AIND Planning project by Folajimi Adekoya

1. **Optimal Plans**
   According to page 82 Artificial intelligence: A modern approach, Breath –first search is optimal if the path cost is a non-decreasing function of the depth of the node. An example of this scenario is when all actions have the same cost. Based on this, I derived my optimal plans from running BFS for problem 1,2 and 3.

Air cargo Problem 1

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

Air cargo Problem 2

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

Air cargo Problem 3

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

2. **Non-heuristic search**

Air cargo problem 1

Algorithm	Node expansions	Goal tests	Time elapsed	Solution optimality
BFS	43	56	0.03285	T
DFGS	21	22	0.01693	F
UCS	55	57	0.04437	T

Air cargo problem 2

Algorithm	Node expansions	Goal tests	Time elapsed	Solution optimality
BFS	3342	4609	15.5744	T
DFGS	624	625	3.99721	F
UCS	4853	4855	12.7341	T

Air cargo problem 3

Algorithm	Node expansions	Goal tests	Time elapsed	Solution optimality
BFS	14663	18098	120.0855	T
DFGS	408	409	1.891418	F
UCS	18164	18166	56.67243	T

Analysis

From the table above we can see for all three problems that the BFS and the UCS provide optimal solutions, but the DFGS does not. It can be seen in the order of time taken to complete a search that the DFGS is the fastest although its solution is not optimal. The BFS is the fastest optimal solution. It can also be seen that the DFGS has a lower node expansion that the other two algorithms. This number is lower probably because explored nodes are removed from memory giving lower node expansion and a faster execution speed.

3. Heuristic search

Air cargo problem 1

A* Search Algorithm	Node expansions	Goal tests	Time elapsed(S)	Solution optimality
h1	55	57	0.044290	
h ignore preconditions	41	43	0.055884	
h pg levelsum	11	13	1.033293	

Air cargo problem 2

A* Search Algorithm	**Node expansions**	**Goal tests**	**Time elapsed(S)**	**Solution optimality**
h1	4853	4855	15.51890	T
h ignore preconditions	1450	1452	5.481276	T
H pg levelsum	86	88	180.2825973	T

Air cargo problem 3

A* Search Algorithm	**Node expansions**	**Goal tests**	**Time elapsed(S)**	**Solution optimality**
h1	18164	18166	65.629012	T
h ignore preconditions	5038	5040	21.77811	T
h pg levelsum	314	316	1028.85284	T

Analysis

From the tables above, it can be seen that for the three problems, A* algorithms levelsum gives the lowest node expansion while h_ignore executed the fastest. The time taken to generate the planning graphs on the average far outweighs the efficiency improvements made. In this case h_ignore is a favourite because of its speed compared to the others.

4. Comparing the best Non-Heuristic and the best Heuristic search

The BFS is considered the best Non-heuristic search while the H_ignore is considered the best heuristic search.

Problem	Algorithim	Node expansions	Goal tests	Time elapsed(S)
1	BFS	43	56	0.03285
	A* h_ignore	41	43	0.055884
2	BFS	3342	4609	15.5744
	A* h_ignore	1450	1452	5.481276
3	BFS	14663	18098	120.0855
	A* h_ignore	5038	5040	21.77811

In all of the three problems, it can be seen in the table above that A* h_ignore has a lower node expansion than BFS for all three problems. The difference in nodes visited increases with increasing problem complexity. I have highlighted in yellow the fastest algorithm in each problem. It can be seen that the BFS is faster with lower problem complexity but as the problem becomes more complicated the A* h_ignore performs better. This means for very simple problems it is better to apply BFS, but for complicated problems it is better to apply A* h_ignore.