# Written Analysis – Planning Search

The current assignment implemented a classical planning problem using the Planning Domain Definition Language. It solved a deterministic air cargo transport system problem with uninformed (non-heuristic) planning searches (Breadth First Search, Depth First Greedy Search, Greedy Best First Graph Search) and heuristic searches using A\* (ignore preconditions, level-sum). Results are found in the tables below with discussion to follow.

## Non-Heuristic Search Results

|  |  |  |  |
| --- | --- | --- | --- |
| Problem 1 | Breadth First Search | Depth First Greedy Search | Greedy Best First Graph Search |
| Plan Length | 6 | 12 | 6 |
| Time (s) | 0.085 | 0.017 | 0.0198 |
| Expansions | 43 | 12 | 7 |
| Goal Tests | 56 | 13 | 9 |
| New Nodes | 180 | 48 | 28 |
| Optimal | Yes | No | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| Problem 2 | Breadth First Search | Depth First Greedy Search | Greedy Best First Graph Search |
| Plan Length | 9 | 346 | 9 |
| Time (s) | 17.56 | 2.85 | 2.62 |
| Expansions | 3401 | 350 | 550 |
| Goal Tests | 4672 | 351 | 552 |
| New Nodes | 31049 | 3142 | 4950 |
| Optimal | Yes | No | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| Problem 3 | Breadth First Search | Depth First Greedy Search | Greedy Best First Graph Search |
| Plan Length | 12 | 3335 | 22 |
| Time (s) | 88.24 | 106.26 | 23.51 |
| Expansions | 14491 | 3491 | 4031 |
| Goal Tests | 17947 | 3492 | 4033 |
| New Nodes | 128184 | 29332 | 35794 |
| Optimal | Yes | No | No |

## Heuristic Search Results

|  |  |  |  |
| --- | --- | --- | --- |
| Problem 1 | A\* h1 | A\* Ignore Preconditions | A\* Level Sum |
| Plan Length | 6 | 6 | 6 |
| Time (s) | 0.113 | 0.074 | 2.489 |
| Expansions | 55 | 41 | 32 |
| Goal Tests | 57 | 43 | 34 |
| New Nodes | 224 | 170 | 138 |
| Optimal | Yes | Yes | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| Problem 2 | A\* h1 | A\* Ignore Preconditions | A\* Level Sum |
| Plan Length | 9 | 9 | 9 |
| Time (s) | 26.634 | 8.946 | 196.141 |
| Expansions | 4761 | 1450 | 168 |
| Goal Tests | 4763 | 1452 | 170 |
| New Nodes | 43206 | 13303 | 1618 |
| Optimal | Yes | Yes | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| Problem 3 | A\* h1 | A\* Ignore Preconditions | A\* Level Sum |
| Plan Length | 12 | 12 | 12 |
| Time (s) | 117.543 | 36.719 | 1482.073 |
| Expansions | 17783 | 5003 | 1037 |
| Goal Tests | 17785 | 5005 | 1039 |
| New Nodes | 155920 | 44586 | 9683 |
| Optimal | Yes | Yes | Yes |

## Optimal Sequence of Actions

### Problem 1

### Load(C1, P1, SFO)

### Load(C2, P2, JFK)

### Fly(P1, SFO, JFK)

### Fly(P2, JFK, SFO)

### Unload(C1, P1, JFK)

### Unload(C2, P2, SFO)

### Problem 2

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)

### Problem 3

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(P1, ATL, JFK)

Fly(P2, ORD, SFO)

Unload(C4, P2, SFO)

Unload(C3, P1, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

## Discussion

The Breadth First Search was the only non-heuristic search algorithm to provide an optimal search result for all three problems, whereas problem optimality was found with all heuristic search algorithms.

With any state space problem, the goal is to find a path connecting the initial state to the goal state. Heuristics make it easier to find a path but they require human ingenuity to define good domain-specific heuristics for search problems. For planning searches adding more edges to the graph or grouping nodes to form an abstraction with fewer states in the state space makes it easier to search1. These are some of the reasons that the heuristic searches were able to find an optimal solution to each problem. Although the heuristic searches found optimal solutions they did so with different times and expansions. The A\* Ignore Preconditions was the fastest for each problem, solving problem 3 (the most difficult problem) in under 40 seconds. Interestingly, the faster algorithm needed 5 times the expansions of the slowest (A\* Level Sum) algorithm. It is difficult to say what heuristic is best, but I would choose A\* Ignore preconditions for the speed. It seems that a greater amount of memory is needed for A\* Ignore preconditions than A\* Level Sum, but the speed to solve the problem is so much greater that the extra memory consumption is worth it.

1. Russell, Stuart, Peter Norvig, and Artificial Intelligence. "A modern approach." Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs 25 (1995): 27.