Analysis for air cargo planning problems

Haoyu Ai

Problem 1

Plan Methods	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
breadth_first_search	43	56	180	6	0.0492
breadth_first_tree_search	1458	1459	5960	6	1.46
depth_first_graph_search	12	13	48	12	0.0133
depth_limited_search	101	271	414	50	0.148
uniform_cost_search	55	57	224	6	0.0547
recursive_best_first_search	4229	4230	17029	6	4.098
greedy_best_first_graph_search	7	9	28	6	0.00813
A*Search(h_1)	55	57	224	6	0.0591
A*Search(h_ignore_preconditions)	41	43	170	6	0.0605
A*Search(h_pg_levelsum)	11	13	50	6	0.641

The optimal plan is:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

Where the “greedy_best_first_graph_search” method is applied. From the table we can see this method performs best among all methods with minimal expansions, goal tests, new nodes, and shortest plan length and time elapsed. For non-heuristic search, all methods can give shortest plan length except depth-first and depth-limited search; for heuristic search using A*, heuristic factor “h_1” performs best with least time elapsed, factor "level-sum" has best efficiency with least expansions, goal tests, new nodes, and shortest plan length but slower than other two heuristic factors. As we can see, all of the methods can find a solution in a reasonable amount of time, because problem 1 is relatively simple.

Problem2

Plan Methods	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
breadth_first_search	3346	4612	30534	9	11.6
breadth_first_tree_search	/	/	/	/	/
depth_first_graph_search	1124	1125	10017	1085	11.1
depth_limited_search	/	/	/	/	/
uniform_cost_search	4853	4855	44041	9	15.9
recursive_best_first_search	/	/	/	/	/
greedy_best_first_graph_search	998	1000	8982	21	3.39
A*Search(h_1)	4853	4855	44041	9	17.9
A*Search(h_ignore_preconditions)	1450	1452	13303	9	6.22

A*Search(h_pg_levelsum)	86	88	841	9	59.6

The optimal plan is:

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)

Where A*Search(h_ignore_preconditions) method is applied. Despite its efficiency, it gives the optimal plan with shortest plan length in shortest time. Since problem 2 has more complexity, "breadth_first_tree_search", "depth_limited_search", and "recursive_best_first_search" can't give any solution even after 10 minutes, so they are skipped. For non-heuristic search, "greedy_best_first_graph_search" searches fastest but doesn't give optimal plan, only "breadth_first_search" and "uniform_cost_search" has optimal plan while the former one performs better for its higher efficiency and shorter searching time; for heuristic search using A*, heuristic "h_ignore_preconditions" outperforms others with least time elapsed, but "h_pg_levelsum" has best efficiency with much less expansions, goal tests, and new nodes.

Problem3

Plan Methods	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
breadth_first_search	14120	17673	124926	12	57.3
breadth_first_tree_search	/	/	/	/	/
depth_first_graph_search	677	678	5608	660	4.98
depth_limited_search	/	/	/	/	/
uniform_cost_search	18223	18225	159618	12	75.3
recursive_best_first_search	/	/	/	/	/
greedy_best_first_graph_search	5578	5580	49150	22	22.1
A*Search(h_1)	18223	18225	159618	12	74.9
A*Search(h_ignore_preconditions)	5040	5042	44944	12	25.1
A*Search(h_pg_levelsum)	325	327	3002	12	290

The optimal plan is:

Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)

Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Where A*Search(h_ignore_preconditions) method is applied. Among all those can give optimal plan with shortest length, it uses least time for searching. Same as problem 2, "breadth_first_tree_search", "depth_limited_search", and "recursive_best_first_search" couldn't complete the searching in 10 minutes, so they are skipped. For non-heuristic search, "depth_first_graph_search" has least searching time, but expensive solution with long plan length, among those who has optimal plan, "breadth_first_search" not only performs faster but also with more efficiency; for heuristic search using A*, again "h_pg_levelsum" is most efficient one, but with large amount of time, "h_ignore_preconditions" performs best for its fastest searching speed to give optical plan.

Question: What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

For heuristic A* search in all three problems, I think "h_ignore_preconditions" is the best heuristic, given the simplicity of its strategy, which relax the problem and only count the number of unsatisfied goals, it is optimistic and has low calculation cost. Therefore, it always can find the optimal plan with shortest action sequence at a relatively fast pace, this is why it outperforms other heuristic factors. For the second question, I think it depends on the problem's complexity, we can see for very simple problem like problem 1, the searching space is simple and small, non-heuristic search methods may win for their lower calculation cost, but when problems tends to be harder, non-heuristic methods would pay large amount of useless effort in a huge search space, while "h_ignore_preconditions" become winner because it guarantees to find a shortest action sequence with higher efficiency if solution exists.