AIND Project 3: Implement a Planning Search

Air Cargo Planning Heuristic Analysis

The following tables show the results gathered after solving the air cargo problems for this project with both uninformed and heuristic based search. The goal of this analysis is to document the results obtained from each search type and find an optimal solution for each air cargo problem, that is; a search algorithm that finds the lowest path among all possible paths from start to goal.

For each set of problems, the optimal and fastest solution has been highlighted with green color.

Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Length**	**Time**	**Optimal**
BFS	43	56	180	6	0,030	True
Uniform Cost	55	57	224	6	0,0381	True
DFS	21	22	84	20	0,015	False
A* Search h1	55	57	224	6	0,042	True
A* Search h_ignore_preconditions	41	43	170	6	0,048	True
A* Search h_pg_levelsum	11	13	50	6	10,370	True

Air Cargo Problem 1 Results

Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Length**	**Time**	**Optimal**
BFS	3343	4609	30509	9	14,186	True
Uniform Cost	4853	4855	44041	9	46,009	True
DFS	624	625	5602	619	3,721	False
A* Search h1	4853	4855	44041	9	49,373	True
A* Search h_ignore_preconditions	1506	1508	13820	9	16,705	True
A* Search h_pg_levelsum	86	88	841	9	1110,260	True

Air Cargo Problem 2 Results

Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Length**	**Time**	**Optimal**
BFS	14663	18098	129631	12	147,072	True
Uniform Cost	18223	18225	159618	12	616,26	True
DFS	408	409	3364	392	2,200	False
A* Search h1	18223	18225	159618	12	654,579	True
A* Search h_ignore_preconditions	5118	5120	45650	12	107,493	True
A* Search h_pg_levelsum					> 10 mins	

Air Cargo Problem 3 Results

Search Strategies Discussion

All three non-heuristic search strategies, that is; breadth first search, uniform cost search, and depth first graph search, find a solution to all air cargo problems. Breadth first search always considers the shortest path first [1] and a result of it it finds a solution to the problem in a reasonable amount of time and in an optimal way.

Depth first graph search does find a quick solution and requires a small amount of memory, but it lacks optimality. It is not optimal because it does not consider if a node is better than another, it simply explores the nodes that take it as deep as possible in the graph even if the goal is to its right [1].

Non-heuristic based search did perform better in problem 1 and 2, which suggest that when working with simple problems using a more elaborated approach, such a A* search with heuristics, is not worth the increase in the solution complexity.

Heuristic based search did perform better as the problem complexity increased. This is more evident in the air cargo problem 3, where the “A* Search with ‘h_ignore_preconditions’” performance was optimal and the fastest amongst those that were optimal. It’s also worth noting that the ‘h_pg_levelsum’ heuristic did in overall perform poorly, most likely due to the heuristic being too complex.

According to the results obtained in this analysis, the breadth first search strategy can solve planning problems both fast and optimality, which makes it a good candidate to start off an analysis when dealing with search planning problems. As the complexity of the problems increase, it might be worth to consider if a heuristic based approach such as “A* Search with ‘h_ignore_preconditions’” can outperform breadth first search and thus be used instead.

Optimal Sequence of Actions

The following table describes an optimal sequence of actions to solve each of the air cargo problems provided using the highlighted approaches from the tables above:

Problem	**Search Type**	**Optimal Sequence of Actions**
Air Cargo Problem 1	BFS	Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)
Air Cargo Problem 2	BFS	Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Fly(P3, ATL, SFO) Unload(C3, P3, SFO)

Air Cargo Problem 3	A* Search h_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) References

References

[1] Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).