Heuristic Analysis

Air Cargo Problem 1					
Algorithm	**Exapnsions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (seconds)**
breadth_first_search	43	56	180	6	0.061
depth_first_graph_search	21	22	84	20	0.016
breadth_first_tree_search	1458	1459	5960	6	0.865
depth_limited_search	101	271	414	50	0.099
uniform_cost_search	55	57	224	6	0.045
recursive_best_first_search h1	4229	4230	17023	6	2.567
greedy_best_first_search h1	7	9	28	6	0.012
astar_search h1	55	57	224	6	0.059
astar_search h_ignore_predictions	41	43	170	6	0.054
astar_search h_pg_levelsum	11	13	50	6	0.482
Air Cargo Problem 2					
breadth_first_search	3343	4609	30509	9	14.967
depth_first_graph_search	624	625	5602	619	3.142
breadth_first_tree_search	Time out				
depth_limited_search	Time out				
uniform_cost_search	4853	4855	44041	9	11.583
recursive_best_first_search h1	Time out				
greedy_best_first_search h1	998	1000	8982	17	2.451
astar_search h1	4853	4855	44041	9	11.527
astar_search h_ignore_predictions	1450	1452	13303	9	3.828
astar_search h_pg_levelsum	86	88	841	9	40.478
Air Cargo Problem 3					
breadth_first_search	1125	14120	124926	12	100.722
depth_first_graph_search	1086	1087	9027	1055	7.941
breadth_first_tree_search	Time out				
depth_limited_search	Time out				
uniform_cost_search	18138	18140	158888	12	50.676
recursive_best_first_search h1	Time out				
greedy_best_first_search h1	5335	5337	47089	32	14.804
astar_search h1	18138	18140	158888	12	51.756
astar_search h_ignore_predictions	5038	5040	44926	12	14.983
astar_search h_pg_levelsum	316	318	2919	12	210.598

Figure 1.0:A comparison of the results from running different planning algorithms with different heuristics on three seperate air cargo problems with increasing complexity. Optimal solutions are highlighted in green.

Optimal Plans

The optimal plan for Problem 1 was the `greedy_best_first_search` algorithm with the `h1` heuristic. As seen in Figure 1.0, It discovered the goal with an ideal plan length (Figure 1.1) in the shortest time.

```
Plan length: 6  Time elapsed in seconds: 0.009350679989438504
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

Figure 1.1: An optimal sequence of actions for Problem 1.

The optimal plan for Problem 2 was the `astar_search` algorithm with the `h_ignore_preconditions heuristic`. It made a midrange amount of expansions and discovered the goal with the ideal path length (Figure 1.2) in the fastest time.

```
Plan length: 9  Time elapsed in seconds: 4.1634205669979565
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
```

Figure 1.2: An optimal sequence of actions for Problem 2.

Much like in Problem 2, the combination of the `astar_search` algorithm and the `h_ignore_preconditions` heuristic yielded optimal results for Problem 3. While expanding more nodes than others, it reached the goal with an ideal plan length (Figure 1.3) in the shortest time.

```
Plan length: 12  Time elapsed in seconds: 17.374589885992464
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C3, P1, JFK)
Unload(C4, P2, SFO)
```

Figure 1.3: An optimal sequence of actions for Problem 3.

Comparison of Breadth First Search, Depth First Search and Uniform Cost Search

The `depth_first_graph_search` (DFS), `breadth_first_search` (BFS) and `uniform_cost_search` (UCS) algorithms successfully found solutions in all three problems. Of the three algorithms, DFS found solutions fastest in all three problems. A shown in Figure 1.0, DFS found a solution up to 10 times faster than UCS and five times faster than BFS.

What DFS made up for in speed, it lacked in overall plan length. DFS returned plan lengths far greater than ideal in all three problems. BFS and UCS both reached the goal with optimal plan lengths in all three problems.

Although DFS returned a less than ideal plan length, it expanded far less nodes than BFS and UCS in all problems. UCS explored the highest number of nodes in all problems.

Out of these three non-heuristic algorithms, UCS would be the ideal pick for each problem. It reaches the goal state faster than BFS and has an optimal plan length when compared to DFS.

A* Search Algorithm with Preconditions and Level-Sum Heuristics versus Non-Heuristic Algorithms

The `astar_search` algorithm with the ignore preconditions and level-sum heuristics both successfully found the goal state for each of the problems.

Figure 1.0 shows the `astar_search` algorithm with the ignore preconditions heuristic reached the goal state faster than the level-sum heuristic in all three problems.

The `astar_search` algorithm with the level-sum heuristic required far less expansions and new nodes than the preconditions heuristic.

Both heuristics achieved an ideal plan length.

Figure 1.0 shows the `astar_search` algorithm with the preconditions heuristic as being the optimal plan for two of the three problems, thus making it the best heuristic for the

problem space. The `greedy_best_first_search` algorithm with the h1 heuristic (not a true heuristic) achieved the best results in Problem 1. However as the problem space grew, `astar_search` with the preconditions heuristic was the clear winner.

Non-heuristic search planning algorithms all achieved great results in Problem 1 but heuristic planning methods seemed to be more optimal for larger problem spaces.

In the case of the `astar_search` algorithm with the preconditions heuristic, as stated in chapter 10.2.3 of *Artificial Intelligence: A Modern Approach* (Norvig et al., 2010), it drops all preconditions from actions.

By dropping the preconditions of each action, it means every action becomes applicable in every state. This information allowed faster searching of the problem space because the number of steps required to solve the relaxed problem is almost the number of unsatisfied goals (Norvig et al., 2010), with adjustments required for actions which achieve multiple goals and those actions which may undo effects of others.