Planning (Lesson 11): Heuristic Analysis

Jörg Döpfert

March 27, 2017

I begin this report with an evaluation of the results of selected search methods. Optimal plans can be found in Section 2, and detailed metrics for all problems in consideration are displayed in Table 1.

1 Evaluation of the search methods

In Figure 1, the run times of selected searches are displayed for all three air cargo problems. Note that only for problem 1 all searches finished in less than 10 minutes (no data was collected for searches that took longer).

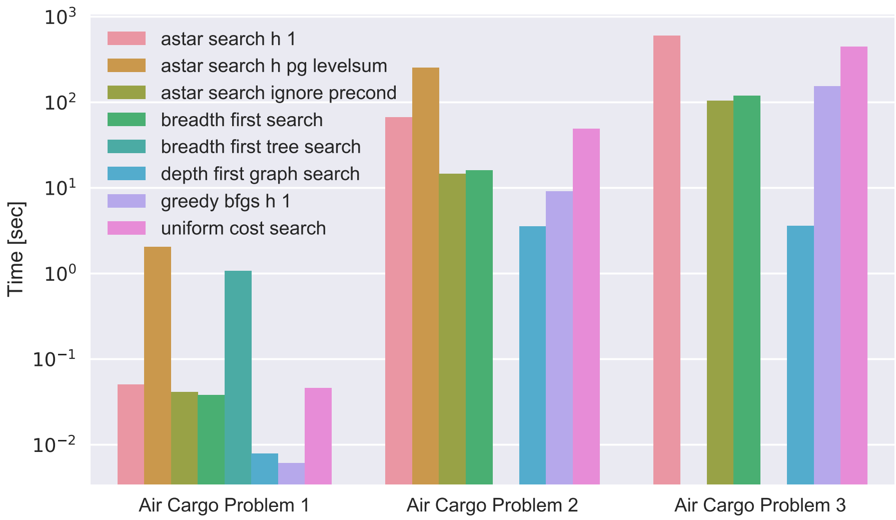

Figure 1: Search run time.

Among the fastest searches for all problems are `depth first graph search` and `greedy best first graph search`. However, those methodes do not find optimal plans in general, as can be seen in Figure 2 (both methods have a larger plan lenghts than the optimal `breadth first search` solution for problems 2 and 3).

The `breadth first search` always finds an optimal plan [1], but at the expense of speed: It is always slower than for example `depth first graph search` since it expands much more nodes (compare Figure 3). The slowest uninformed search seems to be `breadth first tree search`: It expands a lot of nodes, since it does not keep track of nodes that have already been visited.

Compared to `uniform cost search`, which is also guaranteed to find an optimal solution [1], `breadth first search` is a bit faster and also explores less nodes for all problems.

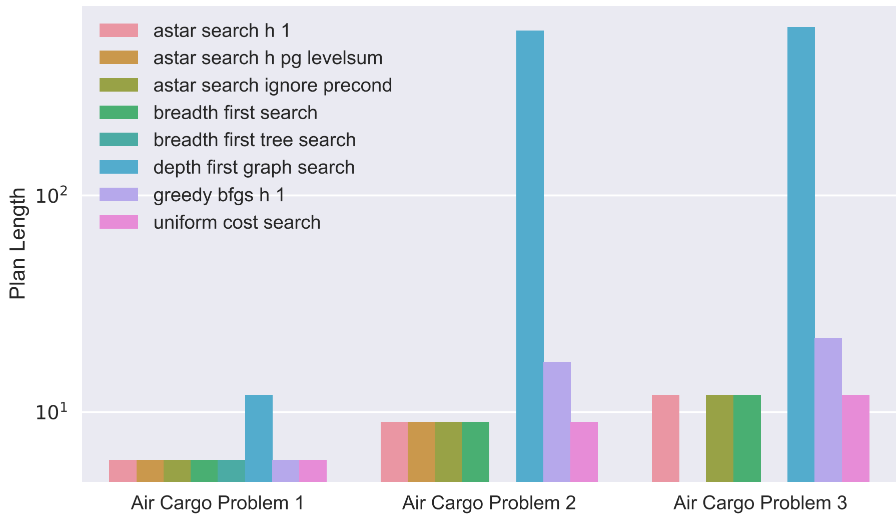

Figure 2: Number of steps in the solution plan

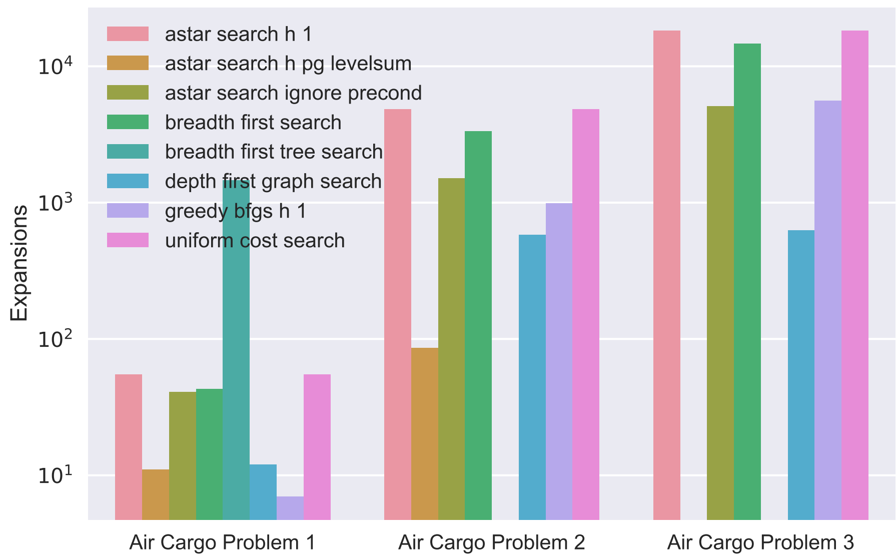

Figure 3: Number of expanded nodes.

The `uniform cost search` and `a star search h1` explore the same amount of nodes, since they are virtually identical: The former choses the node with the cheapest path cost, and the latter choses the node with the cheapest path cost plus a constant.

Among the informed searches, `astar search ignore precond` is much faster than `astar search pg levelsum`. However, the latter expands much less nodes: This is expected, since the `levelsum` heuristic calculated using a planning graph should be much stronger than simply ignoring the action preconditions. The slower runtime can be attributed to the computational cost of constructing the planning graphs.

The informed searches find optimal plans for all problems[1]. For the `astar search ignore precond`, this is a general result, since it uses an admissible heuristic for this problem. The `astar search pg levelsum` is not neccessarily admissible, but has shown good performance according to ref. [1].

The informed searches expand less nodes than the optimal uniformed searches, as

[1]excluding problem 3, where no result could be observed after 10 minutes for the planning graph search.

expected. When it comes to speed, `astar search ignore precond` is every similar to `breadth first search`. However, `astar search pg levelsum` is much slower, which again can probably be attributed to the large overhead for building the planning graphs.

2 Optimal plans

The plans presented here reflect the solutions found with breadth first search (which always finds the optimal solution).

2.1 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)
```

2.2 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)
```

2.3 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)
```

3 All metrics

Metrics for all problems in consideration are displayed in Table 1.

Table 1: Performance of the different custom heuristics.

Problem	Search Method	Plan Lenght	Time [sec]	New Nodes	Goal Tests	Expansions
1	astar search h 1	6	0.05	224	57	55
1	astar search h pg levelsum	6	2.06	50	13	11
1	astar search ignore precond	6	0.04	170	43	41
1	breadth first search	6	0.04	180	56	43
1	breadth first tree search	6	1.08	5960	1459	1458
1	depth first graph search	12	0.01	48	13	12
1	greedy bfgs h 1	6	0.01	28	9	7
1	uniform cost search	6	0.05	224	57	55
2	astar search h 1	9	66.63	44030	4854	4852
2	astar search h pg levelsum	9	253.14	841	88	86
2	astar search ignore precond	9	14.64	13820	1508	1506
2	breadth first search	9	15.96	30509	4609	3343
2	breadth first tree search	-	-	-	-	-
2	depth first graph search	575	3.56	5211	583	582
2	greedy bfgs h 1	17	9.17	8910	992	990
2	uniform cost search	9	49.08	44030	4854	4852
3	astar search h 1	12	597.8	159716	18237	18235
3	astar search h pg levelsum	-	-	-	-	-
3	astar search ignore precond	12	104.66	45650	5120	5118
3	breadth first search	12	119.31	129631	18098	14663
3	breadth first tree search	-	-	-	-	-
3	depth first graph search	596	3.62	5176	628	627
3	greedy bfgs h 1	22	154.53	49429	5616	5614
3	uniform cost search	12	444.72	159716	18237	18235

References

[1] Russell, S. J., and Norvig, P. *Artificial Intelligence: A Modern Approach (2nd Edition).* Prentice Hall, December 2002.