Heuristic Analysis

The below table depicts the various metrics analyzed for different Problems using various Search Techniques.

Problem Name	Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Air Cargo Problem 1	BFS	43	56	180	6	0.04897
	BFS_Tree	1458	1459	5960	6	1.32432
	DFS	12	13	48	12	0.01252
	DFS_Limited	101	271	414	50	0.1287
	UCS	55	57	224	6	0.05753
	Recursive_Best (h_1)	4229	4230	17029	6	3.86346
	Greedy_Best(h_1)	7	9	28	6	0.00787
	A* (h_1)	55	57	224	6	0.05715
	A* (h_ignore_prec)	41	43	170	6	0.06389
	A* (h_pg_levelsum)	11	13	50	6	2.47929
Air Cargo Problem 2	BFS	3346	4612	30534	9	17.9555
	BFS_Tree	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
	DFS	107	108	959	105	0.42442
	DFS_Limited	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
	UCS	4852	4854	44030	9	62.8478
	Recursive_Best (h_1)	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
	Greedy_Best(h_1)	990	992	8910	9	9.6257
	A* (h_1)	4852	4854	44030	9	64.337
	A* (h_ignore_prec)	1506	1508	13820	9	41.1517
	A* (h_pg_levelsum)	86	88	841	9	255.975
Air Cargo Problem 3	BFS	141120	17673	12496	12	128.421

	BFS_Tree	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
	DFS	677	678	5608	660	4.64732
	DFS_Limited	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
	UCS	18235	18237	159716	12	486.412
	Recursive_Best (h_1)	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
	Greedy_Best(h_1)	5615	5617	49438	22	133.118
	A* (h_1)	18235	18237	159716	12	489.375
	A* (h_ignore_prec)	5118	5120	45650	12	109.164
	A* (h_pg_levelsum)	414	416	3818	12	1409.88

Now to report on Detailed analysis of the following searches for all the problems:

1. **Air Cargo Problem 1**

- Solving Air Cargo Problem 1 using breadth_first_search...

```
Expansions   Goal Tests   New Nodes
    43          56          180

Plan length: 6  Time elapsed in seconds: 0.058141184
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

- Solving Air Cargo Problem 1 using depth_first_graph_search...

```
Expansions   Goal Tests   New Nodes
    12          13          48

Plan length: 12  Time elapsed in seconds: 0.020788223999999994
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
```

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

- Solving Air Cargo Problem 1 using uniform_cost_search...

```
Expansions   Goal Tests   New Nodes
   55          57         224

Plan length: 6  Time elapsed in seconds: 0.08141721599999999
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

- Solving Air Cargo Problem 1 using greedy_best_first_graph_search with h_1...

```
Expansions   Goal Tests   New Nodes
   7           9          28

Plan length: 6  Time elapsed in seconds: 0.009100287999999984
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

- Solving Air Cargo Problem 1 using astar_search with h_1...

```
Expansions   Goal Tests   New Nodes
```

```
           55          57          224

        Plan length: 6  Time elapsed in seconds: 0.09375129600000001
        Load(C1, P1, SFO)
        Load(C2, P2, JFK)
        Fly(P1, SFO, JFK)
        Fly(P2, JFK, SFO)
        Unload(C1, P1, JFK)
        Unload(C2, P2, SFO)
```

2. Air Cargo Problem 2

```
    Solving Air Cargo Problem 2 using breadth_first_search...

        Expansions   Goal Tests   New Nodes
          3346        4612       30534

        Plan length: 9  Time elapsed in seconds: 32.901156352
        Load(C1, P1, SFO)
        Load(C2, P2, JFK)
        Load(C3, P3, ATL)
        Fly(P1, SFO, JFK)
        Unload(C1, P1, JFK)
        Fly(P2, JFK, SFO)
        Unload(C2, P2, SFO)
        Fly(P3, ATL, SFO)
        Unload(C3, P3, SFO)


    Solving Air Cargo Problem 2 using depth_first_graph_search...

        Expansions   Goal Tests   New Nodes
          107         108         959

        Plan length: 105  Time elapsed in seconds: 1.3187307520000005
        Fly(P3, ATL, JFK)
        Fly(P2, JFK, ATL)
        Fly(P3, JFK, SFO)
        Fly(P2, ATL, SFO)
        Fly(P1, SFO, ATL)
        Fly(P3, SFO, ATL)
        Fly(P1, ATL, JFK)
        Fly(P3, ATL, JFK)
```

```
Load(C2, P3, JFK)
Fly(P3, JFK, ATL)
Fly(P1, JFK, ATL)
Fly(P3, ATL, SFO)
Fly(P1, ATL, SFO)
Fly(P2, SFO, ATL)
Fly(P3, SFO, ATL)
Fly(P2, ATL, JFK)
Unload(C2, P3, ATL)
Fly(P3, ATL, JFK)
Fly(P2, JFK, ATL)
Fly(P3, JFK, SFO)
Fly(P2, ATL, SFO)
Fly(P1, SFO, ATL)
Fly(P3, SFO, JFK)
Fly(P1, ATL, JFK)
Load(C1, P2, SFO)
Fly(P3, JFK, ATL)
Fly(P1, JFK, ATL)
Fly(P2, SFO, ATL)
Fly(P3, ATL, JFK)
Fly(P2, ATL, JFK)
Fly(P1, ATL, SFO)
Fly(P3, JFK, ATL)
Unload(C1, P2, JFK)
Fly(P3, ATL, SFO)
Fly(P2, JFK, ATL)
Fly(P1, SFO, ATL)
Fly(P2, ATL, SFO)
Fly(P1, ATL, JFK)
Fly(P3, SFO, ATL)
Fly(P2, SFO, ATL)
Fly(P3, ATL, JFK)
Fly(P2, ATL, JFK)
Load(C1, P3, JFK)
Fly(P3, JFK, ATL)
Fly(P2, JFK, ATL)
Fly(P3, ATL, SFO)
Fly(P1, JFK, ATL)
Fly(P2, ATL, JFK)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Fly(P2, JFK, ATL)
Fly(P1, JFK, SFO)
Fly(P2, ATL, SFO)
Fly(P3, SFO, ATL)
Fly(P1, SFO, ATL)
Fly(P3, ATL, JFK)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Fly(P3, JFK, ATL)
Fly(P1, JFK, ATL)
Fly(P3, ATL, SFO)
Fly(P1, ATL, SFO)
Fly(P2, SFO, ATL)
Fly(P3, SFO, ATL)
Fly(P2, ATL, JFK)
Fly(P3, ATL, JFK)
Fly(P1, SFO, ATL)
Fly(P2, JFK, ATL)
Unload(C1, P3, JFK)
Fly(P1, ATL, JFK)
Fly(P2, ATL, JFK)
Fly(P3, JFK, ATL)
Fly(P1, JFK, ATL)
Fly(P3, ATL, SFO)
Fly(P1, ATL, SFO)
Fly(P2, JFK, ATL)
Fly(P3, SFO, ATL)
Fly(P2, ATL, SFO)
Fly(P3, ATL, JFK)
Load(C3, P3, JFK)
Fly(P3, JFK, ATL)
Fly(P2, SFO, ATL)
Fly(P3, ATL, SFO)
Fly(P2, ATL, JFK)
Fly(P1, SFO, ATL)
Fly(P2, JFK, SFO)
Fly(P1, ATL, JFK)
Unload(C3, P3, SFO)
Fly(P1, JFK, SFO)
Fly(P3, SFO, ATL)
Fly(P2, SFO, ATL)
Fly(P3, ATL, JFK)
Fly(P2, ATL, JFK)
Fly(P1, SFO, ATL)
```

```
        Fly(P3, JFK, ATL)
        Fly(P1, ATL, JFK)
        Load(C2, P3, ATL)
        Fly(P3, ATL, JFK)
        Fly(P2, JFK, ATL)
        Fly(P1, JFK, ATL)
        Fly(P2, ATL, SFO)
        Fly(P1, ATL, SFO)
        Fly(P3, JFK, SFO)
        Fly(P2, SFO, ATL)
        Unload(C2, P3, SFO)


Solving Air Cargo Problem 2 using uniform_cost_search...

        Expansions   Goal Tests   New Nodes
          4852        4854       44030

        Plan length: 9  Time elapsed in seconds: 86.751796224
        Load(C1, P1, SFO)
        Load(C2, P2, JFK)
        Load(C3, P3, ATL)
        Fly(P1, SFO, JFK)
        Fly(P2, JFK, SFO)
        Fly(P3, ATL, SFO)
        Unload(C3, P3, SFO)
        Unload(C2, P2, SFO)
        Unload(C1, P1, JFK)


Solving Air Cargo Problem 2 using greedy_best_first_graph_search with h_1...

        Expansions   Goal Tests   New Nodes
          990         992         8910

        Plan length: 9  Time elapsed in seconds: 15.01218304000001
        Load(C1, P1, SFO)
        Load(C2, P2, JFK)
        Load(C3, P3, ATL)
        Fly(P1, SFO, JFK)
        Fly(P2, JFK, SFO)
        Fly(P3, ATL, SFO)
        Unload(C3, P3, SFO)
```

```
    Unload(C2, P2, SFO)
    Unload(C1, P1, JFK)


Solving Air Cargo Problem 2 using astar_search with h_1...

    Expansions   Goal Tests   New Nodes
      4852        4854       44030

    Plan length: 9  Time elapsed in seconds: 82.82224384
    Load(C1, P1, SFO)
    Load(C2, P2, JFK)
    Load(C3, P3, ATL)
    Fly(P1, SFO, JFK)
    Fly(P2, JFK, SFO)
    Fly(P3, ATL, SFO)
    Unload(C3, P3, SFO)
    Unload(C2, P2, SFO)
    Unload(C1, P1, JFK)
```

Air Cargo Problem 3

Skipped to avoid large length of the report.

The Optimal Plans for the Different Problems are :

1. **Air Cargo Problem 1:**

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

2. **Air Cargo Problem 2 :**

   Load(C1, P1, SFO)
   Load(C2, P2, JFK)
   Load(C3, P3, ATL)
   Fly(P1, SFO, JFK)

Fly(P2, JFK, ATL)
Fly(P3, ATL, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, ATL)
Unload(C3, P3, SFO)

3. **Air Cargo Problem 3 :**

Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

Non-Heuristic Search Methods Compare

The three search methods that I will be comparing are BFS, DFS & UCS.

- BFS: This search technique focuses on minimizing the path length while compromising with number of node expansions & time to calculate the best path to the goal state
- DFS: This search technique focuses on minimizing the time elapsed and node expansions while compromising with the path length, since it provides very large path lengths -- due to it’s nature to explore down/deeper along one path/branch rather than going level wise like BFS
- UCS: This search focuses on minimizing the number of expansions. Since using the cost function it tries to minimize the path length/cost which further minimizes the number of node expansions -- unlike BFS which will always expand all nodes on a level even if they have the high cost/path length

A* Search Heuristic Compare

To Compare the Heuristics we try to compare the Time and Metrics recorded for different Air Cargo problems with respect to A* search Technique.

1. Heuristic 1 -- Ignore Preconditions : -

   This Heuristic is one of the few basic heuristic possible in this question. It just tries to record the number of goal states left to be achieved. So it never Over estimates since each action can be completed in minimum 1 number of moves. Assuming this it just returns the count of such number of states yet to be achieved. Although providing an idea about a problem it might not be the best heuristic since it might generally under-estimate the number of actions required to complete a state of goal. Due to this , the number of nodes generally expanded are larger. However since being a simple Heuristic to calculate so the time required to calculate is way less.

2. Heuristic 2 -- level_sum : -

   This Heuristic is not a simple one. It tries to give the best idea of the number of actions required to achieve a particular goal in the list of possible goals. However as being a kind of accurate Heuristic it takes longer time to calculate the value of each node. As a result the number of nodes expanded and explored are quite less.

   It is evident that Heuristic 2 is a harder heuristic to calculate so the search takes longer time , however it gives a more accurate picture of the problem so the search nodes that are expanded are quite less.

Best Heuristic

After referring the table in Page 1, the best heuristic was level_sum since the number of nodes it expanded was quite less , this actually helps in storage methods , since we won’t be needing to store the intermediate states.
However other Heuristics & Non - Heuristic methods have exponentially larger space requirement. Although the time it takes is the largest but the increase in space for other methods is much larger than the level_sum’s increase in time.
In some cases specially easy/small ones the Heuristic & A* searches will take longer and simple searches like Greedy, BFS etc will be way faster and usually also provide the optimal solution.
However in much more complex cases the Heuristic searches work better since they avoid horizon effect.