|  |
| --- |
| **Problem : Air Cargo Problem 1** |
| **Optimal Plan :** |
| Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algo Used** | **Plan Length** | **Expansions** | **Goal Tests** | **New Nodes** | **Optimal** | **Time(seconds)** |
|  |  |  |  |  |  |  |
| **non-heuristic search** |  |  |  |  |  |  |
| breadth\_first\_search | 6 | 43 | 56 | 180 | Y | 0.033105939 |
| depth\_first\_graph\_search | 20 | 21 | 22 | 84 | N | 0.017987239989452064 |
| uniform\_cost\_search | 6 | 55 | 57 | 224 | Y | 0.04212672 |
|  |  |  |  |  |  |  |
| **heuristic search** |  |  |  |  |  |  |
| astar\_search with h\_ignore\_preconditions | 6 | 41 | 43 | 170 | Y | 0.045769307 |
| astar\_search with h\_pg\_levelsum | 6 | 11 | 13 | 50 | Y | 0.48696264 |

In non-heuristic search, breadth first search and uniform cost search come up with the the optimal plan. depth\_first\_graph\_search is the fastest and takes the least memory but does not provides an optimal solution.

In heuristic search, both astar\_search with h\_ignore\_preconditions and astar\_search with h\_pg\_levelsum come up with the optimal plan. astar\_search with h\_ignore\_preconditions is the faster but astar\_search with h\_pg\_levelsum takes less memory.

Overall,in terms of optimal solutions, breadth\_first\_search is the fastest but astar\_search with h\_ignore\_preconditions takes the least memory.

|  |
| --- |
| **Problem : Air Cargo Problem 2** |
| **Optimal Plan :** |
| 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) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algo Used** | **Plan Length** | **Expansions** | **Goal Tests** | **New Nodes** | **Optimal** | **Time(seconds)** |
|  |  |  |  |  |  |  |
| **non-heuristic search** |  |  |  |  |  |  |
| breadth\_first\_search | 9 | 3343 | 4609 | 30509 | Y | 14.363841059006518 |
| depth\_first\_graph\_search | 619 | 624 | 625 | 5602 | N | 3.7531195210031 |
| uniform\_cost\_search | 9 | 4852 | 4854 | 44030 | Y | 12.832434479001677 |
|  |  |  |  |  |  |  |
| **heuristic search** |  |  |  |  |  |  |
| astar\_search with h\_ignore\_preconditions | 9 | 1450 | 1452 | 13303 | Y | 4.49539596500108 |
| astar\_search with h\_pg\_levelsum | 9 | 86 | 88 | 841 | Y | 37.86639750000904 |

In non-heuristic search, breadth first search and uniform cost search come up with the the optimal plan. depth\_first\_graph\_search is the fastest and takes the least memory but does not provides an optimal solution.

In heuristic search, both astar\_search with h\_ignore\_preconditions and astar\_search with h\_pg\_levelsum come up with the optimal plan. astar\_search with h\_ignore\_preconditions is the faster but astar\_search with h\_pg\_levelsum takes less memory.

Overall,in terms of optimal solutions, astar\_search with h\_ignore\_preconditions is the fastest but lowest memory is taken by astar\_search with h\_pg\_levelsum. So heuristic search solutions provide better optimal plans in this case than non-heuristic search ones.

|  |
| --- |
| **Problem : Air Cargo Problem 3** |
| **Optimal Plan :** |
| Load(C1, P1, SFO)  Load(C2, P2, JFK)  Fly(P2, JFK, ORD)  Load(C4, P2, ORD)  Fly(P1, SFO, ATL)  Load(C3, P1, ATL)  Fly(P1, ATL, JFK)  Unload(C1, P1, JFK)  Unload(C3, P1, JFK)  Fly(P2, ORD, SFO)  Unload(C2, P2, SFO)  Unload(C4, P2, SFO) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algo Used** | **Plan Length** | **Expansions** | **Goal Tests** | **New Nodes** | **Optimal** | **Time(seconds)** |
|  |  |  |  |  |  |  |
| **non-heuristic search** |  |  |  |  |  |  |
| breadth\_first\_search | 12 | 14663 | 18098 | 129631 | Y | 120.6063692579919 |
| depth\_first\_graph\_search | 392 | 408 | 409 | 3364 | N | 1.8938232449872885 |
| uniform\_cost\_search | 12 | 18235 | 18237 | 159716 | Y | 59.53526903499733 |
|  |  |  |  |  |  |  |
| **heuristic search** |  |  |  |  |  |  |
| astar\_search with h\_ignore\_preconditions | 12 | 5040 | 5042 | 44944 | Y | 17.715687224001158 |
| astar\_search with h\_pg\_levelsum | 12 | 318 | 320 | 2934 | Y | 196.24829012600821 |

In non-heuristic search, breadth first search and uniform cost search come up with the the optimal plan. depth\_first\_graph\_search is the fastest and takes the least memory but does not provides an optimal solution.

In heuristic search, both astar\_search with h\_ignore\_preconditions and astar\_search with h\_pg\_levelsum come up with the optimal plan. astar\_search with h\_ignore\_preconditions is the faster but astar\_search with h\_pg\_levelsum takes less memory.

Overall,in terms of optimal solutions, astar\_search with h\_ignore\_preconditions is the fastest but lowest memory is taken by astar\_search with h\_pg\_levelsum. So heuristic search solutions provide better optimal plans in this case than non-heuristic search ones.

**Conclusion**

In **non-heuristic searches**, breadth first search and uniform cost search provide optimal solutions as breadth first search expand shortest nodes first and uniform cost search expand cheapest node first.

Depth first search on the other hand, since it explores all the nodes till the final depth, does not gives optimal solution but performs better on space requirements.

**Reference: Search Lesson (Search Comparison Videos (20-22) from AIND)**

In **heuristic search**, A\* will find the lowest cost path if the heuristic function for a state is less than the true cost of the path to the goal through that state.

And since the heuristic functions here are admissible(optimistic), they never overestimate the distance to the goal which gives an optimal solution.

**Reference: Search Lesson (A\* Search (24-29) and Optimistic Heuristic(30) Videos from AIND)**

**It seems overall, as problems become more complex, heuristic searches provide better solutions. For simpler problems, non-heuristic search seem to provide good solutions.**