Analysis of heuristics

Here are the optimal solutions to problems 1, 2 and 3:

problem 1:

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

problem 2:

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

problem 3:

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

follow the data of the solutions of problems 1,2 and 3 using breadth_first_search, in all cases the search is optimal

data from problem 1:

Expansions	Goal Tests	New Nodes
43	56	180

Plan length: 6 Time elapsed in seconds: 0.043315038466222144

data from problem 2:

Expansions	Goal Tests	New Nodes
3343	4609	30509

Plan length: 9 Time elapsed in seconds: 12.843819233324716

data from problem 3:

Expansions	Goal Tests	New Nodes
14663	18098	129631

Plan length: 12 Time elapsed in seconds: 48.05282167525151

and now follow the solution data of problems 1,2 and 3 using depth_first_graph_search, in all cases the search is not optimal

data from problem 1:

Expansions	Goal Tests	New Nodes
21	22	84

Plan length: 20 Time elapsed in seconds: 0.022777216374368955

data from problem 2:

Expansions	Goal Tests	New Nodes
624	625	5602

Plan length: 619 Time elapsed in seconds: 5.136034336524431

data from problem 3:

Expansions	Goal Tests	New Nodes

408 409 3364

Plan length: 392 Time elapsed in seconds: 2.8621150974313836

finally follow the data of the solutions of problems 1,2 and 3 using uniform_cost_search, in all cases the search is optimal

data from problem 1:

Expansions Goal Tests New Nodes
55 57 224

Plan length: 6 Time elapsed in seconds: 0.05318403462025565

data from problem 2:

Expansions Goal Tests New Nodes
4853 4855 44041

Plan length: 9 Time elapsed in seconds: 13.940575055846013

data from problem 3:

Expansions Goal Tests New Nodes

18235 18237 159716

Plan length: 12 Time elapsed in seconds: 60.181018654814366

the data can be summarized in the following table:

algorithm		Problem 1			Problem 2			Problem 3	
	Expansions Goal Tests New Nodes	Plan length	Time(S)	Expansions Goal Tests New Nodes	Plan length	Time	Expansions Goal Tests New Nodes	Plan length	Time
Breadth first search	43 56 180	6	0.0433	3343 4609 30509	9	12.8438	14663 18098 129631	12	48.0528
Depth first search	21 22 84	20	0.0227	624 625 5602	619	5.1360	408 409 3364	392	2.8621

Uniform cost search	55 57 224	6	0.0531	4853 4855 44041	9	13.9405	18235 18237 159716	12	60.1810

Analyzing the data, we can see that the depth search is the fastest of all, and it is also the one that consumes fewer computational resources because it expands fewer nodes, performs less objective tests and creates fewer new nodes, although it is not ideal. first solution, we can see the size of the solutions found, since it is much larger than the solutions found by other algorithms. The wide search performed better than the uniform cost search because, in all cases, it was faster and used fewer computational resources, as we can see from the data in the table above.

The results obtained for problems 1,2 and 3 using the algorithm A * using the "ignore prerequisites" heuristic, in all cases the search is optimal:

data from problem 1:

Expansions Goal Tests New Nodes

41 43 170

Plan length: 6 Time elapsed in seconds: 0.055000395237495

data from problem 2:

Expansions Goal Tests New Nodes

1450 1452 13303

Plan length: 9 Time elapsed in seconds: 4.976795072563467

data from problem 3:

Expansions Goal Tests New Nodes

5040 5042 44944

Plan length: 12 Time elapsed in seconds: 19.886966776549812

Finally, the results obtained for problems 1,2 and 3 are presented below using the A * algorithm using the "sum of level" heuristics, which, like the previous one, is also optimal

data from problem 1:

Expansions Goal Tests New Nodes

11 13 50

Plan length: 6 Time elapsed in seconds: 0.6397213934089655

data from problem 2:

Expansions Goal Tests New Nodes

86 88 841

Plan length: 9 Time elapsed in seconds: 56.141793936918276

data from problem 3:

Expansions Goal Tests New Nodes

325 327 3002

Plan length: 12 Time elapsed in seconds: 308.44176304424826

the data for the algorithm A * can be summarized in the table below:

heuristic		Problem 1			Problem 2			Problem 3	
	Expansions Goal Tests New Nodes	Plan length	Time(S)	Expansions Goal Tests New Nodes	Plan length	Time	Expansions Goal Tests New Nodes	Plan length	Time
ignore previous conditions	41 43 170	6	0.0550	1450 1452 13303	9	4.9767	5040 5042 44944	12	19.8869
level sum	11 13 50	6	0.6397	86 88 841	9	56.1417	325 327 3002	12	308.4417

Analyzing the data, we can notice that the heuristic ignores preconditions is significantly faster than the heuristic of sum of levels, although it consumes more computational resources, because it expands more nodes, performs more objective tests and creates new nodes, we can consider that the heuristic of igonating the preconditions is better than the heuristic of sum of levels, since it had a better

performance in these problems, because it is more important to find the solution more quickly than to consume additional resources like the memory for example, comparing the heuristic to pre-conditions with the non-heuristic searches, we noticed that for problem 2 it presented a better performance in relation to time, and for problem 3 it was not faster than the search in depth, in relation to computational resources, it presented a better performance than all search algorithms, in all problems with the exception of in-depth search, we expanded menons nodes , performed fewer goal tests, and created fewer nodes.