P2, SFO) load(C3, P1, JFK) load(C4, P2, SFO)

Comparing and Contrasting non-heuristic search result metrics

Depth-first search has less number of node expansions and time spent than breadth-first search. The number and time difference is much bigger when it comes to more complexing problem. However, optimality of depth-first search is really poor. Even for the easiest problem (Problem 1), depth-first search is not able to give optimal plan.
Depth-limit search expanses much more nodes and costs much more time than depth-first search (even more than 20 minutes to solve Problem 2), yet its plan does not improve much, which is still far from the optimal one.
Breadth-first search performs pretty good in solving the problems. We can tell from the above table that it is able to give optimal plan for each problem in a rather small amount of time with less number of node expansions.
Uniform cost search also gives optimal plan for each problem, with a bit larger number of node expansions than breadth-first search. For easier problems (Problem 1 and 2), Uniform cost search costs a bit more time than breadth-first search, yet it has much less time complexity for complexing problems (Problem 3).
Therefore, for easier problems, breadth-first search should be recommended, while uniform cost search for more complexing problems.

Comparing and Contrasting heuristic search result metrics

From the table in Part 2 we can tell that heuristic searches all give optimal plans, so the only things that matter are time spent and number of node expansions.
"level-sum" heuristic performs the best in number of node expansions but worst in time spent. "ignore preconditions" heuristic has the fastest computation and a decent number of node expansions. "h_1" heuristic expanses the most number of nodes and also spends more time than "ignore preconditions". To conclude, "level-sum" has the highest time complexity but lowest space complexity, "ignore preconditions" is opposite to "level-sum", and "h_1" is in the middle of both complexities.
Therefore, if a problem is to be solved with limited space and unlimited time, "level-sum" should be recommended. When time is limited but space is not, "ignore preconditions" is recommended.

Comparing and Contrasting non-heuristic with heuristic

Heuristic search planning methods are not always better than non-heuristic ones. When solving simpler problems, non-heuristic (simpler) method, especially breadth-first search, performs better in both time and space complexity. Heuristic methods are better in solving more complexing problems, and depending on which complexity (time/space) is more important, "level-sum" or "ignore preconditions" heuristic is recommended.