Shane Harmon Udacity AIND Project 3 Algorithm/Heuristic exploration.
Project 3 Path Planning

Because A* is guaranteed to be optimal I will use it to determine the optimal plans for each of the problems.

Optimal plan for problem 1 A* h_1

Expansions	Goal Tests	New Nodes
43	56	180

Plan length: 6 Time elapsed in seconds: 0.02899806573987007
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Optimal plan for problem 2 A* h_1

Expansions	Goal Tests	New Nodes
43	56	180

Plan length: 6 Time elapsed in seconds: 0.02899806573987007
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Optimal plan for problem 3 A* h_1

Expansions	Goal Tests	New Nodes
18053	18055	158200

Plan length: 12 Time elapsed in seconds: 36.0016542407684
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(C1, P1, JFK)
Unload(C2, P2, SFO)
Unload(C3, P1, JFK)

Unload(C4, P2, SFO)

Compare and contrast non-heuristic search result metrics :

breadth_first_search problem 1

Expansions	Goal Tests	New Nodes
55	57	224

Plan length: 6 Time elapsed in seconds: 0.026286126114428043

depth_first_graph_search problem 1

Expansions	Goal Tests	New Nodes
43	56	180

Plan length: 6 Time elapsed in seconds: 0.026731600053608418

uniform_cost_search   problem 1

Expansions	Goal Tests	New Nodes
55	57	224

Plan length: 6 Time elapsed in seconds: 0.026286126114428043

breadth_first_search problem 2

Expansions	Goal Tests	New Nodes
2844	4033	23627

Plan length: 9 Time elapsed in seconds: 4.659194177016616

depth_first_graph_search problem 2

Expansions	Goal Tests	New Nodes
44	45	275

Plan length: 38 Time elapsed in seconds: 0.06145387468859553

uniform_cost_search   problem 2

Expansions	Goal Tests	New Nodes
4280	4282	35043

Plan length: 9 Time elapsed in seconds: 6.539343740791082

breadth_first_search problem 3

Expansions	Goal Tests	New Nodes
14663	18098	129631

Plan length: 12 Time elapsed in seconds: 27.948343598283827

depth_first_graph_search problem 3

Expansions	Goal Tests	New Nodes
627	628	5176

Plan length: 596 Time elapsed in seconds: 2.0642239288426936

uniform_cost_search problem 3

Expansions	Goal Tests	New Nodes
18053	18055	158200

Plan length: 12 Time elapsed in seconds: 35.31203967612237

For problem one all algorithms performed similarly with depth first search expanding a few less nodes. Depth first search performed better for these problems in general, as the complexity of the problems increased Depth First search time, expansions, goal tests, and new nodes was far less than Breadth First Search and Uniform Cost Search. The trade off is that depth first is not guaranteed to find a solution, where breadth first search is. None of these algorithms are guaranteed to find an optimal solution, and the general performance of these algorithms can't be deduced from these problems as they are have fairly easily found solutions, or are fairly small, but their performance here can be a good indicator of the pros and cons of each.

astar_search h_ignore_preconditions problem 1:

Expansions	Goal Tests	New Nodes
41	43	170

Plan length: 6 Time elapsed in seconds: 0.025575700215995312

astar_search h_pg_levelsum problem 1

Expansions	Goal Tests	New Nodes
11	13	50

Plan length: 6 Time elapsed in seconds: 0.36015109717845917

astar_search h_ignore_preconditions problem 2:

Expansions	Goal Tests	New Nodes
1272	1274	10732

Plan length: 9 Time elapsed in seconds: 2.413294856902212

astar_search h_pg_levelsum problem 2:

Expansions	Goal Tests	New Nodes
172	174	1314

Plan length: 9 Time elapsed in seconds: 32.87612859904766

astar_search h_ignore_preconditions problem 3:

Expansions	Goal Tests	New Nodes
5038	5040	44926

Plan length: 12 Time elapsed in seconds: 11.193901665043086
astar_search h_pg_levelsum problem 3

Expansions	Goal Tests	New Nodes
317	319	2928

Plan length: 12 Time elapsed in seconds: 179.88011232530698

For all three problems it seems that each heuristic has a different strength. Ignore Preconditions heuristic seems to run much faster then the Levelsum heuristic at the expense of using more system resources, more expansions, goal tests and nodes. The Levelsum seems to use less resources at the expense of time. Each one may be useful for different applications.

I would say the best heuristic for these problems is the Ignore Preconditions heuristic, because it runs faster and utilizes a reasonable amount of resources. It performs better than all of the non-heuristic methods because it is not blindly searching, paths that lead astray are penalized and it is guaranteed to find an optimal solution.