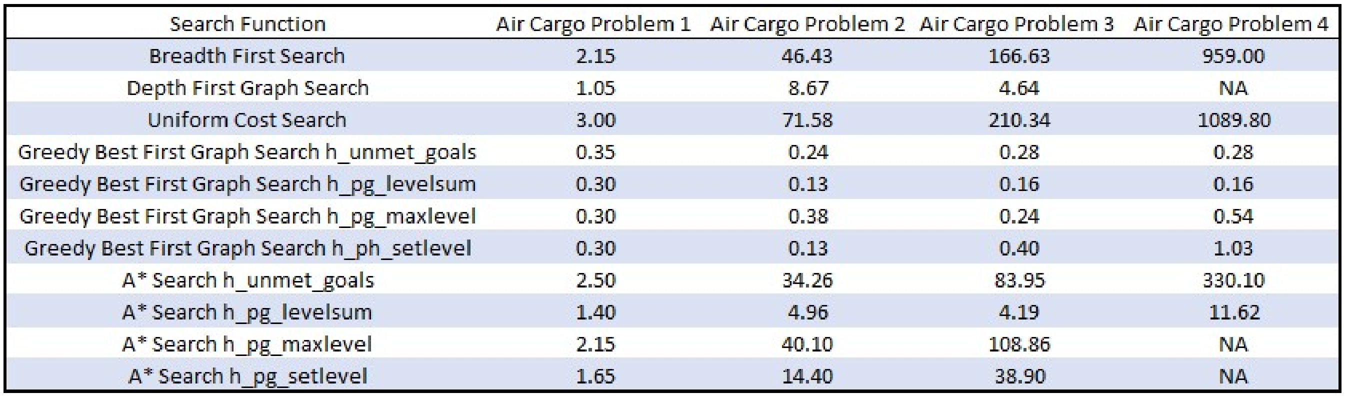
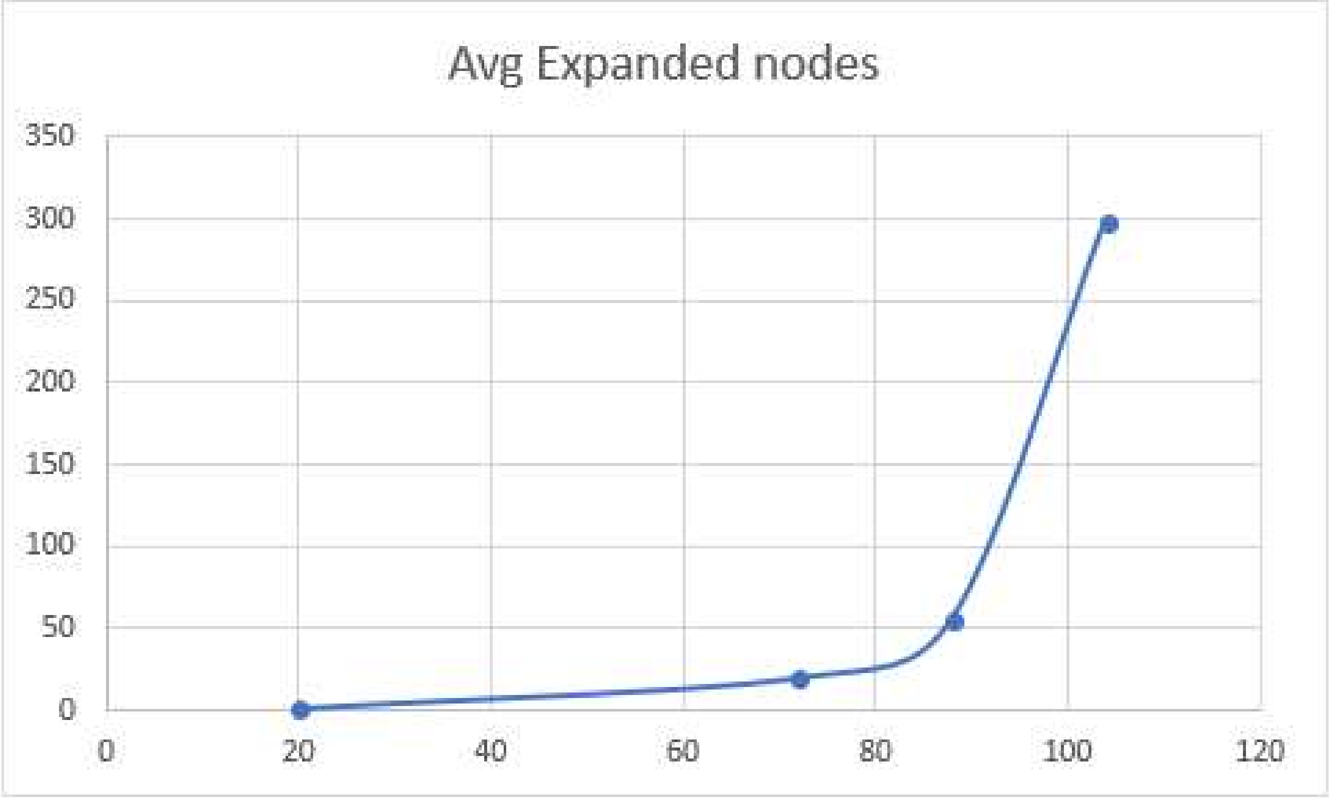
Forward Planning Agent Analysis

Number of nodes expanded against the number of actions in the domain

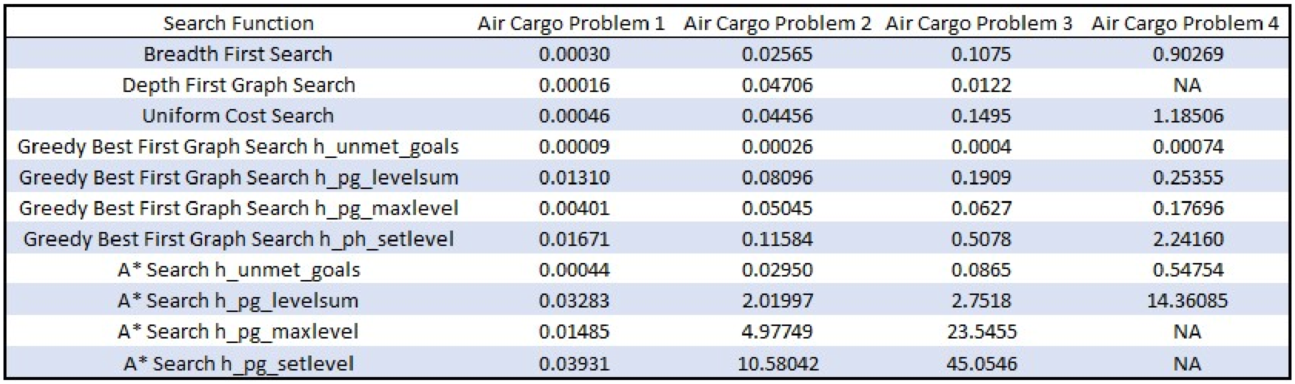


Search Function	Air Cargo Problem 1	Air Cargo Problem 2	Air Cargo Problem 3	Air Cargo Problem 4
Breadth First Search	2.15	46.43	166.63	959.00
Depth First Graph Search	1.05	8.67	4.64	NA
Uniform Cost Search	3.00	71.58	210.34	1089.80
Greedy Best First Graph Search h_unmet_goals	0.35	0.24	0.28	0.28
Greedy Best First Graph Search h_pg_levelsum	0.30	0.13	0.16	0.16
Greedy Best First Graph Search h_pg_maxlevel	0.30	0.38	0.24	0.54
Greedy Best First Graph Search h_ph_setlevel	0.30	0.13	0.40	1.03
A* Search h_unmet_goals	2.50	34.26	83.95	330.10
A* Search h_pg_levelsum	1.40	4.96	4.19	11.62
A* Search h_pg_maxlevel	2.15	40.10	108.86	NA
A* Search h_pg_setlevel	1.65	14.40	38.90	NA

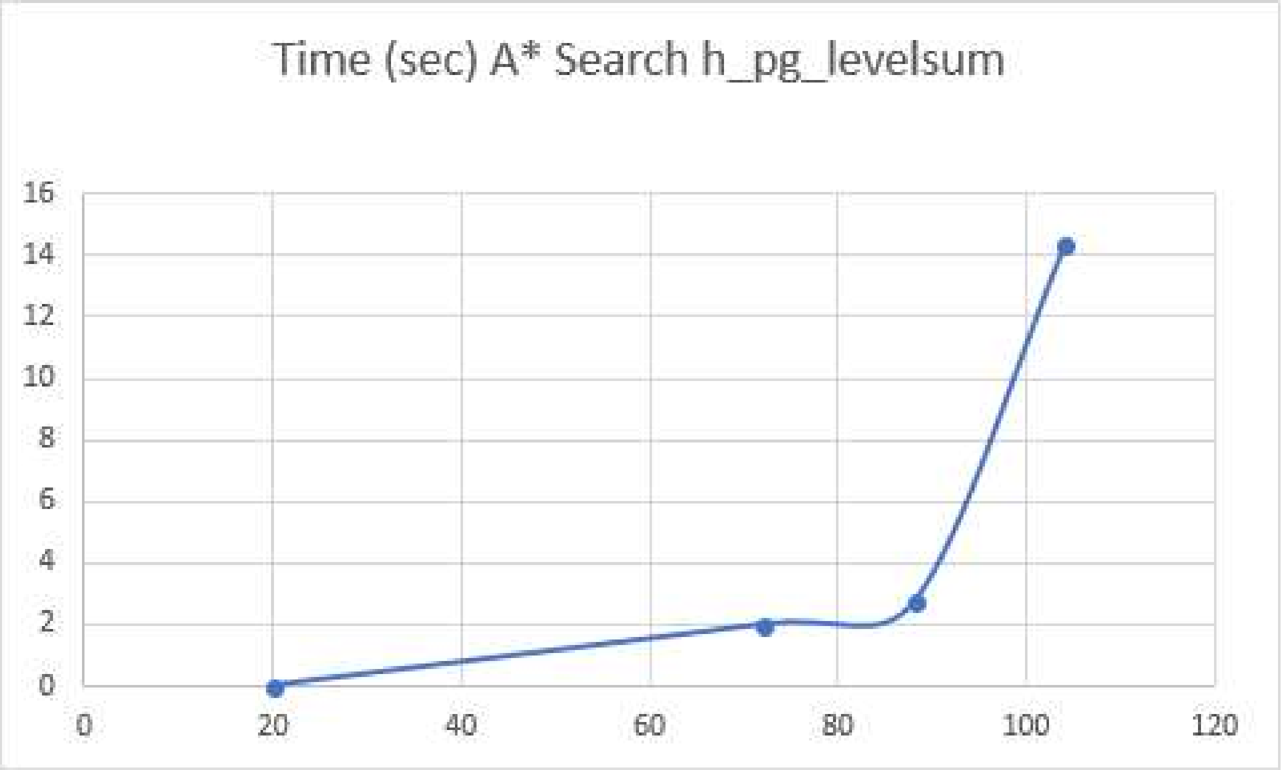


The first chart shows the number of expanded nodes per the number of actions for each problem. The second chart shows the averages of the heuristics for each problem. We can see from the second chart that as the number of actions increase the number of expanded nodes fallows an exponential curve. This is to be expected as the number of actions increase the number of possibility's also increases.

Amount of time against the number of actions for each domain

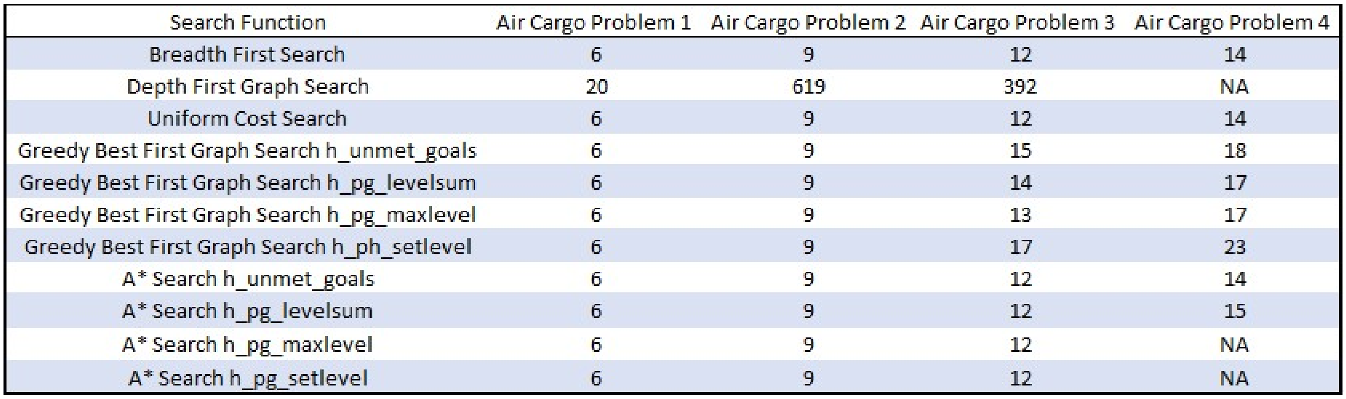


Search Function	Air Cargo Problem 1	Air Cargo Problem 2	Air Cargo Problem 3	Air Cargo Problem 4
Breadth First Search	0.00030	0.02565	0.1075	0.90269
Depth First Graph Search	0.00016	0.04706	0.0122	NA
Uniform Cost Search	0.00046	0.04456	0.1495	1.18506
Greedy Best First Graph Search h_unmet_goals	0.00009	0.00026	0.0004	0.00074
Greedy Best First Graph Search h_pg_levelsum	0.01310	0.08096	0.1909	0.25355
Greedy Best First Graph Search h_pg_maxlevel	0.00401	0.05045	0.0627	0.17696
Greedy Best First Graph Search h_ph_setlevel	0.01671	0.11584	0.5078	2.24160
A* Search h_unmet_goals	0.00044	0.02950	0.0865	0.54754
A* Search h_pg_levelsum	0.03283	2.01997	2.7518	14.36085
A* Search h_pg_maxlevel	0.01485	4.97749	23.5455	NA
A* Search h_pg_setlevel	0.03931	10.58042	45.0546	NA



The first chart shows the time is seconds for each heuristic to run for each problem. The second chart shows the time for the A star search using the planning graph level sum heuristic. This gives us a similar picture as before with the number of expanded nodes. As the number of actions increase so does the time to search through the tree of possibility's. We can expect extremely long processing time with problems with over 100 possible actions.

Length of plans for each heuristic for each problem



Search Function	Air Cargo Problem 1	Air Cargo Problem 2	Air Cargo Problem 3	Air Cargo Problem 4
Breadth First Search	6	9	12	14
Depth First Graph Search	20	619	392	NA
Uniform Cost Search	6	9	12	14
Greedy Best First Graph Search h_unmet_goals	6	9	15	18
Greedy Best First Graph Search h_pg_levelsum	6	9	14	17
Greedy Best First Graph Search h_pg_maxlevel	6	9	13	17
Greedy Best First Graph Search h_ph_setlevel	6	9	17	23
A* Search h_unmet_goals	6	9	12	14
A* Search h_pg_levelsum	6	9	12	15
A* Search h_pg_maxlevel	6	9	12	NA
A* Search h_pg_setlevel	6	9	12	NA

Summery

- The algorithm that would be most appropriate for a small service which has to run in real time would be the Greedy Best First Graph Search using the heuristic of unmet goals. This always returns a plan in the under one second for all domain sizes. Returning a plan in under one second is important for the ability to have the perception of running in real time.
- The algorithm that would be most appropriate for a large service would be the A Star Search. This search most often outputs the plan with the most efficient plan.
- The algorithm that would be most appropriate the return only optimal plans is the A Star Search. Given the data from the charts it always returns plans with the least amount of moves.