Optimal Plans

Problem 1

This is one possible optimal plan from breadth first search:
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Problem 2

From A* search with ignore preconditions.
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

From A* search with ignore preconditions
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)

Comparison of non-heuristic search result metrics

For problem 1, the non-heuristic search strategies produced a solution for the air cargo problem. Breadth first search (BFS) and Uniform cost search (UCS) found a optimal solution both in similar search time. However Depth first graph search (DFGS) is faster than the other two, but the solution found is not optimal (20 actions needed) [1]. Depth first search only considers the expanding in the depth of the state graph, at the end it doesn't necessary finds the optimal solution but only a solution. However the amount of memory required in DFGS is close to half that the needed in BFS and UCS [2].

For problem 2 and problem 3 results are very similar in the sense that BFS and UCS were able to find an optimal solution both in 9 and 12 steps, however BFS required less search time as well as less memory for new nodes. DFGS was not able to find a optimal plan, however the search time is smaller.

So in general if the problem is more complex, DFGS will find a plan no necessary the best, but faster and with less memory requirements than the other non-heuristic search methods BFS and UCS.

Problem 1

Search	Expanded Nodes	# of goal test	New nodes	Search time	Plan length
BFS	41	56	180	0.03	6
DFGS	21	22	84	0.013	20
UCS	55	57	224	0.036	6

Problem 2

Search	Expanded Nodes	# of goal test	New nodes	Search time	Plan length
BFS	3343	4609	30509	5.9121	9
DFGS	624	625	5602	2.604	619
UCS	4852	4854	44030	8.754	9

Problem 3

Search	Expanded Nodes	# of goal test	New nodes	Search time	Plan length
BFS	14663	18098	129631	32.381	12
DFGS	408	409	3364	1.44	392
UCS	18235	18237	159716	45.824	12

Comparison of search result metrics using A* with Ignore preconditions and Level Sum

When using the A* search with ignore preconditions and level sum, they both were able to find an optimal solution for problems 1 and 2, however for problem 3 A* Level sum, took so long > 30min. However A* with ignoring preconditions, the search result was the fastest search time and with optimal solution from all other 9 search strategies.

Problem 1

Search	Expanded Nodes	# of goal test	New nodes	Search time	Plan length
A* Ignore Preconditions	41	43	170	0.033	6
A* Levelsum	11	13	50	0.784	6

Problem 2

Search	Expanded Nodes	# of goal test	New nodes	Search time	Plan length
A* Ignore Preconditions	1450	1452	13303	3.438	9
A* Levelsum	86	88	841	135.56	9

Problem 3

Search	Expanded Nodes	# of goal test	New nodes	Search time	Plan length
A* Ignore Preconditions	5040	5042	44944	12.925	12
A* Levelsum	**NA**	**NA**	**NA**	**NA**	**NA**

Which are the fastest and slowest uniformed algorithms?

Fastest: DFGS.

Slowest: BFTS.

From lesson Search, We know that BFS will always find an optimal solution and will always expands the shortest paths before [1], in BFTS this means that will expand every node in the level order of the tree until finds a goal node [1], this is why it takes longer to find a solution. However DFGS does not guarantee optimal path and that's the reason of why has a longer plan length. DFGS is the fastest in the three problems because the number of expanded nodes are the lowest, however not for every problem it can be faster than BFTS [3].

Which are the fastest and slowest heuristic searches?

Fastest: For all problems Greedy Best first graph search was the fastest, however at problem 3 A* with heuristic ignore preconditions was closed as fast.

Slowest: Recursive Best First Search.

Greedy first graph search is a very goal directed approach and always tries to minimize the search cost, however it is not an optimal plan [5]. Recursive Best First Search was the slowest of all heuristic search because of its nature, suffers from node regeneration overhead, which makes the search take longer [6] .

Final Words and conclusion

From all 10 search strategies we can conclude that BFS was the best non-heuristic method, always able to an optimal plan with still ok time search and also memory requirements (2^n) [1]. From all other heuristic searches A* with Ignore preconditions performed the best, even better than BFS, this was clearly seen in problem 3, where the complexity was higher than in problem 1 and 2. Ignore preconditions heuristic can estimate the minimum number of actions needed to the goal state by ignoring the preconditions for an action to be done [4]. From the tests we found that A* with ignore preconditions can be the used as first search strategie when the problem is complex.

The full results from the different search strategies is shown below

Problem	Search Type	# Expanded nodes	Number of goal test	New nodes	Search time	Plan lenght
Air Cargo 1	breadth first search	41	56	180	0.03	6
	breadth first tree search	1458	1459	5960	0.83	6
	depth first graph search	21	22	84	0.013	20
	depth limited search	101	271	414	0.0803	50
	uniform cost search	55	57	224	0.036	6
	recursive best first search h1	4229	4230	17023	2.38	6
	greedy best first graph search h1	7	9	28	0.005	6
	A* search h1	55	57	224	0.038	6
	A* search h ignore preconditions	41	43	170	0.033	6
	A* search h pg levelsum	11	13	50	0.784	6
Air Cargo 2	breadth first search	3343	4609	30509	5.9121	9
	breadth first tree search					
	depth first graph search	624	625	5602	2.604	619
	depth limited search	222719	2053741	2054119	673.409	50
	uniform cost search	4852	4854	44030	8.754	9
	recursive best first					

	search h1					
	greedy best first graph search h1	990	992	8910	1.766	21
	A* search h1	4852	4854	44030	8.738	9
	A* search h ignore preconditions	1450	1452	13303	3.438	9
	A* search h pg levelsum	86	88	841	135.56	9
Air Cargo 3	breadth first search	14663	18098	129631	32.381	12
	breadth first tree search					
	depth first graph search	408	409	3364	1.44	392
	depth limited search					
	uniform cost search	18235	18237	159716	45.824	12
	recursive best first search h1					
	greedy best first graph search h1	5613	5615	49420	12.563	21
	A* search h1	18235	18237	159716	40.111	12
	A* search h ignore preconditions	5040	5042	44944	12.925	12
	A* search h pg levelsum					

References

1. Search Comparison , retrieve from https://www.youtube.com/watch?time_continue=59&v=RMt_NiyY4nU.

2. Lesson Search 7, Tree Search Continued.
3. **https://stackoverflow.com/questions/47222855/in-what-sense-is-dfs-faster-than-bfs.**
4. Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (2003). *Artificial intelligence: a modern approach* (Vol. 2, No. 9). Upper Saddle River: Prentice hall.
5. https://cseweb.ucsd.edu/classes/sp07/cse150/lectures-pdf/l.newsearch.2.pdf
6. Hatem, M., Kiesel, S., & Ruml, W. (2015, January). Recursive Best-First Search with Bounded Overhead. In *AAAI* (pp. 1151-1157).