Part 1 - Planning problems
Metrics for non-heuristic planning solution searches

Here are some metrics for the air cargo problems 1, 2 and 3.

Expansions: number of node expansions required.
Goal tests: number of goal tests.
Time Elapsed: time for the search to complete
Optimality: the plan length, number of actions to achieve all goals

Air Cargo Problem	Search Function	Expansions	Goal Tests	Time Elapsed (sec)	Optimality
1	breadth_first (1)	43	56	0.0258	6
1	depth_first_graph (3)	12	13	0.0066	12
1	uniform_cost (5)	55	57	0.0328	6
1	A* / ignore_preconditions heuristic (9)	41	43	0.0251	6
1	A* / levelsum heuristic (10)	11	13	3.9286	6
2	breadth_first (1)	3343	4609	11.5382	9
2	depth_first_graph (3)	1669	1670	12.0039	1444
2	uniform_cost (5)	4852	4854	36.7260	9
2	A* / ignore_preconditions heuristic (9)	1506	1508	10.4294	9
2	A* / levelsum heuristic (10)	86	88	1069.5043	9
3	breadth_first (1)	14663	18098	85.9720	12
3	depth_first_graph (3)	592	593	2.4736	571
3	uniform_cost (5)	18235	18237	317.9204	12
3	A* / ignore_preconditions heuristic (9)	5118	5120	68.6039	12
3	A* / levelsum heuristic (10)	NA	NA	+45 minutes	NA

Table 1

We can see that in the three problems, depth first graph search was the only function that did not find an optimal solution. According to the video lectures, depth first search goes all the way to the bottom of a branch until a dead end is met or a solution is found, then it proceeds to look into the second branch

and so on, in this case, all of the branches will yield a solution because some actions can later be reversed by certain actions in the future, but not all of these solutions will be optimal, because in a certain branch, some actions might not have been essential to reach the objective, or they could even be detrimental. Breadth first search does better in these problems because it looks at all branches up to a level, then all branches on the next level, until it finds a solution, so the solution this process yields will always be the shortest (in terms of actions executed). So in this case, even though depth first search found a solution much faster for problems 1 and 3, I would discard it because it is not optimal, the time it would take to actually do all the actions that this function suggests would take much more time than the time it takes for the other functions to find a better solution. In this situation, when we're talking about expensive and relatively long flights, finding an optimal solution that reduces the number of actions required is much more important than the time it takes for the function to calculate a solution.

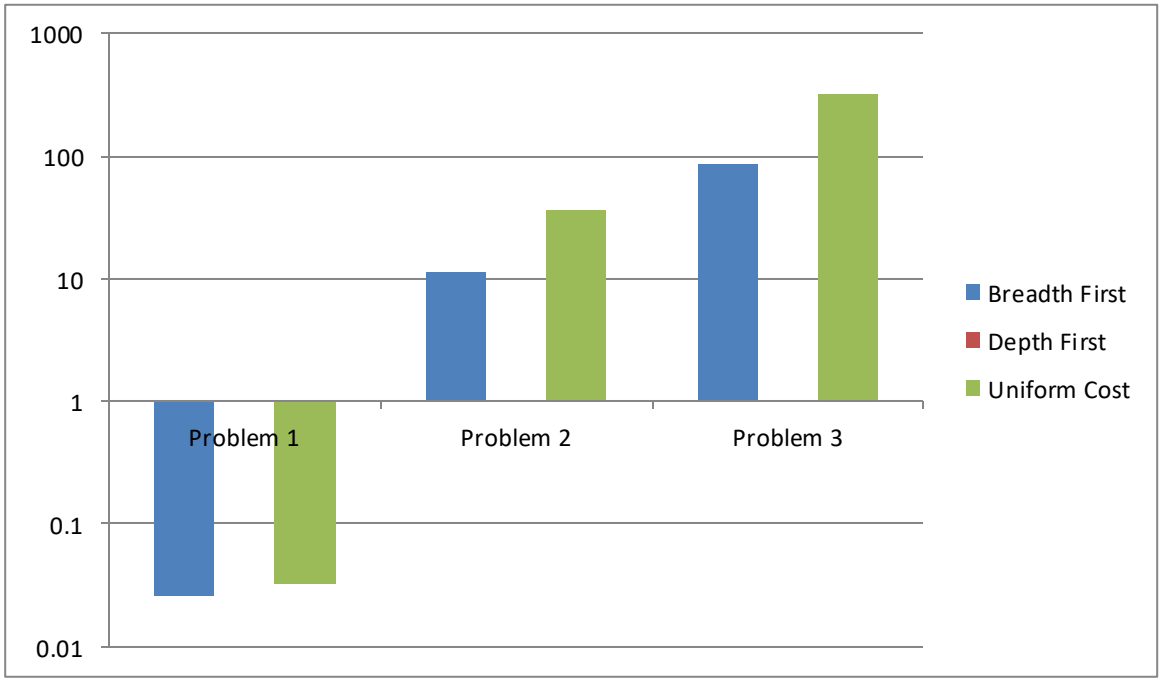

Figure 1. Elapsed Time (seconds), log 10. Non-heuristic.
Note: only searches that yielded optimal results are showed

Although both breadth first and uniform cost search functions found an optimal solution, breadth first found this solution faster, expanded fewer nodes and for the three problems executed fewer goal tests than uniform cost search. Between these three non-heuristic planning solution searches, I would definitely choose breadth first search.

To make a final decision, we need to take a look to planning graphs with search functions using domain-independent heuristics.

Part 2 - Domain-independent heuristics
Metrics of A* searches with domain-independent

Looking at the results of these heuristics in Table 1, we can say that although the levelsum heuristic had to expand much fewer nodes, the time it took for the algorithm to find a solution was many times greater. For Air Cargo Problem 3, the process had to be stopped because it had been running for more than 45 minutes. Lastly, the solutions found were equally optimal for problems 1 and 2. I would definitely choose the A* search using the ignore preconditions heuristic.

To find an optimal plan for these 3 problems, I would argue that the most important factor is reducing the plan length to a minimum, because this would imply the biggest trade off of resources. I consider that the process' time elapsed is second priority, because although we do want the optimal solution, it would offer us any advantage if the process takes more than we can afford to, for example, if we need to plan a route for the next week, but the process of searching the optimal solution with a certain search function and a certain heuristic takes about 2 weeks, then it is not viable to use it. Number of node expansions and goal tests can come into a third priority order because although these are important measures, they can be taken into account measuring the time elapsed.

For problem 1, I would argue that the optimal plan would be using an A* search with the ignore precondition heuristic, yielding the fewest nodes expanded with around 0.02 seconds of time elapsed. The plan that this process yields has length 6 and is:
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)

For problem 2, I consider that the optimal plan would be using an A* search with the ignore preconditions heuristic, yielding the fewest nodes expanded with around 11 seconds of time elapsed. The plan that this process yields has length 9 and is:
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

For problem 3, I would argue that the optimal plan would be using an A* search with the ignore preconditions heuristic, yielding the fewest nodes expanded with around 69 seconds of time elapsed. The plan that this process yields has length 12 and is:
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)

As we can see, the best searching function and heuristic was A* search with the ignore precondition heuristic. Compared to the non-heuristic search planning, in general, this heuristic using A* expanded fewer nodes, had fewer goals tested and yielded equally optimal plans in less time, which makes A* with ignore the precondition heuristic a better option than its non-heuristic counterparts.

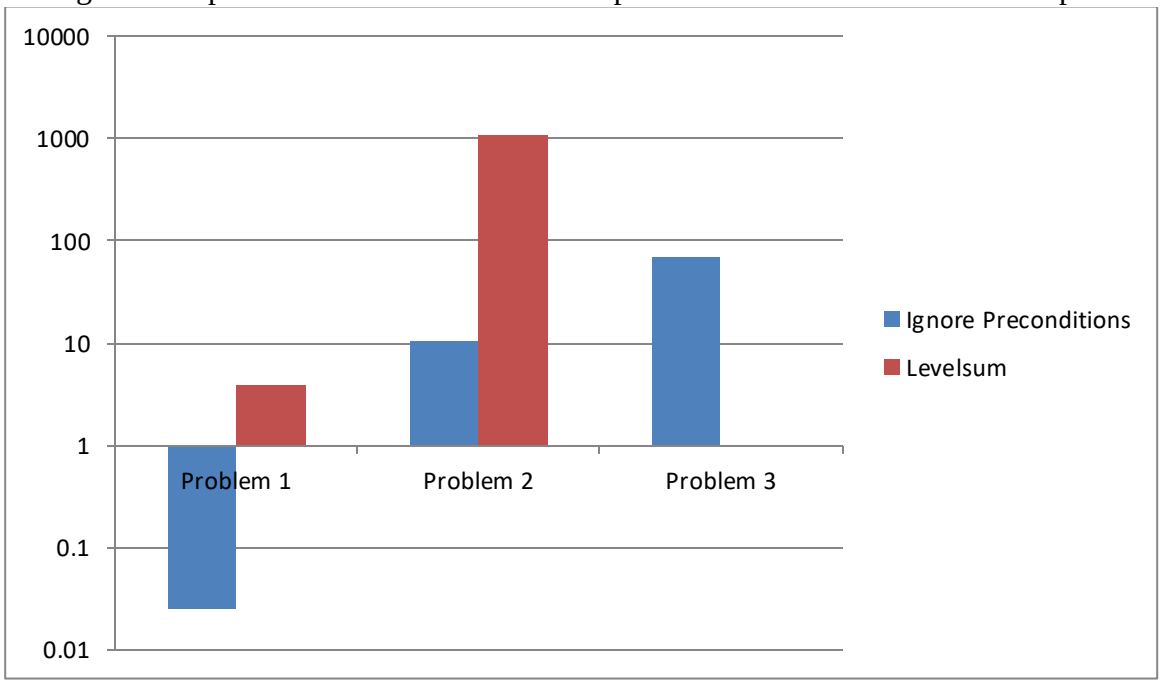

Figure 2. Elapsed Time (seconds), log 10. A search.*
Note: Problem 3 – levelsum not showed because it took too long to complete.