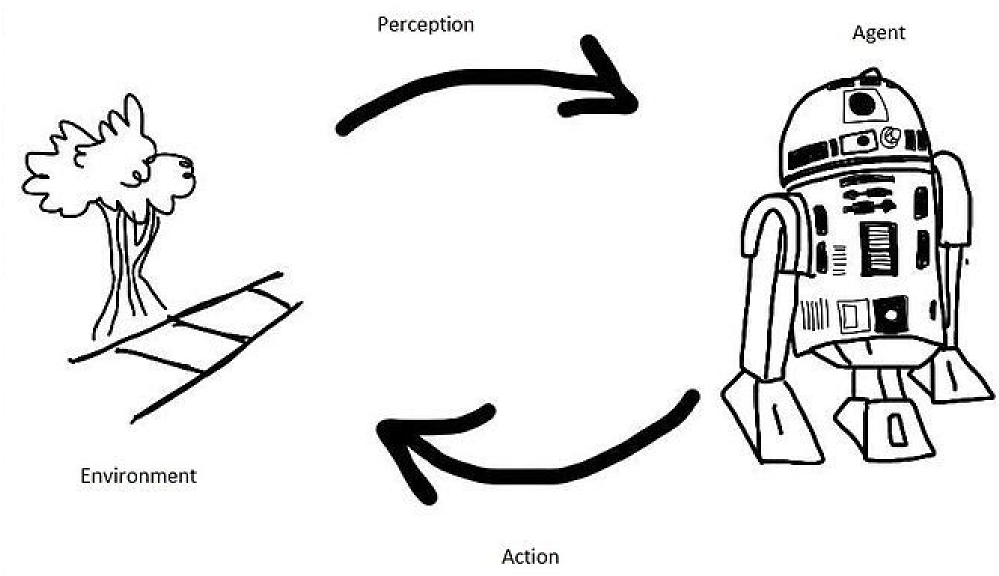
Build a Forward-Planning Agent



Author: Marvin Lüthe

Table of Contents

Introduction

This project uses a variety of search and planning algorithms to solve air cargo problems with different degrees of complexity. These air cargo problems basically consist in loading cargo on airplanes, transporting them from one airport to another and unloading it at the target location. We will run algorithms on four different problems and discuss the results with respect to the search complexity, search time and optimality of the solution.

Search Algorithms

- **Breadth-First Search**: searches the shallowest nodes in the search tree first
- **Depth-First Search**: searches the deepest nodes in the search tree first
- **Uniform-Cost Search**: expands nodes according to their path costs from the root node: $f(n) = g(n), \text{ where } g(n): \text{ path cost}$
- **Greedy-Best-First Graph Search**: expands most promising node according to a specified heuristic: $f(n) = h(n), \text{ where } h(n): \text{ estimated cost to the goal state}$
- **A-Star Search**: searches the lowest-cost path tree from the start node to the target node according to:
  $f(n) = g(n) + h(n), \text{ where } g(n): \text{ path cost and } h(n): \text{ estimated cost to the goal state}$

Heuristics

- **H-Unmet- Goals:** estimates the minimum number of actions that must be carried out from the current state in order to satisfy all of the goal conditions by ignoring the preconditions required for an action to be executed
- **H-PG-Level-Sum:** estimates the sum of the number of actions that must be carried out from the current state in order to satisfy each individual goal condition
- **H-PG-Max-Level:** estimates the maximum level cost out of all the individual goal literals; the level cost is the first level where a goal literal appears in the planning graph
- **H-PG-Set-Level:** estimates the level cost in the planning graph to achieve all of the goal literals such that none of them are mutually exclusive

Experiment Results

Air Cargo Problem 1: 20 actions in the domain
Air Cargo Problem 2: 72 actions in the domain
Air Cargo Problem 3: 88 actions in the domain
Air Cargo Problem 4: 104 actions in the domain

The following table shows the results of the experiments:

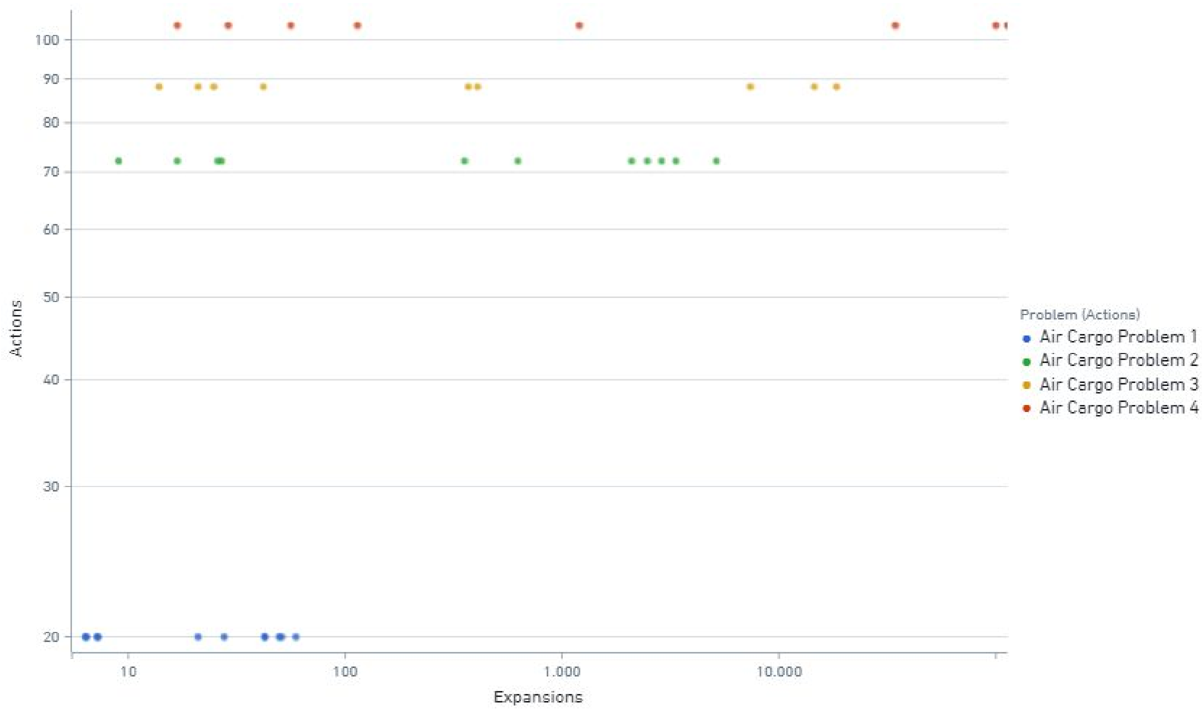
Problem	Search algorithm	Heuristics	Plan length	Expansions	Goal tests	New nodes	Time
Air Cargo Problem 1	Breadth-First Search		6	43	56	178	0.02
Air Cargo Problem 1	Depth-First Search		20	21	22	84	0.007
Air Cargo Problem 1	Uniform-Cost Search		6	60	62	240	0.19
Air Cargo Problem 1	Greedy-Best-First Graph Search	H-Unmet-Goals	6	7	9	29	0.002
Air Cargo Problem 1	Greedy-Best-First Graph Search	H-PG-Level-Sum	6	6	8	28	0.94
Air Cargo Problem 1	Greedy-Best-First Graph Search	H-PG-Max-Level	6	6	8	24	0.31
Air Cargo Problem 1	Greedy-Best-First Graph Search	H-PG-Set-Level	7	7	9	31	1.03
Air Cargo Problem 1	A-Star Search	H-Unmet-Goals	6	50	52	206	0.12
Air Cargo Problem 1	A-Star Search	H-PG-Level-Sum	6	28	30	122	0.78
Air Cargo Problem 1	A-Star Search	H-PG-Max-Level	6	43	45	180	0.7
Air Cargo Problem 1	A-Star Search	H-PG-Set-Level	6	51	53	208	1.84
Air Cargo Problem 2	Breadth-First Search		9	3343	4609	30503	0.40
Air Cargo Problem 2	Depth-First Search		619	624	625	5602	0.63
Air Cargo Problem 2	Uniform-Cost Search		9	5154	5156	46618	0.74
Air Cargo Problem 2	Greedy-Best-First Graph Search	H-Unmet-Goals	9	17	19	170	0.2
Air Cargo Problem 2	Greedy-Best-First Graph Search	H-PG-Level-Sum	9	9	11	86	5.19
Air Cargo	Greedy-Best-	H-PG-Max	9	27	29	249	10.85

Problem 2	First Graph Search	-Level					
Air Cargo Problem 2	Greedy-Best-First Graph Search	H-PG-Set-Level	10	26	28	232	26.36
Air Cargo Problem 2	A-Star Search	H-Unmet-Goals	9	2467	2469	22522	0.8
Air Cargo Problem 2	A-Star Search	H-PG-Level-Sum	9	357	359	3426	139.32
Air Cargo Problem 2	A-Star Search	H-PG-Max-Level	9	2887	2889	26594	846.12
Air Cargo Problem 2	A-Star Search	H-PG-Set-Level	9	2102	2104	19395	1671.14
Air Cargo Problem 3	Breadth-First Search		12	14663	18098	129625	1.21
Air Cargo Problem 3	Depth-First Search		392	408	409	3364	0.36
Air Cargo Problem 3	Uniform-Cost Search		12	18510	18512	161936	2.28
Air Cargo Problem 3	Greedy-Best-First Graph Search	H-Unmet-Goals	15	25	27	230	0.05
Air Cargo Problem 3	Greedy-Best-First Graph Search	H-PG-Level-Sum	14	14	16	126	12.97
Air Cargo Problem 3	Greedy-Best-First Graph Search	H-PG-Max-Level	13	21	23	195	17.97
Air Cargo Problem 3	Greedy-Best-First Graph Search	H-PG-Set-Level	18	42	44	405	89.24
Air Cargo Problem 3	A-Star Search	H-Unmet-Goals	12	7388	7390	65711	1.42
Air Cargo Problem 3	A-Star Search	H-PG-Level-Sum	12	369	371	3403	235.13
Air Cargo Problem 4	Breadth-First Search		14	99736	114953	944130	6.18
Air Cargo Problem 4	Uniform-Cost Search		14	113339	113341	1066413	10.47
Air Cargo Problem 4	Greedy-Best-First Graph	H-Unmet-Goals	18	29	31	280	0.06

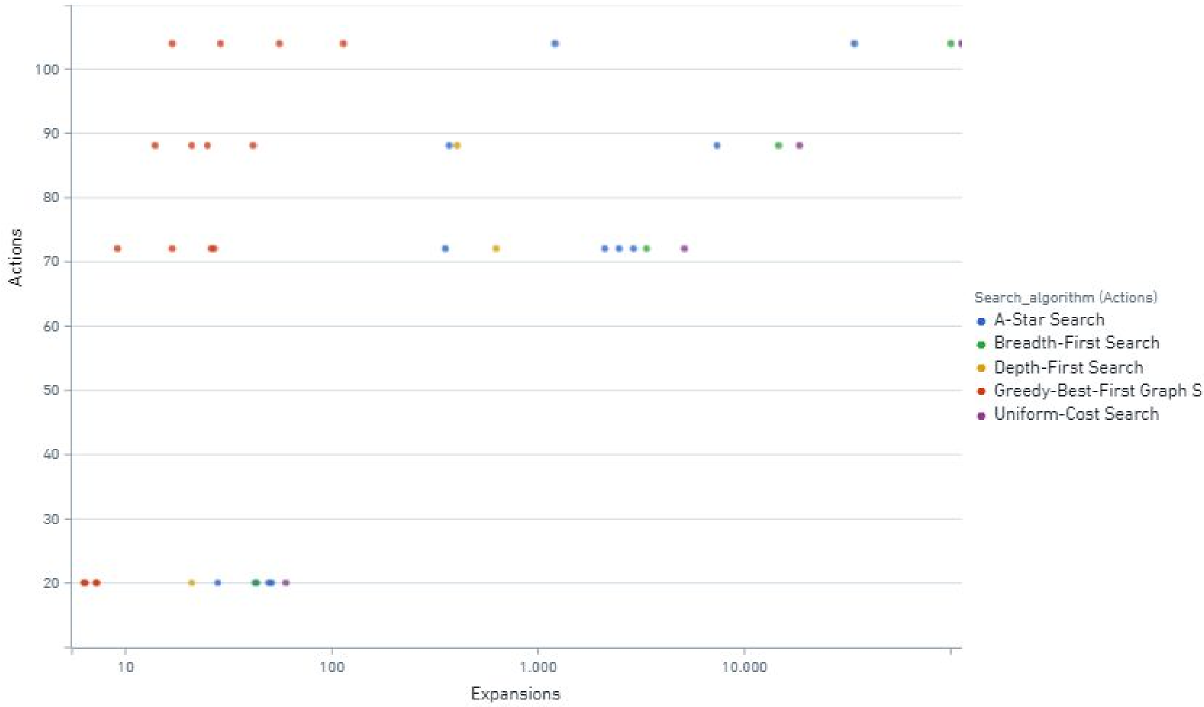
	Search						
Air Cargo Problem 4	Greedy-Best-First Graph Search	H-PG-Level-Sum	17	17	19	165	22.73
Air Cargo Problem 4	Greedy-Best-First Graph Search	H-PG-Max-Level	17	56	58	580	63.29
Air Cargo Problem 4	Greedy-Best-First Graph Search	H-PG-Set-Level	24	114	116	1229	352.78
Air Cargo Problem 4	A-Star Search	H-Unmet-Goals	14	34330	34332	328509	5.45
Air Cargo Problem 4	A-Star Search	H-PG-Level-Sum	15	1208	1210	12210	2208.54

Search Complexity

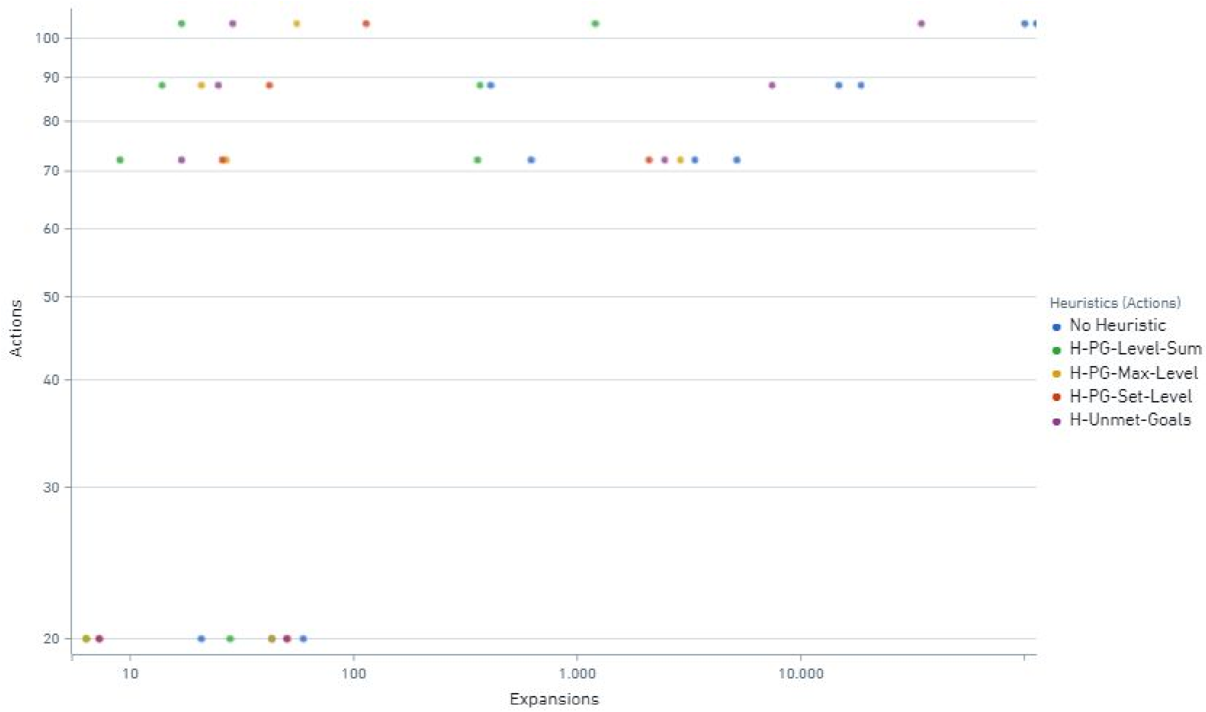
The complexity of the problem (number of actions in the domain) has a significant impact on the amount of node expansions during the search through the state space. The following chart depicts that the more complex problems lead to an exponential higher number of node expansions (consider the logarithmic scale).



The number of node expansions increases with the number of possible actions in the domain depending on the given problem. The Greedy-Best-First Graph Search algorithm expands the fewest nodes compared to the remaining search algorithms. The uninformed search algorithm Uniform-Cost Search expands the most nodes.



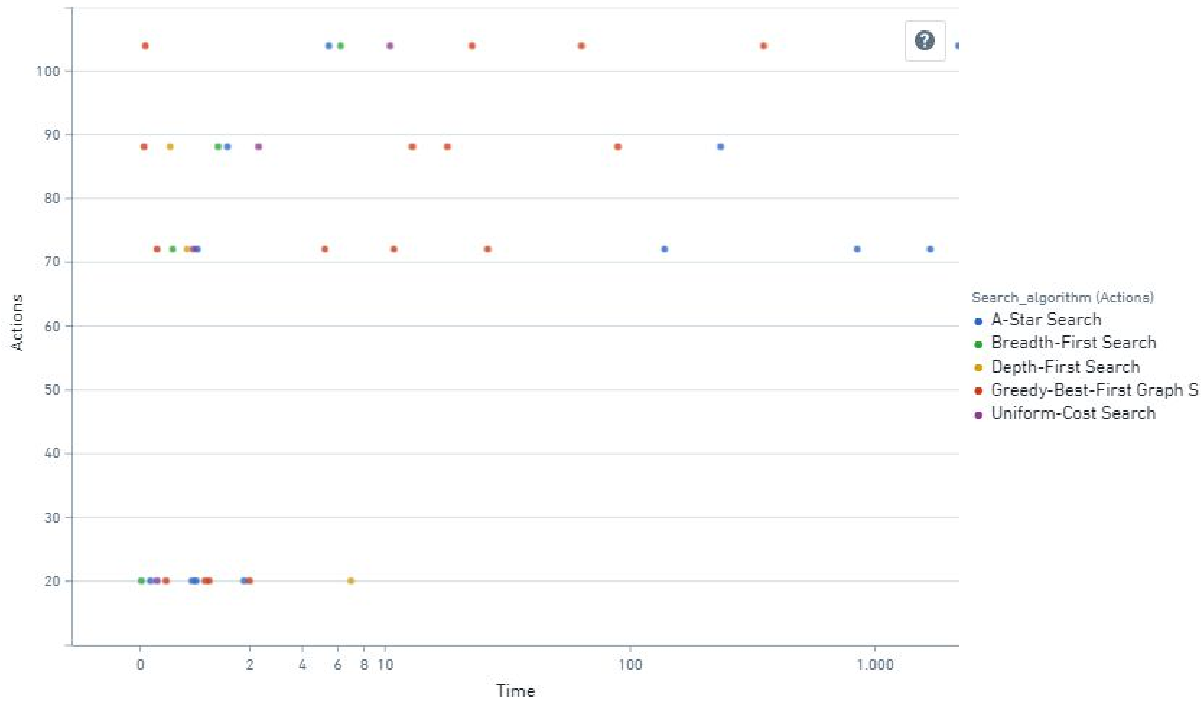
On the next chart we can see that algorithms without heuristics tend to expand more nodes than informed algorithms.



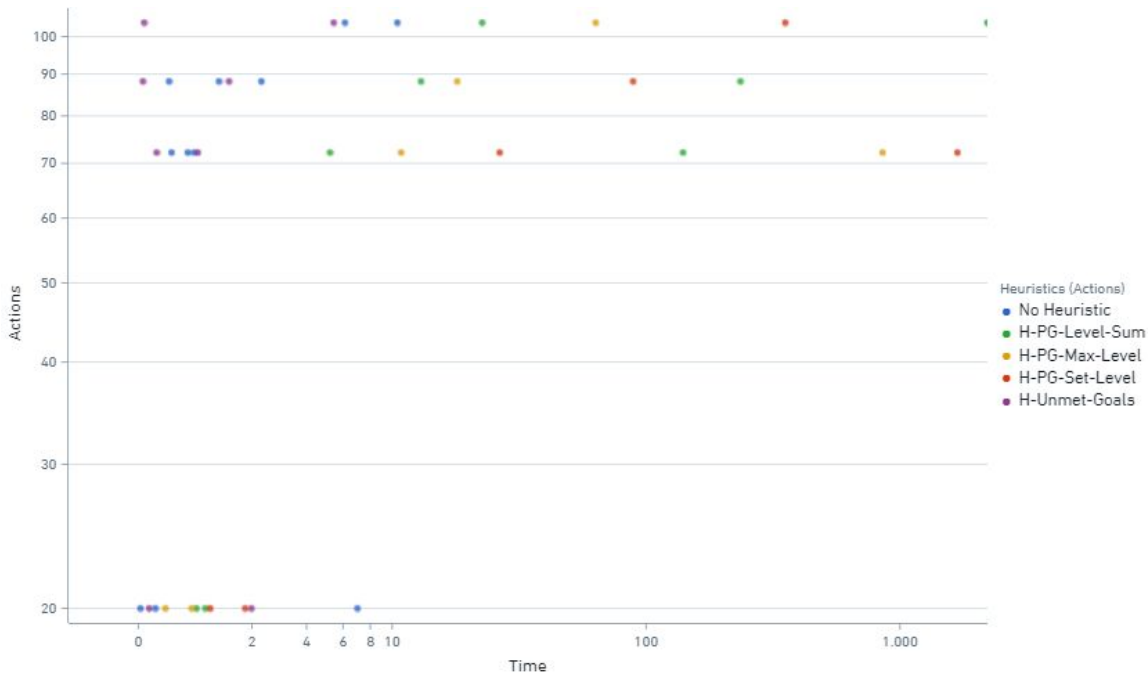
Search Time

As we can already expect the computation time for the search grows with the complexity of the problem. While the low complex Air Cargo Problem 1 (20 actions in domain size) can be solved by all algorithms in less than ten seconds, the high complex Air Cargo Problem 4 (104 actions in domain size) needs up to two thousand seconds for the chosen algorithms. This would increase even further if we solved this problem with the A-Star search algorithm using the heuristics **H-PG-Max-Level and H-PG-Set-Level.** Due to the computational effort, I rejected solving the Air Cargo Problem 4 with these search algorithms.

The following picture shows how the search time evolves with different problems and algorithms. For the Air Cargo Problem 1, **Depth-First Search** is the most time consuming as it keeps on expanding new nodes until it reaches a goal state.
For the more complex problems the informed searches **A-Star** and **Greedy-Best-First Graph Search** are more costly as they compute their heuristics every time new nodes are added to the priority queue. The computation of these heuristics means a high effort except for the heuristic **H-Unmet-Goals** as this heuristic does not use the planning graph data structure.



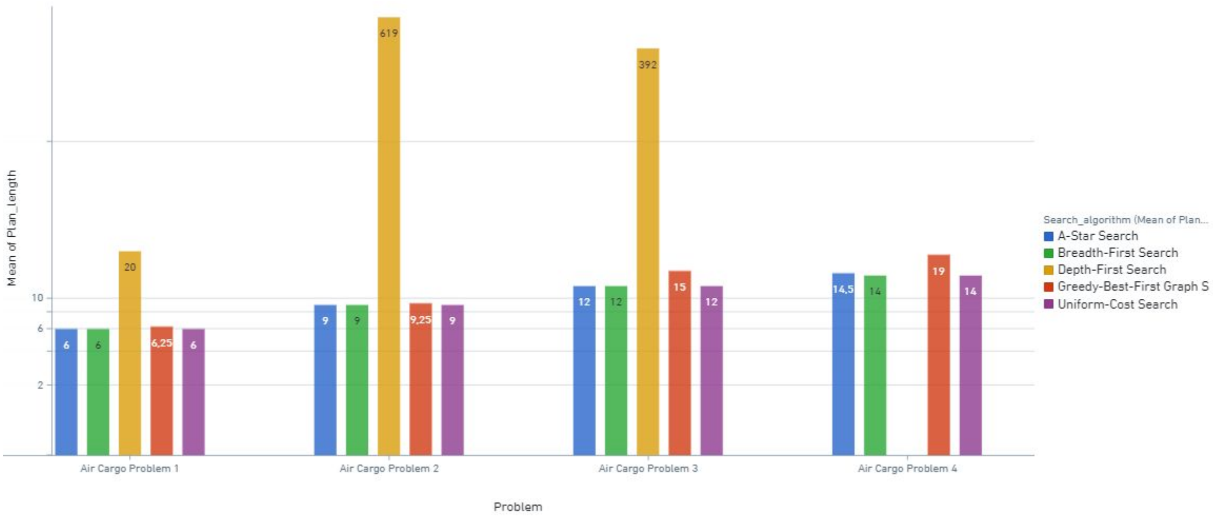
The following picture shows that the heuristics which were derived from the planning graph led to a significant increase in computation time.



The following heatmap underlines our findings in a succinct way. It becomes clear how the combination of search algorithm and heuristic does have a very huge impact on the search time. The A-Star Search in combination with planning graph heuristics is especially time consuming. The H-PG-Set-Level heuristic should not be used for very complex problems as it's inherent logic is more demanding than the less complex H-PG-Max-Level and H-PG-Level-Sum heuristics.

Solution Optimality

The plan length provides us a good estimate about the optimality of the returned solution. Depth-First Search does not return good results as we can see that these solutions consist of much more actions than the solutions returned by the other algorithms.
Apart from that, we cannot detect, for the given problems, large differences in the quality of the solutions found by the algorithms Breadth-First, Uniform-Cost, A-Star and Greedy-Best-First Graph Search.
We would expect that the A-Star algorithm always returns optimal plans. However, the results reveal that it did not always return the shortest length path. For the Air Cargo Problem 4, A-Star returned a solution consisting of 15 actions while the minimum possible length is 14. Accordingly, we can infer that the heuristic H-PG-Level-Sum is not admissible for this problem.



Conclusion

Which algorithms would be most appropriate for planning in a very restricted domain?

In a very restricted domain we should work with informed search algorithms like A-Star Search. In a restricted domain, we can handle the computational effort to run this algorithm. The reason for choosing A-Star is that if the heuristic function h(n) is admissible, A-Star guarantees to find the lowest coth path to the goal.

Which algorithms would be most appropriate for planning in very large domains?

In very large domains we will need to work with informed search algorithms using a “simple” heuristic, thus resulting in low computational effort like the heuristic H-Unmet-Goals. Furthermore, we can use uninformed search algorithms but we have to make sure that the returned plan does not contain a considerable amount of useless actions as returned by Depth-First Search in our experiment.

Which algorithms would be most appropriate for planning where it is important to find only optimal plans?

In order to return optimal plans we can use informed search algorithms e.g. A-Star. Here, we need to make sure that the provided heuristic h(n) is admissible. If h(n) is admissible, A-Star will find the shortest-length path while expanding a minimum number of paths possible.