Heuristic Analysis

# Introduction

This analysis is a companion to the “Implement a Planning Search” Project as part of Udacity’s Artificial Intelligence NanoDegree. In this project, there were three problems, defined in classical Planning Domain Definition Language (PDDL), in the Air Cargo Domain, where there are three possible actions: Load(Cargo, Plane, Airport), Unload(Cargo, Plane, Airport) and Fly(Plane, From, To). The Goals in each problem were to have specific Cargo instances Unloaded at specific Airports.

# Air Cargo Problem #1:

## Initial State:

Cargo 1, Plane 1 at SFO; Cargo 2, Plane 2 at JFK.

## Goals:

Cargo 1 at JFK; Cargo 2 at SFO.

## Best Plan (6 Actions):

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 1 | Expansions | Goal Tests | New Nodes | Time Elapsed (seconds) | Plan Length | Optimal Plan? |
| Breadth First Search | 43 | 56 | 180 | 0.044 | 6 | Yes |
| Breadth First Tree Search | 1458 | 1459 | 5960 | 1.369 | 6 | Yes |
| Depth First Graph Search | 21 | 22 | 84 | 0.022 | 20 | No |
| Depth Limited Search | 101 | 271 | 414 | 0.137 | 50 | No |
| Uniform Cost Search | 55 | 57 | 224 | 0.058 | 6 | Yes |
| Recursive Best First Search with h\_1 | 4229 | 4230 | 17023 | 4.15 | 6 | Yes |
| Greedy Best First Graph Search with h\_1 | 7 | 9 | 28 | 0.008 | 6 | Yes |
| A\* Search with h\_1 | 55 | 57 | 224 | 0.056 | 6 | Yes |
| A\* Search with h\_ignore\_preconditions | 41 | 43 | 170 | 0.061 | 6 | Yes |
| A\* Search with h\_pg\_levelsum | 11 | 13 | 50 | 0.814 | 6 | Yes |

For Problem #1, 8 out of the 10 Search Algorithm/Heuristic Combinations found the optimal solution (plan length == 6). Depth First Graph Search and Depth Limited Search both returned non-optimal solutions (plan length > 6). **The best-performing algorithm was Greedy Best First Graph Search with h\_1,** which returned the optimal plan in ~0.008 seconds after completing 7 expansions, performing 9 goal tests, and exploring 28 nodes. Recursive Best First Search with h\_1 took the longest to return the optimal plan at ~4.150 seconds (~519 x longer) after completing 4229 expansions, performing 4230 goal tests, and exploring 17023 nodes. Breadth First Search, Uniform Cost Search, and all of the A\* Searches would all also be reasonable methods to solve this problem, as they all returned the optimal solution in less than 1 second.

# Air Cargo Problem #2:

## Initial State:

Cargo 1, Plane 1 at SFO; Cargo 2, Plane 2 at JFK; Cargo 3, Plane 3 at ATL.

## Goals:

Cargo 1 at JFK; Cargo 2 at SFO; Cargo 3 at SFO.

## Best Plan (9 Actions):

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Problem 2 | Expansions | Goal Tests | New Nodes | Time Elapsed (seconds) | Plan Length | Optimal Plan? |
| Breadth First Search | 3343 | 4609 | 30509 | 19.089 | 9 | Yes |
| Breadth First Tree Search | Failed | Failed | Failed | > 10 Minutes | Failed | No |
| Depth First Graph Search | 624 | 625 | 5602 | 4.622 | 619 | No |
| Depth Limited Search | Failed | Failed | Failed | > 10 Minutes | Failed | No |
| Uniform Cost Search | 4852 | 4854 | 44030 | 17.06 | 9 | Yes |
| Recursive Best First Search with h\_1 | Failed | Failed | Failed | > 10 Minutes | Failed | No |
| Greedy Best First Graph Search with h\_1 | 990 | 992 | 8910 | 3.39 | 15 | No |
| A\* Search with h\_1 | 4852 | 4854 | 44030 | 17.522 | 9 | Yes |
| A\* Search with h\_ignore\_preconditions | 1450 | 1452 | 13303 | 6.306 | 9 | Yes |
| A\* Search with h\_pg\_levelsum | 86 | 88 | 841 | 69.861 | 9 | Yes |

For Problem #2, 5 out of the 10 Search Algorithm/Heuristic Combinations found the optimal solution (plan length == 9). Depth First Graph Search and Greed Best First Graph Search with h\_1 both returned non-optimal solutions (plan length > 9). Breadth First Tree Search, Depth Limited Search and Recursive Best First Graph Search with h\_1 did not return any solutions within a 10 minute time frame. **The best-performing algorithm was A\* Search with h\_ignore\_preconditions,** which returned the optimal plan in ~6.306 seconds after completing 1450 expansions, performing 1452 goal tests, and exploring 13303 nodes. A\* Search with h\_pg\_levelsum took the longest to return the optimal plan at ~69.861 seconds (~11 x longer) after completing 86 expansions, performing 88 goal tests, and exploring 841 nodes. So despite the fact that it explored fewer nodes, the h\_pg\_levelsum heuristic did not return a result faster, as it is much more computationally expensive when compared to the h\_ignore\_preconditions heuristic.

# Air Cargo Problem #3:

## Initial State:

Cargo 1, Plane 1 at SFO; Cargo 2, Plane 2 at JFK; Cargo 3 at ATL; Cargo 4 at ORD.

## Goals:

Cargo 1, Cargo 3 at JFK; Cargo 2, Cargo 4 at SFO.

## Best Plan (12 Actions):

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1 ATL, JFK)

Unload(C1, P1, JFK)

Unload(C3, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Unload(C2, P2, SFO)

Unload(C4, P2, SFO)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Problem 3 | Expansions | Goal Tests | New Nodes | Time Elapsed (seconds) | Plan Length | Optimal Plan? |
| Breadth First Search | 14663 | 18098 | 129631 | 139.126 | 12 | Yes |
| Breadth First Tree Search | Failed | Failed | Failed | > 10 Minutes | Failed | No |
| Depth First Graph Search | 408 | 409 | 364 | 2.533 | 392 | No |
| Depth Limited Search | Failed | Failed | Failed | > 10 Minutes | Failed | No |
| Uniform Cost Search | 18235 | 18237 | 159716 | 74.771 | 12 | Yes |
| Recursive Best First Search with h\_1 | Failed | Failed | Failed | > 10 Minutes | Failed | No |
| Greedy Best First Graph Search with h\_1 | 5614 | 5616 | 49429 | 22.892 | 22 | No |
| A\* Search with h\_1 | 18235 | 18237 | 159716 | 74.046 | 12 | Yes |
| A\* Search with h\_ignore\_preconditions | 5040 | 5042 | 44944 | 23.717 | 12 | Yes |
| A\* Search with h\_pg\_levelsum | 315 | 317 | 2902 | 332.739 | 12 | Yes |

For Problem #3, 5 out of the 10 Search Algorithm/Heuristic Combinations found the optimal solution (plan length == 12). As in Problem #2, Depth First Graph Search and Greed Best First Graph Search with h\_1 both returned non-optimal solutions (plan length > 9), and Breadth First Tree Search, Depth Limited Search and Recursive Best First Graph Search with h\_1 did not return any solutions within a 10 minute time frame. Once again, **the best-performing algorithm was A\* Search with h\_ignore\_preconditions,** which returned the optimal plan in ~23.717 seconds after completing 5040 expansions, performing 5042 goal tests, and exploring 44944 nodes. A\* Search with h\_pg\_levelsum took the longest to return the optimal plan at ~332.739 seconds (~14 x longer) after completing 315 expansions, performing 317 goal tests, and exploring 2902 nodes.

# Summary:

The best performing Search Algorithm/Heuristic Combination across all three problems was the A\* Search with h\_ignore\_preconditions, which always returned the optimal solution in either the shortest time frame and/or a reasonable time frame. A\* Search with h\_1(no heuristic) and A\* Search with h\_pg\_levelsum also always returned the optimal solution, but h\_1 took longer because it always expanded more nodes, and levelsum took longer because it was more computationally expensive. Out of the “Uninformed” Search Algorithms, Breadth First Search and Uniform Cost Search also always returned the optimal solution, and were comparable to A\* Search with h\_1 in time elapsed. As the problems became more complex, Uniform Cost Search began to out-perform Breadth First Search as it expanded less nodes.