# Air Cargo Planning Search Heuristic Analysis

The results for both uninformed and heuristic based search for the 3 air cargo problem are tabulated below:

| Search Type | Expansions | Goal Tests | New Nodes | Plan Length | Time(s) | Optimal |
|---|---|---|---|---|---|---|
| **Breadth First Search** | 43 | 56 | 180 | 6 | 0.096 | Yes |
| **Depth First Graph Search** | 12 | 13 | 48 | 12 | 0.0081 | No |
| **Uniform Cost Search** | 55 | 57 | 224 | 6 | 0.0396 | Yes |
| **A* Search h_1** | 55 | 57 | 224 | 6 | 0.0458 | Yes |
| **A* Search Ignore Pre-conditions** | 41 | 43 | 170 | 6 | 0.0535 | Yes |
| **A* Search h_pg_levelsum** | 11 | 13 | 50 | 6 | 1.569 | Yes |

Air Cargo Problem 1

| Search Type | Expansions | Goal Tests | New Nodes | Plan Length | Time(s) | Optimal |
|---|---|---|---|---|---|---|
| **Breadth First Search** | 3401 | 4672 | 31049 | 9 | 15.396 | Yes |
| **Depth First Graph Search** | 350 | 351 | 3142 | 346 | 1.683 | No |
| **Uniform Cost Search** | 4761 | 4763 | 43206 | 9 | 16.802 | Yes |
| **A* Search h_1** | 4761 | 4763 | 43206 | 9 | 11.582 | Yes |
| **A* Search Ignore Pre-conditions** | 1450 | 1452 | 13303 | 9 | 4.309 | Yes |
| **A* Search h_pg_levelsum** | 86 | 88 | 841 | 9 | 382.488 | Yes |

Air Cargo Problem 2

| Search Type | Expansions | Goal Tests | New Nodes | Plan Length | Time(s) | Optimal |
|---|---|---|---|---|---|---|
| **Breadth First Search** | 14491 | 17947 | 128184 | 12 | 121.496 | Yes |
| **Depth First Graph Search** | 1948 | 1949 | 16253 | 1878 | 27.782 | No |
| **Uniform Cost Search** | 17783 | 17785 | 155920 | 12 | 54.524 | Yes |
| **A* Search h_1** | 17783 | 17785 | 155920 | 12 | 64.24 | Yes |
| **A* Search Ignore Pre-conditions** | 5003 | 5005 | 44586 | 12 | 19.506 | Yes |
| **A* Search h_pg_levelsum** | 311 | 313 | 2863 | 12 | 1806.99 | Yes |

Air Cargo Problem 3

**Search Strategies Analysis**

All three uninformed search strategies find the solution to the three problems. Only Breadth first search and Uniform Cost Search find an optimal solution. Depth first graph search is the fastest and takes the

least memory. However, depth first graph search explores a node without considering if it is better or not and hence gives a non-optimal solution [1].

All 3 heuristic based searches provide an optimal solution but A* search with level sum fails to get the solution within the 10 min time limit for Problem 3. Among the three, A* search with ignore pre conditions is the fastest while A*search with level sum takes the least memory.

Non heuristic based searches performed better in Problems 1 and 2 while heuristic based search performed better in Problem 3 indicating that heuristic based searches are better for complex problems. As the problem complexity grows more information is required for better performance.

Overall, A* search with level sum was able to minimize the number of nodes to be examined but at a heavy calculation cost. A* search with h1 finds an optimal solution within time but examines a very large number of nodes. A* search with ignore preconditions is a better heuristic as it takes both less time and memory [1]. Breath first search strategy can solve problem both optimally and fast and hence it can be used to start analysis when solving planning search problems.

As discussed in the video lectures, it is sometimes better to use a cheap to calculate heuristic over an expensive one (Artificial Intelligence Nanodegree, Build a Game-Playing Agent).

Optimal Sequence of Actions

The following table shows the optimal sequence of actions for the three problems:

| | |
|---|---|
| Air Cargo Problem 1 | Load(C2, P2, JFK)<br>Load(C1, P1, SFO)<br>Fly(P2, JFK, SFO)<br>Unload(C2, P2, SFO)<br>Fly(P1, SFO, JFK)<br>Unload(C1, P1, JFK) |
| Air Cargo Problem 2 | Load(C3, P3, ATL)<br>Load(C2, P2, JFK)<br>Load(C1, P1, SFO)<br>Fly(P3, ATL, SFO)<br>Unload(C3, P3, SFO)<br>Fly(P2, JFK, SFO)<br>Unload(C2, P2, SFO)<br>Fly(P1, SFO, JFK)<br>Unload(C1, P1, JFK) |
| Air Cargo Problem 3 | Load(C2, P2, JFK)<br>Fly(P2, JFK, ORD)<br>Load(C4, P2, ORD)<br>Fly(P2, ORD, SFO)<br>Unload(C4, P2, SFO)<br>Load(C1, P1, SFO)<br>Fly(P1, SFO, ATL)<br>Load(C3, P1, ATL)<br>Fly(P1, ATL, JFK)<br>Unload(C3, P1, JFK) |

| | |
|---|---|
| | Unload(C2, P2, SFO)<br>Unload(C1, P1, JFK) |

References:


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