# Heuristic analysis

## Part 1. Planning problems

After experimenting with 3 given PDDL problems (their states and goals), metrics were documented for non-heuristic planning solution searches.

*Table 1*

Breadth first tree search was aborted in problems 2 and 3 because it exceeded 10 minutes, similarly depth limited search was aborted in problem 3 as it exceeded the same time limit as well; those are marked as *N/A* in the *Table 1*.

After running tests following optimal plans were computed (*Table 2*).

|  |  |  |
| --- | --- | --- |
| Problem 1 | Problem 2 | Problem 3 |
| Load(C1, P1, SFO)  Load(C2, P2, JFK)  Fly(P2, JFK, SFO)  Unload(C2, P2, SFO)  Fly(P1, SFO, JFK)  Unload(C1, P1, JFK) | Load(C1, P1, SFO)  Load(C2, P2, JFK)  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) | Load(C1, P1, SFO)  Fly(P1, SFO, ATL)  Load(C3, P1, ATL)  Fly(P1, ATL, JFK)  Load(C2, P1, JFK)  Unload(C3, P1, JFK)  Unload(C1, P1, JFK)  Fly(P1, JFK, ORD)  Load(C4, P1, ORD)  Fly(P1, ORD, SFO)  Unload(C4, P1, SFO)  Unload(C2, P1, SFO) |

*Table 2*

*Figure 1*

If algorithm is chosen based solely on time it takes to find the solution then depth first graph search is the fastest one from non-heuristic solutions (*Figure 1*), but it does not find the optimal solution, just the first matching one, e.g. in Problem 3 its plan is 875 steps (*Table 1*), where uniform cost search finds a solution with just 12 steps. Instead I would recommend using uniform cost search as it outperforms breadth first search and finds shortest search plan.

## Part 2. Domain-independent heuristics

Same problems that were mentioned above were tested with graph search algorithms that used heuristics (*Table 3*).

*Table 3*

*Figure 2*

Although A\* ignore preconditions search is faster (*Figure 2*) it expands far more nodes that A\* level-sum (*Table 3*) because when algorithm ignores preconditions there are more actions allowed (edges created). Furthermore it does not calculate optimal plan length in Problem 3, this might be because A\* level sum is more accurate that ignore preconditions heuristic.

## Summary

Both non-heuristic and heuristic searches find optimal plans but there is no clear winner when comparing their execution for Problem 3. Best non-heuristic is a uniform cost search which finds **optimal plan** (12 steps) by expanding 7302 nodes and does it in 23 seconds. Fastest heuristic search is A\* ignore preconditions search, it expands less nodes (2829) and does it under 11 seconds but it finds **non optimal plan** (13 steps), it adds more edges to the graph, making it strictly easier to find a path [1]! Last heuristic search is A\* level sum, which finds **optimal plan** (12 steps) “it is much more accurate than the number- of-unsatisfied-goals heuristic” [2] and it does so by expanding only 167 nodes but takes longer and finishes in 78 seconds.

## References

[1] Stuart J. Russell and Peter Norvig, Artificial Intelligence. A Modern Approach. Third Edition, 376 10.2.3 Heuristics for planning

[2] Stuart J. Russell and Peter Norvig, Artificial Intelligence. A Modern Approach. Third Edition, 382 10.3.1 Planning graphs for heuristic estimation