Optimal problem 1 plan:
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Optimal problem 2 plan:
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Optimal problem 3 plan:
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

	Problem 1	Problem 2	Problem 3
Breadth-first	Optimal 0.028 seconds 43 expansions	Optimal 11.890 seconds 3343 expansions	Optimal 89.474 seconds 14663 expansions
Breadth-first tree	Optimal 0.836 seconds 1458 expansions	Takes >10 mins	Takes >10 mins
Depth-first graph	Not Optimal 0.007 seconds 12 expansions	Not Optimal 3.487 seconds 582 expansions	Not Optimal 2.941 seconds 627 expansions
Depth limited	Not Optimal 0.078 seconds 101 expansions	Not Optimal 1,027.468 seconds 222719 expansions	Takes > 10 minutes
Uniform cost	Optimal 0.033 seconds 55 expansions	Optimal 11.365 seconds 4823 expansions	Optimal 49.091 seconds 18223 expansions
Recursive best first h1	Optimal 2.460 seconds 4229 expansions	Takes >10 minutes	Takes > 10 minutes
Greedy best first graph h1	Optimal 0.005 seconds 7 expansions	Not Optimal 0.866 seconds 385 expansions	Not Optimal 15.284 seconds 5578 expansions
A* h1	Optimal 0.033 seconds 55 expansions	Optimal 11.373 seconds 4823 expansions	Optimal 50.602 seconds 18223 expansions
A* ignore preconditions	Optimal 0.033 seconds 41 expansions	Optimal 4.011 seconds 1421 expansions	Optimal 16.110 seconds 5040 expansions
A* pg levelsum	Optimal 0.478 seconds 11 expansions	Optimal 41.437 seconds 86 expansions	Optimal 210.801 seconds 316 expansions

The uniform cost search and breadth first searches give optimal results, but start to take a significant amount of time and expansions when the problem space grows a small amount, when compared with depth first graph search. Depth first graph search, while finding solutions much faster, particularly for problem 3, do not provide optimal plans, and don't give consistent plans between runtimes. In real life situations like air cargo where pointless flights cost much more time and resources than a bit of extra cpu time, this flaw would make depth first graph search unacceptable.

I found greedy best first first graph search using h1 to be interesting. It always gave optimal results for problem 1. For problems 2 and 3, while not being optimal, greedy best first graph search with h1 consistently gave the same plan, unlike the depth first graph search which would give arbitrary plans with huge number of steps. Looking at the plan though, there is some questionable logic with loading cargo and flying it to a different airport to be picked up by a different plane and flown to the final airport, when flying directly would certainly be faster. Greedy best first graph search finishes the fastest of the heuristic searches for problems 2 and 3.

A* ignore preconditions and A* pg levelsum seem like the two heuristic strategies I would actually choose to use in a real life instance of this problem. Both give optimal solutions in reasonable amounts of time. Of all the searches that give optimal results, A* ignore preconditions has the fastest run time and A* pg levelsum is the most efficient in terms of expansions, for problems 2 and 3. Greedy best first search graph beats A* ignore preconditions and A* pg levelsum for expansions and runtime, but I'd argue that since this case is so trivial in complexity, and runs so quickly across all searches, that it isn't a good indicator of which algorithm is the best choice. Whether A* pg levelsum or A* ignore preconditions is the best search depends on whether runtime or expansions is more important in a given situation, but as runtime seems to be more important in most real life applications, I'd vote for A* ignore preconditions as the best search of them all.