**Planning Search Analysis**

by Ewa Tulodziecka

In this project, a planning search agent has been implemented to solve deterministic logistic planning problems in an Air Cargo transport system. For the sake of the problem, a Planning Graph has been applied as a method with automatic heuristics embedded in A* search whose search result metrics have been compared against several uninformed non-heuristic search methods (incl. breadth-first, depth-first, uniform cost etc.).

## Optimal Plan for the problems

The 3 given problems in Air Cargo domain have the following action schema:

```
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))
```

And their initial states and goals are as follows:

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

The goals above can be achieved by many different algorithms but the optimal plans for them are the following:

| Problem 1 | Problem 2 | Problem 3 |
|---|---|---|
| Plan length: 6 | Plan length: 9 | Plan length: 12 |
| Load(C1, P1, SFO)<br>Load(C2, P2, JFK)<br>Fly(P2, JFK, SFO)<br>Unload(C2, P2, SFO)<br>Fly(P1, SFO, JFK)<br>Unload(C1, P1, JFK) | Load(C1, P1, SFO)<br>Load(C2, P2, JFK)<br>Load(C3, P3, ATL)<br>Fly(P2, JFK, SFO)<br>Unload(C2, P2, SFO)<br>Fly(P1, SFO, JFK)<br>Unload(C1, P1, JFK)<br>Fly(P3, ATL, SFO)<br>Unload(C3, P3, SFO) | Load(C1, P1, SFO)<br>Load(C2, P2, JFK)<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) |

# Results

## Non-heuristic searches

Problem 1 results

| Search algorithm | Optimal | Plan Length | Node Expansions | Time elapsed |
|---|---|---|---|---|
| Breadth First Search | Yes | 6 | 43 | 0.03 |
| Breadth First Tree Search | Yes | 6 | 1458 | 0.97 |
| Depth First Graph Search | No | 12 | 12 | 0.01 |
| Depth Limited Search | No | 50 | 101 | 0.09 |
| Uniform Cost Search | Yes | 6 | 55 | 0.04 |

Problem 2 results

| Search algorithm | Optimal | Plan Length | Node Expansions | Time elapsed |
|---|---|---|---|---|
| Breadth First Search | Yes | 9 | 3343 | 14.00 |
| Breadth First Tree Search | - | - | - | - |
| Depth First Graph Search | No | 466 | 476 | 2.36 |
| Depth Limited Search | - | - | - | - |
| Uniform Cost Search | Yes | 9 | 4834 | 13.63 |

Problem 3 results

| Search algorithm | Optimal | Plan Length | Node Expansions | Time elapsed |
|---|---|---|---|---|
| Breadth First Search | Yes | 12 | 14663 | 104.97 |
| Breadth First Tree Search | - | - | - | - |
| Depth First Graph Search | No | 1442 | 1511 | 12.85 |
| Depth Limited Search | - | - | - | - |
| Uniform Cost Search | Yes | 12 | 18221 | 55.32 |

## Non-heuristic search algorithms analysis

Breadth First Tree Search and Depth Limited Search have been discarded in the problem 2 and 3 comparisons as they were not able to find a solution within a reasonable amount of time. Thus, for the $2^{nd}$ and $3^{rd}$ problems only Breadth First Search (BFS), Depth First Graph Search (DFS) and Uniform Cost Search (UCS) have been used.

For all the three problems BFS and UCS yielded an optimal action plan. In terms of execution time and memory usage, the best performing algorithm was Depth First Graph Search as it is the fastest one and uses the least memory (indicated by node expansions metric) while executing (in all three problems in question). However, the plan lengths obtained by DFS were significantly longer than those generated by the two other algorithms (Problem 1: 12 vs 6, Problem 2: 466 vs 9, Problem 3: 1442 vs 12).

If the optimal plan length is a matter of concern then BFS is the recommended algorithm due to the fact that it is complete and optimal (however, it may have a long running time if the problem's branching factor is large) and takes less memory and time (except for the problem 3 where BFS takes almost twice as much time as UCS) than UCS does. On the other hand, if speed and memory are major concerns then the best choice would be DFS which yields non-optimal plans (in problem 2 and 3 the yielded plans are much longer then the optimal ones). Thus, the DFS choice comes at a greater cost which results from execution of significantly longer action plans.

## Informed (heuristic) searches

Problem 1

| Search algorithm | Optimal | Plan Length | Node Expansions | Time elapsed |
|---|---|---|---|---|
| A* Search with h1 heuristic | Yes | 6 | 55 | 0.04 |
| A* Search with Ignore Preconditions heuristic | Yes | 6 | 41 | 0.04 |
| A* Search with Level Sum heuristic | Yes | 6 | 11 | 0.88 |

Problem 2

| Search algorithm | Optimal | Plan Length | Node Expansions | Time elapsed |
|---|---|---|---|---|
| A* Search with h1 heuristic | Yes | 9 | 4834 | 12.40 |
| A* Search with Ignore Preconditions heuristic | Yes | 9 | 1450 | 4.35 |
| A* Search with Level Sum heuristic | Yes | 9 | 86 | 182.64 |

Problem 3

| Search algorithm | Optimal | Plan Length | Node Expansions | Time elapsed |
|---|---|---|---|---|
| A* Search with h1 heuristic | Yes | 12 | 18221 | 55.69 |
| A* Search with Ignore Preconditions heuristic | Yes | 12 | 5040 | 17.47 |
| A* Search with Level Sum heuristic | Yes | 12 | 314 | 988.62 |

## Informed (heuristic) search algorithms analysis

In this case all three heuristics yielded an optimal action plan, however, their performance in terms of memory usage and running time substantially vary.

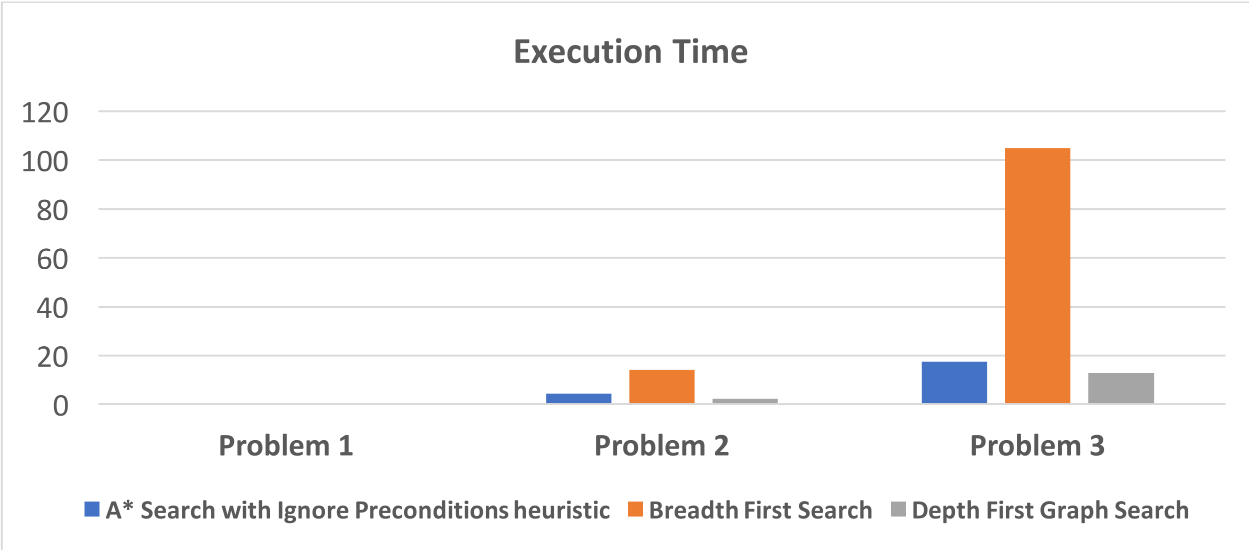
From the execution time perspective, the best search strategy in all three problems under consideration is A* Search with Ignore Preconditions heuristic. Its time required to reach a goal was substantially shorter than those required by h_1 and Level Sum heuristics. The second best strategy in terms of running time was h1 heuristic which is essentially the same algorithm as UCS. The worst performance in that matter has been observed for Level Sum heuristic which execution times for problem 2 and 3 were 3min and 16.5min accordingly. This stems from the high and complex computation incurred by the heuristic in order to limit search space significantly, and thus the node expansions. Due to the same fact, the Level Sum heuristic has been proved to be the best solution when memory usage is a matter of concern.

When all the aforementioned factors taken into account together, the Ignore preconditions heuristic is recommended as it yields an optimal solution in the lowest execution time with a reasonable number of nodes expanded.

## Informed vs Uninformed Search Strategies

In this section, the best heuristic, namely Ignore Preconditions heuristic, will be evaluated against the best non-heuristic search planning methods based on each of metrics considered – memory usage and execution time.

The following figure presents DFS (the best non-heuristic methods in terms of execution time), BFS (the best optimal non-heuristic method from all considered) and A* with Ignore Preconditions heuristic. What can be noticed is the fact that the execution time difference between A* with the heuristic is not much higher than the one of DFS. And it is also worth pointing out the fact, that the A* with the heuristic yields the optimal plan what is not the case for DFS. This shows how the A* with the heuristic is efficient (great time saving in comparison to BFS) and how little cost in terms of running time is incurred to achieve the optimal plan.




In the next figure, the A* search, BFS and UCS are compared based on the number of node expansions. DFS has been excluded as it performs the worst in that metric. For all three problems, the A* search with the Ignore Preconditions heuristic outperforms the two other algorithms. The discrepancy between them widens as the problem complexity expands. This figure clearly shows the competitive advantage of the A* with the heuristic over the rest in the memory usage metric. This leads us to conclusion that **A* Search with Ignore Preconditions heuristic** is the best choice for each of the Air Cargo problems analyzed. Also, it illustrates the benefits arising from using informed search algorithm with custom heuristics over uninformed search methods when the aim is to obtain the optimal plan. The performance benefits, namely memory usage and running time, are considerable, also the fact that the trade-off between those two can be adjusted by a proper heuristic is in favor of informed search strategies.