Heuristic Analysis – Air Cargo Planning Problems

The three problem formulated for this project were solved using multiple planning and search algorithms. The algorithms were of two classes, namely, uninformed and heuristic-based. For each problem, as many search strategies as was feasible (i.e., they could be solved in a reasonable amount of time) were tested. The full results and a discussion of the performance of these algorithms are discussed in this report.

The results for the three problems are shown below.

Problem 1	Expansions	Goal tests	New nodes	Plan length	Time(seconds)	Relative Optimality
Breadth-first	43	56	180	6	0.019	Y
Breadth-first tree search	1458	1459	5960	6	0.581	Y
Depth-first graph search	12	13	48	12	0.005	N
Depth-limited search	101	271	414	50	0.064	N
Uniform cost	55	57	224	6	0.025	Y
Rec. best-first with h_1	4229	4230	17023	6	1.738	Y
Greedy best-first graph search with h_1	7	9	28	6	0.00294	Y
A* with h_1	55	57	224	6	0.0264	Y
A* with h_ignore_precond.	41	43	170	6	0.0288	Y
A* with h_pg_levelsum	11	13	50	6	10.896	Y

Table 1 - Problem 1 – The red-highlighted boxes correspond to uninformed algorithms and the blue-highlighted boxes correspond to heuristic-based searches

Problem 2	Expansions	Goal tests	New nodes	Plan length	Time(seconds)	Relative Optimality
Breadth first	3343	4609	30509	9	11.752	Y
Depth-first graph search	476	477	4253	466	2.023	N
Uniform cost	4852	4854	44030	9	9.129	Y
Greedy best-first graph search with h_1	990	992	8910	21	1.825	Y
A* with h_1	4852	4854	44030	9	9.139	Y
A* with h_ignore_precond.	1450	1452	13303	9	3.449	Y
A* with h_pg_levelsum	86	88	841	9	5565.67	Y

Table 2 - Problem 2 – The red-highlighted boxes correspond to uninformed algorithms and the blue-highlighted boxes correspond to heuristic-based searches

Problem 3	Expansions	Goal tests	New nodes	Plan length	Time(seconds)	Relative Optimality
Breadth first	14663	18098	129631	12	100.236	Y
Depth-first graph search	1511	1512	12611	1442	12.447	N
Uniform cost	18234	18236	159707	12	47.204	Y
Greedy best-first graph search with h_1	5605	5607	49360	22	14.425	N
A* with h_1	18234	18236	159707	12	47.276	Y
A* with h_ignore_precond.	5040	5042	44944	12	14.818	Y
A* with h_pg_levelsum	N/A	N/A	N/A	N/A	>>10 mins	N/A

Table 3 - Problem 3– The red-highlighted boxes correspond to uninformed algorithms and the blue-highlighted boxes correspond to heuristic-based searches

Discussion of the results

Note that I have called it relative optimality as I don’t know if it’s possible to achieve the goal with fewer steps than was reached with the algorithms. I therefore define a relatively optimal solution as one that required the lowest number of steps of the algorithms tested. The judgement is based on the relative optimality and the time required reaching the solution.

Problem 1:

For this problem, both uninformed and heuristic based algorithms were able to reach the same optimality. Among uninformed algorithms, both breadth-first and Uniform cost reached the optimal solution in the shortest time. The breadth-first search was marginally faster. Considering the heuristic-based searches, Greedy best-first graph search with h_1 performed optimally in the shortest time.

Optimal overall algorithm: Greedy best-first graph search with h_1

The optimum solution achieved using the above algorithm is:

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem 2:

Again, both types of algorithms were able to achieve the goals. Among uninformed searches, Uniform-cost algorithm performed the best reaching the optimal solution in the shortest time. Breadth-first search was a close second. The best heuristic-based algorithm was “A* with h_ignore_precond”.

Optimal overall algorithm: A* with h_ignore_preconditions

The optimum solution achieved using the above algorithm 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)

Problem 3:

Uniform-cost algorithm reached the optimal solution the fastest among the uninformed searches. “A* with h_ignore_preconditions” was the best among heuristic-based algorithm and overall.

Optimal overall algorithm: A* with h_ignore_precond.

The optimum solution achieved using the above algorithm 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)

Comparison between different algorithms

It appears for all 3 problems, uninformed algorithms such as “breadth-first” and “Uniform-cost” are sufficient to achieve optimal solution. Since breadth-first search always considers the shortest path, it seems to be a good strategy to try it first for all problems. Depth-first search on the other hand doesn’t appear to find the optimum solution consistently, even though it’s very fast. So in cases where an optimum solution is not necessary, and when the size the problem is very large, it could be considered a quicker way to get a solution. However, for the air cargo problem, I don’t imagine the solutions are practically acceptable since they require many more steps. In depth-first the optimality is not reached since the nodes aren’t compared with the aim of picking the best ones. The goal is just to reach higher depths until a solution is reached.

Heuristic-based strategies always performed best according to solutions achieved. However, the gains in time were only marginal in problems 1 and 2, whereas in problem 3, the time was considerably lower. I conclude that the more difficult the problem gets, the more we should consider using heuristic-based approaches whereas if the problems are simple enough, the additional implementation complexity is not justified. Note that “h_pg_levelsum” heuristic did not perform well in terms of time even though it eventually reached the optimal solution in problems 1 and 2 (It was taking too long for problem 3). The amount of time it required was orders of magnitude larger than other heuristic-based and uninformed searches. It is therefore no considered feasible for at least these types of problems.

In conclusion, I would recommend using an uninformed algorithm such as “breadth-first” or “Uniform-cost” to start. If the solution is acceptable we could stop there. However, as the complexity of the problem increases, it’s worth considering heuristic-based searches. In my tests, the “A* with h_ignore_preconditions” performed the best for problems 2 and 3, whereas, “Greedy best-first graph search with h_1”, was best for problem 1. However, the Greedy search didn’t reach an optimal solution from problems 2 and 3. I therefore recommend using the “A* with h_ignore_preconditions” algorithm.