Udacity AIND Planning Heuristic Analysis

This project is to solve deterministic logistic planning problems for an Air Cargo transport system using a planning search agent.

Planning problems:

We have three planning problems with gradually more variables. In problem 1, we have 2 cargos, 2 planes, and 2 airports. In problem 2, we have 3 cargos, 3 planes, and 3 airports. In problem 3, we have 4 cargos, 2 planes, and 4 airpots. It is expected the computation expanse increase from problem 1 to problem 3.

We are comparing 10 search algorithm on the three problems. With 5 non-heuristic based plans and 5 heuristic based plans.

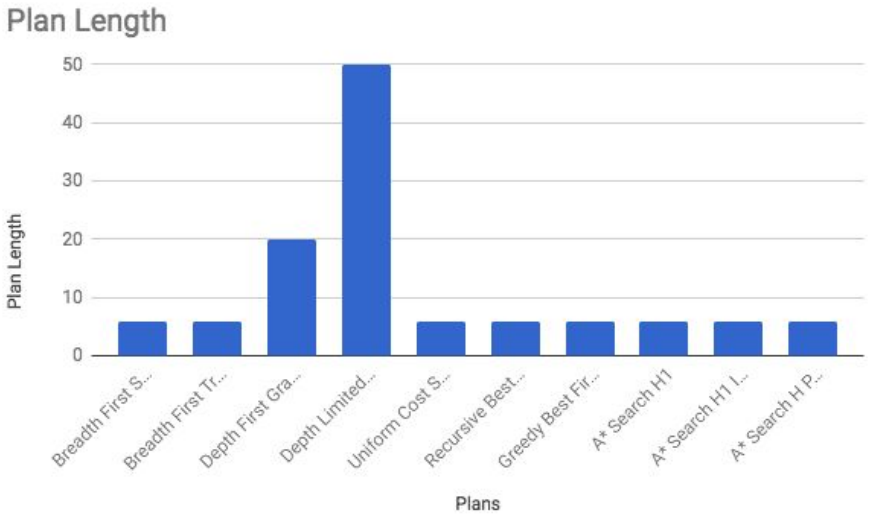
Results:

As expected, the complexity of solving the problems increases, this can be reflected by their optimal plan length, problem 1 has an optimal of 6, problem 2 has an optimal of 9, and problem 3 has an optimal of 12.

Problem 1

			Expansions	Goal Tests	New Notes	Plan Length	Time elapsed	Optimality
1	Non heuristic	Breadth First Search	43	56	180	6	0.05	Yes
2		Breadth First Tree Search	1458	1459	5960	6	1.48	Yes
3		Depth First Graph Search	21	22	84	20	0.02	No
4		Depth Limited Search	101	271	414	50	0.13	No
5		Uniform Cost Search	55	57	224	6	0.06	Yes
6	Heuristic	Recursive Best First Search h_1	4229	4230	17023	6	4.14	Yes
7		Greedy Best First Search h_1	7	9	28	6	0.007	Yes
8		A* Search h_1	55	57	224	6	0.05	Yes
9		A* Search h_ignore_preconditions	41	43	170	6	0.06	Yes
10		A* Search h_pg_levelsum	11	13	50	6	1.38	Yes

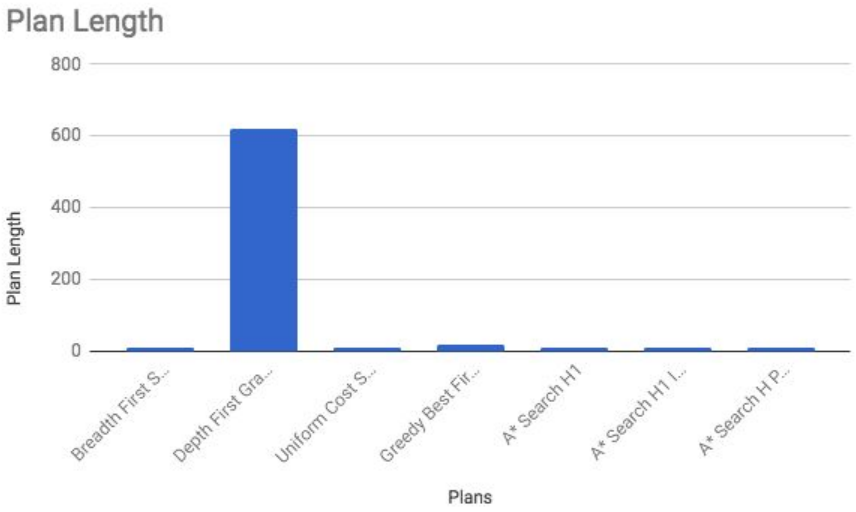
For problem 1, three of the non-heuristic based search found a minimum length plan, in this case, six: breadth first search, breadth first tree search, and uniform cost search. All of the heuristic based searches reached a plan length of six, which means all of the heuristic based searches are optimal in terms of problem 1.



Problem 2

			Expansions	Goal Tests	New Notes	Plan Length	Time elapsed	Optimality
1	Non heuristic	Breadth First Search	3343	4609	30509	9	22.09	Yes
2		Breadth First Tree Search						More than 10min
3		Depth First Graph Search	624	625	5602	619	5.38	No
4		Depth Limited Search						More than 10min
5		Uniform Cost Search	4853	4855	44041	9	21.49	Yes
6	Heuristic	Recursive Best First Search h_1						More than 10min
7		Greedy Best First Search h_1	998	1000	8982	21	3.82	No
8		A* Search h_1	4853	4855	44041	9	19.3	Yes
9		A* Search h_ignore_preconditions	1450	1452	13303	9	6.55	Yes
10		A* Search h_pg_levelsum	86	88	841	9	281.34	Yes

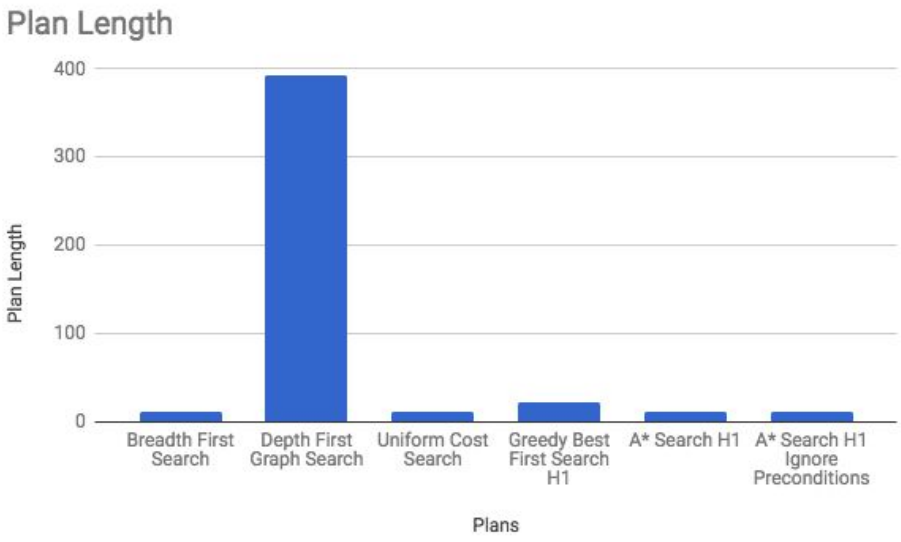
For problem 2, two non-heuristic based plan and one heuristic based plan were taking more than 10 minutes to compute. For the three remaining non-heuristic based plans, two found optimal: breadth first search and uniform cost search. For the four remaining heuristic based plans, three A* plans all found optimal.



Problem 3

			Expansions	Goal Tests	New Notes	Plan Length	Time elapsed	Optimality
1	Non heuristic	Breadth First Search	14663	18098	129631	12	188.08	Yes
2		Breadth First Tree Search						More than 10min
3		Depth First Graph Search	408	409	3364	392	2.9	No
4		Depth Limited Search						More than 10min
5		Uniform Cost Search	18222	18224	159608	12	91.11	Yes
6	Heuristic	Recursive Best First Search h_1						More than 10min
7		Greedy Best First Search h_1	5569	5571	49084	22	25.09	No
8		A* Search h_1	18222	18224	159608	12	88.24	Yes
9		A* Search h_ignore_preconditions	5040	5042	44944	12	27.45	Yes
10		A* Search h_pg_levelsum						More than 10min

For problem 3, again, the same two non-heuristic based plan and one heuristic based plan were taking more than 10 minutes to compute. For the three remaining non-heuristic based plans, two found optimal: breadth first search and uniform cost search. For the four remaining heuristic based plans, three A* plans all found optimal.



Analysis:

As regards to non-heuristic based plans, both breadth first search and uniform cost search were able to reach optimal for all three problems, with very similar time elapsed. It is understandable that depth first graph search and depth limited search were not able to find optimal, because the plan will explore the nodes depth wise, and it will return the first goal state encounter. This goal state might or might not be optimal. While breadth first search always considers the shortest path first, thus reaches optimal. Depth first search does require less time elapsed compare to breadth first search, which became more obvious in problem 2 and 3.

Comparing the two plans reaching optimal, uniform cost search had very minimal advantage on time elapsed with problem 1, and slightly bigger advantage with problem 2, and when it reaches to problem 3, uniform cost search was able to reach a more significant shorter time elapsed. In conclusion, both breadth first search and uniform cost search were good choices if plan length is critical, with uniform cost search having more advantage with more complex problems. If the time elapse is critical instead of plan length, depth first graph search would be a good option.

In terms of heuristic based plans, A* searches generally perform better. With A* h_1 and A* h_ignore_preconditions were able to reach optimal for all three problems. A* h_1 returns a constant value 1. So the estimation to the goal for each note is going to be the same. A* depends on both path cost and heuristic to select note expansion. In this case, we had constant heuristic value, thus A* h_1 rely more on path cost. With A* h_ignore_preconditions, every action is applicable in every step so that makes that single goal fluent could be reachable in one step. A* h_levelsum had significant less note expansion but significant higher time elapse. A* h_pg_levelsum performed similar to other plans with problem 1, and with problem 2, it required a dramatic more time elapsed to solve the problem, and with problem 3, it was not able to reach the solution within 10 minutes. This is most likely due to the heuristic being complex.

Comparing the two plans reaching optimal, for problem 1, A* h_1 and A* h_ignore_preconditions had very similar performance, with A* h_ignore_preconditions had 0.01 longer time elapsed, but less expansions and notes used. A* h_ignore_preconditions had less time elapsed for problem 2 and 3, with problem 3 had more significant time saving. Thus, we reached a similar conclusion with heuristic based plans, that we have two plans A* h_1 and A* h_ignore_preconditions that are good choices, with A* h_ignore_preconditions performances better with more complex problems.

Overall, non-heuristic based plans and heuristic based plans had similar performance with problem 1, with problem 2 and 3, heuristic plans tend to have less time elapsed. With more complex the problem, more evident the time saved. Considering all plans, A* h_ignore_preconditions is the best option, it produced constant good results, with always reaching optimal within relatively short time elapsed. It will also allows us to scale to a more complex problem if needed. Depth First Graph Search could also be a good choice if optimality is not critical, with shorter time elapse and less note expansion.

This exercise illustrates that heuristic based plan generally perform better than non-heuristic based plans, with more complex the problem the stronger the evidence. Heuristic based plans are also more flexible with modification for best interest, either optimality or memory expenses, which is not possible with non-heuristic based plans.