# *Planning search analysis.*

*Jose Tomas Rubio Lorente*

In this document we will analyse the heuristics used and results obtained in  AIND-Planing project.

These are the **optimal solutions** obtained for each problem:

**Problem 1**

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

**Problem 2**

Load(C2, P2, JFK)
Load(C1, P1, SFO)
Load(C3, P3, ATL)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

**Problem 3**

Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

## Performance comparison:

## Uninformed searches:

Problem 1:

| **Planning algorithm** | **Optimal** | **Path length** | **Node expansions** | **Goal tests** | **New nodes** | **Time elapsed (seconds)** |
|---|---|---|---|---|---|---|
| Breath first search | YES | 6 | 43 | 56 | 180 | 0.028 |
| Depth first graph search | NO | 12 | 12 | 13 | 48 | 0.008 |
| Uniform cost search | YES | 6 | 55 | 57 | 224 | 0.035 |
| Greedy best first grapph search with h1 | YES | 6 | 7 | 9 | 21 | 0.007 |

Problem 2:

| **Planning algorithm** | **Optimal** | **Path length** | **Node expansions** | **Goal tests** | **New nodes** | **Time elapsed (seconds)** |
|---|---|---|---|---|---|---|
| Breath first search | YES | 9 | 3343 | 4609 | 30509 | 13.957 |
| Depth first graph search | NO | 575 | 582 | 583 | 5211 | 3.116 |
| Uniform cost search | YES | 9 | 4853 | 4855 | 44041 | 12.216 |
| Greedy best first grapph search with h1 | NO | 17 | 998 | 1000 | 8982 | 2.540 |

Problem 3:

| **Planning algorithm** | **Optimal** | **Path length** | **Node expansions** | **Goal tests** | **New nodes** | **Time elapsed (seconds)** |
|---|---|---|---|---|---|---|
| Breath first search | YES | 12 | 14663 | 18098 | 129631 | 102.52 |
| Depth first graph search | NO | 596 | 627 | 628 | 5176 | 3.270 |
| Uniform cost search | YES | 12 | 18151 | 18153 | 159038 | 53.463 |
| Greedy best first grapph search with h1 | NO | 27 | 5398 | 5400 | 47665 | 15.933 |

As shown in the tables, **Breath first search** and **Uniform cost search** are the only ones giving an **optimal** action plan.

In terms of execution speed and memory usage, **Depth first search** is the **fastest** and uses **less memory** by far, but it gives not-optimal solutions with huge difference in the plan length compared to the other three algorithms analyzed. These behaviour of not finding an optimal path is not a surprise according to the video lectures (Lesson 8, section 25), as DFS is not complete and could even not find a path.

If we want to get always the optimal solution, we should use **Breath first search** (BFS) or **Uniform cost search** (UCS) as they are complete and optimal. BFS is slower than UCS, but uses less memory. So the recommendation in this case is using Uniform cost search if execution time is critical and we can assume the memory usage.

In case having an optimal plan is not critical, the recommendation is to use **Greedy best first graph search with h1**, as it is more efficient than BFS and UCS in terms of memory usage and execution time, and the solution plan length is much shorter than DFS.

## Informed searches:

Now is the turn of analysing A* algorithm with different heuristics:

Problem 1:

| Planning algorithm | Optimal | Path length | Node expansions | Goal tests | New nodes | Time elapsed (seconds) |
|---|---|---|---|---|---|---|
| A* ignore preconditions | YES | 6 | 41 | 43 | 170 | 0.038 |
| A* level-sum | YES | 6 | 11 | 13 | 50 | 0.940 |

Problem 2:

| Planning algorithm | Optimal | Path length | Node expansions | Goal tests | New nodes | Time elapsed (seconds) |
|---|---|---|---|---|---|---|
| A* ignore preconditions | YES | 9 | 1450 | 1452 | 13303 | 4.361 |
| A* level-sum | YES | 9 | 86 | 88 | 841 | 149.980 |

Problem 3:

| Planning algorithm | Optimal | Path length | Node expansions | Goal tests | New nodes | Time elapsed (seconds) |
|---|---|---|---|---|---|---|
| A* ignore preconditions | YES | 12 | 5038 | 5040 | 44926 | 17.115 |
| A* level-sum | YES | 12 | 316 | 318 | 2919 | 793.535 |

In case of heuristics searches, both algorithms analysed get to an optimal solution, as they are both optimal and complete. Regarding their performance, A* ignore preconditions heuristic executes much faster than A* with level sum heuristic, but it spent about 20 times more memory so the recommendation is to use ignore preconditions if the time is critical and memory usage is supported, or use level-sum if there are limitations in the amount of memory available.

The best algorithm for solving this problem seems to be A* with ignore preconditions, as it gets always to an optimal plan and uses less time and memory than other non-heuristics algorithms with optimal solution, except for problem 1 where the complexity is lower and the difference in time is reduced to milliseconds.