Air Cargo Planning Heuristic Analysis

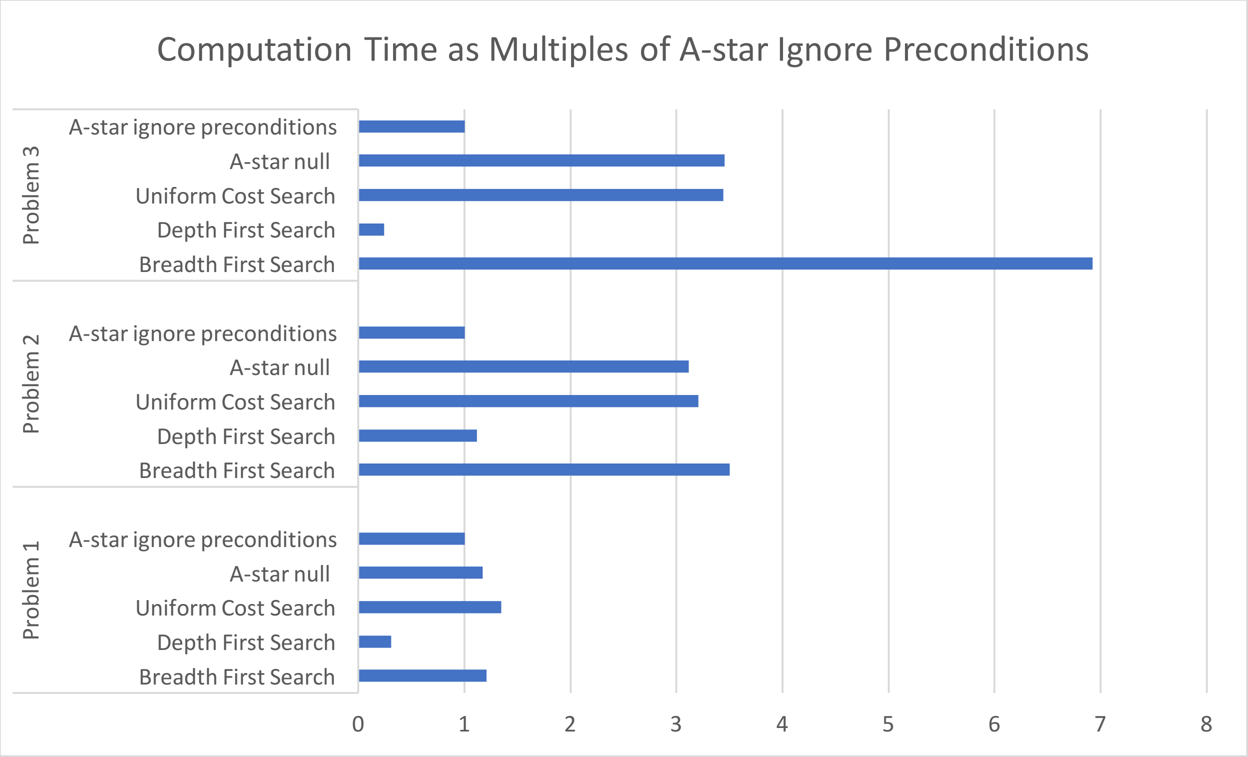
The following tables summarize the performances of uninformed planning algorithms (Breadth-first, Depth-first search and Uniform cost search) as well as of automatic heuristics with A-star search (including null, ignore-conditions and level-sum).

Problem 1	Expansions	Goal Tests	New Nodes	Plan length	Time(s)
Breadth First Search	43	56	180	6	0.035
Depth First Search	12	13	48	12	0.009
Uniform Cost Search	55	57	224	6	0.039
A-star null	55	57	224	6	0.034
A-star ignore preconditions	41	43	170	6	0.029
A-star levelsum	11	13	50	6	0.817

Problem 2	Expansions	Goal Tests	New Nodes	Plan length	Time(s)
Breadth First Search	3346	4612	30534	9	12.25
Depth First Search	624	625	5602	619	3.92
Uniform Cost Search	4853	4855	44041	9	11.21
A-star null	4853	4855	44041	9	10.91
A-star ignore preconditions	1450	1452	13303	9	3.5
A-star levelsum	86	88	841	9	71.21

Problem 3	Expansions	Goal Tests	New Nodes	Plan length	Time(s)
Breadth First Search	14120	17673	124926	12	93.57
Depth First Search	677	678	5608	660	3.28
Uniform Cost Search	18223	18225	159618	12	46.5
A-star null	18223	18225	159618	12	46.67
A-star ignore preconditions	5040	5042	44944	12	13.52
A-star levelsum	325	327	3002	12	348.9

In all three problems, A-star with ignore preconditions heuristics has the best performance in in terms of computation time as well as the number of node expansions. Other than level-sum heuristics, A-star search generally performs better than non-heuristics based search.
The level over-performance is more pronounced as the complexity increases. The following graph shows computation time of different algorithms expressed as a multiple of the time for A-star ignored preconditions heuristic. The multiples increase as we progress from the simple problem 1 to complicated problem 3.



According to Norvig and Russell’s textbook, breadth-first always considers shortest path first. Therefore, it always finds the optimal solution and in a relative fashion.
Depth-first is very fast, however, it’s not optimal. Because it tries to traverse the graph as deep as possible even if the goal is to its right, and it does not compare if a node is better than the other.

The breadth-first and uniform-cost search could be a great start for solving simple planning problems but A-star search with ignore preconditions seem to be the overwhelming option when the complexity of the problem reaches a certain level.

Optimal Sequences

Problem 1

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem 2

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 3

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)