# Planning Project Heuristic Analysis

An air cargo problem was given defined in classical PDDL.

The action schema is as follows:

Action(Load(c, p, a),

PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)

EFFECT: ¬ At(c, a) ∧ In(c, p))

Action(Unload(c, p, a),

PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)

EFFECT: At(c, a) ∧ ¬ In(c, p))

Action(Fly(p, from, to),

PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)

EFFECT: ¬ At(p, from) ∧ At(p, to))

Three different problems were given with the following initial states and goals:

| Problem 1 (P1) | Problem 2 (P2) | Problem 3 (P3) |
|---|---|---|
| Init(At(C1, SFO) ∧ At(C2, JFK)<br>∧ At(P1, SFO)<br>∧ At(P2, JFK)<br>∧ Cargo(C1) ∧ Cargo(C2)<br>∧ Plane(P1) ∧ Plane(P2)<br>∧ Airport(JFK)<br>∧ Airport(SFO))<br>Goal(At(C1, JFK) ∧ At(C2, SFO)) | Init(At(C1, SFO) ∧ At(C2, JFK)<br>∧ At(C3, ATL) ∧ At(P1, SFO)<br>∧ At(P2, JFK) ∧ At(P3, ATL)<br>∧ Cargo(C1) ∧ Cargo(C2)<br>∧ Cargo(C3) ∧ Plane(P1)<br>∧ Plane(P2) ∧ Plane(P3)<br>∧ Airport(JFK)<br>∧ Airport(SFO)<br>∧ Airport(ATL))<br>Goal(At(C1, JFK) ∧ At(C2, SFO)<br>∧ At(C3, SFO)) | Init(At(C1, SFO) ∧ At(C2, JFK)<br>∧ At(C3, ATL) ∧ At(C4, ORD)<br>∧ At(P1, SFO) ∧ At(P2, JFK)<br>∧ Cargo(C1) ∧ Cargo(C2)<br>∧ Cargo(C3) ∧ Cargo(C4)<br>∧ Plane(P1) ∧ Plane(P2)<br>∧ Airport(JFK)<br>∧ Airport(SFO)<br>∧ Airport(ATL)<br>∧ Airport(ORD))<br>Goal(At(C1, JFK) ∧ At(C3, JFK)<br>∧ At(C2, SFO)<br>∧ At(C4, SFO)) |

## Optimal plans for Problems 1, 2, and 3

Using the run_search script, the metrics for various search algorithms were gathered and used to compare non-heuristic search and heuristic search strategies for each problem.

The optimal plan for each problem is given below:

| P1 | P2 | P3 |
|---|---|---|
| Load(C2, P2, JFK)<br>Load(C1, P1, SFO)<br>Fly(P2, JFK, SFO)<br>Unload(C2, P2, SFO)<br>Fly(P1, SFO, JFK)<br>Unload(C1, P1, JFK) | Load(C2, P2, JFK)<br>Load(C1, P1, SFO)<br>Fly(P2, JFK, ATL)<br>Load(C3, P2, ATL)<br>Fly(P1, SFO, JFK)<br>Unload(C1, P1, JFK)<br>Fly(P2, ATL, SFO)<br>Unload(C2, P2, SFO)<br>Unload(C3, P2, SFO) | Load(C2, P2, JFK)<br>Load(C1, P1, SFO)<br>Fly(P2, JFK, ORD)<br>Load(C4, P2, ORD)<br>Fly(P1, SFO, ATL)<br>Load(C3, P1, ATL)<br>Fly(P1, ATL, JFK)<br>Unload(C1, P1, JFK)<br>Unload(C3, P1, JFK)<br>Fly(P2, ORD, SFO)<br>Unload(C2, P2, SFO)<br>Unload(C4, P2, SFO) |

### Comparison of non-heuristic search result metrics

| | Node Expansions | | | Goal Tests | | | Time elapsed in seconds | | | Optimal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P1 | P2 | P3 | P1 | P2 | P3 | P1 | P2 | P3 | P1 | P2 | P3 |
| breath_first_search | 43 | 769 | 14663 | 56 | 979 | 18098 | 0.0412 | 2.819 | 131.568 | Yes | Yes | Yes |
| depth_first_graph_search | 12 | 41 | 627 | 13 | 42 | 628 | 0.0124 | 0.131 | 4.515 | Yes | No | No |
| uniform_cost_search | 55 | 992 | 18223 | 57 | 994 | 18225 | 0.0481 | 4.532 | 497.198 | Yes | Yes | Yes |

For problem 1, the best non-heuristic search planning method is Breath First Graph Search. It is the fastest, it's optimal, it expands the fewest number of nodes and tests the fewest goals.

For problems 2 and 3, the best choice is Breath First Search because it is optimal even though it's slower than Breath First Graph Search but the latter is not optimal.

Breath First Search is the best non-heuristic search planning method overall, among the three that were tested. It is optimal for all three problems and faster than Uniform Cost Search in all cases.

Depth First Graph Search was not chosen as best despite being the fastest overall because it is not optimal for two out of three problems. As stated in chapter 3 of "Artificial Intelligence: A Modern Approach", depth-first search is neither complete nor optimal but breadth-first search and uniform-cost search are both complete and optimal. Therefore the results obtained are not surprising.

## Comparison of heuristic search result metrics

| | Node Expansions | | | Goal Tests | | | Time elapsed in seconds | | | Optimal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P1 | P2 | P3 | P1 | P2 | P3 | P1 | P2 | P3 | P1 | P2 | P3 |
| astar_search h_ignore_preconditions | 41 | 452 | 5118 | 43 | 454 | 5120 | 0.060 | 2.924 | 113.314 | Yes | Yes | Yes |
| astar_search h_pg_levelsum | 11 | 93 | | 13 | 95 | | 4.869 | 1391.208 | | Yes | Yes | |
| astar_search h1 | 55 | 992 | 18223 | 57 | 994 | 18225 | 0.057 | 5.344 | 595.854 | Yes | Yes | Yes |

For problem 1, A* search h1 is the fastest. It does however have more node expansions and goal tests than A* search with “level sum” but the latter is much slower.

For problem 2, A* with “ignore preconditions” is the fastest, it also has fewer node expansions and goal tests than the second faster heuristic which is A* search h1.

For problem 3, A* with “ignore preconditions” is the best heuristic. It is faster than A* search h1, with fewer node expansions and goal tests. A* with “level sum” runs longer than 10 minutes, in fact I had to terminate it after about an hour because it never finished running.

These metrics show that A* with the “ignore preconditions” is the best heuristic overall if we consider execution time, no extension and goal test and the fact that it does provide a result in a somewhat timely fashion for all three problems.

The A* search with “level sum” is the slowest of all three heuristic planning methods for all three problems.

## Are heuristic search planning methods better than non-heuristic search planning methods for all problems?

For problem 1, non-heuristic search planning methods performs better than all heuristic methods in terms of speed. The fastest heuristic method is slower than the slowest of the three non-heuristic methods tested. All search methods are optimal for problem 1.

For problem 2, only two non-heuristic methods are optimal, the Breath First Search and Uniform Cost Search. Breath First Search is faster than the best heuristic method, A* Search with "ignore preconditions". However, Breath First Search expands more nodes 769 versus 452 for A* with "ignore preconditions" and it performs more goal tests, 979 versus 454.

For problem 3, the best optimal search planning method is the heuristic A* Search with "ignore preconditions". The fastest method is the non-heuristic Depth First Graph Search, it is however, not optimal.

If only one method must be chosen, A* Search with "ignore preconditions" is the best choice. It is the best search planning method for problems 2 and 3 based on the tests. While it is not the top performing method for problem 1 it is optimal for that problem and is faster than most other methods. This is consistent with what is stated in section 11.2 of "Artificial Intelligence: A Modern Approach" 2nd edition,

a good heuristic makes forward and backward search efficient and can be used with A* search to find optimal solutions.