Planing search Heuristic Analysis

by Artem Odintsov

In this paper I would like to present the results of solving Air Cargo problem. To solve this problem there were used two types of search such as uniformed search and heuristic based search. The main goal of this analyzes is to find an optimal path for three different problems using variance of algorithms such as Breadth first search, depth first search, uniform search, astar search.

First of all lets present the tree problems that we should solve:

Problem 1:

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))

Problem 2:

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))

Problem 3:

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 results for each of the problems:

Air cargo problem 1	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**	**Optimal**
Breadth_first_search	43	56	180	6	0.0252	TRUE
Breadth_first_tree_search	1458	1459	5960	6	0.5728	TRUE
Bepth_first_graph_search	21	22	84	20	0.0125	FALSE
Depth_limited_search	101	271	414	50	0.0544	FALSE
Uniform_cost_search	55	57	224	6	0.0239	TRUE
Recursive_best_first_search h_1	4229	4230	17023	6	1.6283	TRUE
Greedy_best_first_graph_search h_1	**7**	**9**	**28**	**6**	**0.0035**	**TRUE**
Astar_search h_1	55	57	224	6	0.0242	TRUE
Astar_search h_ignore_preconditions	41	43	170	6	0.0248	TRUE
Astar_search h_pg_levelsum	11	13	50	6	0.3234	TRUE

Air cargo problem 2	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**	**Optimal**
Breadth_first_search	3346	4612	30534	9	5.0651	TRUE
Breadth_first_tree_search	N/A	N/A	N/A	N/A	N/A	FALSE
Bepth_first_graph_search	107	108	959	105	0.1985	FALSE
Depth_limited_search	N/A	N/A	N/A	N/A	N/A	FALSE
Uniform_cost_search	4853	4855	44041	9	7.1371	TRUE
Recursive_best_first_search h_1	N/A	N/A	N/A	N/A	N/A	FALSE
Greedy_best_first_graph_search h_1	998	1000	8982	17	1.4476	TRUE
Astar_search h_1	4853	4855	44041	9	7.1383	FALSE
Astar_search h_ignore_preconditions	**1450**	**1452**	**13303**	**9**	**2.6029**	**TRUE**
Astar_search h_pg_levelsum	86	88	841	9	26.4141	TRUE

Air cargo problem 3	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**	**Optimal**
Breadth_first_search	14120	17673	124926	12	24.976	TRUE
Breadth_first_tree_search	N/A	N/A	N/A	N/A	N/A	FALSE
Bepth_first_graph_search	292	293	2388	288	0.6978	FALSE
Depth_limited_search	N/A	N/A	N/A	N/A	N/A	FALSE
Uniform_cost_search	18223	18225	159618	12	32.1392	TRUE
Recursive_best_first_search h_1	N/A	N/A	N/A	N/A	N/A	FALSE
Greedy_best_first_graph_search h_1	557	5580	49150	14	10.0108	FALSE
Astar_search h_1	18223	18225	159618	12	31.5693	TRUE
Astar_search h_ignore_preconditions	**5040**	**5042**	**44944**	**12**	**9.9608**	**TRUE**
Astar_search h_pg_levelsum	325	327	3002	12	137.1723	TRUE

From the results presented above we can see that no all the algorithms are optimal or even can find the solution for a particular problem due to the fact that it takes too much time for them to find a solution. For example, for the problem 2 breadth first tree, depth limited, and recursive best first search took too much time to solve the problem and might not be considered as complete or optimal. We also make a conclusion that only some of the algorithms are optimal and can find the shortest path. The best of them are presented in the table below.

Air cargo problem 1	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**
Greedy_best_first_graph_search h_1	47	49	178	6	0.0248
Astar_search h_ignore_preconditions	41	43	170	6	0.0248
Astar_search h_pg_levelsum	11	13	50	6	0.3234

Air cargo problem 2	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**
Astar_search h_ignore_preconditions	1450	1452	13303	9	2.6029
Astar_search h_pg_levelsum	86	88	841	9	26.4141

Air cargo problem 3	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**
Astar_search h_ignore_preconditions	5040	5042	44944	12	9.9608
Astar_search h_pg_levelsum	325	327	3002	12	137.1723

If we compare heuristic search for given three problem we can see that for the *Problem 2* and *Problem 3* they were optimal and managed to find the shortest path. In the table above we can see for both problem 2 and 3 A*-search with level-sum heuristic managed to find th shortest path. Despite the fact that A* with heuristic takes much longer than non-heuristic search algorithms such as breadth first search or uniform cost search, it takes less nodes for expansion and can find an optimal path.

Despite the fact that *Greedy best first graph search* showed the best result for the *Problem 1* it doesn't work well for the rest of the problems. From the results above we also can see that *Breadth first tree search*, *Recursive best first search with h1*, and *Depth limited search* showed the worst performance, that is it took too long for them to find any paths. If we keep comparing non-heuristic and heuristic algorithm we can conclude that heuristic algorithms show a better performance for more complicated problems. Although it takes much long for *A* level-sum search* to find an optimal path but it requires less memory than the rest of the algorithms.

Problem 1	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time elapsed**
Breadth_first_search	43	56	180	6	0.0252
Uniform_cost_search	55	57	224	6	0.0239
Astar_search h_1	55	57	224	6	0.0242
Astar_search h_ignore_preconditions	41	43	170	6	0.0248
Astar_search h_pg_levelsum	11	13	50	6	0.3234
Problem 2					
Breadth_first_search	3346	4612	30534	9	5.0651
Uniform_cost_search	4853	4855	44041	9	7.1371
Astar_search h_1	4853	4855	44041	9	7.1383
Astar_search h_ignore_preconditions	1450	1452	13303	9	2.6029
Astar_search h_pg_levelsum	86	88	841	9	26.4141
Problem 2					
Breadth_first_search	14120	17673	124926	12	24.976
Uniform_cost_search	18223	18225	159618	12	32.1392
Astar_search h_1	18223	18225	159618	12	31.5693
Astar_search h_ignore_preconditions	5040	5042	44944	12	9.9608
Astar_search h_pg_levelsum	325	327	3002	12	137.1723

Conclusion

To sum up from the result above we clearly can see the benefits of heuristics algorithms over non-heuristics algorithms not only in terms of speed but also in terms of memory usage. Another advantage of heuristic algorithms is it’s flexibility it allows us to find a trade-off between speed and memory usage simply choosing different heuristics.