Solving air cargo transport via PDDL

Three problems to solve:

1. Preconditions: cargo C1 is at SFO, cargo C2 at JFK. Planes P1 at SFO and P2 at JFK. Goal: C1 at JFK, C2 at SFO.

2. Preconditions: cargos C1 at SFO, C2 at JFK, C3 at ATL. Planes P1 at SFO, P2 at JFK, P3 at ATL. Goal: C1 at JFK, C2 at SFO and C3 at SFO.

3. Preconditions: cargos C1 at SFO, C2 at JFK, C3 at ATL and C4 at ORD. Planes P1 at SFO and P2 at JFK. Goal: C1 and C3 at JFK, C2 and C4 at SFO.

Solutions

Uniform cost search provided following solutions:

Problem 1: Load(C1, P1, SFO), Load(C2, P2, JFK), Fly(P1, SFO, JFK), Fly(P2, JFK, SFO), Unload(C1, P1, JFK), Unload(C2, P2, SFO).

Problem 2: Load(C1, P1, SFO), Load(C2, P2, JFK), Load(C3, P3, ATL), Fly(P1, SFO, JFK), Fly(P2, JFK, SFO), Fly(P3, ATL, SFO), Unload(C1, P1, JFK), Unload(C3, P3, SFO), Unload(C2, P2, SFO).

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

Data from the experiments

Expanded nodes data

	Search method	Air_cargo_1	Air_cargo_2	Air_cargo_3
0	BFS	43.0	3343.0	14663.0
5	DFS	12.0	1669.0	592.0
10	UCS	55.0	4853.0	18164.0
15	Heur_zero	55.0	4853.0	18164.0
20	Ignore_prec_heur	41.0	1450.0	5038.0
25	Astar	11.0	86.0	307.0

New nodes data

	Search method	Air_cargo_1	Air_cargo_2	Air_cargo_3
3	BFS	180.0	30509.0	129631.0
8	DFS	48.0	14863.0	4927.0
13	UCS	224.0	44041.0	159147.0
18	Heur_zero	224.0	44041.0	159147.0
23	Ignore_prec_heur	170.0	13303.0	44924.0
28	Astar	50.0	841.0	2825.0

Time elapsed data

	Search method	Air_cargo_1	Air_cargo_2	Air_cargo_3
2	BFS	0.07	31.14	272.00
7	DFS	0.02	28.19	6.60
12	UCS	0.09	26.96	144.00
17	Heur_zero	0.10	27.51	121.14
22	Ignore_prec_heur	0.07	8.54	34.53
27	Astar	1.30	109.90	524.71

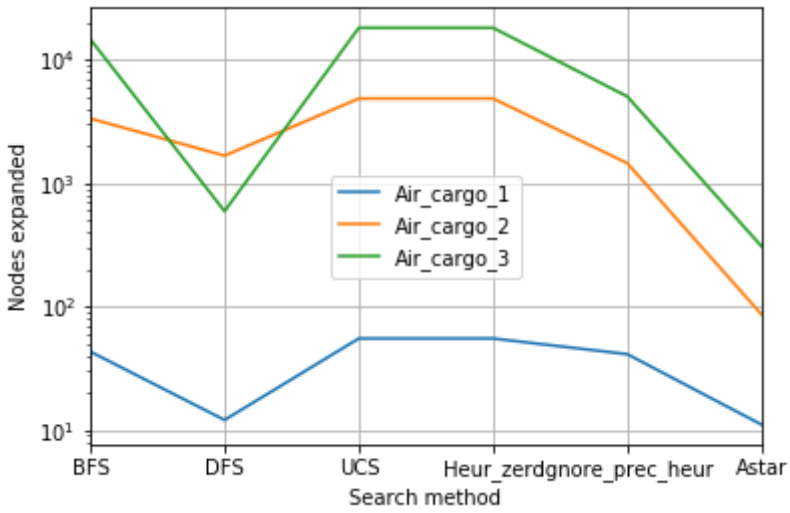
Plan length data

	Search method	Air_cargo_1	Air_cargo_2	Air_cargo_3
4	BFS	6.0	9.0	12.0
9	DFS	12.0	1444.0	571.0
14	UCS	6.0	9.0	12.0
19	Heur_zero	6.0	9.0	12.0
24	Ignore_prec_heur	6.0	9.0	12.0
29	Astar	6.0	9.0	12.0

Search methods performance

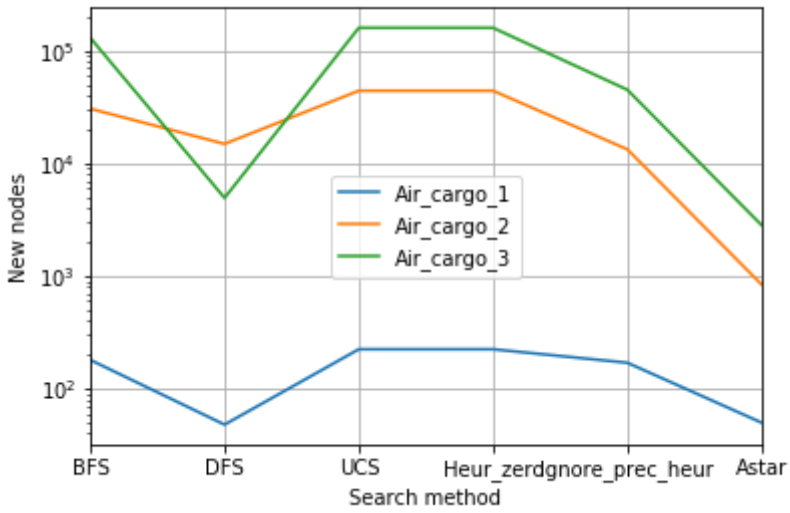
The performance is analyzed by looking into expanded nodes, number of goal tests, time elapsed and optimality

Expanded nodes:



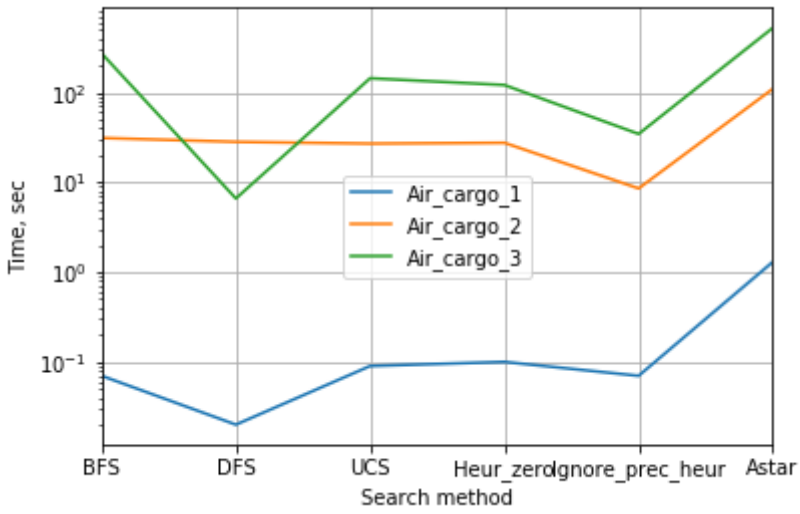
Across all problems Astar method expanded the least amount of nodes: 11, 86 and 307 for Air_cargo problems 1, 2 and 3. The second best method was depth-first-search (marked as DFS). Note, also that uniform cost search (UCS) and Astar with zero heuristics provided the same results as expected. Those two methods expanded the most of nodes 55, 4853 and 18164.

New nodes:



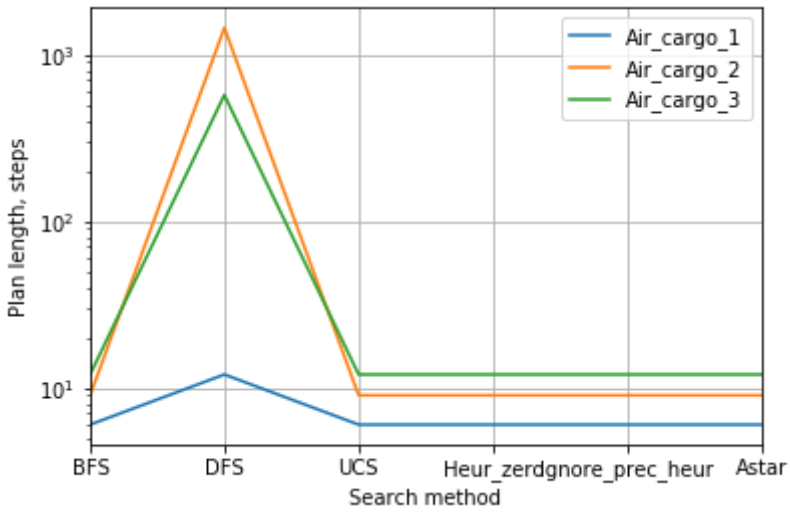
New nodes follow the same pattern as nodes expanded in the first figure.

Time taken by computer to complete the search across different methods and problems



Astar method lapsed about 10 times longer compared to other methods (about 20 times compared to DFS).

Optimality



Here optimality of the search method is expressed in the terms of plan length. Meaning how many steps it takes to solve the problem in the provided solution. Air_cargo_1 takes 6, Air_cargo_2 takes 9 and Air_cargo_3 takes 12 steps in optimal solution. The last figure demostrates that all methods except DFS performed optimally, delivering the shortest path (optimized) solution for the given problem.

S. J. Russell and P. Norvig explained similar conclusions in their book Artificial Inteligence A Modern Approach (3rd edition). In chapter 3, they described the search techniques that were used in this study. They noted that depth-first-search is not optimal.

Conclusions

1. Three uninformed search (depth-first-search, breadth-first-search and uniform-cost-search) an three informed search (Astar with zero heuristics, ignore-preconditions heuristics and planning-graph-levelsum heuristics) were used to solve three plane-cargo-airports planning problems. All the used methods found solutions for all the given problems.

2. Astar with planning-graph-levelsum heuristics expanded the least amount of nodes to find the solution, however it was also the slowest method across all the given problems.

3. All methods except depth-first-search provided the optimal solution.

4. Among uninformed searches depth-first-search expanded the smallest number of nodes, had the shortest lapsed time, however provided solution that was not optimal.

5. Among Astar various heuristics, planning-graph-levelsum heuristics expanded the smallest number of nodes. However, that heuristics had the highest computation costs for these problems and therefore was the slowest. Both Astar's with non-zero heuristics expanded fewer nodes compared to breadth-first-search and uniform-cost-search algorithms.

6. Depth-first-search provided realtively quick solution in terms of expanded nodes and lapsed time. However, the solution was not optimal.

7. All searches lapsed less than 10 minutes.

8. From this study results levelsum heuristics is recommended as it expanded the least amount of nodes. However, if the computation time is an issue it could be better to use ignore-precondition-heurestics.