Heuristic analysis

Plan search run stats

1. Air Cargo problem 1

Search	Expansion	Goal test	New nodes	Plan length	Time
breadth_first_search	43	56	180	6	0.0367849220019707
breadth_first_tree_search	1458	1459	5960	6	1.1180712259993015
depth_first_graph_search	12	13	48	12	0.011657097998977406
depth_limited_search	101	271	414	50	0.10886529499839526
uniform_cost_search	55	57	224	6	0.04530639400036307
recursive_best_first_search with h_1	4229	4230	17029	6	3.263332430000446
greedy_best_first_graph_search with h_1	7	9	28	6	0.007309614000405418
astar_search with h_1	55	57	224	6	0.045446995998645434
astar_search with h_ignore_preconditions	41	43	170	6	0.04675115500140237
astar_search with h_pg_levelsum	11	13	50	6	1.1896226000026218

Optimal Plan Length

6

Optimal Plan

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

2. Air Cargo problem 2

Search	Expansion	Goal test	New nodes	Plan length	Time
breadth_first_search	3343	4609	30509	9	10.447076509997714
breadth_first_tree_search	inf	inf	inf	inf	inf
depth_first_graph_search	582	583	5211	575	3.68269315499856
depth_limited_search	inf	inf	inf	inf	inf
uniform_cost_search	4852	4854	44030	9	15.361362896999935
recursive_best_first_search with h_1	inf	inf	inf	inf	inf
greedy_best_first_graph_search with h_1	990	992	8910	17	2.9553077760001543
astar_search with h_1	4852	4854	44030	9	14.810009461999925
astar_search with h_ignore_preconditions	1450	1452	13303	9	5.031621488000155
astar_search with h_pg_levelsum	86	88	841	9	245.46225985299998

*inf refer to too long

Optimal Plan Length

9

Optimal Plan

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

3. Air Cargo problem 3

Search	Expansion	Goal test	New nodes	Plan length	Time
breadth_first_search	14491	17947	128184	12	52.591107399000066
breadth_first_tree_search	inf	inf	inf	inf	inf
depth_first_graph_search	1948	1949	16253	1878	23.938193622999734
depth_limited_search	inf	inf	inf	inf	inf
uniform_cost_search	17783	17785	155920	12	60.425305381000726
recursive_best_first_search with h_1	inf	inf	inf	inf	inf
greedy_best_first_graph_search with h_1	4031	4033	35794	22	15.452620782999475
astar_search with h_1	17783	17785	155920	12	63.65132430700032
astar_search with h_ignore_preconditions	5003	5005	44586	12	19.68687148799927
astar_search with h_pg_levelsum	311	313	2863	12	1091.688766022

*inf refers to too long

Optimal Plan Length

12

Optimal Plan

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)

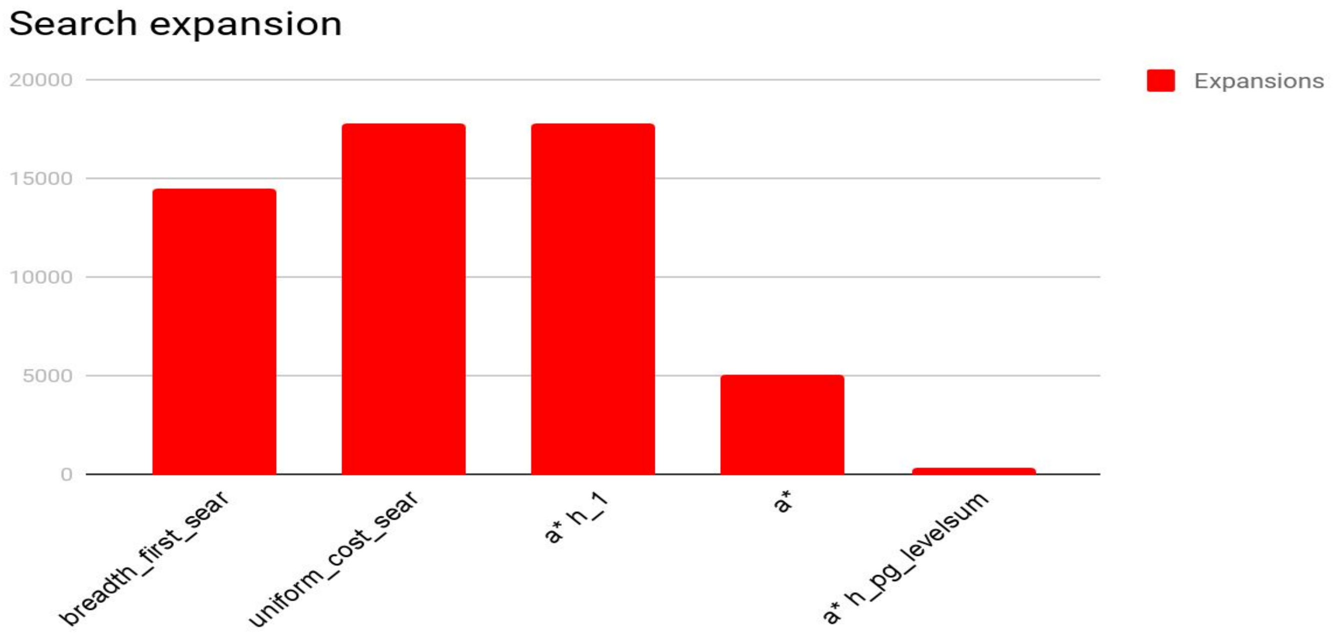
Search Strategies Discussion

All three non-heuristic search strategies, that is; breadth first search, uniform cost search, and depth first graph search, find a solution to all air cargo problems. Breadth first search always considers the shortest path first and a result of it it finds a solution to the problem in a reasonable amount of time and in an optimal way. Depth first graph search does find a quick solution and requires a small amount of memory, but it lacks optimality. It is not optimal because it does not consider if a node is better than another, it simply explores the nodes that take it as deep as possible in the graph even if the goal is to its right . Non-heuristic based search did perform better in problem 1 and 2, which suggest that when working with simple problems using a more elaborated approach, such a A* search with heuristics, is not worth the increase in the solution complexity. Heuristic based search did perform better as the problem complexity increased. This is more evident in the air cargo problem 3, where the “A* Search with ‘h_ignore_preconditions’” performance was optimal and the fastest amongst those that were optimal. It’s also worth noting that the ‘h_pg_levelsum’ heuristic did in overall perform poorly, most likely due to the heuristic being too complex. According to the results obtained in this analysis, the breadth first search strategy can solve planning problems both fast and optimality, which makes it a good candidate to start off an analysis when dealing with search planning problems. As the complexity of the problems increase, it might be worth to consider if a heuristic based approach such as “A* Search with ‘h_ignore_preconditions’” can outperform breadth first search and thus be used instead.

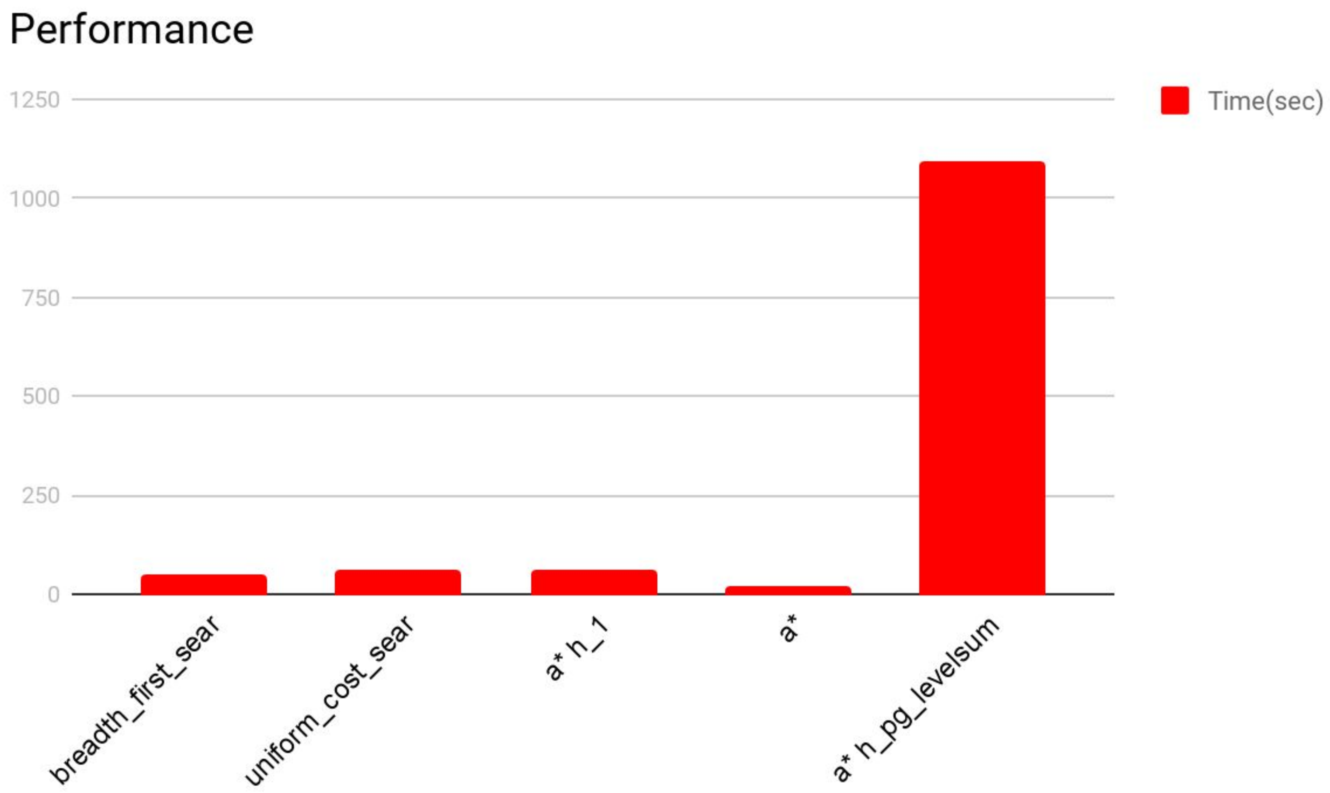
Comparison Graph

Breadth first search vs Uniform cost Search/A* Search with h1 vs A* with h_ignore precondition vs A* with h_pg_levelsum using air cargo problem 3 data

Node expansion



Performance: (in sec)


Performance
Time(sec)
breadth_first_sear
uniform_cost_sear
a* h_1
a*
a* h_pg_levelsum