Problem 1 optimal solution (Plan length: 6):

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

Metrics- Search |	Expansions	New Nodes	Plan length (Optimality)	Time elapsed (s)
Breadth_first	43	180	6	0.0445
Depth_first_graph	12	48	12	0.0135
Uniform_cost	55	224	6	0.0605
Astar with h_1	55	224	6	0.0587
Astar with h_ignore_precond	41	170	6	0.0571
Astar with h_pg_levelsum	11	50	6	1.293

For problem 1, breadth_first_search was the best overall non-heuristic search method, as it has a small number of expansions, new nodes, and takes a decent amount of time to carry out, while achieving 100% optimality. Depth_first_graph_search, while not achieving 100% optimality, should be noted as it has the best metrics among the 3 non-heuristic search methods by a large margin.

Among heuristic search methods, Astar search with h_ignore_preconditions has the best balance between expansions, new nodes, and time elapsed, while achieving 100% optimality. While Astar search with h_pg_levelsum is considerably less resource intensive than the other heuristic search methods, the long time elapsed makes it unfeasible for use in problem 1.

Problem 2 optimal solution (Plan length: 9):

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

Metrics- Search |	Expansions	New Nodes	Plan length (Optimality)	Time elapsed (s)
Breadth_first	3343	30509	9	20.5
Depth_first_graph	582	5211	575	4.41
Uniform_cost	4853	44041	9	17.1
Astar with h_1	4853	44041	9	18.0
Astar with h_ignore_precond	1450	13303	9	6.32
Astar with h_pg_levelsum	86	841	9	284

In tackling problem 2, uniform_cost_search was the best overall non-heuristic search method. Although it has the most number of expansions and new nodes, and takes a decent amount of time to carry out, it achieves 100% optimality in less time than breadth_first_search. Depth_first_graph_search, while producing poor optimality, should be noted as it has the best metrics among the 3 non-heuristic search methods by a large margin.

Among heuristic search methods, Astar search with h_ignore_preconditions has the best statistics in expansions, new nodes, and time elapsed, all while achieving 100% optimality. While Astar search with h_pg_levelsum is less resource intensive by large oders of magnitude than the other heuristic search methods, the extremely long time elapsed makes it unfeasible for use in problem 2.

Problem 3 optimal solution (Plan length: 12):

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

Metrics- Search |	Expansions	New Nodes	Plan length (Optimality)	Time elapsed (s)
Breadth_first	14663	129631	12	152
Depth_first_graph	627	5176	596	4.83
Uniform_cost	18223	159618	12	74.3
Astar with h_1	18223	159618	12	76.9
Astar with h_ignore_precond	5040	44944	12	28.9
Astar with h_pg_levelsum	320	2956	12	1760

Looking at problem 3, uniform_cost_search was the best overall non-heuristic search method. Although it has the most number of expansions and new nodes, and takes a decent amount of time to carry out, it achieves 100% optimality in less time than breadth_first_search. Depth_first_graph_search, while producing very poor optimality, should be noted as it has the best metrics among the 3 non-heuristic search methods by a large margin, and takes significantly less time than the other methods.

Among heuristic search methods, Astar search with h_ignore_preconditions has the most balanced metrics in expansions, new nodes, and takes the least amount of time to complete. While Astar search with h_pg_levelsum is less resource intensive by large oders of magnitude than the other heuristic search methods, the extremely long time elapsed makes it unfeasible for use in problem 3.

Overall, the best heuristic used in these problems was h_ignore_preconditions with Astar search. It also consistently outperformed all non-heuristic search planning methods for all the problems as it had the best search metrics in each problem.

References

1. Russell, S, and P Norvig. Artificial Intelligence: A Modern Approach, Chapter 10: Classical Planning
2. Udacity AIND Lesson 8, videos 23, 26, 27