Christopher Verceles

Planning Project

Heuristics Analysis

After running various search functions (breadth first, depth first graph, uniform cost, level sum, ignore preconditions) on the 3 air cargo problems, I observed that depth first graph search produced the fastest, longest path length result, while A* search with the ignore preconditions heuristic produced the 2nd fastest, but shortest path length result. Level sum applied to A* search always gave the same path length as ignore preconditions with significantly less expansions, goal tests and new nodes visited, but always at the cost of significantly slower runtime.

I came up with an equation that would help to describe optimality in each case. It works by assigning a lower score to more optimal searches and a higher score to more resource intensive searches. The factors of this optimality equation, shown below, are the number of nodes expanded, number of goal tests performed, number of new nodes visited, time elapsed raised to 2.5 and the square of the resulting plan length. I put the most emphasis on the time elapsed and the resulting plan length, in order to keep a potential user's attention and to avoid longer than needed solutions.

$$OptimalityScore = Expansions + GoalTests + NewNodes + (Time^{2}.5) + (PlanLength^{2})$$

The data tables below describe my observations, along with the optimality equation used to choose the best path from among the different search technique results. You may notice that, for Air Cargo 1 the optimality equation chooses the plan of level sum instead of ignore preconditions, but this only seems to be true for relatively small search spaces, unlike those we encounter in real life. Please note, more illustrative figures follow afterwards.

Air Cargo 1

	Breadth First Search	Depth First Graph Search	Uniform Cost Search	Ignore Preconditions + A* Search	Level Sum + A* Search
Expansions	43	21	55	41	**11**
Goal Tests	56	22	57	43	**13**
New Nodes	180	84	224	170	**50**
Time Elapsed (s)	0.033	0.0185	0.043	0.0535	**0.573**
Plan Length	6	20	6	6	**6**
AirCargo1 Search Cost Score	78.75004945666	131.7500116377	93.00009585420	72.50016551008	**27.56213360663**

Optimal Plan from A* Level Sum: Load(C1, P1, SFO), Fly(P1, SFO, JFK), Load(C2, P2, JFK), Fly(P2, JFK, SFO), Unload(C1, P1, JFK), Unload(C2, P2, SFO)

Air Cargo 2

	Breadth First Search	Depth First Graph Search	Uniform Cost Search	Ignore Preconditions + A* Search	Level Sum + A* Search
Expansions	3343	624	4852	**1450**	86
Goal Tests	4609	625	4854	**1452**	88
New Nodes	30509	5602	44030	**13303**	841
Time Elapsed (s)	15.915	3.855	13.167	**4.833**	48.369
Plan Length	9	619	9	**9**	9
AirCargo2 Search Cost Score	9888.113534872	97510.294591305	13611.52405979	**4084.3375455190**	4341.7830322368

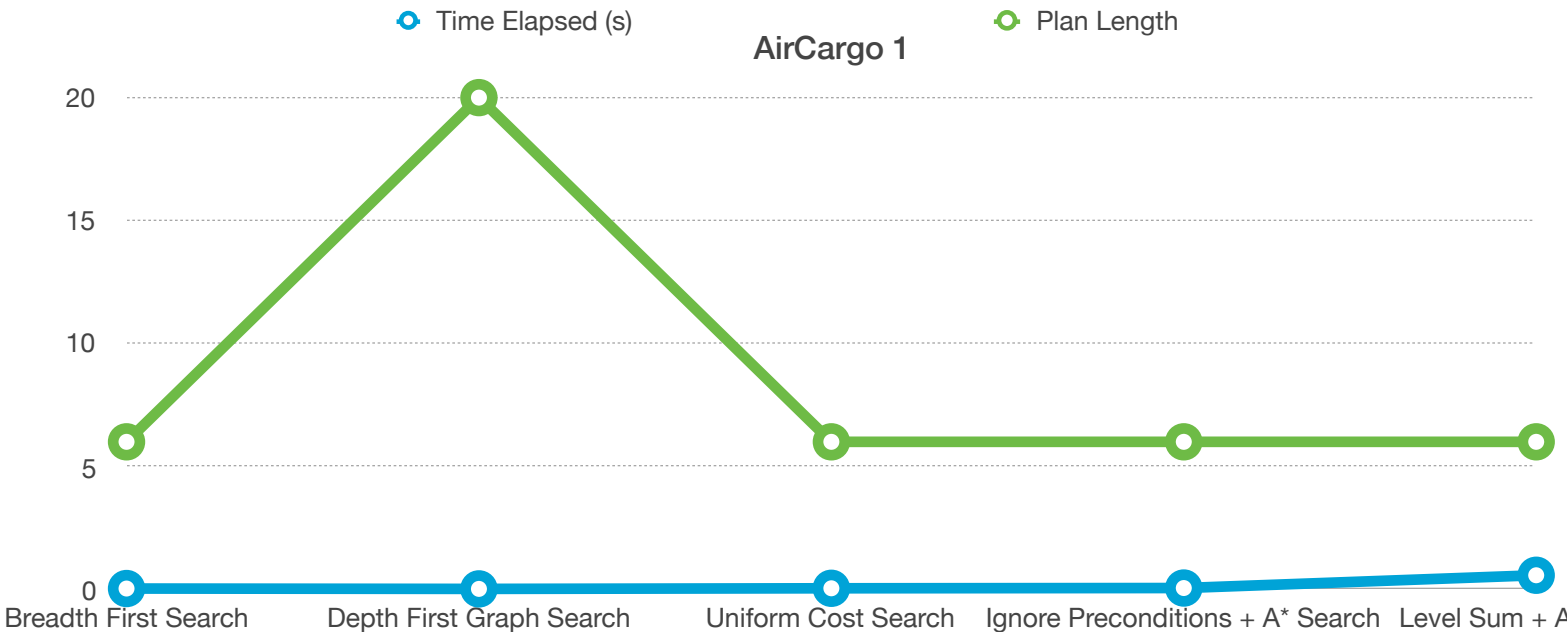
Optimal Plan from A* Ignore Preconditions: Load(C3, P3, ATL), Fly(P3, ATL, SFO), Unload(C3, P3, SFO), Load(C2, P2, JFK), Fly(P2, JFK, SFO), Unload(C2, P2, SFO), Load(C1, P1, SFO), Fly(P1, SFO, JFK), Unload(C1, P1, JFK)

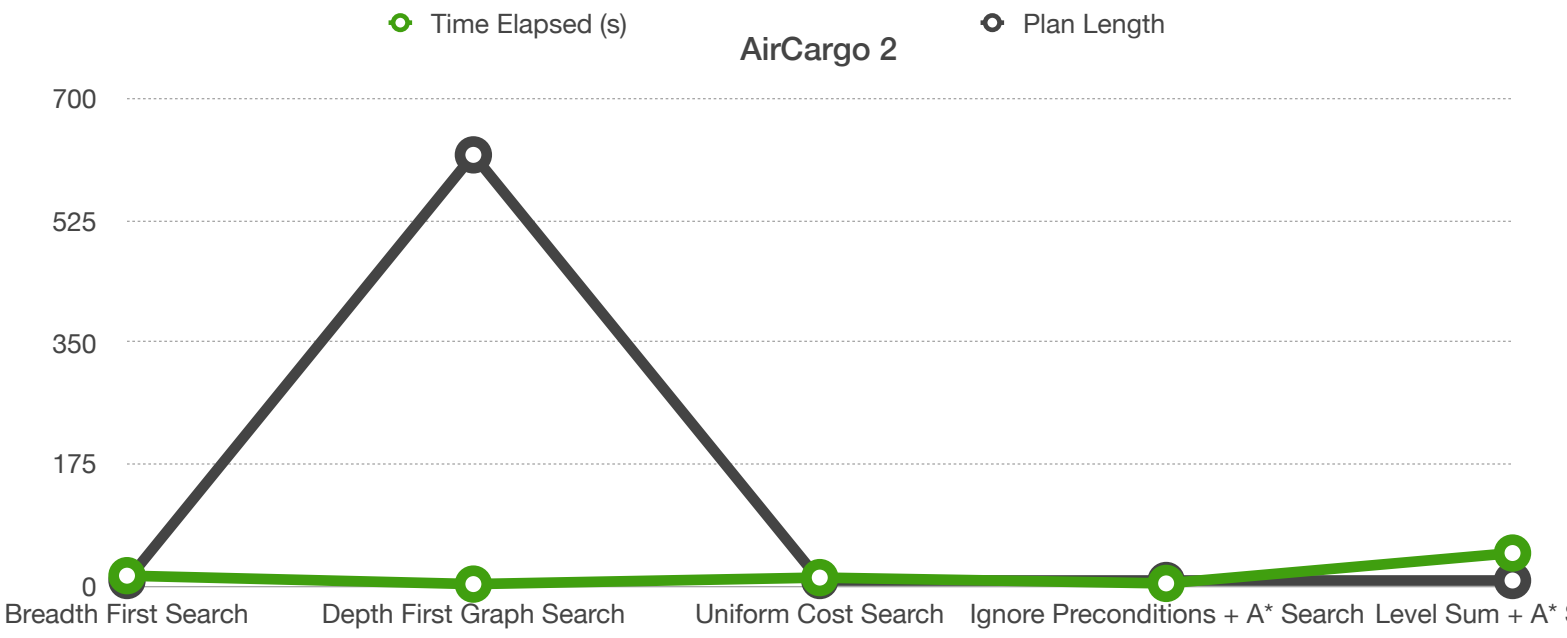
Air Cargo 3

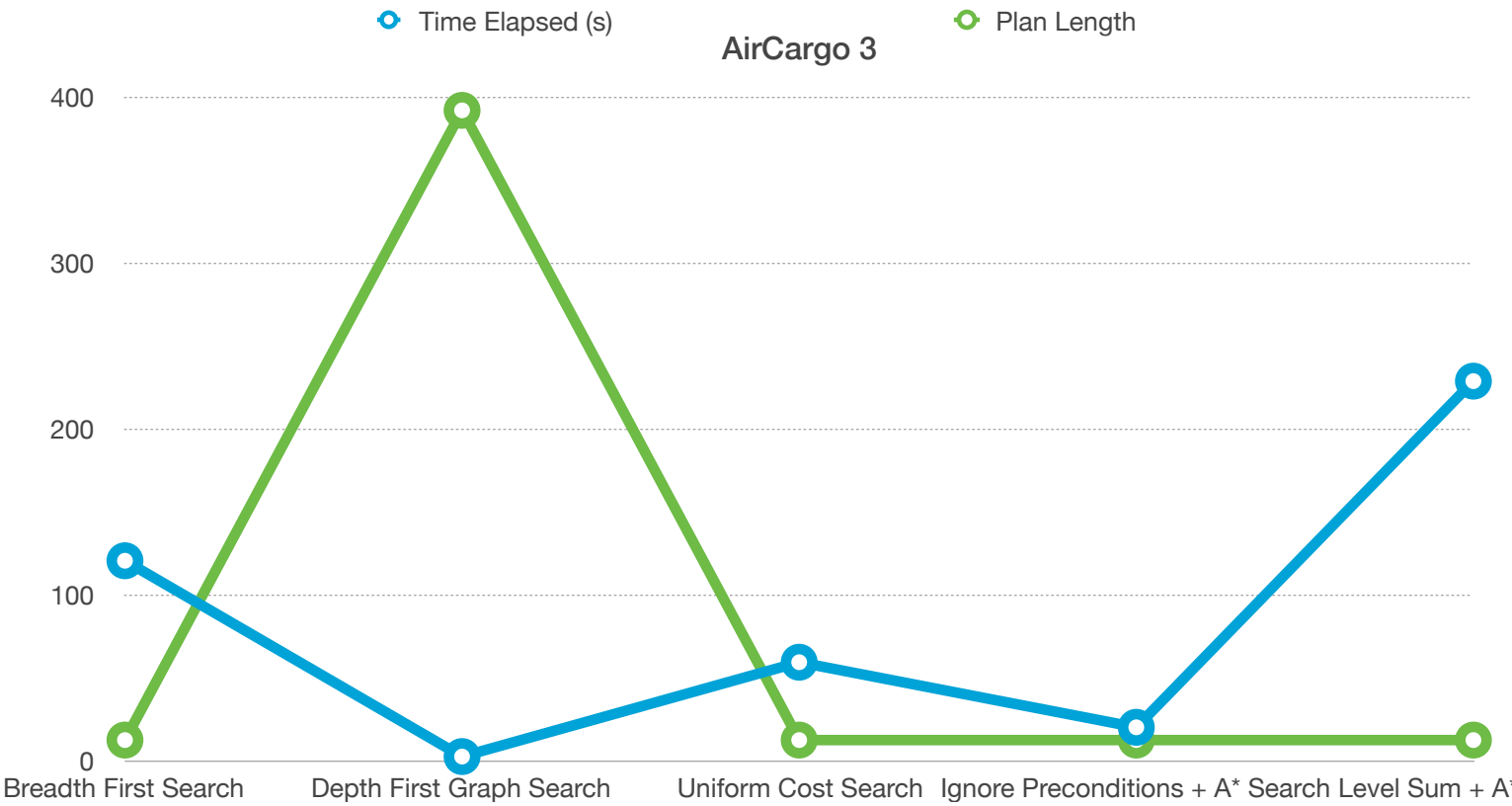
	Breadth First Search	Depth First Graph Search	Uniform Cost Search	Ignore Preconditions + A* Search	Level Sum + A* Search
Expansions	14663	408	18235	**5040**	316
Goal Tests	18098	409	18237	**5042**	318
New Nodes	129631	3364	159716	**44944**	2912
Time Elapsed (s)	120.215	2.051	58.976	**19.706**	228.604
Plan Length	12	392	12	**12**	12
AirCargo3 Search Cost Score	80246.902096735	39462.756101217	55760.721346864	**14223.459248086**	198460.00843756

Optimal Plan from A* Ignore Preconditions: 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)

The following illustrations support my optimality equation in showing that A* applied to the Ignore Preconditions Heuristic consistently produces one of the, if not the, shortest plan length(s) in the relatively short amounts of time.







In conclusion, it's easy to see that A* search applied to the ignore preconditions heuristic provides a more practical and efficient result in terms of keeping a user's attention and finding the shortest path of actions towards a certain goal in general situations. If, on the other hand, resources such as memory space become more scarce, then it would be prudent to switch to an A* search applied to the level sum heuristic at the expense of the length of time to find the same short solution. As a side note, it may be argued that depth first graph search will save the most time and space, but it seems to stop once it finds a minimum viable solution. This just returns an inefficient and long path cost.