Planning Heuristic Analysis

Problem 1 optimal plan	Problem 2 optimal plan	Problem 3 optimal plan
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	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(C3, P3, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	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(C4, P2, SFO) Unload(C3, P1, JFK) Unload(C1, P1, JFK) Unload(C2, P2, SFO)

Uninformed non-heuristic search such as **Breadth First Search (BFS)**, **Depth First Search (DFS)** and **Uniform Cost Search (UCS)** do not utilize any domain specific knowledge about the problem. Only BFS and UCS result in complete and optimal plans while DFS's solution is not guaranteed to be optimal. UCS performs better than BFS in terms of time but expands more nodes. BFS is preferred if memory is a concern but UCS is preferred in terms of time performance.

Heuristic search such as **Astar_search_with_h_ignore_preconditions** and **Astar_search_with_h_pg_levelsum** result in complete and optimal plans. **Astar_search_with_h_ignore_preconditions** adds nodes to the planning graph by ignoring preconditions and thus allowing a greater number of possible paths to explore. **Astar_search_with_h_pg_levelsum** is a heuristic to estimate the cost of reaching the goal.

The levelsum heuristic expands far fewer nodes but is much slower than the ignore preconditions heuristic, BFS and UCS and is advantageous when memory is the constraint. The ignore preconditions heuristic outperforms BFS and UCS both in terms of time and node expanded. Thus, **Astar_search_with_h_ignore_preconditions** is the recommended search strategy for the Cargo problem.

Problem 1	Search	Expansions	Goal Tests	New Nodes	Plan length	Time (s)	Optimal
1	breadth_first_search	43	56	180	6	0.037	Yes
2	breadth_first_tree_search	1458	1459	5960	6	1.145	Yes
3	depth_first_graph_search	12	13	48	12	0.010	No
4	depth_limited_search	101	271	414	50	0.112	No
5	uniform_cost_search	55	57	224	6	0.051	Yes
6	recursive_best_first_search with h_1	4229	4230	17029	6	3.537	Yes
7	greedy_best_first_graph_search with h_1	7	9	28	6	0.007	Yes
8	astar_search with h_1	55	57	224	6	0.054	Yes
9	astar_search with h_ignore_preconditions	41	43	170	6	0.048	Yes
10	astar_search with h_pg_levelsum	11	13	50	6	0.998	Yes

Problem 2	Search	Expansions	Goal Tests	New Nodes	Plan length	Time (s)	Optimal
1	breadth_first_search	3343	4609	30509	9	17.629	Yes
2	breadth_first_tree_search						
3	depth_first_graph_search	582	583	5211	575	3.843	No
4	depth_limited_search						
5	uniform_cost_search	4852	4854	44030	9	16.314	Yes
6	recursive_best_first_search with h_1						
7	greedy_best_first_graph_search with h_1	990	992	8910	21	3.309	No
8	astar_search with h_1	4852	4854	44030	9	16.246	Yes
9	astar_search with h_ignore_preconditions	1450	1452	13303	9	5.783	Yes
10	astar_search with h_pg_levelsum	86	88	841	9	98.450	Yes

Problem 3	Search	Expansions	Goal Tests	New Nodes	Plan length	Time (s)	Optimal
1	breadth_first_search	14663	18098	128605	12	136.277	Yes
2	breadth_first_tree_search						
3	depth_first_graph_search	3664	3665	29381	195	21.909	No
4	depth_limited_search						
5	uniform_cost_search	18235	18237	158272	12	71.806	Yes
6	recursive_best_first_search with h_1						
7	greedy_best_first_graph_search with h_1	5631	5633	48822	26	21.593	No
8	astar_search with h_1	18235	18237	158272	12	71.302	Yes
9	astar_search with h_ignore_preconditions	5040	5042	44769	12	23.807	Yes
10	astar_search with h_pg_levelsum	393	395	3508	12	685.226	Yes