PHILLIP DUDERO
4/27/18

AIND Project: Implement a Planning Search

Heuristic Analysis

Part 1: Uninformed Planning Searches

Uninformed planning searches were performed on three air cargo problems, where the number of planes, cargo and airports are, respectively, (2,2,2), (3,3,3), and (2,4,4). Since timing is a part of these tests, the first test I performed was a timing dependence on the environment. I determined that, on my machine at least, a python 3.6-based environment yielded substantially lower times than a python 3.5-based environment. All times reported here are from the 3.6 environment.

Tables 1-3 shows the results of these searches. Note that "h_1" is implemented as always returning a constant, and so renders the search that uses it "uninformed". This is demonstrated by the fact that "uniform cost search", which is categorized by the AIMA text as uninformed, and "astar search with h_1" always yield exactly the same metrics. Inspecting the code also proves that the underlying implementation is identical when using h_1.

Note also that "breadth first tree search" and "recursive best first search with h_1" created the most nodes and therefore took the longest time of all the algorithms. When "breadth first tree search" was tried on problem 2 (twice), it failed to complete even an overnight run. Since "recursive best first search with h1" took even longer on Problem 1, it wasn't even attempted on Problems 2 or 3. When the computing time took much longer than is feasible for this project, the algorithm was omitted.

Table 1: Uninformed Planning Search results, Air Cargo Problem 1

	expansions	goal tests	new nodes	plan length	seconds
greedy_best_first_graph_search with h_1	7	9	28	6	0.02
depth_first_graph_search	12	13	48	12	0.03
breadth_first_search	43	56	180	6	0.11
uniform_cost_search	55	57	224	6	0.14
astar_search with h_1	55	57	224	6	0.14
depth_limited_search	101	271	414	50	0.29
breadth_first_tree_search	1458	1459	5960	6	3.68
recursive_best_first_search with h_1	4229	4230	17029	6	10.52

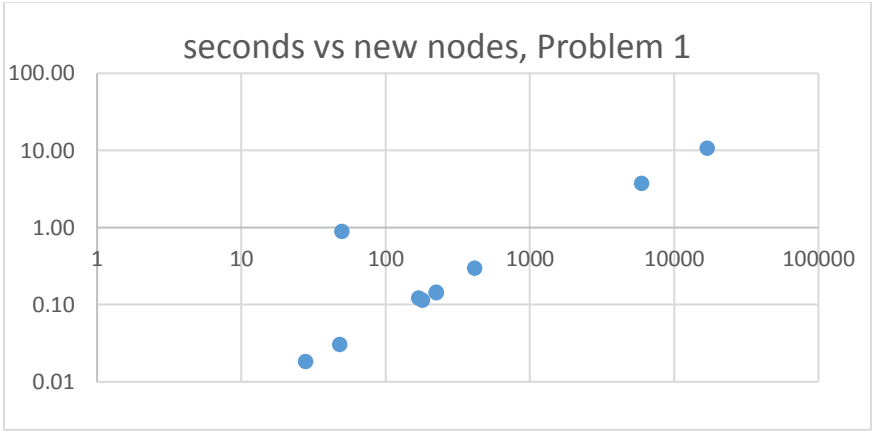
Table 2: Uninformed Planning Search results, Air Cargo Problem 2

	expansions	goal tests	new nodes	plan length	seconds
depth_first_graph_search	582	583	5211	575	9.6
greedy_best_first_graph_search with h_1	998	1000	8982	21	12.8
breadth_first_search	3343	4609	30509	9	44.1
uniform_cost_search	4853	4855	44041	9	64.0
astar_search with h_1	4853	4855	44041	9	64.6
depth_limited_search	222719	2053741	2054119	50	3593.1

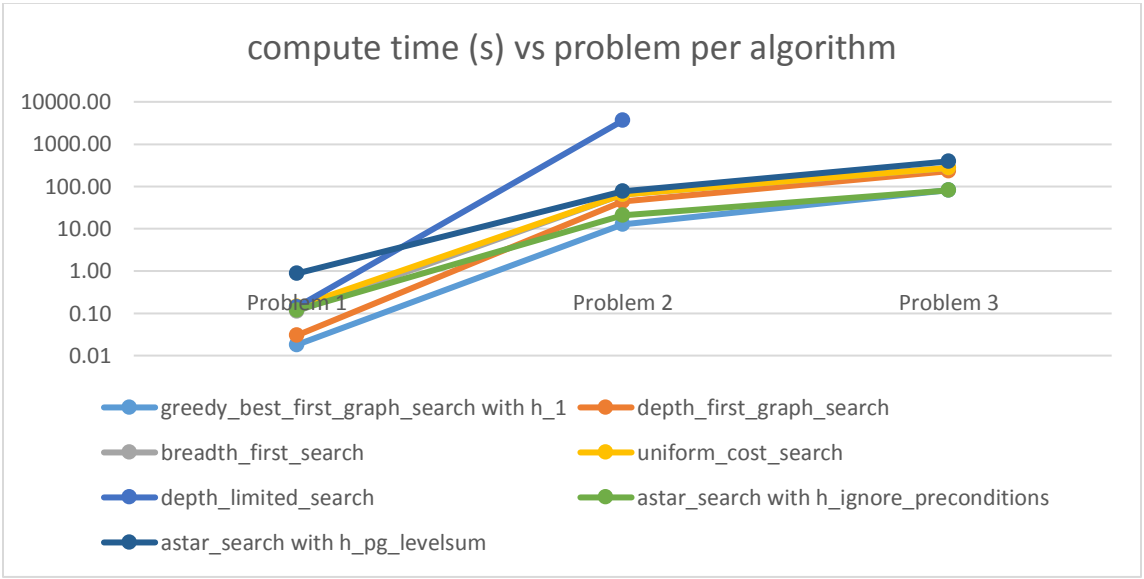
Table 3: Uninformed Planning Search results, Air Cargo Problem 3

	expansions	goal tests	new nodes	plan length	seconds
depth_first_graph_search	627	628	5176	596	10.9
greedy_best_first_graph_search with h_1	5398	5400	47665	26	81.6
breadth_first_search	14663	18098	129631	12	229.9
uniform_cost_search	18151	18153	159038	12	272.3
astar_search with h_1	18151	18153	159038	12	273.6

I plotted the times and "new nodes" columns for Problem 1 against each other to get a sense of the data. The relationship appears to be linear, so the two can be used interchangeably to investigate other relationships. The outlier comes from astar_search with h_pg_levelsum, discussed in Parts 2 and 3.



I also plotted the dependence of compute time against the problem number for each algorithm:



Part 2: Informed (heuristic) Planning Searches

After implementing "ignore preconditions" and "levelsum" heuristic functions, planning searches were performed on the same three air cargo problems. Tables 4-6 duplicate Tables 1-3 with the heuristic search results added in bold for comparison.

Table 4: Heuristic Planning Search results, Air Cargo Problem 1

	expansions	goal tests	new nodes	plan length	seconds
greedy_best_first_graph_search with h_1	7	9	28	6	0.02
depth_first_graph_search	12	13	48	12	0.03
astar_search with h_pg_levelsum	**11**	**13**	**50**	**6**	**0.87**
astar_search with h_ignore_preconditions	**41**	**43**	**170**	**6**	**0.12**
breadth_first_search	43	56	180	6	0.11
uniform_cost_search	55	57	224	6	0.14
astar_search with h_1	55	57	224	6	0.14
depth_limited_search	101	271	414	50	0.29
breadth_first_tree_search	1458	1459	5960	6	3.68
recursive_best_first_search with h_1	4229	4230	17029	6	10.52

Table 5: Heuristic Planning Search results, Air Cargo Problem 2

	expansions	goal tests	new nodes	plan length	seconds
astar_search with h_pg_levelsum	**86**	**88**	**841**	**9**	**77.2**
depth_first_graph_search	582	583	5211	575	9.6
greedy_best_first_graph_search with h_1	998	1000	8982	21	12.8
astar_search with h_ignore_preconditions	**1450**	**1452**	**13303**	**9**	**20.5**
breadth_first_search	3343	4609	30509	9	44.1
uniform_cost_search	4853	4855	44041	9	64.0
astar_search with h_1	4853	4855	44041	9	64.6
depth_limited_search	222719	2053741	2054119	50	3593.1

Table 6: Heuristic Planning Search results, Air Cargo Problem 3

	expansions	goal tests	new nodes	plan length	seconds
astar_search with h_pg_levelsum	**314**	**316**	**2894**	**12**	**391.2**
depth_first_graph_search	627	628	5176	596	10.9
astar_search with h_ignore_preconditions	**5038**	**5040**	**44926**	**12**	**80.7**
greedy_best_first_graph_search with h_1	5398	5400	47665	26	81.6
breadth_first_search	14663	18098	129631	12	229.9
uniform_cost_search	18151	18153	159038	12	272.3
astar_search with h_1	18151	18153	159038	12	273.6

Part 3: Analysis

Table 7 lists the optimal plans for Air Cargo Problems 1, 2, and 3. Multiple algorithms came up with the same length plan, as seen in the tables above, and "optimality" is defined here as number of action steps only. So although the order did vary, order of steps only matters in terms of correctness. One representative plan is chosen to represent the results in Table 7.

Table 7: Optimal Plans by Problem

Problem 1	**Problem 2**	**Problem 3**
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Load(C2, P2, JFK)	Fly(P1, SFO, ATL)
Fly(P1, SFO, JFK)	Load(C3, P3, ATL)	Load(C3, P1, ATL)
Fly(P2, JFK, SFO)	Fly(P1, SFO, JFK)	Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)
Unload(C2, P2, SFO)	Fly(P3, ATL, SFO)	Load(C2, P2, JFK)
	Unload(C1, P1, JFK)	Fly(P2, JFK, ORD)
	Unload(C2, P2, SFO)	Load(C4, P2, ORD)
	Unload(C3, P3, SFO)	Fly(P2, ORD, SFO)
		Unload(C2, P2, SFO)
		Unload(C3, P1, JFK)
		Unload(C4, P2, SFO)

Among the non-heuristic algorithms, greedy best first and depth-first searches were the fastest, but failed to find optimal plans every time; depth-first search particularly generated highly un-optimal plans. Breadth-first and uniform cost searches performed comparably, always finding an optimal plan, and were the fastest uninformed algorithms to do so. The other algorithms fell off the charts due to their inefficiency.

Between the two informed, or heuristic searches, "ignore preconditions" and "levelsum", both always found optimal plans. "Levelsum" is quite impressive in terms of its ability to minimize the number of goal tests, expansions, and new nodes. As seen in the graphs, with the uninformed algorithms there was a direct relationship between new nodes and computation time, regardless of the details of the algorithm. But the "levelsum" algorithm could not translate its search efficiency into superior speed, presumably because the generation of and search through a planning graph severely impacted its performance. This is seen both in the first graph above, where the outlier is the "levelsum" algorithm, and in the second graph, where the "levelsum" time performance proved worse than all the uninformed algorithms that didn't fall of the table.

As a result, I'd have to say that the best overall algorithm was the "ignore preconditions" heuristic algorithm, which for Problems 2 and 3 produced an optimal plan in the least amount of time. Perhaps my implementation of the "levelsum" heuristic was suboptimal, and with performance enhancements the "levelsum" algorithm can be made more competitive.