Heuristic analysis

Problem 1 involves 2 pieces of cargo, 2 airplanes and 2 airports. **Problem 2** involves 3 pieces of cargo, 3 airplanes and 3 airports**. Problem 3** involves 4 pieces of cargo, 4 airplanes and 4 airports. Hence, they are increasing in levels of complexity and the timings of the experiment reflect this reality.

Optimal plan for **Problem 1** is - **breadth_first_search**

```
D:\Sola\Udacity\AIND\AIND-Planning-master>python run_search.py -p 1 -s 1

Solving Air Cargo Problem 1 using breadth_first_search...

Expansions   Goal Tests   New Nodes
    43          56          180

Plan length: 6  Time elapsed in seconds: 0.030297248042767753
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

Optimal plan for **Problem 2** is - **greedy_best_first_graph_search with h_1**

```
D:\Sola\Udacity\AIND\AIND-Planning-master>python run_search.py -p 2 -s 7

Solving Air Cargo Problem 2 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   550         552         4950

Plan length: 9  Time elapsed in seconds: 1.2043000047802412
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

Optimal plan for **Problem 3** is - **astar_search with h_ignore_preconditions**

```
D:\Sola\Udacity\AIND\AIND-Planning-master>python run_search.py -p 3 -s 9

Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   5003        5005       44586

Plan length: 12  Time elapsed in seconds: 13.708246864300012
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)
```

In the tables and graphs, I have plotted Performance as **(1/elapsed time x 1000)** for each search method, for comparability. For optimality, I use least plan length, and if there are any ties, I use least expansion or elapsed time. Optimal searches are highlighted in green in the tables.

Findings indicated that Breadth-first and A* heuristic searches provide optimal results and depth-first does NOT guarantee optimality.

If the number of expansion is taken as a measure of performance **(i.e. 1/expansions * 1000,000)**, reflecting the intuition that the higher the expansions, the worse the performance - i.e. the search with the least number of expansions is the best performer since it is taking fewer steps before hitting the goal) it is found that this measure follows the same trend as the measure of performance as a function of elapsed time.

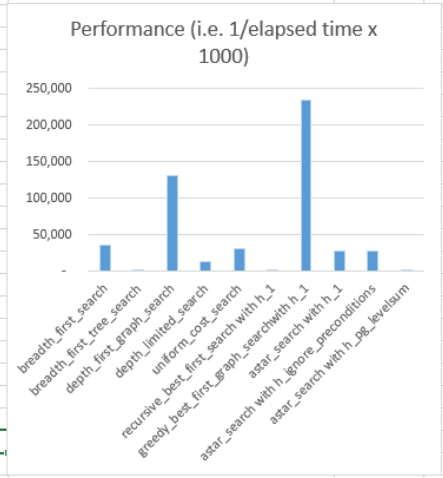
A* with level-sum heuristics vs **A* with ignore preconditions**

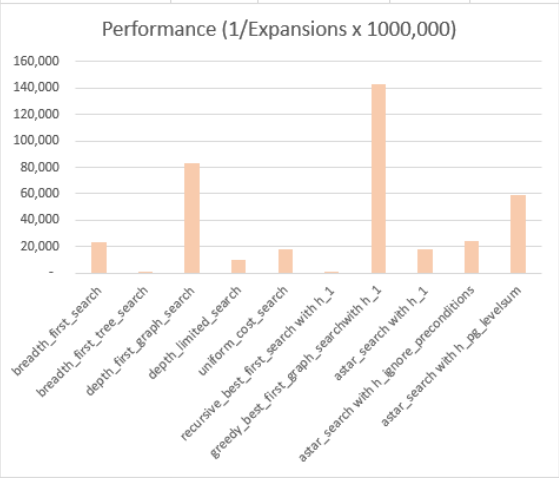
	Plan length	**Time elapsed**
Problem 1		
astar_search with h_ignore_preconditions	6	0.035815069
astar_search with h_pg_levelsum	6	0.540050718
Problem 2		
astar_search with h_ignore_preconditions	9	3.568888215
astar_search with h_pg_levelsum	10	276.214318
Problem 3		
astar_search with h_ignore_preconditions	12	13.74856524
astar_search with h_pg_levelsum	13	649.09177

In all three problems A* with ignore conditions performed better than A* with level- sum both in terms of optimality and speed.

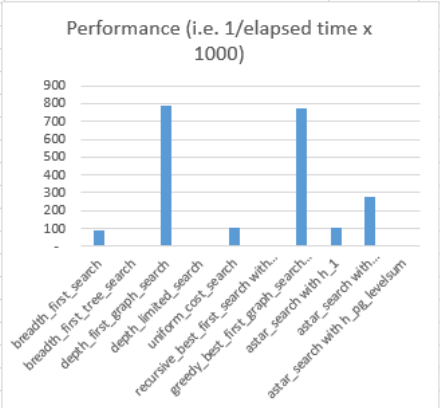
Overall from the figures, the best search method is the **astar_search with h_ignore_preconditions** because it achieves optimality in all the three problems with appropriate time performance. In some of the problem it was not the best in terms of time performance (e.g. slower that the non-heuristic **breadth_first_search** in problem 1) this is in line with the table **Comparing uninformed search strategies** in section 3.4.7 of Norvig and Russells' Artificial Intelligence - A Modern Approach, third Edition where BFS is shown to be faster than other search methods.

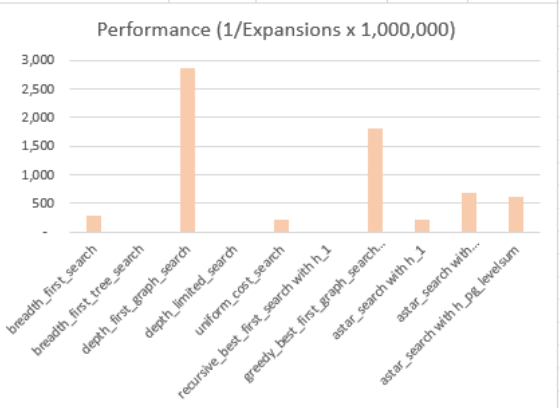
	Solving Air Cargo Problem 1							
	Serach method	Plan length	Time elapsed	Performance (1/elapsed time x 1000)	Expansions	Performance (1/Expansions x 1000)	Goal Tests	New Nodes
1	breadth_first_search	6	0.028292707	35,345	43	23,256	56	180
2	breadth_first_tree_search	6	0.804330899	1,243	1458	686	1459	5960
3	depth_first_graph_search	12	0.007681966	130,175	12	83,333	13	48
4	depth_limited_search	50	0.081726718	12,236	101	9,901	271	414
5	uniform_cost_search	6	0.033003023	30,300	55	18,182	57	224
6	recursive_best_first_search with h_1	6	2.276110685	439	4229	236	4230	17029
7	greedy_best_first_graph_searchwith h_1	6	0.004280489	233,618	7	142,857	9	28
8	astar_search with h_1	6	0.036617833	27,309	55	18,182	57	224
9	astar_search with h_ignore_preconditions	6	0.035815069	27,921	41	24,390	43	170
10	astar_search with h_pg_levelsum	6	0.540050718	1,852	17	58,824	19	71





	Solving Air Cargo Problem 2							
	Serach method	Plan length	Time elapsed	Performance (1/elapsed time x 1000)	Expansions	Performance (1/Expansions x 1,000,000)	Goal Tests	New Nodes
1	breadth_first_search	9	11.6895522	86	3401	294	4672	31049
2	breadth_first_tree_search	NO RESPONXE AFTER 10MINS						
3	depth_first_graph_search	346	1.26594576	790	350	2,857	351	3142
4	depth_limited_search	NO RESPONX						
5	uniform_cost_search	9	9.83264533	102	4761	210	4763	43206
6	recursive_best_first_search with h_1	NO RESPONX						
7	greedy_best_first_graph_search with h_1	9	1.29638365	771	550	1,818	552	4950
8	astar_search with h_1	9	9.47093633	106	4761	210	4763	43206
9	astar_search with h_ignore_preconditions	9	3.56888821	280	1450	690	1452	13303
10	astar_search with h_pg_levelsum	10	276.214318	4	1604	623	1606	14564





	Solving Air Cargo Problem 3							
	Serach method	Plan length	Time elapsed	Performance (1/elapsed time x 1000)	Expansions	Performance (1/Expansions x 1,000,000)	Goal Tests	New Nodes
1	breadth_first_search	12	84.6760679	12	14491	69	17947	128184
2	breadth_first_tree_search	NO RESPONX						
3	depth_first_graph_search	1878	16.243707	62	1948	513	1949	16253
4	depth_limited_search	NO RESPONX						
5	uniform_cost_search	12	40.7464876	25	17783	56	17785	155920
6	recursive_best_first_search with h_1	NO RESPONX						
7	greedy_best_first_graph_search with h_1	22	9.46143708	106	4031	248	4033	35794
8	astar_search with h_1	12	41.1189822	24	17783	56	17785	155920
9	astar_search with h_ignore_preconditions	12	13.7485652	73	5003	200	5005	44586
10	astar_search with h_pg_levelsum	13	649.09177	2	2911	344	2913	25097