Heuristic Analysis for Air Cargo Planning Problem

Tharathorn (Joy) Rimchala

Udacity Artificial Intelligence Nanodegree
Project 3: Logic and Planning

Summary

In this project, the five non-heuristic (uninformative) and five heuristic (informative) search algorithms are experimented on three Air Cargo Planning Problems. *Problem 1* starts with 2 cargoes at 2 airports (1 cargo at each airport) that need to be airport swapped by two planes starting at those 2 airports. *Problem 2* is slightly more complicated than *Problem 1*, starting with 3 cargoes at 3 different airports to be moved by 3 planes (one at each airports) with two cargoes ending in the same airport. *Problem 3* has additional constraints in two aspects: 1. it has 4 cargoes, 2 airports and 2 plans, thus not all cargoes can be moved at the same time and the order in which the cargoes are moved become important. In my experiment, all the search algorithms are tested on all the problems and the results are summarized below. For searches that did not take longer than 10 minutes, the search algorithms were run 5 times and the elapsed time are average across those runs.

Results

Provide an optimal plan for *Problems 1*, *2*, and *3*.

An optimal plan for *Problem 1* is:

Load(C1, P1, SFO),

Fly(P1, SFO, JFK),

Load(C2, P2, JFK),

Fly(P2, JFK, SFO),

Unload(C1, P1, JFK),

Unload(C2, P2, SFO)

An optimal plan for *Problem 2*:

Load(C1, P1, SFO),

Fly(P1, SFO, JFK),

Load(C2, P2, JFK),

Fly(P2, JFK, SFO),

Load(C3, P3, ATL),

Fly(P3, ATL, SFO),

Unload(C1, P1, JFK),

Unload(C2, P2, SFO),

Unload(C3, P3, SFO)

An optimal plan for *Problem 3* is as follow:

Load(C1, P1, SFO),

Load(C2, P2, JFK),

Fly(P1, SFO, ATL),

Load(C3, P1, ATL),

Fly(P2, JFK, ORD),

Load(C4, P2, ORD),

Fly(P1, ATL, JFK),

Fly(P2, ORD, SFO),

Unload(C1, P1, JFK),

Unload(C2, P2, SFO),

Unload(C3, P1, JFK),

Unload(C4, P2, SFO)

Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3.

Based on experimental results in this project, the overall best non-heuristic search algorithms for *Problem 1* is *Breadth First Search (BFS)*, while *Uniform Cost Search(UCS)* is the best non-heuristic searches for *Problem 2* and *3*.

Since *Problem 1* is the simplest planning problem among the three problems and all algorithms in this project found an optimal plan, we can compare the search time for this problem as a high level metric to rank the algorithms. Based on this criteria, depth first graph search may appear to perform the best, beating all other non-heuristic searches by at least 10 fold (Table 1) shorter in search time (while *breadth first tree search* is the worst). However, for problem 2 (Table 2), *DFS* solution's plan length is 1085, while the optimal plan length is 9 (i.e >100 times longer than the optimal plan length). Similarly, for problem 3 (Table 3), *DFS*'s plan length is 787, while the optimal plan length is 12.

Among the non-heuristic search algorithms, only *breath first search (BFS)*, *depth first graph search (DFS)*, and *uniform cost search (UCS)* found solutions to all three Air Cargo Planning problems. Among these, *DFS* consistently found a solution in the least search time, however *DFS*'s solution is not necessary the most optimal (i.e. shortest) plan.

UCS expands search frontier based on cost rather than search tree level, it could find solutions faster than *BFS* in complex graph in which path cost connecting nodes are vastly different.

Table 1: Problem 1 : Non-heuristic Search Result Metrics

Algorithm Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapse
breadth first search	43	56	180	6	0.0312 s
breadth first tree search	1458	1459	5960	6	0.9829 s
depth first graph search	12	13	48	6	0.0092 s
depth limited search	101	271	414	6	0.0950 s
uniform cost search	55	57	224	6	0.0391 s

Table 2: Problem 2 : Non-heuristic Search Result Metrics

Algorithm Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapse
breadth first search	3346	4612	30534	9	14.7918 s
breadth first tree search					> 10 mins
depth first graph search	1124	1125	10017	1085	8.5868 s
depth limited search	213491	1967093	1967471	50	1241.3935 s
uniform cost search	4853	4855	44041	9	14.1403 s

Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for *Problems 1*, *2*, and *3*.

Based on experimental results in this project, the overall best heuristic search algorithms for Problem 1, 2 and 3 is A^* *with h_ignore_precondition*.

Among the heuristic search algorithms used in this project, *recursive best first search with* h_1 is the worst, taking much longer than even some non-heuristic searches to find a solution to problem 1, and taking longer than 10 minutes (Table 4) for *Problem 2* and *3* (Tables 5 and 6 respectively). *Greedy best first graph search (greedy BFS)* suffers the same flaw as *DFS* for non-heuristic search. While it found optimal plans in the shortest times, it's solutions are not necessarily optimal. For Problem 2 and 3 for example, greedy *BFS*'s optimal plans' lengths are 27 and 13 respectively while the optimal plans' lengths are 9 and 12 respectively.

Comparing A^* searches with different heuristic functions, A^* *with h_ignore_precondition* perform the best. In all problems, A^* *with h_ignore_precondition* expands 3 - 10 times more nodes than A^* *with h_pg_levelsum*, while taking $\approx$ 20 times less search time and both algorithms found optimal plan for all the planning problems.

Table 3: Problem 3 : Non-heuristic Search Result Metrics

Algorithm Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapse
breadth first search	11248	14988	88531	12	93.6088 s
breadth first tree search					> 10 mins
depth first graph search	1022	1023	4475	787	3.6517 s
depth limited search					> 10 mins
uniform cost search	15722	15724	121875	12	76.5495 s

Table 4: Problem 1 : Heuristic Searches Result Metrics

Algorithm Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapse
recursive best first search with h_1	4229	4230	17029	6	2.9373 s
greedy best first graph search with h_1	7	9	28	6	0.0078 s
A^* with h_1 (constant heuristic)	55	57	224	6	0.0485 s
A^* with h_ignore_precondition	41	43	170	6	0.0487 s
A^* with h_pg_levelsum	11	13	50	6	0.7982 s

What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

Between the two heuristic functions, the *h_ignore_precondition* is the best for the Air Cargo Planning problems in this project. The *h_ignore_precondition* heuristic function is achieved by relaxing all the preconditions and simply counting the number of unsatisfied goal, which is comparatively much simpler than the *h_pg_levelsum* heuristic function.

In a simple planning problem like *Problem 1*, there are multiple solutions so the problem. As such the algorithms that expand faster. As a results, when comparing between the best non-heuristics and heuristic searches, *UCS* slightly outperforms the *A^* with h_ignore_precondition* in search time (0.039 s vs. 0.049 s) because it expands much fewer nodes. In more complex problems e.g. *Problem 2* and *3*, *A^* with h_ignore_precondition* out perform *UCS* by large margins (4.616 s vs. 14.140 s for *Problem 2* and 13.039 s vs. 76.550 s for *Problem 3*). This can be explained by two reasons. First, *Problem 2* and *Problem 3* have more objects than *Problem 1* making the search graph larger. Second, the solutions space for *Problem 2* and *3* are more restrictive than in *Problem 1* because it requires multiple cargoes need to be delivered to the

Table 5: Problem 2 : Heuristic Searches Result Metrics

Algorithm Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapse
recursive best first search with h_1					> 10 mins
greedy best first graph search with h_1	998	1000	28	27	3.7398 s
A^* with h_1 (constant heuristic)	4853	4855	44041	9	15.0751 s
A^* with h_ignore_precondition	1450	1452	13303	9	4.6158 s
A^* with h_pg_levelsum	86	88	841	9	88.1002 s

Table 6: Problem 3 : Heuristic Searches Result Metrics

Algorithm Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapse
recursive best first search with h_1					> 10 mins
greedy best first graph search with h_1	2596	2598	17918	13	10.8707 s
A^* with h_1 (constant heuristic)	15722	15724	121875	12	47.5062 s
A^* with h_ignore_precondition	3762	3764	29998	12	13.0388 s
A^* with h_pg_levelsum	262	264	2399	12	277.8655 s

same airports, and in *Problem 3*, there are fewer planes than cargoes (thus resource contention).

Conclusion and Discussion

Experiments with the five non-heuristic and five heuristic search algorithms are experimented on three Air Cargo Planning Problems show that for simple planning problem, certain non-heuristic search algorithms namely *UCS* and *BFS* achieve comparable performances to heuristic searches. For complex problems where the search space is much larger than the number of optimal solutions, informative searches with good heuristic functions vastly outperforms uninformative searches.

Greedy algorithms with and without heuristic functions (*greedy BFS* and *DFS* in this project respectively) may found solutions much faster than non-greedy searches. However, it is consistently observed that those solutions are far from optimal.