Heuristic Analysis

Air Cargo Problem 1
States: 12
State Space: 4096

	Expansions	Goal Tests	New Nodes	Time	Optimal
breadth_first_search	43	56	180	0.027	YES
depth_first_graph_search	12	13	48	0.0079	NO
uniform_cost_search	55	57	224	0.032	YES
h_pg_levelsum	39	41	158	1.0	YES

Optimal Solution

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

Air Cargo Problem 2
States: 27
State Space: 134,217,728

	Expansions	Goal Tests	New Nodes	Time	Optimal
breadth_first_search	3346	4612	30534	12.41	YES
depth_first_graph_search	1124	1125	10017	7.37	NO
uniform_cost_search	4853	4855	44041	10.39	YES
h_pg_levelsum	1129	1131	10232	390.46	YES

Optimal Solution

Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
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 3
States: 32
State Space: 4,294,967,296

	Expansions	Goal Tests	New Nodes	Time	Optimal
breadth_first_search	14120	17673	124926	92.10	YES
depth_first_graph_search	667	678	5608	3.36	NO
uniform_cost_search	18223	18225	159618	58.375	YES
h_ignore_preconditions	5040	5042	44944	15.80	YES
h_pg_levelsum	2026	2028	17933	1308.92	YES

Optimal Solution

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

Unload(C1, P1, JFK)

In general, depth_first_graph_search could not return optimal result. In our problem sets, because the events do not depend on each other, so BFS simulates exploration in parallel, that does find optimal solution. uniform_cost_search achieved the best performance without any heuristics.

A* search with simple ignore preconditions heuristic greatly improved the performance. By comparing with level sum, although ignore preconditions did more expansions, but the simplicity of the heuristic algorithm enabled much better search performance.

PlanningGraph with level sum heuristic achieves the best performance in terms of exploration. But due to the complexity of the algorithm, the running time is not optimal. I did the a profiling session and figured out method *update_a_mutex* took a big amount of time to execute, it could be the potential place to improve performance.

In general, ignore preconditions is the best heuristic. It is fast, but may not guaranteed to find optimal solution.