Task: Run uninformed planning searches for air\_cargo\_p1, air\_cargo\_p2, and air\_cargo\_p3; provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm. Include the result of at least three of these searches, including breadth-first and depth-first, in your write-up (breadth\_first\_search and depth\_first\_graph\_search).

Table 1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Problem | Search Type | Node Expansions | Goal Tests | Time Elapsed | Runtime/Node Optimality? | Plan Optimality? |
| P1 | Breadth\_first\_search | 43 | 56 | 0.115 seconds | Least optimal for P1 (of three surveyed), second-most nodes expanded/tested | Plan length of 6 - optimal |
| P1 | Depth first graph search | 21 | 22 | 0.050 seconds | Most optimal for P1 (of three surveyed), fewest nodes expanded/tested | Plan length of 20 – not optimal |
| P1 | astar\_search h\_1 | 55 | 57 | 0.106 seconds | Second most optimal for P1 (of three surveyed), most nodes expanded/tested | Plan length of 6 - optimal |
| P2 | Breadth\_first\_search | 3343 | 4609 | 4.836 seconds | Second most optimal for P2 (of three surveyed), second-most nodes expanded/tested | Plan length of 9 - optimal |
| P2 | Depth first graph search | 624 | 625 | 1.001 seconds | Most optimal for P2 (of three surveyed), fewest nodes expanded/tested | Plan length of 619 – not optimal |
| P2 | astar\_search h\_1 | 4852 | 4854 | 5.171 seconds | Least optimal for P2 (of three surveyed), second-most nodes expanded/tested | Plan length of 9 - optimal |
| P3 | Breadth\_first\_search | 14663 | 18098 | 22.058 seconds | Most optimal for P3 (of three surveyed), second-most nodes expanded/tested | Plan length of 12 - optimal |
| P3 | Depth first graph search | 408 | 409 | 0.724 seconds | Least optimal for P3 (of three surveyed), fewest nodes expanded/tested | Plan length of 392 – not optimal |
| P3 | astar\_search h\_1 | 18235 | 18237 | 20.493 | Second most optimal for P3 (of three surveyed), most nodes expanded/tested | Plan length of 12 - optimal |

Overall summary – for all of the problems considered, breadth first search and astar\_search h1 consistently found the most optimal plan lengths, with astar\_search h\_1 typically being faster for problems of a large size.

P3 the order is revised, with breadth-first search most optimal, followed by astar\_search h1 and finally

Task: Run A\* planning searches using the heuristics you have implemented on air\_cargo\_p1, air\_cargo\_p2 and air\_cargo\_p3. Provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm and include the results in your report.

8. astar\_search h\_1

9. astar\_search h\_ignore\_preconditions

10. astar\_search h\_pg\_levelsum

Table 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Problem | Search Type | Node Expansions | Goal Tests | Time Elapsed | Optimality |
| P1 | astar\_search h\_1 | 55 | 57 | 0.109 s | Plan length 6, second fastest, second-most nodes expanded/tested |
| P1 | astar\_search h\_ignore\_preconditions | 41 | 43 | 0.086 s | Plan length 6, fastest, fewest nodes expanded/tested |
| P1 | astar\_search h\_pg\_levelsum | 55 | 57 | 1.319 s | Plan length 6, slowest, second-most nodes expanded/tested |
| P2 | astar\_search h\_1 | 4852 | 4854 | 5.599 s | Plan length 9, second fastest, second-most nodes expanded/tested |
| P2 | astar\_search h\_ignore\_preconditions | 1450 | 1452 | 2.101 s | Plan length 9, fastest, fewest nodes expanded/tested |
| P2 | astar\_search h\_pg\_levelsum | 4852 | 4854 | 106.047 s | Plan length 9, slowest, second-most nodes expanded/tested |
| P3 | astar\_search h\_1 | 18235 | 18237 | 20.347 s | Path length of 12, second fastest, second-most nodes expanded/tested |
| P3 | astar\_search h\_ignore\_preconditions | 5040 | 5042 | 6.675 s | Path length of 12, fastest, fewest nodes expanded/tested |
| P3 | astar\_search h\_pg\_levelsum | 18235 | 18237 | 563.163 s | Path length of 12, slowest, second-most nodes expanded/tested |

Overall, the astar\_search h\_ignore\_preconditions seems to be a sensible choice among astart algorithms as it scales well to large datasets such as problem P3 and still finds optimal path lengths in a reasonable time.

* Provide an optimal plan for Problems 1, 2, and 3.

An optimal plan for Problem 1 is:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

An optimal plan for Problem 2 is:

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)

An optimal plan for Problem 3 is:

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)

Fly(P1, ATL, JFK)

Unload(C4, P2, SFO)

Unload(C3, P1, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

* Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3. Include breadth-first, depth-first, and at least one other uninformed non-heuristic search in your comparison; Your third choice of non-heuristic search may be skipped for Problem 3 if it takes longer than 10 minutes to run, but a note in this case should be included.

Please see table 1

* Compare and contrast heuristic search result metrics using A\* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

Please see table 2

* What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

The best heuristic used in these problems was the “ignore preconditions” heuristic – it consistently returned solutions more rapidly than the “level sum” heuristic while maintaining similar quality of results (same path length). It was better than the non-heuristic search methods such as, for instance, breadth-first search or depth-first graph search; the former saw significant increases in runtime as problems got larger, while the latter found paths that were much too long and were thus not optimal.

* Provide tables or other visual aids as needed for clarity in your discussion.

Please see tables 1 and 2 above.

**Task: Provide a reason for the observed results using at least one appropriate justification from the video lessons or from outside resources**

In Russel and Norvig (2002) the authors note, “The most obvious difficulty with… [standard search algorithms—depth-first, A∗, and so on]…is that the problem-solving agent can be overwhelmed by irrelevant actions” (p. 375). This can clearly be seen in our results for depth first search on problem 3, which produced a path length of 392, as opposed to the optimal length of 12. We note that DFS could not find the optimal path as it had no domain-specific heuristic to guide it towards an optimal answer; instead it simply traversed further and further until it found a functional path, but it had no to to know whether or not this was a “good” path.

We note that ignore preconditions runs faster than pg\_levelsum because the latter must compute an estimated cost for each goal in a conjunctive goal clause and sum them, whereas the former can simply assume that each individual goal can be reached regardless of the current state.

Citations:

Russell, S. J. & Norvig, P. (2002), *Artificial Intelligence: A Modern Approach (2nd Edition)* , Prentice Hall .