Optimal plans and collected test metrics are presented in tables below for each of the 3 air cargo planning problems

Air Cargo Problem 1:

Optimal plan found (Plan length : 6) :

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

Search algorithm	Expansions	Goal Tests	New Nodes	Plan length	Time(seconds)
Breadth_first_search	43	56	180	6	0.27
depth_first_graph_search	21	22	84	20	0.11
depth_limited_search	101	271	414	50	0.52
uniform_cost_search	55	57	224	6	0.24
recursive_best_first_search with h_1	4229	4230	17023	6	16.13
greedy_best_first_graph_search with h_1	7	9	28	6	0.035
astar_search with h_1	55	57	224	6	0.22
astar_search with h_ignore_preconditions	41	43	170	6	0.27
astar_search	11	13	50	6	13.91

with h_pg_levelsum					

SUMMARY FOR AIR CARGO PROBLEM 1:

For air cargo problem 1, the search space is very small esp. when compared to problem 3. In this case, greedy best first graph search appears to be optimal with respect to time, node expansions and plan length compared to the other uninformed search methods as well even the automated heuristic planning algorithms. However this is a case that may apply only to small search spaces. In larger search spaces, the greedy best first graph will not be optimal as it may lead to unwanted mode expansions before reaching optimality. Breadth first search(Shortest first search explained in video 8 of “Search” lesson) which expands frontier nodes first is also optimal in this small space along with depth first search which goes as deep as possible before considering other nodes at frontier. There DFS will never be as optimal as the search space grows bigger. Uniform cost search(see videos 16 -18) which expands node with cheapest total cost will be slower than BFS which always considers shortest path. A* with h_1 always has the same node expansions, nodes and plan length as uninformed uniform cost search as it (A* search) also finds the path with cheapest cost. (Videos 27 onward in “Search” lesson)

Among the automated heuristic planning methods, A* with h_pg_levelsum (Sec 10.3.1 of Artificial Intelligence, Russell/Norvig) is optimal wrt(with respect to) node expansions but not wrt time elapsed because of the use of planning graph representation to estimate the sum of all actions from state to each individual goal which increased processing time. So it was orders of magnitude more time expensive than A* with h_ignore_preconditions (Sec 10.2.3 of AI, Russell//Norvig) which only “estimates” minimum number of actions from state to goal conditions by ignoring all preconditions.

Air Cargo Problem 2:

Optimal plan found (Plan length: 9) :

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

Search algorithm	Expansions	Goal Tests	New Nodes	Plan length	Time(seconds)
Breadth_first_search	3343	4609	30509	9	75.52
depth_first_graph_search	624	625	5602	619	20.18
uniform_cost_search	4852	4854	44030	9	269.49
recursive_best_first_search with h_1	TL	TL	TL	TL	TL
greedy_best_first_graph_search with h_1	990	992	8910	17	42.094
astar_search with h_1	4852	4854	44030	9	242.51
astar_search with h_ignore_preconditions	1506	1508	13820	9	78.91
astar_search with h_pg_levelsum	86	88	841	9	1433.82

TL = Too Long (execution time dismissed as way over several hours for comparison)

SUMMARY FOR AIR CARGO PROBLEM 2:

For air cargo problem 2, BFS was most optimal as it found optimal plan in the least time while greedy best first graph search failed to find optimal plan. DFS as expected is even less optimal in this search space.

Among the automated heuristic planning methods, A* with h_pg_levelsum is again as expected optimal wrt(with respect to) node expansions to all methods but much more time expensive than A* with h_ignore_preconditions for the same reason as given in air cargo problem 1 summary above.

Air Cargo Problem 3:

Optimal plan found (Plan length: 12) :

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

Search algorithm	Expansions	Goal Tests	New Nodes	Plan length	Time(seconds)
Breadth_first_search	14663	18098	129631	12	565.16
depth_first_graph_search	408	409	3364	392	10.35
uniform_cost_search	18235	18237	159716	12	2095.85
recursive_best_first_search with h_1	TL	TL	TL	TL	TL
greedy_best_first_graph_search with h_1	5614	5616	49429	22	635.10
astar_search with h_1	18235	18237	159716	12	2050.84
astar_search with h_ignore_preconditions	5118	5120	45650	12	543.82
astar_search with h_pg_levelsum	408	410	3758	12	12104.53

TL = Too Long (execution time dismissed as way over several hours for comparison)

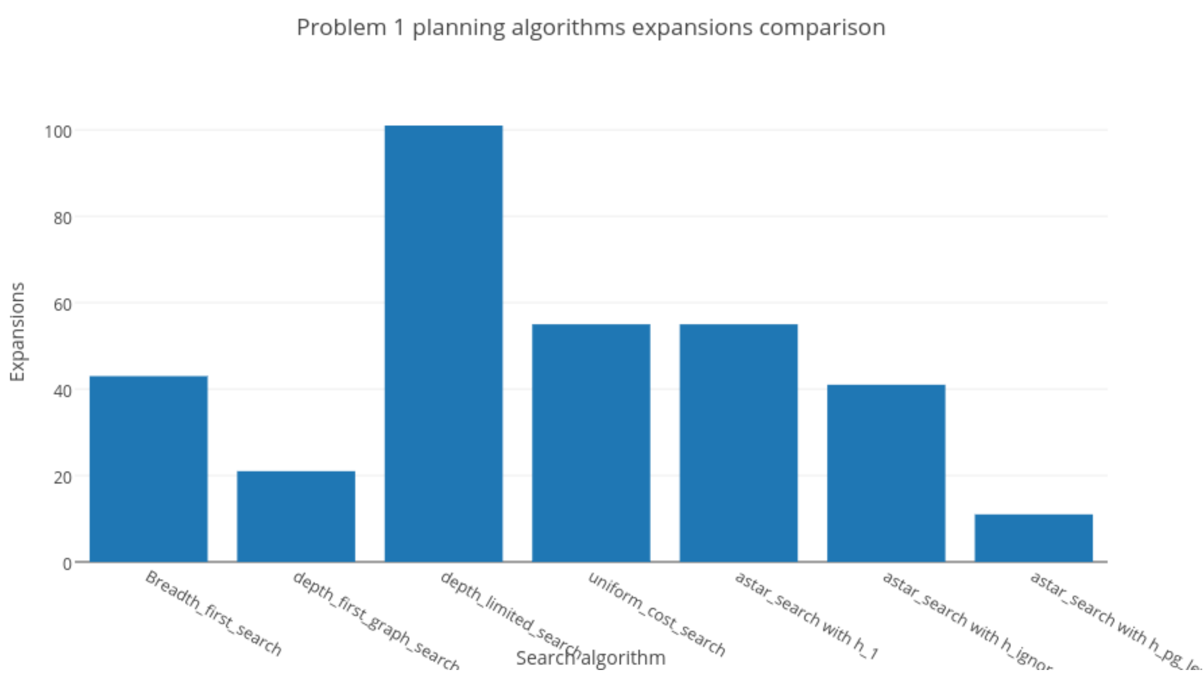
SUMMARY FOR AIR CARGO PROBLEM 3:

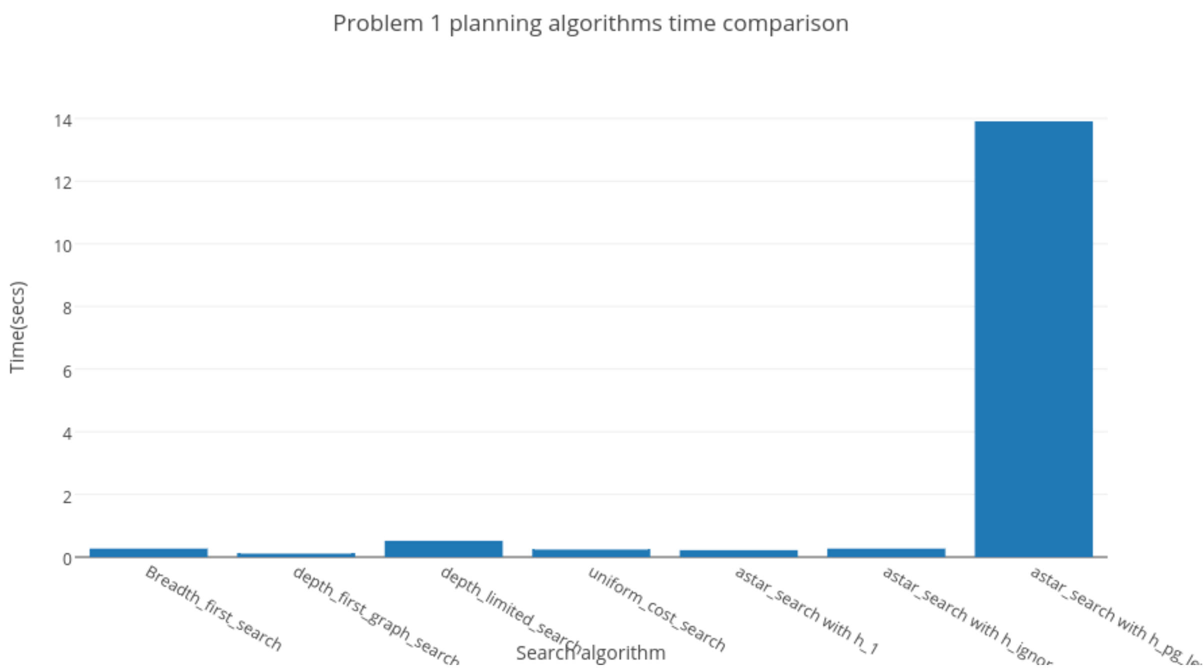
For air cargo problem 3, A* with h_ignore_preconditions was most optimal wrt all other methods as it found optimal plan in the least time. BFS is again most optimal among the uninformed planning methods and again greedy best and DFS as expected are even less optimal in this search space and fail to find optimal plan in this larger space.

Among the automated heuristic planning methods, A* with h_pg_levelsum is again as expected optimal wrt(with respect to) node expansions to all methods but much more time expensive than A* with h_ignore_preconditions for the same reason as given in air cargo problem 1 summary above.

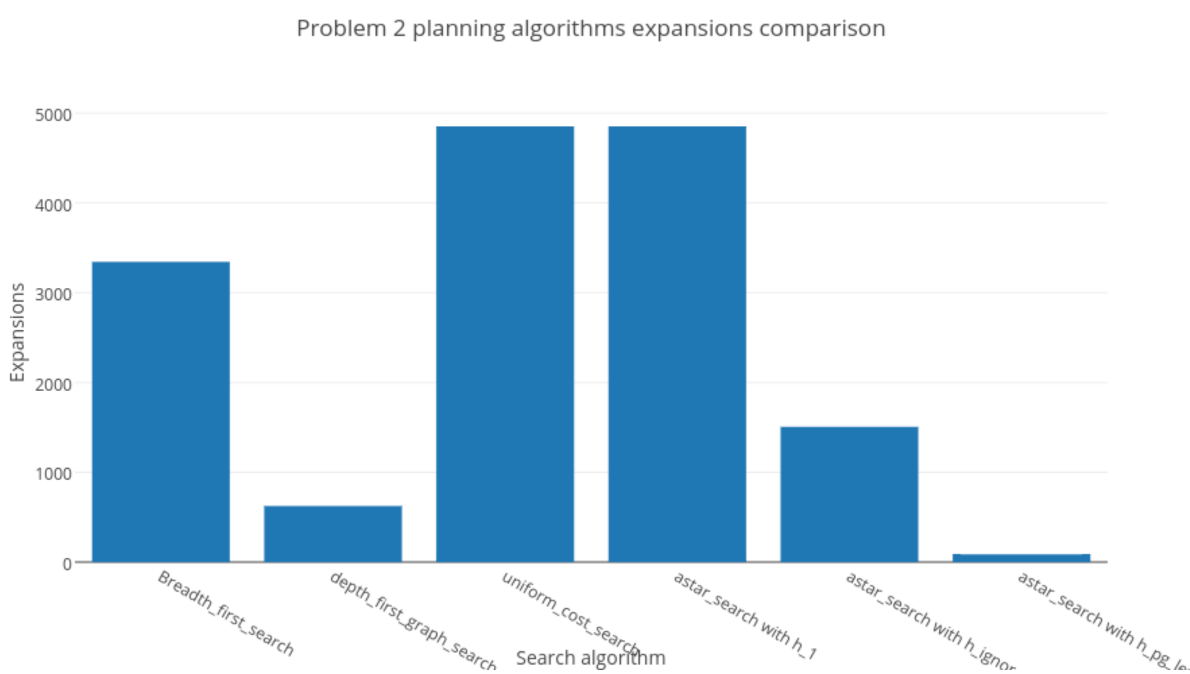
A graphic comparison is also shown below for the number of node expansions and time taken to run 3 of these uninformed and automatic heuristic based planning algorithms for each of the three air cargo problems presented

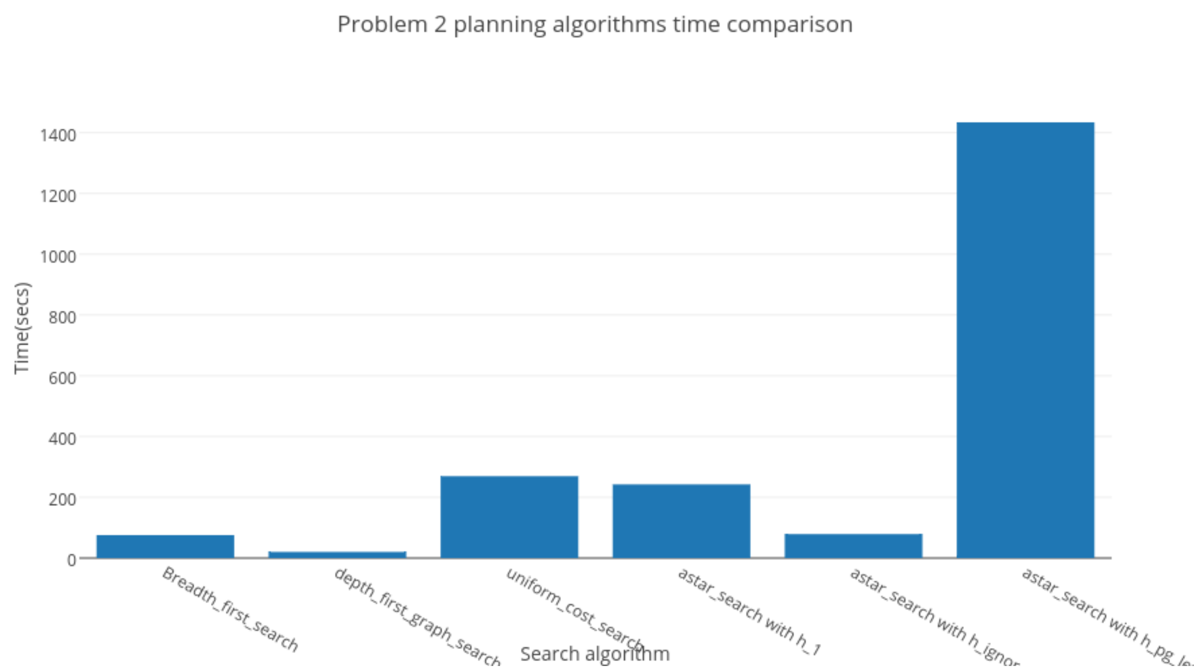
Air cargo Problem 1 node expansions and time taken comparison:



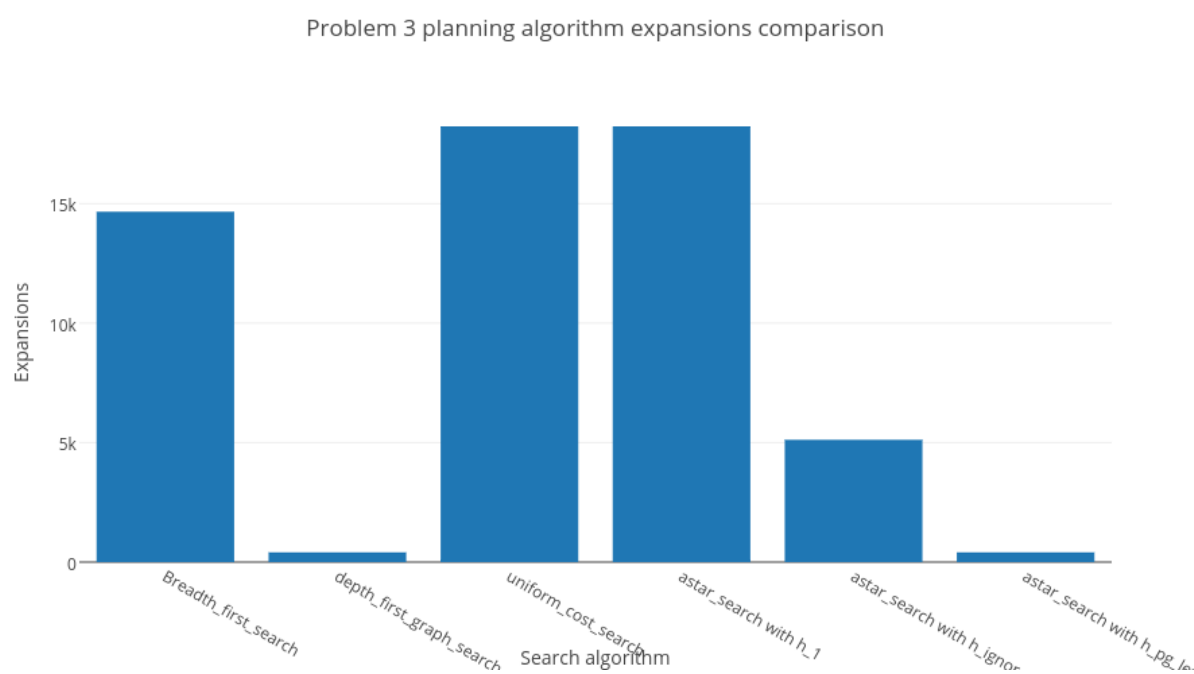


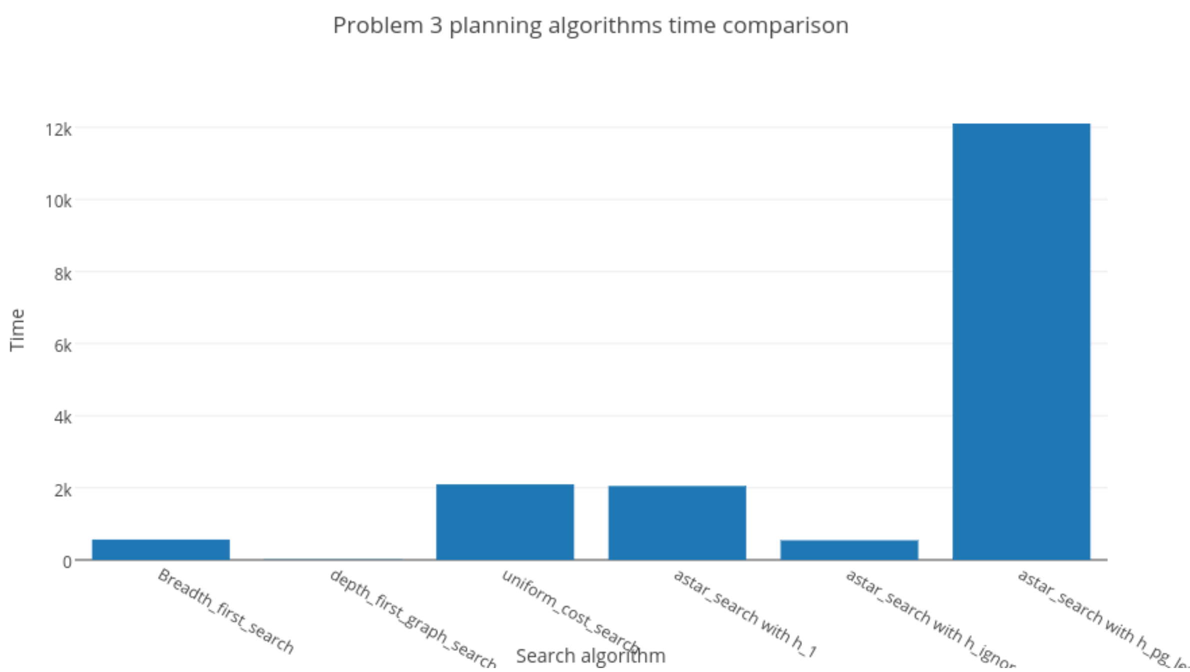
Air cargo problem 2 node expansions and time taken for comparison:





Air cargo problem 3 node expansions and time taken for comparison:




Problem 3 planning algorithms time comparison
Time
12k
10k
8k
6k
4k
2k
0
Breadth_first_search
depth_first_graph_search
uniform_cost_search
astar_search with h_1
Search algorithm

CONCLUSION:

It is clear than in smaller search spaces as in problem 1 and 2, the use of non-heuristic uninformed searches can be optimal and preferred wrt automated heuristic based searches in both node expansion and time costs. As search space gets larger, BFS which is plan optimal for uninformed searches in smaller state spaces slows down considerably while DFS and greedy best search are always non-plan optimal. Uniform cost search and A* search becomes more effective. The use of automated heuristic planning methods with A* are more plan optimal in larger search spaces with the use of planning graphs(level sum heuristic) or with A* search with h_ignore_preconditions where not every node of the space needs to be expanded if a cost estimate can be made before expanding or searching further. This will reduce either one or both the number of node expansions and time taken to reach optimal solution with the use of automated heuristics. A better method in future may be a cross-pollination of uninformed methods and automated heuristic methods to reach optimality in mode expansions and time costs based on search space size and complexity. Among the current problems discussed, using problem 3 which is more representative of complex AI suitable problems, it is clear that A* with h_ignore_preconditions is the most optimal as it “estimates” minimum number of actions from state to goal conditions by ignoring all preconditions, thus reducing processing time and limiting unnecessary node expansions as seen with the uninformed searches.

Bibliographic and Video References used in the analysis above:

Quiz: Tree Search, Udacity Artificial Intelligence course Jan 2017, Lesson 7 (Search), Video 8, minutes 1:46 – 3:11
Quiz: Uniform Cost Search, Udacity Artificial Intelligence course Jan 2017, Lesson 7(Search), Video 16-21
Quiz: A * Search, Udacity Artificial Intelligence course Jan 2017 Video 27-32 of Lesson 7 (Search)

Russell, S. and Norvig, P. Artificial Intelligence, A Modern Approach, Third Edition, Sec 10.3.1 Planning graphs for heuristic estimation, LEVEL SUM HEURISTIC

Russell, S. and Norvig, P. Artificial Intelligence, A Modern Approach, Third Edition, Sec 10.2.3 Heuristics for planning, IGNORE PRECONDITIONS HEURISTIC