Heuristic analysis

# Optimal solutions

Uniform cost search is guaranteed to find an optimal solution—and so is breadth first search in this case, because our path cost is simply the number of nodes. We have used this algorithm to find the solutions reproduced below.

Clearly depth first search will not guarantee an optimal solution.

Given an admissible heuristic, such as *ignore pre-conditions,* A\* search will give an optimal result. The *level sum* heuristic is only admissible for goals that are independent, which I haven’t guaranteed. Despite this it found optimal solutions for these three problems.

## Problem 1

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 2

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

## Problem 3

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1, ATL, JFK)

Unload(C3, P1, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Unload(C4, P2, SFO)

Unload(C2, P2, SFO)

# Results

*Breadth first*, *depth first* and *uniform cost* are the uninformed searches I tested. *A\* ignore pre-conditions* and *A\* level sum* are the heuristic searches.

Time elapsed

|  |  |  |  |
| --- | --- | --- | --- |
|  | Problem 1 | Problem 2 | Problem 3 |
| Breadth first | 0.03 | 10 | 90 |
| Depth first | 0.006 | 3 | 3 |
| Uniform cost | 0.03 | 10 | 40 |
| A\* ignore pre-cond | 0.02 | 3 | 10 |
| A\* level sum | 0.6 | 80 | 200 |

Figures are in seconds, to one significant figure. I have quoted the median time of three runs.

## Node expansions, goal tests

|  |  |  |  |
| --- | --- | --- | --- |
|  | Problem 1 | Problem 2 | Problem 3 |
| Breadth first | 43, 56 | 3343, 4609 | 14663, 18098 |
| Depth first | 12, 13 | 582, 583 | 627, 628 |
| Uniform cost | 55, 57 | 4853, 4855 | 16961, 16963 |
| A\* ignore pre-cond | 41, 43 | 1450, 1452 | 4444, 4446 |
| A\* level sum | 11, 13 | 86, 88 | 229, 231 |

# Comparison of algorithms

## Uninformed searches

Depth first search doesn’t expand any nodes that aren’t part of the solution. In this sense, it is more efficient than the other algorithms.

Uniform cost search and breadth first search are very similar algorithms in this case. The key difference is that breadth first search is performing goal tests before nodes are added to the frontier, rather than when they are expanded. This gives it a small advantage in the number of nodes explored.

Uniform cost search is faster than breadth first search on Problem 3. This is because it benefits from a better engineered implementation in this project.

## Heuristic searches

The level sum heuristic was very slow; much slower than ignore pre-conditions. This is thanks to the time it spends building planning graphs.

The planning graphs do make level sum far more accurate than ignore pre-conditions—it visited nowhere near as many nodes.

# Discussion

In Problem 1 ignore pre-conditions was the fastest, though uniform cost’s performance was similar.

In the bigger problems 2 and 3 ignore pre-conditions was again the fastest, although not the most efficient in terms of nodes explored.

Despite these results, level sum remains a more accurate heuristic, and for even bigger problems I would expect it to overtake ignore pre-conditions. Exploring nodes takes exponential time, and the planning graphs save us that, at the cost of polynomial time for their creation.

# Conclusions

The winning algorithm was A\* ignore pre-conditions. It produced an optimal solution for each problem in the fastest time.