

Heuristic Analysis

Air Cargo Problem 1:

<i>Search</i>	<i>Expansions</i>	<i>Goal Tests</i>	<i>New Nodes</i>	<i>Plan length</i>	<i>Time(s)</i>
breadth_first_search	43	56	180	6	0.035
depth_first_graph_search	21	22	84	20	0.015
uniform_cost_search	55	57	224	6	0.041
astar_search h_ignore_preconditions	41	43	170	6	0.032
astar_search h_pg_levelsum	11	13	50	6	1.83

For a simple problem like this with two cargos, two planes and two airports, the performance's comparable between the first four search algorithms. A* level-sum's more efficient with less expansions, goal tests and new nodes. However, this comes at a cost as it takes 1.83 seconds compared to 0.062 execution time (mean) for others.

I'd recommend breadth first search for a simple problem like this.

Air Cargo Problem 2:

<i>Search</i>	<i>Expansions</i>	<i>Goal Tests</i>	<i>New Nodes</i>	<i>Plan length</i>	<i>Time(s)</i>
breadth_first_search	3343	4609	30509	9	9.14
depth_first_graph_search	624	625	5602	619	3.46
uniform_cost_search	4853	4855	44041	9	11.47
astar_search h_ignore_preconditions	1450	1452	13303	9	4.24
astar_search h_pg_levelsum	86	88	841	9	162.54

This problem's more complex with three cargos, three planes and three airports. Depth first search is the fastest but has a sub-optimal plan length. Breadth first search's outperformed by A* ignore preconditions algorithm - likely due to the fact that adding a heuristic reduces the possibilities that need to be considered. This's evident by the lesser number of expansions: 1450 vs 3343. This performance gap should widen as the problem complexity increases.

I'd recommend A* ignore preconditions for a problem like this.

Air Cargo Problem 3:

<i>Search</i>	<i>Expansions</i>	<i>Goal Tests</i>	<i>New Nodes</i>	<i>Plan length</i>	<i>Time(s)</i>
breadth_first_search	14663	18098	129631	12	40.54
depth_first_graph_search	408	409	3364	392	1.69
uniform_cost_search	17797	17799	156081	12	51.74
astar_search h_ignore_preconditions	5034	5036	44886	12	20.32
astar_search h_pg_levelsum	314	316	2894	12	approx. 15 min

This is a more complex problem with four cargos, two planes and four airports. Similar to problem 2's performance, bread first search is penalized for the higher number of expansions (when compared to algorithms with heuristics), which leads to a longer time. Depth first search continues to be fast, but yields sub-optimal results. As mentioned before, as complexity increases, heuristics become more important. A* level-sum is the most efficient but the slowest taking 15 minutes.

I'd recommend A* ignore preconditions for a problem like this.

Optimal Solutions:

Air Cargo Problem 1 (using breadth_first_search in 0.03s):

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

Air Cargo Problem 2 (using astar_search h_ignore_preconditions in 4.24s):

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

Air Cargo Problem 3 (using astar_search h_ignore_preconditions in 20.32s):

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