# Comparison of Search Algorithms used in the Implement a Planning Search Project

by Ben Blazado

The following searches were performed on each of the three air cargo problems:

- A* with h_1, h_ignore_conditions, h_level_sum
- Breadth-First
- Depth-First
- Uniform Cost

The following sections are organized by problem, and the search results are summarized in a table this is ordered in ascending time (i.e. searches that take longer are at the bottom rows of the table).

## Air Cargo Problem 1

The goal for this problem was to move two pieces of cargo from two airports with two aircraft to two other airports. This problem was the simplest compared to the other two as it involved fewer cargo, aircraft, airports, and goals. The results of each search are below:

| Search | Heuristic | Exp. | Goal Tests | New Nodes | Plan Length | Time (Secs) |
|---|---|---|---|---|---|---|
| Depth First | | 21 | 22 | 84 | 20 | 0.03 |
| A* | h_ignore_preconditions | 41 | 43 | 170 | 6 | 0.06 |
| Breadth First | | 43 | 56 | 180 | 6 | 0.07 |
| Uniform Cost | | 55 | 57 | 224 | 6 | 0.08 |
| A* | h_1 | 55 | 57 | 224 | 6 | 0.08 |
| A* | h_pg_levelsum | 28 | 30 | 114 | 6 | 1.03 |

Depth First finds the solution the fastest however produces plan length of 20 steps, while the rest have a plan length of 6. A* with h_ignore_preconditions is the second fastest search. Although A* with h_pg_level_sum comes in last, it produces the second lowest number of new nodes. Note that Uniform Cost and A* with h_1 produce the same benchmarks, which is expected since they are essentially equivalent searches: h_1 produces a uniform cost of h_const = 1 for all nodes.

It is no surprise that Depth First was the fastest as it prioritizes going deep in the search tree to find a solution. While going deep in a search tree can find the solution the fastest, the solution may not be optimal. In this case, the resulting solution is not optimal with a 20 step-plan as compared to a 6-step plan for the others.

For the purposes of selecting one optimal search for this problem, A* with h_ignore_preconditions is selected as it was the second fastest and produced the plan length of 6:

- Load(C1, P1, SFO)
- Fly(P1, SFO, JFK)
- Unload(C1, P1, JFK)
- Load(C2, P2, JFK)
- Fly(P2, JFK, SFO)
- Unload(C2, P2, SFO)

Note that because the state space is small for this problem, the other searches also produce a plan length of 6 and a time of approximately 1 second or less and so can be considered optimal as well for this small problem.

## Air Cargo Problem 2

The goal for this problem was to move three pieces of cargo from three airports with three aircraft to three other airports. With more objects and more goals, this problem is more complex than the first. The results of each search are below:

| Search | Heuristic | Exp. | Goal Tests | New Nodes | Plan Length | Time (Secs) |
|---|---|---|---|---|---|---|
| Depth First |  | 624 | 625 | 5602 | 619 | 4.45 |
| A* | h_ignore_preconditions | 1450 | 1452 | 13303 | 9 | 6.89 |
| Breadth First |  | 3343 | 4609 | 30509 | 9 | 20.13 |
| A* | h_1 | 4853 | 4855 | 44041 | 9 | 22.55 |
| Uniform Cost |  | 4853 | 4855 | 44041 | 9 | 22.72 |
| A* | h_pg_levelsum | 659 | 661 | 5977 | 9 | 279.79 |

As in the previous problem, Depth First finds the solution the fastest, but results in a plan length of 619 which is not optimal (the other searches produced a plan length of 9). Also, Uniform Cost produces the exact benchmarks (to the second) as A* with h_1 as h_1 is simply a constant (i.e. the cost of a node in Uniform Cost is the same as the cost of a node in A* with h_1). A* with h_pg_levelsum comes in last, taking approximately 280 seconds but again, producing the second lowest new nodes (5977).

A* with h_ignore_preconditions appears to be the optimal search, again the second fastest in finding the solution (approximately 7 seconds) with a 9-step plan:

- 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)

## Air Cargo Problem 3

The goal for this problem was to move four pieces of cargo, using two aircraft, among four airports. While the number of aircraft is lower, the complexity of the problem is again increased due to the increased number of cargo pieces and airports.

| Search | Heuristic | Exp. | Goal Tests | New Nodes | Plan Length | Time (Secs) |
|---|---|---|---|---|---|---|
| Depth First | | 408 | 409 | 3364 | 392 | 2.47 |
| A* | h_ignore_preconditions | 5040 | 5042 | 44944 | 12 | 25.57 |
| A* | h_1 | 18235 | 18237 | 159716 | 12 | 90.42 |
| Uniform Cost | | 18235 | 18237 | 159716 | 12 | 90.75 |
| Breadth First | | 14663 | 18098 | 129631 | 12 | 120.16 |
| A* | h_pg_levelsum | 1133 | 1135 | 10025 | 12 | 806.62 |

As in the previous two problems and as explained in the first problem, Depth First finds the solution the fastest, but also results in a plan that is not optimal (392 steps compare to 12 steps for the other searches). Benchmarks for Uniform Cost and A* with h_1 remain equivalent as expected. A* with h_pg_levelsum comes in last again, taking over 10 minutes to complete, but again producing the second lowest number of new nodes.

A* with h_ignore_preconditions appears to be the optimal search, again the second fastest in finding the solution (approximately 26 seconds) with a 12-step plan:

- 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)

## Conclusions

A* with h_pg_levelsum appears to have the best heuristic in terms of producing fewer new nodes and having a smaller expansion than the other A* searches, Breadth First, and Uniform Cost. A* with h_pg_levelsum, for problem 3, produced only a quarter of new nodes (10025) as compared to A* with h_ignore_preconditions (44944). Unfortunately, creating the plan graph to compute the heuristic value was just too time-consuming. Although A* with h_ignore_preconditions was more focused on the goal, it took too long to create the data (the planning graph) that allowed it to better focus on the goal, and so this search was not selected as the optimal search for the air cargo problems.

Depth First appears to be the fastest search to find the goal, but the plan it produces is not an optimal plan. While searching deep allows for quickly achieving a state that satisfies all goals, the cost to arrive at such a state is largely ignored and the plan produced will likely not be as optimal. In problem 3 for instance, although Depth First found a state that satisfied all goals in 2.47 seconds, but the plan to achieve that goal was 392 steps, compared to 12 steps for all the other searches.

A* with h_ignore_preconditions is the optimal search across all the problem sets tested. The heuristic allowed the search to focus on the goal by assigning lower costs to nodes that had less goals needed to complete, enabling the search to head in the right direction, and arriving more quickly to the goal. Although A* with h_ignore_preconditions may not have been as focused as A* with h_pg_levelsum, A* with h_ignore_preconditions was significantly faster at finding the solution (for problem 3, 25.57 seconds compared 806.62 seconds for A* with h_pg_levelsum) and produced a plan with a smaller number of steps (for problem 3, 12 steps compared 392 steps for A* with h_pg_levelsum).