Optimal Plans and Search Results for Problems

by Emmanuel Ik Ezenwere

A performance comparison of the the various planning search algorithms with some air cargo problems.

Air Cargo Problem 1

	Node Expansions	Goal Tests	New Nodes	Time Elapsed	plan length	Optimality
breadth_first_search	43	56	180	0.057325824978761375	6	Yes
breadth_first_tree_search	1458	1459	5960	1.0518679219821934	6	Yes
depth_first_graph_search	21	22	84	0.01610736097791232	20	No
depth_limited_search	101	271	414	0.1074302629858721	50	No
uniform_cost_search	55	57	224	0.04229367699008435	6	Yes
recursive_best_first_search h_1	4229	4230	17023	3.0306091850216035	6	Yes
greedy_best_first_graph_search h_1	7	6	28	0.0062948649865575135	6	Yes
astar_search h_1	55	57	224	0.04528658799245022	6	Yes
astar_search h_ignore_preconditions	41	43	170	0.044782328011933714	6	Yes
astar_search h_pg_levelsum	11	13	50	1.0411832180107012	6	Yes

Analysis for Air Cargo Problem 1

Optimal Plan

The plan below is the optimal plan because it accomplishes the goal; getting C1 to JFK & C2 to SFO in the fewest number of steps.

- Load(C1, P1, SFO)
- Load(C2, P2, JFK)
- Fly(P1, SFO, JFK)
- Fly(P2, JFK, SFO)
- Unload(C1, P1, JFK)
- Unload(C2, P2, SFO)

The greedy_best_first_graph_search h_1 performed best; it accomplished the goal with the minimum number of steps, node expansion and time. An interesting observation is that the uniform_cost_search algorithm performs remarkably similar to the astar_search h_1, they both have the same node expansions, goal tests, new nodes & plan length only that the uniform_cost_search was slightly faster on this problem. Of all the variations of astar_search with heuristics the variation with ignore_preconditions was fastest, that makes sense given we ignored preconditions which leads to a reduced search space.

Depth focused search algorithms such as Depth first graph search and Depth limited search took more steps than necessary to reach the Goal hence they were the only non-optimal algorithms.

Air Cargo Problem 2

	Node Expansions	Goal Tests	New Nodes	Time Elapsed(s)	Plan length	Optimality
breadth_first_search	3346	4612	30534	9.39584339101	9	Yes
breadth_first_tree_search			Time Out (>10 mins)			No
depth_first_graph_search	107	108	959	0.37344751600176096	105	No
depth_limited_search			Time Out (>10 mins)			No
uniform_cost_search	4853	4855	44041	16.815798768016975	9	Yes
recursive_best_first_search h_1			Time Out (>10 mins)			No
greedy_best_first_graph_search h_1	998	1000	8982	2.945279206993291	20	No
astar_search h_1	4853	4855	44041	14.31560180100496	9	Yes
astar_search h_ignore_preconditions	1450	1452	13303	5.383887563017197	9	Yes
astar_search h_pg_levelsum			Time Out (>10 mins)			No

Analysis fo Air Cargo Problem 2

Optimal Plan

The plan below is the optimal plan because it accomplishes the goal; getting C1 to JFK, C2 to SFO & C3 to SFO in the fewest number of steps.

- 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)

The astar_search ignore preconditions performed best,. it accomplished the goal with the minimum number of steps, node expansion and time. An surprising observation is that the greedy_best_first_graph_search algorithm took more steps than necessary to solve the goal and yet was able to reach the goal in a relatively short time. I would conclude that the greedy_best_first_graph_search algorithm is a really fast algorithm.

Air Cargo Problem 3

	Node Expansions	Goal Tests	New Nodes	Time Elapsed(s)	Plan length	Optimality
breadth_first_search			Time Out (>10 mins)			No
breadth_first_tree_search			Time Out (>10 mins)			No
depth_first_graph_search	292	293	2388	1.4583688709826674	288	No
depth_limited_search			Time Out (>10 mins)			No
uniform_cost_search	18075	18077	158390	66.34240781498374	12	Yes
recursive_best_first_search h_1			Time Out (>10 mins)			No
greedy_best_first_graph_search h_1	5084	5086	45059	17.663082393992227	22	No
astar_search h_1	18075	18077	158390	60.07788403297309	12	Yes
astar_search h_ignore_preconditions	5029	5031	44840	19.628851997025777	12	Yes
astar_search h_pg_levelsum			Time Out (>10 mins)			No

Analysis fo Air Cargo Problem 3

Optimal Plan

The plan below is the optimal plan because it accomplishes the goal; getting C1 & C3 to JFK, and C2 & C4 to SFO in the fewest number of steps.

- 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)

The astar_search ignore preconditions performed best,. it accomplished the goal with the minimum number of steps, node expansion and time. I would say that the ignore preconditions heuristics gives the astar_search algorithm a performance boost as is evident from the table. It performs faster that all other variants of astar_search.

Justification for the obtained results

astar_search with the ignore preconditions heuristic had the best general performance, the ignore preconditions heuristic enabled a faster search due to a smaller search space. It never overestimated the Goal unlike the greedy_best_first_graph_search with h_1 which usually would accomplish a sub-goal, undo this progress before finally accomplishing the goal, this lead to a more than necessary path length