Research review of Forward Planning Agent
Experimental Results & Report
By Marco Tiseo

Analyze the number of nodes expanded against number of actions in the domain for AirCargo Proble 1,2,3,4.

AirCargo 1

Legend Chart	Algorithm	Actions	Expansions
1	breadth_first_search	20	43
2	depth_first_graph_search	20	60
3	uniform_cost_search	20	21
4	greedy_best_first_graph_search h_unmet_goals	20	7
5	greedy_best_first_graph_search h_pg_levelsum	20	6
6	greedy_best_first_graph_search h_pg_maxlevel	20	6
7	greedy_best_first_graph_search h_pg_setlevel	20	6
8	astar_search h_unmet_goals	20	50
9	astar_search h_pg_levelsum	20	28
10	astar_search h_pg_maxlevel	20	43
11	astar_search h_pg_setlevel	20	33

Table 1

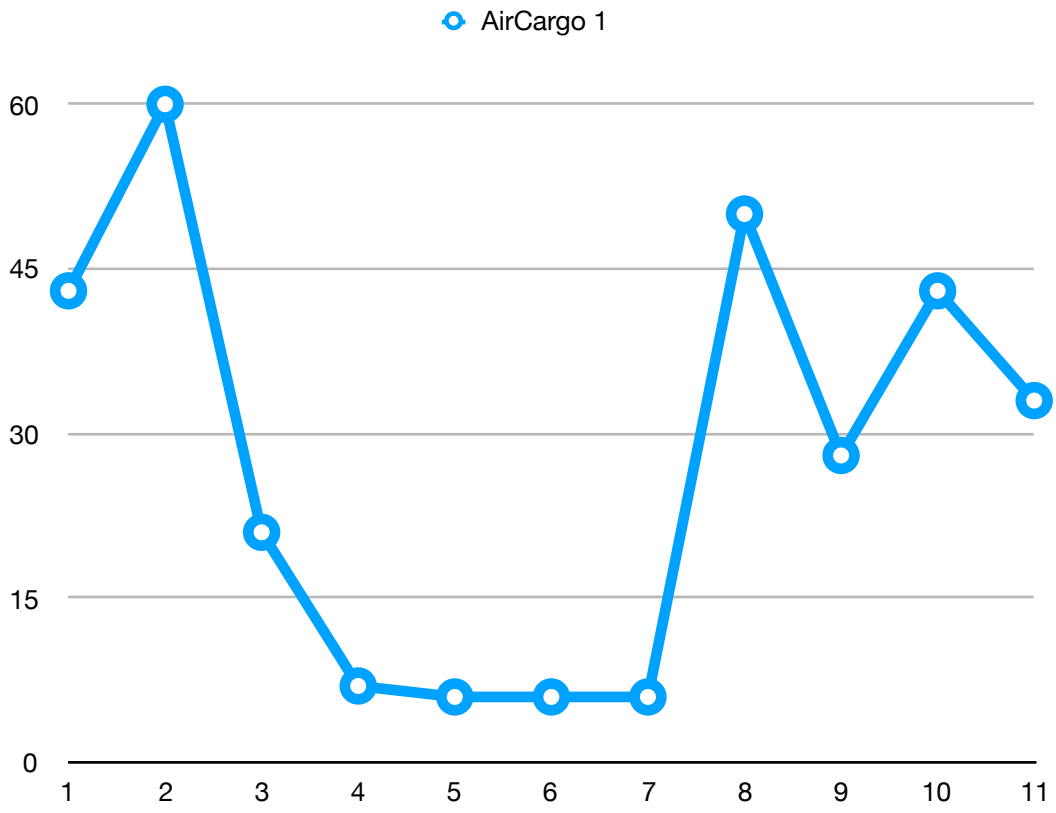

Chart 1

AirCargo 2

Legend Chart	Algorithm	Actions	Expansions
1	breadth_first_search	72	62
2	depth_first_graph_search	72	624
3	uniform_cost_search	72	5154
4	greedy_best_first_graph_search h_unmet_goals	72	17
5	greedy_best_first_graph_search h_pg_levelsum	72	9
6	greedy_best_first_graph_search h_pg_maxlevel	72	27
7	greedy_best_first_graph_search h_pg_setlevel	72	9
8	astar_search h_unmet_goals	72	2467
9	astar_search h_pg_levelsum	72	357
10	astar_search h_pg_maxlevel	72	2887
11	astar_search h_pg_setlevel	72	33

Table2

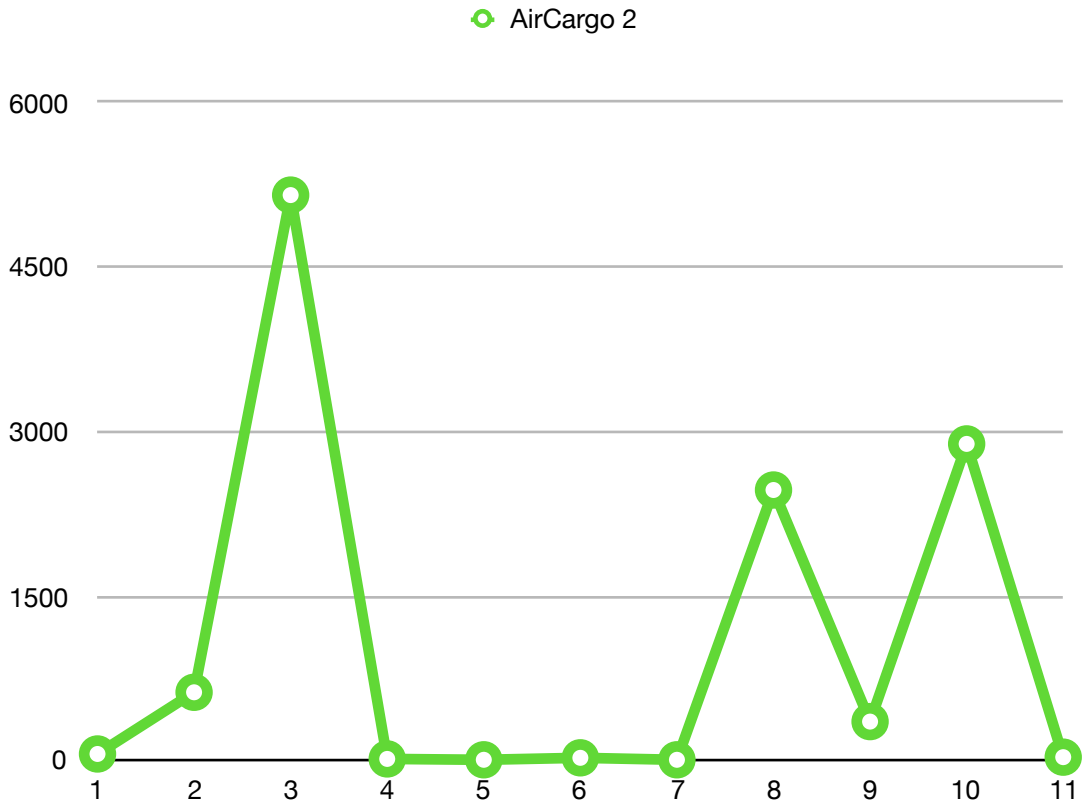

Chart 2

AirCargo 3

Legend Chart AirCargo3	Algorithm	Actions	Expansions
1	breadth_first_search	88	14663
2	greedy_best_first_graph_search h_unmet_goals	88	25
3	greedy_best_first_graph_search h_pg_levelsum	88	14
4	astar_search h_unmet_goals	88	7388
5	astar_search h_pg_levelsum	88	369

Table 3

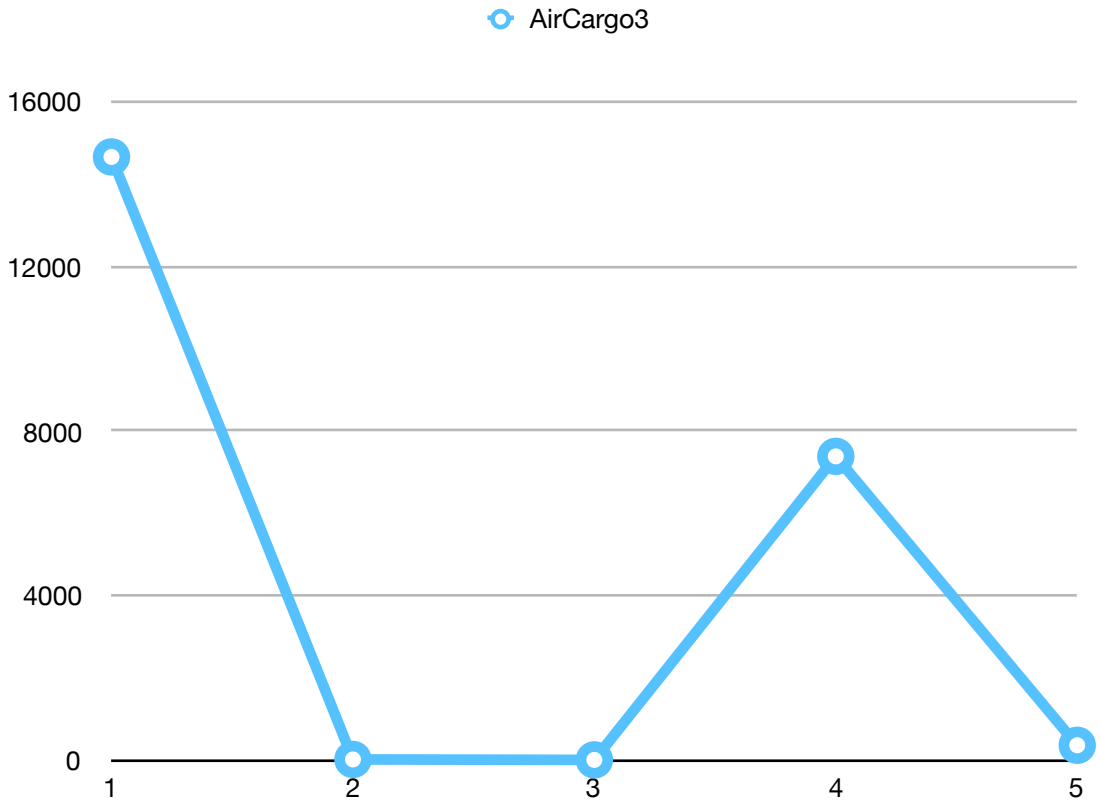

Chart 3

AirCargo 4

Legend Chart AirCargo4	Algorithm	Actions	Expansions
1	breadth_first_search	88	99736
2	greedy_best_first_graph_search h_unmet_goals	88	25
3	greedy_best_first_graph_search h_pg_levelsum	88	17
4	astar_search h_unmet_goals	88	34330
5	astar_search h_pg_levelsum	88	1208

Table 4

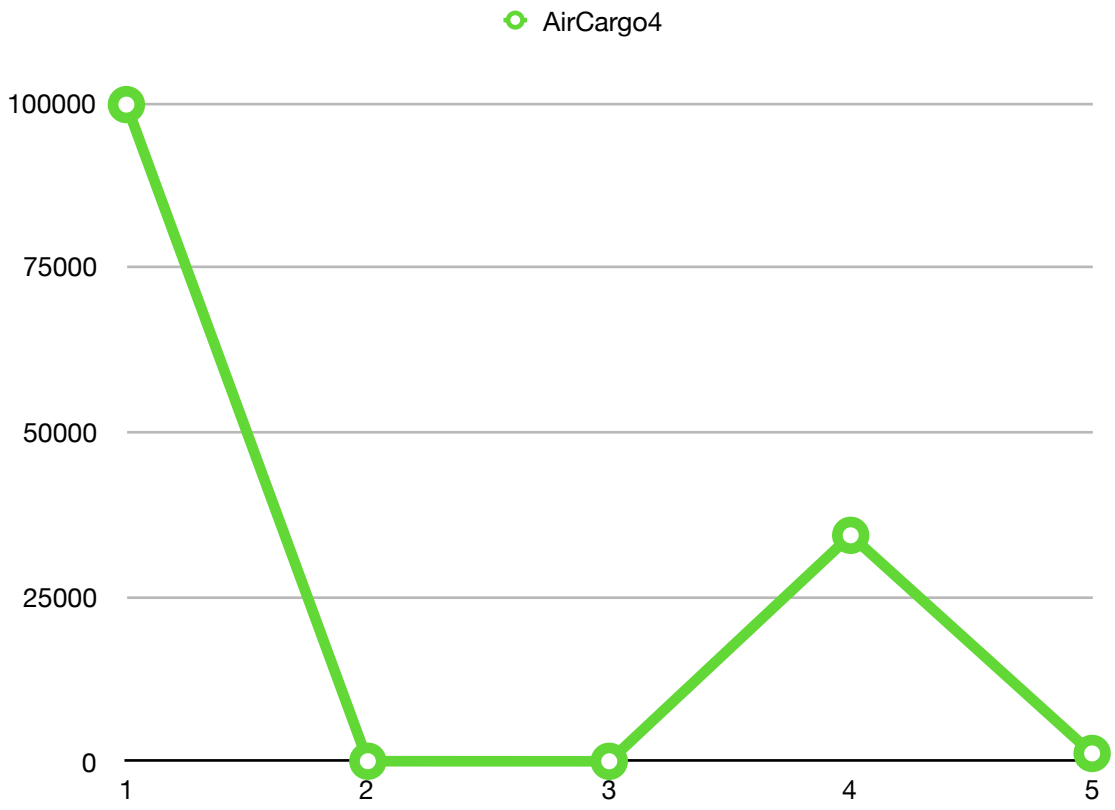

Chart 4

Discussion about results that analyse the number of nodes expanded against the number of actions.

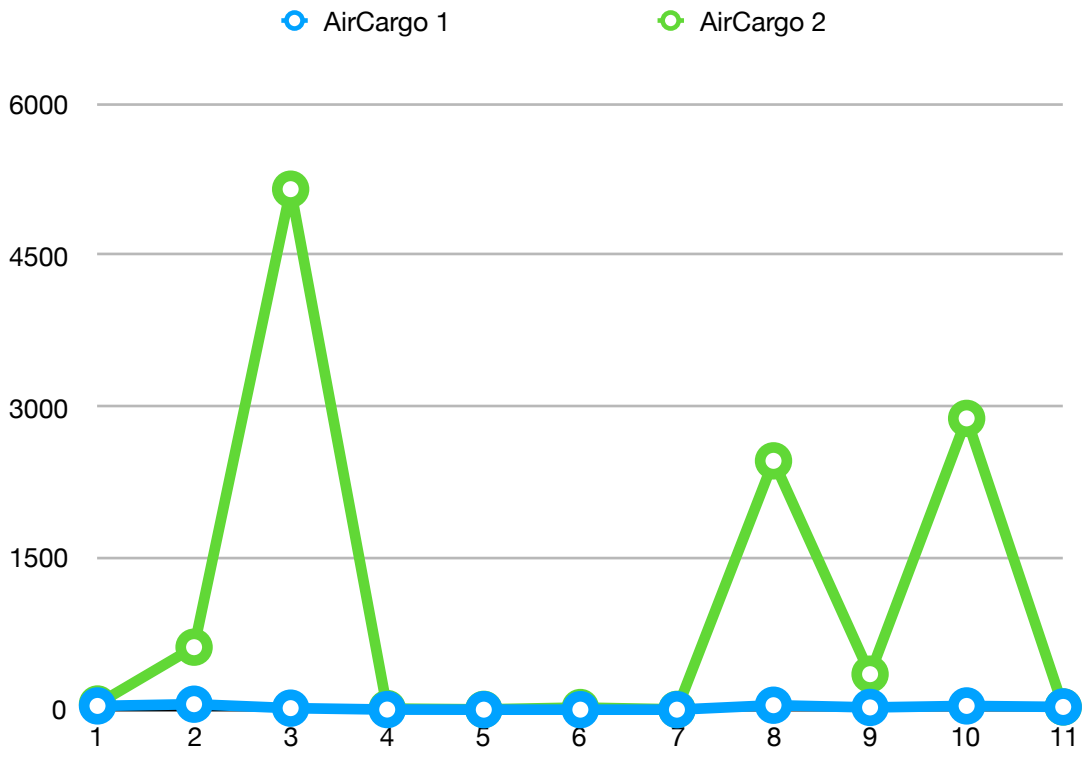

Chart 5

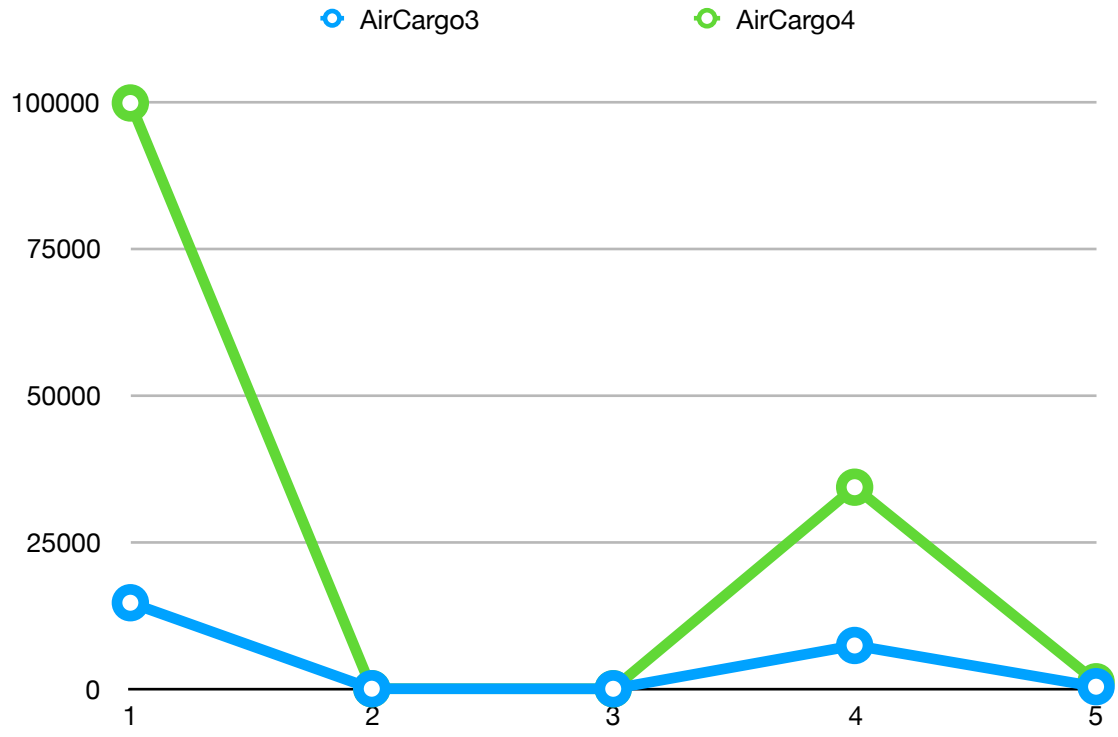

Chart 6

The most expensive algorithms in relation to the number of actions performed against expanded nodes are certainly uninformed search algorithms and some A* algorithms, these are not optimal and must be discarded due to planning problems it needs to visit a considerable number of nodes as the problem grows, thus it increase the computation time. From the charts we can see instead that the best algorithms are the gready ones, because the growth of the nodes visited remain constant. However it is to be taken into consideration “astar_search h_pg_setlevel” which turns out to be the best and the most constant when actions grow. To conclude if you choose an appropriate heuristic A* turns out to be excellent, while the gready algorithms are to be preferred because they are able to find optimal solutions in polynomial time, uninformed search instead is to be considered only for problems with few actions or when there is no information and a heuristic can not be used.

Analyze the search time against the number of actions in the domain for AirCargo 1,2,3,4.

AirCargo 1

Legend Chart	Algorithm	Actions	Expansions
1	breadth_first_search	20	0.008344940026290715
2	depth_first_graph_search	20	0.0047410299885086715
3	uniform_cost_search	20	0.012221363023854792
4	greedy_best_first_graph_search h_unmet_goals	20	0.0023372030118480325
5	greedy_best_first_graph_search h_pg_levelsum	20	0.46097055199788883
6	greedy_best_first_graph_search h_pg_maxlevel	20	0.35225533798802644
7	greedy_best_first_graph_search h_pg_setlevel	20	0.6133407220477238
8	astar_search h_unmet_goals	20	0.010809921950567514
9	astar_search h_pg_levelsum	20	1.1539088310091756
10	astar_search h_pg_maxlevel	20	1.2088912730105221
11	astar_search h_pg_setlevel	20	1.3886346130166203

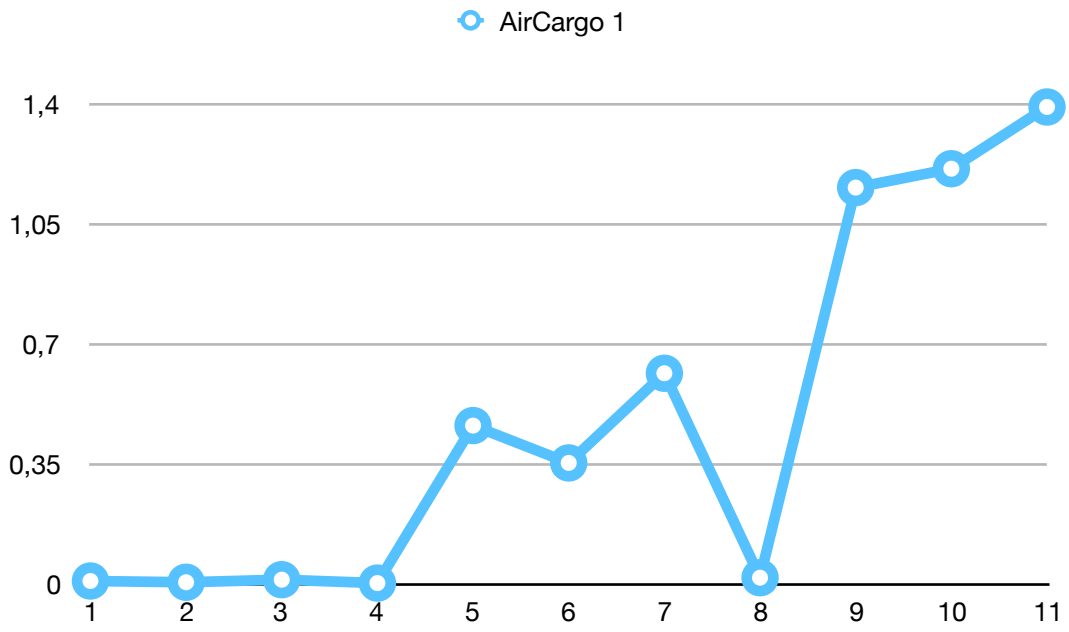

Table 5

Chart 7

AirCargo 2

Legend Chart	Algorithm	Actions	Expansions
1	breadth_first_search	72	2.07123582897475
2	depth_first_graph_search	72	2.8012673199991696
3	uniform_cost_search	72	3.4372634880128317
4	greedy_best_first_graph_search h_unmet_goals	72	0.023458998010028154
5	greedy_best_first_graph_search h_pg_levelsum	72	10.448376726999413
6	greedy_best_first_graph_search h_pg_maxlevel	72	20.795809456030838
7	greedy_best_first_graph_search h_pg_setlevel	72	14.891949646989815
8	astar_search h_unmet_goals	72	2.2651135710184462
9	astar_search h_pg_levelsum	72	268.6259330919711
10	astar_search h_pg_maxlevel	72	1549.4727375460207
11	astar_search h_pg_setlevel	72	1367.4398460480152

Table 6

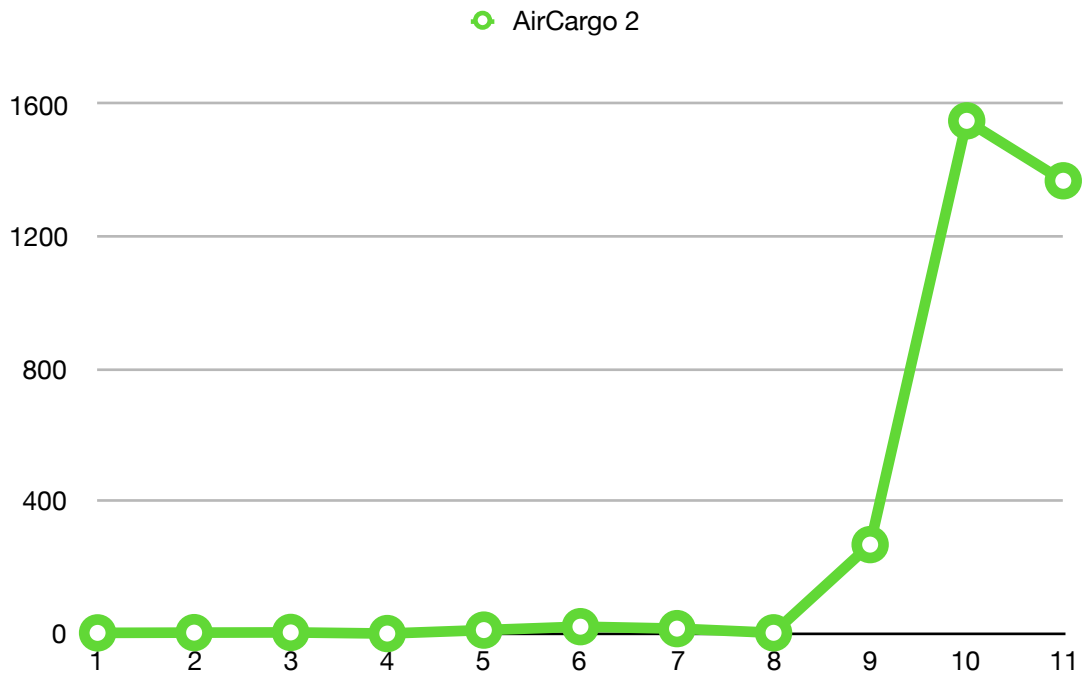

Chart 8

AirCargo 3

Legend Chart	Algorithm	Actions	Expansions
1	breadth_first_search	88	11.44255031296052
4	greedy_best_first_graph_search h_unmet_goals	88	0.0411085769883357
5	greedy_best_first_graph_search h_pg_levelsum	88	13.476647987
8	astar_search h_unmet_goals	88	8.625326932989992
9	astar_search h_pg_levelsum	88	430.32003477198305

Table 7

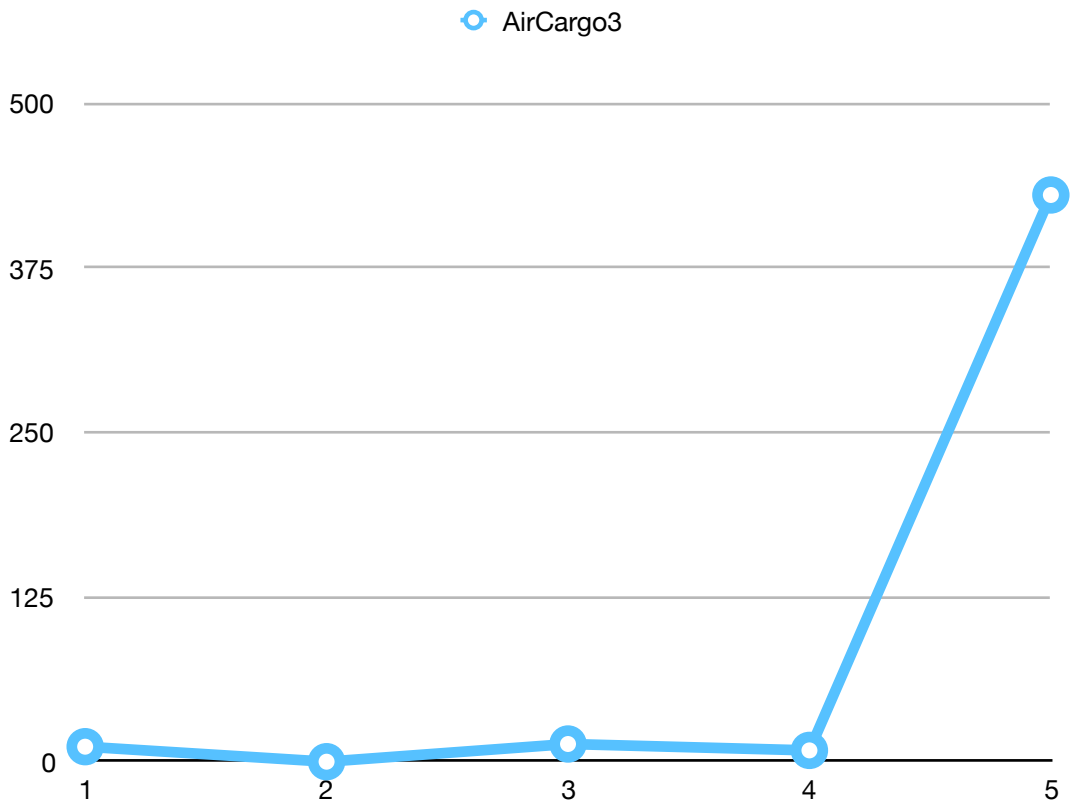

Chart 9

AirCargo 4

Legend Chart	Algorithm	Actions	Expansions
1	breadth_first_search	104	104.97104673401918
4	greedy_best_first_graph_search h_unmet_goals	104	0.06365173496305943
5	greedy_best_first_graph_search h_pg_levelsum	104	42.49239284702344
8	astar_search h_unmet_goals	104	56.51207398495171
9	astar_search h_pg_levelsum	104	2453.366108573973

Table 8

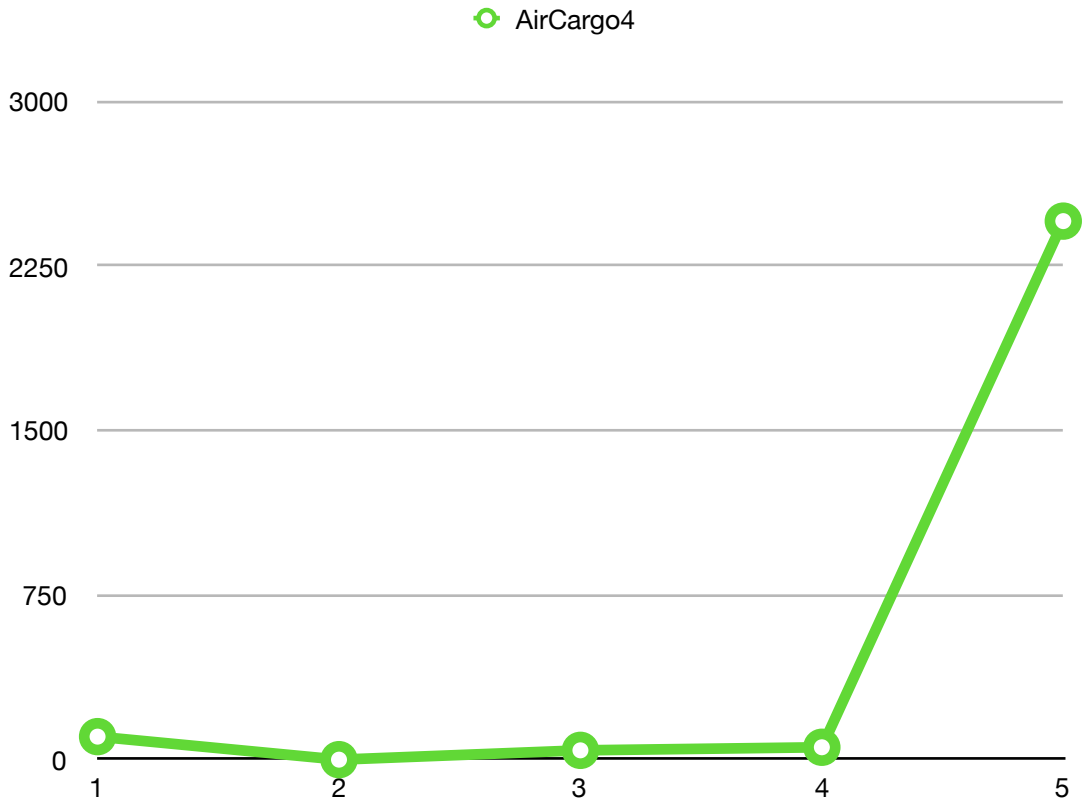

Chart 10

Discussion about results that analyze the search time against the number of actions and the growth trends as the problem size increases.

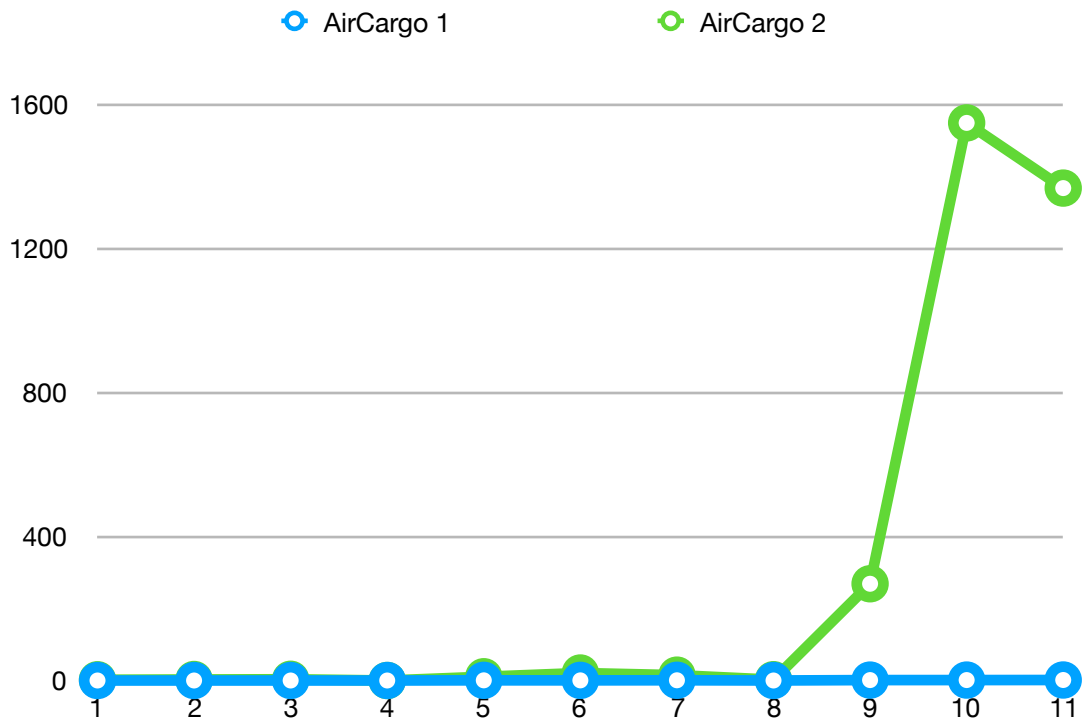

Chart 11

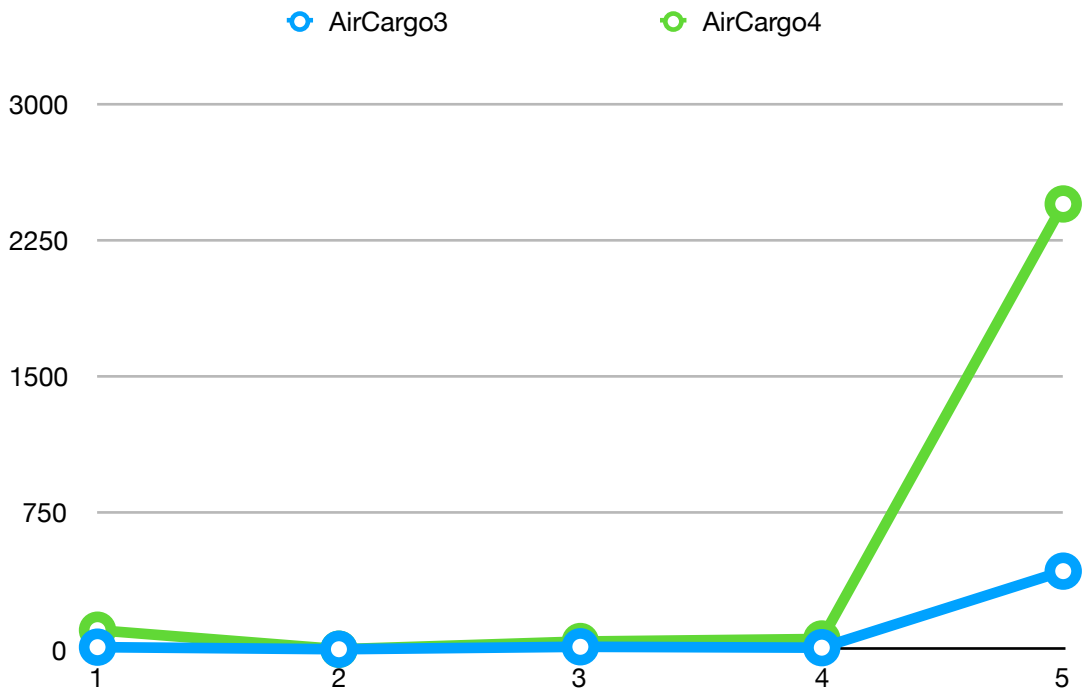

Chart 12

as you can see in AirCargo 1 and 2 the uninformed search algorithms are fast for small problems but they grow a lot as the actions increase, the same applies to A * search which in terms of time against the growth of the actions turns out to be the worst, while the gready algorithms remain constant even if for small problems we can see that the algorithms uninformed search do better in terms of time. Regarding Aircargo 3 and 4 instead we can see that the uninformed search algorithms turn out worse than the first experiment, the gready algorithms instead remain costly and excellent while only some A* are excellent this emphasizes how the choice of a good heuristic influences the final result.

This section includes a tables to analyze the length of the plans returned by each algorithm on all search problems.

AirCargo 1

Algorithm	Plans
breadth_first_search	6
depth_first_graph_search	20
uniform_cost_search	6
greedy_best_first_graph_search h_unmet_goals	6
greedy_best_first_graph_search h_pg_levelsum	6
greedy_best_first_graph_search h_pg_maxlevel	6
greedy_best_first_graph_search h_pg_setlevel	6
astar_search h_unmet_goals	6
astar_search h_pg_levelsum	6
astar_search h_pg_maxlevel	6
astar_search h_pg_setlevel	6

Table 9

AirCargo 2

Algorithm	Plans
breadth_first_search	9
depth_first_graph_search	620
uniform_cost_search	9
greedy_best_first_graph_search h_unmet_goals	9
greedy_best_first_graph_search h_pg_levelsum	9
greedy_best_first_graph_search h_pg_maxlevel	9
greedy_best_first_graph_search h_pg_setlevel	9
astar_search h_unmet_goals	9
astar_search h_pg_levelsum	9
astar_search h_pg_maxlevel	9
astar_search h_pg_setlevel	9

Table 10

AirCargo 3

Algorithm	Plans
breadth_first_search	12
greedy_best_first_graph_search h_unmet_goals	15
greedy_best_first_graph_search h_pg_levelsum	14
astar_search h_unmet_goals	12
astar_search h_pg_levelsum	12

Table 11

AirCargo 4

Algorithm	Plans
breadth_first_search	14
greedy_best_first_graph_search h_unmet_goals	18
greedy_best_first_graph_search h_pg_levelsum	17
astar_search h_unmet_goals	14
astar_search h_pg_levelsum	15

Table 12

Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

the search algorithms not informed in a restricted domain behave well expanding a few nodes and the execution time is reasonable but this also applies to gready.

Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)

For a very high-grade domain and for many actions the gready algorithms with good heuristics behave well.

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

Definitely A * but it is important to find a heuristic that minimizes the execution time.

Conclusion

In general, use search algorithms not informed only when you do not have information about the problem and for problems with few actions. Use gready and A* with a heuristic function when the problem is complex and we want to find an optimal solution. Remember that a heuristic function estimates the cost of the node and therefore a heuristic value will be much better when it is lower. Using a heuristic evaluation function to reduce the complexity of a problem is of paramount importance in problems with exponential growth since the work must be done by a computer and there will be a point in time beyond which the problem becomes insoluble.