**Heuristic Analysis**

### Air Cargo Results Comparison Matrix

The table below summarizes the results obtained from running 6 types of searches: 3 non-heuristic, and 3 heuristic approaches for each of 3 air cargo problems: P1, P2, and P3.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Problem #** | **Search Type** | **Expansion** | **Goal Tests** | **New Nodes** | **Plan Length** | **Time**  **(s)** | | **P1** | *breadth\_first\_search* | 43 | 56 | 180 | 6 | 0.028 | |  | *depth\_first\_graph\_search* | 12 | 13 | 48 | 12 | 0.007 | |  | *uniform\_cost\_search* | 55 | 57 | 224 | 6 | 0.036 | |  | *using astar\_search with h\_1* | 55 | 57 | 224 | 6 | 0.035 | |  | *h\_ignore\_preconditions* | 41 | 43 | 170 | 6 | 0.027 | |  | *h\_pg\_levelsum* | 11 | 13 | 50 | 6 | 1.055 | | **P2** | *breadth\_first\_search* | 3343 | 4609 | 30509 | 9 | 12.95 | |  | *depth\_first\_graph\_search* | 1669 | 1670 | 14863 | 1444 | 12.35 | |  | *uniform\_cost\_search* | 4853 | 4855 | 44041 | 9 | 11.88 | |  | *using astar\_search with h\_1* | 4853 | 4855 | 44041 | 9 | 12.10 | |  | *h\_ignore\_preconditions* | 1450 | 1452 | 13303 | 9 | 4.016 | |  | *h\_pg\_levelsum* | 86 | 88 | 841 | 9 | 184.48 | | **P3** | *breadth\_first\_search* | 13601 | 16987 | 106167 | 12 | 91.16 | |  | *depth\_first\_graph\_search* | 1292 | 1293 | 5744 | 875 | 3.68 | |  | *uniform\_cost\_search* | 17004 | 17006 | 132060 | 12 | 48.37 | |  | *using astar\_search with h\_1* | 17004 | 17006 | 132060 | 12 | 45.79 | |  | *h\_ignore\_preconditions* | 6550 | 6552 | 52021 | 13 | 18.23 | |  | *h\_pg\_levelsum* | 381 | 383 | 2565 | 12 | 701.63 |   Table :Results of different Search Types on P{1,2,3} |

The optimal solutions (Table: 2) in terms of Plan Length & Time Taken are highlighted in yellow.

In terms of path length & time taken, **Breadth First Search (BFS)** and **Uniform Cost Search** (Weighted BFS/Dijkstra’s) performed better. BFS performs faster, uses less memory, and is guaranteed to find the shortest path and hence it is the most optimal solution and recommended search strategy. **Depth First Graph Search (DFS)** on the other hand is the fastest, and uses the least amount of memory, but doesn’t find the optimal solution as it keeps on probing deeper into the tree resulting in larger path lengths (from above table).

Non-heuristic searches (BFS) performed better with simple problems, such as P1 and P2, while heuristic approach (**A\* Search**) performed better when the problem space was large indicating that heuristic search scales much better for complicated problems (real life problems).

### Optimal Solutions

|  |  |  |
| --- | --- | --- |
| **Problem** | **Optimal Search Type** | **Optimal Solution** |
| P1 | Breadth First Search | Load(C2, P2, JFK)  Load(C1, P1, SFO)  Fly(P2, JFK, SFO)  Unload(C2, P2, SFO)  Fly(P1, SFO, JFK)  Unload(C1, P1, JFK) |
| P2 | Uniform Cost Search | 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(C1, P1, JFK)  Unload(C2, P2, SFO)  Unload(C3, P3, SFO) |
| P3 | A\* Search with h1 | Load(C1, P1, SFO)  Fly(P1, SFO, ATL)  Load(C3, P1, ATL)  Fly(P1, ATL, JFK)  Load(C2, P1, JFK)  Unload(C1, P1, JFK)  Unload(C3, P1, JFK)  Fly(P1, JFK, ORD)  Load(C4, P1, ORD)  Fly(P1, ORD, SFO)  Unload(C2, P1, SFO)  Unload(C4, P1, SFO) |

Table 2: Optimal Solutions

Reference:

* Stuart J. Russel, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd edition)