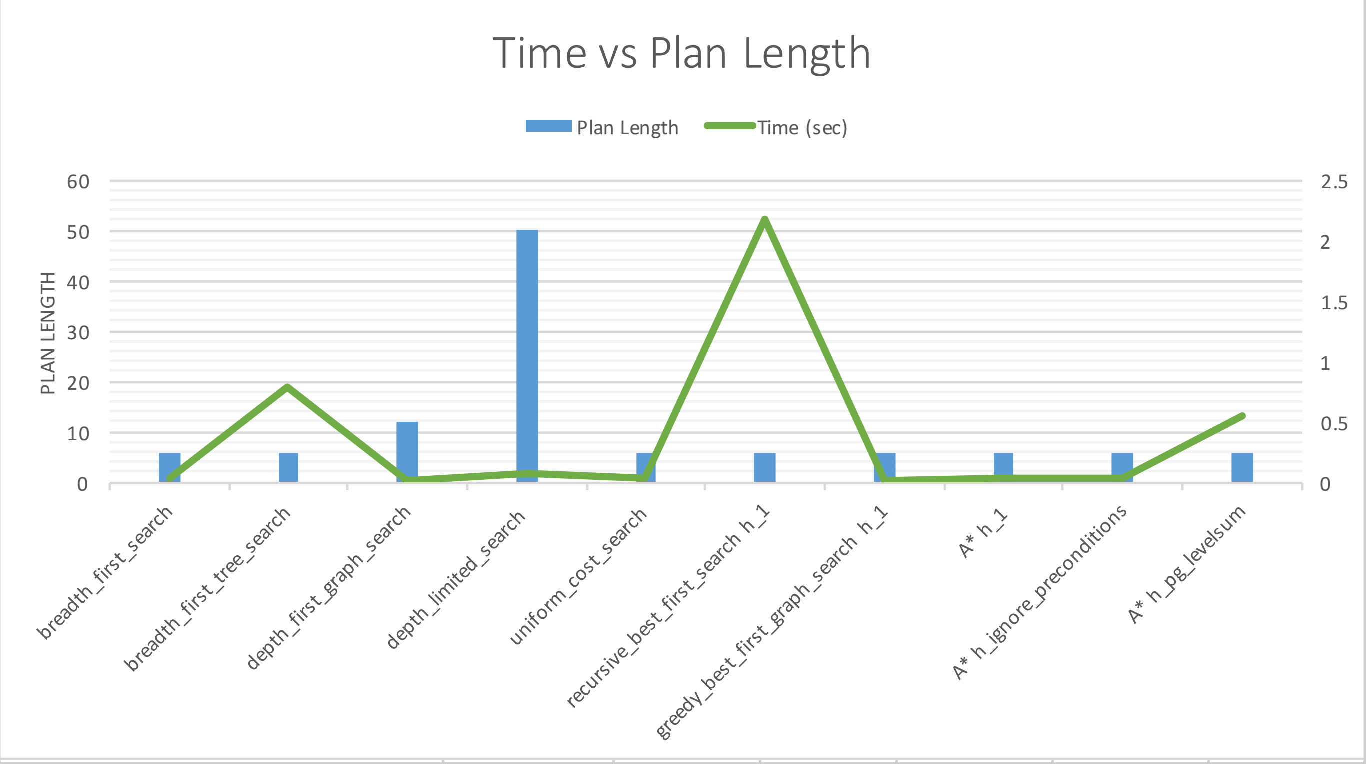
Air Cargo Problem Heuristic Analysis

Problem 1:

Search Algorithm	Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (sec)
breadth_first_search	non-heuristic	43	56	180	6	0.029
breadth_first_tree_search		1458	1459	5960	6	0.783
depth_first_graph_search		12	13	48	12	0.008
depth_limited_search		101	271	414	50	0.081
uniform_cost_search		55	57	224	6	0.033
recursive_best_first_search h_1		4229	4230	17029	6	2.184
greedy_best_first_graph_search h_1		7	9	28	6	0.004
A* h_1		55	57	224	6	0.033
A* h_ignore_preconditions	heuristic	41	43	170	6	0.024
A* h_pg_levelsum		11	13	50	6	0.545

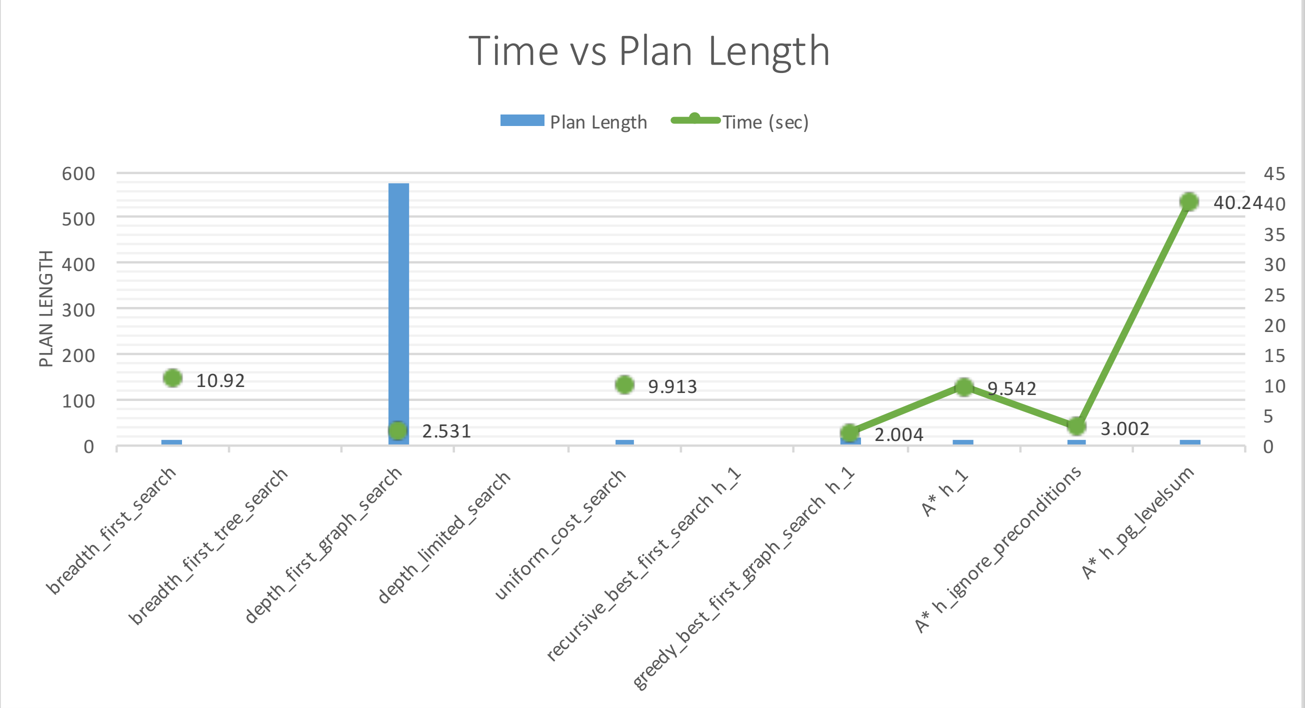


All the search algorithms were able to find the solution in this case. recursive_best_first_search h_1 alone took more than 2 seconds to find the solution. However, all the search algorithms except depth_first_graph_search and depth_limited_search were able to find the solution in under one second and a plan length of 6.

The best search algorithm for this problem was ***greedy_best_first_graph_search h_1.*** It solved the problem in 0.004 seconds with 7 expansions.

Problem 2:

Search Algorithm	Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (sec)
breadth_first_search	non-heuristic	3343	4609	30509	9	10.92
breadth_first_tree_search	non-heuristic	too long				
depth_first_graph_search	non-heuristic	582	583	5211	575	2.531
depth_limited_search	non-heuristic	too long				
uniform_cost_search	non-heuristic	4853	4855	44041	9	9.913
recursive_best_first_search h_1	non-heuristic	too long				
greedy_best_first_graph_search h_1	non-heuristic	998	1000	8982	17	2.004
A* h_1	non-heuristic	4853	4855	44041	9	9.542
A* h_ignore_preconditions	heuristic	1450	1452	13303	9	3.002
A* h_pg_levelsum	heuristic	86	88	841	9	40.24

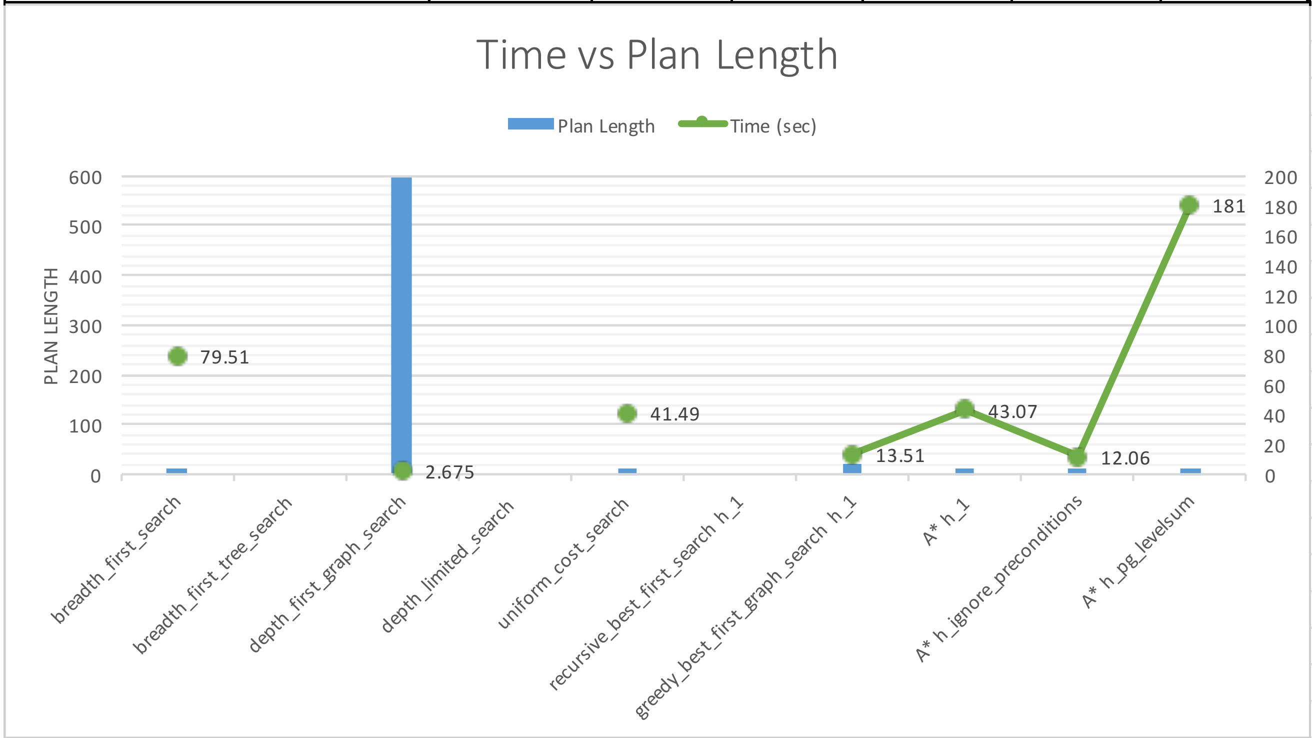


In case of problem 2, breadth_first_tree_search, depth_limited_search and recursive_best_first_search h_1 were not able to find the solution in under 10 minutes. Apart from depth_first_graph_search, greedy_best_first_graph_search h_1 and above mentioned three search algorithms all the other search algorithms were able to find the solution with a plan length of 9.

The best algorithm for this problem was A* h_ignore_preconditions which got to the solution in 3.002 seconds with 1450 expansions. breadth_first_search got to the solution in 10.92 seconds with 3343 expansions. uniform_cost_search and A* h_1 were the next best after the above two search algorithms.

Problem 3:

Search Algorithm	Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (sec)
breadth_first_search	non-heuristic	14663	18098	129631	12	79.51
breadth_first_tree_search	non-heuristic	too long				
depth_first_graph_search	non-heuristic	627	628	5176	596	2.675
depth_limited_search	non-heuristic	too long				
uniform_cost_search	non-heuristic	18223	18225	159618	12	41.49
recursive_best_first_search h_1	non-heuristic	too long				
greedy_best_first_graph_search h_1	non-heuristic	5579	5581	49159	22	13.51
A* h_1	non-heuristic	18223	18225	159618	12	43.07
A* h_ignore_preconditions	heuristic	5040	5042	44944	12	12.06
A* h_pg_levelsum	heuristic	317	319	2925	12	181



In case of problem 3, similar to problem 2 breadth_first_tree_search, depth_limited_search and recursive_best_first_search h_1 were not able to find the solution in under 10 minutes. Apart from depth_first_graph_search, greedy_best_first_graph_search h_1 and above mentioned three search algorithms all the other search algorithms were able to find the solution with a plan length of 12.

The best algorithm for this problem was A* h_ignore_preconditions which got to the solution in 12.06 seconds with 5040 expansions. breadth_first_search got to the solution in 79.51 seconds with 14663 expansions. Similar to problem 2 uniform_cost_search and A* h_1 were the next best after the above two search algorithms.

Observations:

- In case of small problems all the search algorithms were able to find the solution but as the problem gets larger breadth_first_tree_search, depth_limited_search and recursive_best_first_search h_1 were not able to find the optimal solution in limited time.
- breadth_first_search performed well but A* h_ignore_preconditions beat the algorithm by finding the optimal plan with least expansions. However, breadth_first_search is the best non-heuristic search algorithm.
- The heuristic approach got to the optimal plan with less number of expansions compared to non-heuristic approach.
- The above tests were conducted on Macbook Pro [i7, 2.8 GHz and 16 GB RAM]