Air Cargo Problem Search algorithm Analysis

metrics for non-heuristic planning solution searches

Air Cargo Problem 1: 2 cargos, 2 planes, 2 airports, 2 goals					
search functions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
breadth_first_search	43	56	180	6	0.0332
breadth_first_tree_search	1458	1459	5960	6	1.0179
depth_first_graph_search	12	13	48	12	0.0102
depth_limited_search	101	271	414	50	0.1013
uniform_cost_search	55	57	224	6	0.0479
astar_search h_1	55	57	224	6	0.0399
astar_search h_ignore_preconditions	41	43	170	6	0.0543
astar_search h_pg_levelsum	55	57	224	6	3.3985

Air Cargo Problem 2: 3 cargos, 3 planes, 3 airports, 3 goals					
search functions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
breadth_first_search	3343	4609	30509	9	14.395
breadth_first_tree_search	aborted				
depth_first_graph_search	1669	1670	14863	1444	13.435
depth_limited_search	aborted				
uniform_cost_search	4853	4855	44041	9	46.869
astar_search h_1	4853	4855	44041	9	42.949
astar_search h_ignore_preconditions	1506	1508	13820	9	13.567
astar_search h_pg_levelsum	86	88	841	9	122.697

Air Cargo Problem 3: 4 cargos, 2 planes, 4 airports, 4 goals					
search functions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
breadth_first_search	14663	18098	129631	12	105.607
breadth_first_tree_search	aborted				
depth_first_graph_search	592	593	4927	571	3.161
depth_limited_search	aborted				
uniform_cost_search	18233	18235	159697	12	348.119
astar_search h_1	18233	18235	159697	12	367.201
astar_search h_ignore_preconditions	5188	5120	45650	12	83.312
astar_search h_pg_levelsum	405	407	3724	12	901.016

Algorithm Analysis

For the 3 air cargo problem use ACP1,2,3 for short.

1,graph search exploring much less node compare to tree search, cause in graph search,newly generated nodes that match previously generated nodes—ones in the
explored set or the frontier—be discarded instead of being added to the frontier, graph search also take much less time. as it shown in ACP2 and ACP3, tree
search been aborted after waiting is long.

2, Breadth_first_search use the graph search idea in it’ s function implementation, In AP1-3,Breadth_first_search find the optimal plan also take elapsed is small, Depth_limited_search and depth_first_graph_search doesn’ t have a good performance in finding the optional plan-length, some failed even as the state space become larger, Like in ACP3, plan length for depth_first_graph_search is 571, which can not be used in real problem situation even it’ t time elapsed is very short. Uniform_cost_search gives comparable performances, but in non-heuristic search category breadth-first-search would be the best choice among those search methods.

3, The ignore pre-conditions heuristic drops all preconditions from actions. Every action becomes applicable in every state, and any single goal fluent can be achieved in one step. While the level sum heuristic, following the subgoal independence assumption, returns the sum of the level costs of the goals. In ACP1-3, Astar_search h_ignore_preconditions and astar_search h_pg_levelsum, both can find the optimal plan length for ACP1-3, level sum heuristic search expands much less node but also takes much more time compare to ignore preconditions heuristic search. So Astar_search h_ignore_preconditions will be best choice in heuristic search category.

4, Compare all search algorithms, for ACP1, the best choice will be breadth-first-search, Astar_search h_ignore_preconditions also a good choice but for limited states like ACP1, breadth-first-search will be more fit.