Optimal Sequences

Problem 1: 6 Actions
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Problem 2: 9 Actions
Load(C1,P1,SF0)
Load(C2,P2,JFK)
Load(C3,P3,ATL)
Fly(P1,SFO,JFK)
Fly(P2,JFK,SFO)
Fly(P3,ATL,SFO)
Unload(C1,P1,JFK)
Unload(C2,P2,SFO)
Unload(C3,P3,SFO)

Problem 3: 12 Actions
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)

Heuristic Analysis

N/A -- RunTime greater than 2 hours

Time (secs) /Expansions	BFSearch	BFTreeSrch	DFGrphSrch	DFLimitSrch	UniformCost	RecursivBFS	GreedyBFS-h1	A*-h1	A*-ignore precondition	A*-h_pg_level
Prob 1	0.08 / 43	2.29 /1458	0.03 / 21	0.18 / 101	0.10 / 55	6.64 / 4229	0.01 / 55	0.10 / 55	0.08 / 33	2.12 / 29
Prob 2	25 / 3343	N/A	5.3 / 624	N/A	60 / 4853	N/A	10 / 998	59.1 / 4853	30.8 / 2076	N/A
Prob 3	152 / 14663	N/A	3.02 / 408	N/A	436/18223	N/A	136 / 4933	491 / 18223	359 / 11905	N/A

Number of Steps in Solution

(Optimal)	BFSearch	BFTreeSrch	DFGrphSrch	DFLimitSrch	UniformCost	RecursivBFS	GreedyBFS-h1	A*-h1	A*-ignore precondition	A*-h_pg_level
Prob1 (6)	6	6	20	50	6	6	6	6	6	6
Prob2 (9)	9	-	619	-	9	-	15	9	9	-
Prob3(12)	12	-	392	-	12	-	22	12	13	-

Of the deterministic algorithms, Breadth-First-Search consistently provides optimal solutions and performs better than heuristic search like Uniform Cost, with respect to run time and node expansions. Of the heuristic methods, A*-h1 and A*-ignore-preconditions and uniform cost perform very well, and provide optimal solutions. However, A*-ignore-preconditions explores the fewest number of nodes and completes the fastest without sacrificing accuracy. Uniform Cost and A*-h1 expand the exact same number of nodes and run the same number of goal tests.

For very small problem sizes, most of the search techniques perform well and provide optimal solutions, including Depth-first Graph Search and Depth-first Tree Search. These two search techniques provide poor solutions that are very far from optimal, even if their run time is reasonable.

Differences between the search techniques come to the fore, as the problem size increases, and when problem constraints are increased.   For example, Comparing the run times of the techniques that result in optimal solutions, with BFS taking least time and A*-h1 and uniform cost taking approximately equal times for all problems (and taking most time to reach the goal state).
BFS < A*-ignore-precondition < A*-h1 == Uniform Cost

The superior performance of the A*-ignore-precondition algorithm is because of the weakening of the preconditions to derive an admissible heuristic that does not over-estimate the distance from the current state to a goal, as described in Section10.2.3 of the AIMA textbook.