**HEURISTIC ANALYSIS**

**FOR DETERMINISTIC LOGISTIC PLANNING PROBLEMS FOR AN AIR CARGO TRANSPORT SYSTEM USING A PLANNING SEARCH AGENT**

**Uninformed Non-Heuristic Search:**

The following table shows the result of the uninformed planning searches namely Breadth First Search(BFS), Depth First Search(DFS), and Uniform Cost Search (UCS).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Problems | Search | Plan Length | Time Elapse(secs) | Node Expended | Goal Test | Optimal |
| 1 | BFS | 6 | 0.02429 | 43 | 56 | Yes |
| 2 | BFS | 9 | 5.57490 | 3343 | 4609 | Yes |
| 3 | BFS | 12 | 27.594 | 14663 | 18098 | Yes |
| 1 | DFS | 12 | 0.00648 | 12 | 12 | No |
| 2 | DFS | 575 | 2.09727 | 582 | 583 | No |
| 3 | DFS | 1451 | 8.0761 | 1501 | 1502 | No |
| 1 | UCS | 6 | 0.026224 | 55 | 57 | Yes |
| 2 | UCS | 9 | 7.73486 | 4853 | 4855 | Yes |
| 3 | UCS | 12 | 34.8446 | 18223 | 18225 | Yes |

**Time Elapsed:**

The following graph shows the time taken by different problems for various searches.

Each of the problem takes the least time for DFS followed by BFS & UCS. So DFS is the fastest search among the three.

**Node Expanded:**

The graph shows the number of node expanded under various searches.

The UCS has the most number of nodes expanded for all the problems followed by BFS. DFS has the least nodes expanded which is the reason for its fastest execution time.

**Path Length:**

The optimality of the solution will depend on the path length. Least the length of the path greater its Optimality.

Since both the searches BFS &UCS have same length of path they are optimal for our solution, whereas the DFS has lengthy path, that means it is less likely for the optimality.

**Heuristic Search:**

The following heuristics will be discussed:

* h\_1: returns 1
* h\_ignore\_preconditions: obtained from the suggestion given in the book “AIMA-3rd edition”
* h\_pg\_level\_search: obtained through planning graph

We will be applying A\* search over these heuristics for the 3 different problems that were discussed earlier.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Problems | Search | Plan Length | Time Elapse**(**sec) | Node Expended | Goal Test | Optimal |
| P1 | h\_1 | 6 | 0.02707 | 55 | 57 | Yes |
| P1 | h\_ignore\_preconditions | 6 | 0.020481 | 41 | 43 | Yes |
| P1 | h\_pg\_level\_sum | 6 | 0.794172 | 11 | 13 | Yes |
| P2 | h\_1 | 9 | 7.988941 | 4853 | 4855 | Yes |
| P2 | h\_ignore\_preconditions | 9 | 2.50792 | 1450 | 1452 | Yes |
| P2 | h\_pg\_level\_sum | 9 | 89.24429 | 86 | 88 | Yes |
| P3 | h\_1 | 12 | 56.03783 | 18233 | 18225 | Yes |
| P3 | h\_ignore\_preconditions | 12 | 12.90056 | 5040 | 5042 | Yes |
| P3 | h\_pg\_level\_sum | 12 | 737.4154 | 315 | 317 | Yes |

The above table shows the metrics for the heuristic search. The values for the columns Node expanded and goal tests are similar.

**Time Elapsed:**

The time taken to execute the planning is the highest for the h\_pg\_level\_sum heuristic. The reason for it is the high computation it undergoes. The heuristic h\_ignore\_preconditions turns out to be the fastest to execute.

**Nodes Expanded:**

H\_1 heuristic has the most number of nodes expanded while h\_pg\_level\_sum has the least.

**Path Length:**

Since all the heuristic have same path length they all are admissible for the optimality of the solution.

**Optimality of A\* Search:**

A\* is admissible and considers fewer nodes than any other admissible search algorithm with the same heuristic. This is because A\* uses an "optimistic" estimate of the cost of a path through every node that it considers—optimistic in that the true cost of a path through that node to the goal will be at least as great as the estimate.

**Conclusion:**

**Overall, the best heuristic is the h\_ignore\_preconditions, which compared to other heuristics took the least time for the execution at the same time has the significant number of nodes expanded. The worst among the six heuristics is the h\_pg\_level\_sum because of its slowness and had least node expanded.**

**Optimal Plans:**

The optimal plans for the 3 problems using best heuristic i.e. h\_ignore\_preconditions are carried out by A\* search.

**Problem1:**

Load (C1, P1, SFO)

Fly (P1, SFO, JFK)

Unload (C1, P1, JFK)

Load (C2, P2, JFK)

Fly (P2, JFK, SFO)

Unload (C2, P2, SFO)

**Problem2:**

Load (C3, P3, ATL)

Fly (P3, ATL, SFO)

Unload (C3, P3, SFO)

Load (C1, P1, SFO)

Fly (P1, SFO, JFK)

Unload (C1, P1, JFK)

Load (C2, P2, JFK)

Fly (P2, JFK, SFO)

Unload (C2, P2, SFO)

**Problem 3:**

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 (C1, P1, JFK)

Unload (C2, P2, SFO**)**