Air Cargo Problem Heuristic Analysis

Mu Chen for Udacity AIND

Optimal Plans:

Problem 1:

- 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:

- Load(C1, P1, SFO)
- Load(C2, P2, JFK)
- Load(C3, P3, ATL)
- Fly(P1, SFO, JFK)
- Fly(P2, JFK, SFO)
- Fly(P3, ATL, SFO)
- Unload(C3, P3, SFO)
- Unload(C2, P2, SFO)
- Unload(C1, P1, JFK)

Problem 3:

- 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(P2, ORD, SFO)
- Unload(C4, P2, SFO)
- Fly(P1, ATL, JFK)
- Unload(C3, P1, JFK)
- Unload(C2, P2, SFO)
- Unload(C1, P1, JFK)

None Heuristic Search Results:

Problem 1	Number of node expansions	Number of goal tests	Time elapsed	Path length
Breadth First Search	43	56	0.0527	6
Depth First Graph Search	12	13	0.0097	12
Uniform Cost Search	55	57	0.0452	6

Problem 2	Number of node expansions	Number of goal tests	Time elapsed	Path length
Breadth First Search	3343	4609	16.5605	9
Depth First Graph Search	1669	1670	15.6563	1444
Uniform Cost Search	4853	4855	14.7535	9

Problem 3	Number of node expansions	Number of goal tests	Time elapsed	Path length
Breadth First Search	14120	17673	118.3977	12
Depth First Graph Search	677	678	4.2909	660
Uniform Cost Search	18223	18225	63.5671	12

Analysis:

The Depth First Graph Search always finds a solution by going straight into the search tree regardless of path length. It is fast in most cases since it requires the least number of node expansions but when the solution node is at the far end of the branch selected it can also take a long time before ending up with a very long path length. On the contrary, Breadth First Search takes the longest time to find the solution by visiting significantly more nodes than Depth First Graph Search, although the path length found by Breadth First Search is always optimal. The Uniform Cost Search also finds the optimal solution in all three problems. It visits more nodes than the other two algorithms while it gets a close match with Breadth First Search in terms of running time, indicating that it takes less time to process each node.

For relative simple problems, Bread First Search achieves optimal solution within a good time and with less node expansions than Uniform Cost Search. For complex problems (Problem 3) Uniform Cost Search takes only half the computation time by visiting a slightly larger number of nodes compared to Breadth First Search. Depth First Graph Search can only be used when any solution is acceptable regardless its optimality.

Heuristic Search Results:

Problem 1	**Number of node expansions**	**Number of goal tests**	**Time elapsed**	**Path length**
A Star Search with h_1	55	57	0.0444	6
A Star Search with h_ignore_preconditions	55	57	0.0479	6
A Star Search with h_pg_levelsum	39	41	1.0084	6

Problem 2	**Number of node expansions**	**Number of goal tests**	**Time elapsed**	**Path length**
A Star Search with h_1	4853	4855	14.3121	9
A Star Search with h_ignore_preconditions	4853	4855	15.2244	9
A Star Search with h_pg_levelsum	1129	1131	338.0472	9

Problem 3	**Number of node expansions**	**Number of goal tests**	**Time elapsed**	**Path length**
A Star Search with h_1	18223	18225	64.6966	12
A Star Search with h_ignore_preconditions	18223	18225	68.0604	12
A Star Search with h_pg_levelsum	2026	2028	1216.6103	12

Analysis:

All three heuristics found the optimal solution while the level sum heuristic always take the longest time. Although it visits less number of nodes before finding the solution, the computation of the heuristic score for each node takes too long, making it an unsatisfied solution for complex problems. The other two functions, a constant heuristic and a heuristic that ignores any precondition, performs very similarly in both number of node expansions and the time complexity except that the ignore precondition function takes a little longer time in computation.

Overall, the best algorithms in terms of computation time are A Star Search with h_1 and Uniform Cost Search since they find the optimal solution in less time than any other searching algorithm we tested. The Uniform Cost Search is essentially the same as A Star with h_1, with the only difference that it does not include a constant heuristic. However, these two methods do visit more nodes than Breadth First Search and they did not outperform BFS too much for relatively simple problems (Problem 1 & 2). Thus, the best algorithm for simple problem would be the non-heuristic Breadth First Search while the Uniform Cost Search and A Star Search with h_1 would be most suited for complex problems.