ANALYSIS OF PERFORMANCE BETWEEN NON-HEURISTIC SEARCHES AND HEURISTIC SEARCHES

I. Optimal Plan for Each Cargo Problem

a) Air Cargo Problem 1:

Methods: A Star search with Ignore Preconditions heuristic

Summary:

Plan Length	Duration (s)	Expansion	New Nodes	Goal Test
6	0.03	41	173	43

Steps:

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

Note:

A Star search with planning graph and level sum heuristic has better performance considering memory-space wise, but running time is 0.6 which is 20 times slower.

b) Air Cargo Problem 2:

Methods: A Star search with Planning Graph Level Sum Heuristic

Summary:

Plan Length	Duration (s)	Expansion	New Nodes	Goal Test
9	55	86	841	88

Steps:

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

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

Note:

Although A Star search with planning graph using level sum heuristic suffers from long runtime duration (6 times slower than Ignore Precondition Heuristic), it makes better use of memory space with 15 times less number of new nodes created.

c) Air Cargo Problem 3:

Methods: A Star search with Ignore Preconditions heuristic

Summary:

Plan Length	Duration (s)	Expansion	New Nodes	Goal Test
12	35	5040	44944	5042

Steps:

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

Note:

Similar as above, this time we select the A Star Search with ignore precondition heuristic based on execution speed (35 seconds) comparing with the A Star Search Planning Graph with

level sum heuristic takes almost 5 minutes to complete. However, A Star Search Planning Graph manage the memory space much better with 7 times less new nodes created. The choice for the optimal solution can be considered as a trade off between speed and memory space.

II. Compare and contrast Non-Heuristic Search

Below tables contain key elements for comparing Non-Heuristic search:

Figure1: Air Cargo Problem 1

	Plan Length	**Running Time (s)**	**Expansions**	**Goal Tests**	**New Nodes**
Breadth-First Search	6	0.04	43	56	180
Depth First Search	20	0.01	21	22	84
Uniform Cost Search	6	0.05	55	57	224

Figure 2 : Air Cargo Problem 2

	Plan Length	**Running Time (s)**	**Expansions**	**Goal Tests**	**New Nodes**
Breadth-First Search	9	24.38	3346	4612	30534
Depth First Search	105	0.66	107	108	959
Uniform Cost Search	9	27.03	4853	4855	44041

Figure 3 : Air Cargo Problem 3

	Plan Length	Running Time (s)	Expansions	Goal Tests	New Nodes
Breadth-First Search	12	165.63	14120	17673	124926
Depth First Search	288	2.17	292	293	2388
Uniform Cost Search	12	129.30	18221	18223	159599

Based on the data from the above three tables:

- Depth-first search (DFS) cannot find the optimal solution although the running time is faster than the remaining two search algorithms in all cases.
- DFS memory-space complexity also increase as the complexity of the problem increase but not as much as the remaining algorithms.
- Both breadth-first search (BFS) and Uniform Cost Search (UCS) can find the optimal solution.
- The running time for BFS and UCS increase dramatically as the complexity of the problem increase. Also, memory-space complexity also take a huge jump for BFS and UCS as the complexity of the problem increase.
- BFS is slightly faster than UCS in the Air Cargo Problem 2 (about 3 seconds) UCS has to create more new nodes and do more operations like expansion and goal tests.
- It is really interesting that UCS is faster than BFS (about 24 seconds) in the Air Cargo Problem 3 (most complicated problem) although it has to create more new nodes 159599 in comparing with 124926 as well as making more expansion operations and goal tests.

III. Compare and contrast Heuristic Search

Below tables contain key elements for comparing Non-Heuristic search:

Figure 4 : Air Cargo Problem 1

	Plan Length	Running Time	Expansions	Goal Tests	New Nodes
A Star Ignore Precondition Heuristic	6	0.038	41	43	170
A Star Planning Graph Level Sum Heuristic	6	0.62	11	13	50

Figure 5 : Air Cargo Problem 2

	Plan Length	Running Time	Expansions	Goal Tests	New Nodes
A Star Ignore Precondition Heuristic	9	8.21	1450	1452	13303
A Star Planning Graph Level Sum Heuristic	9	55.43	86	88	841

Figure 6 : Air Cargo Problem 3

	Plan Length	Running Time	Expansions	Goal Tests	New Nodes
A Star Ignore Precondition Heuristic	12	35.17	5040	5042	44944
A Star Planning Graph Level Sum Heuristic	12	278.62	315	317	2902

Based on the data from the above three tables:

- Both A Star Ignore Preconditions Heuristic (AIP) and A Star Planning Graph with Level Sum Heuristic (APG) can find the optimal solutions for all of the problems.

- It is really interesting to see that AIP is approximately 7 times faster than APG in Air Cargo Problem 2 and Air Cargo Problem 3. As AIP heuristic relaxed the preconditions of the problem, it is understandable to see that AIP create approximately 15 times more nodes as well as doing approximately 16 times more expansions and goal test in both cases in compare with APG.
- We can see that APG is very memory-space efficient with number of nodes created, number of expansions and goal tests operations to come up with the optimal solution. However, it spends a lot of time building the Planning Graph data structure and relates all the possible internal states with various kinds of mutexes and preconditions that make the running time suffer quite dramatically.

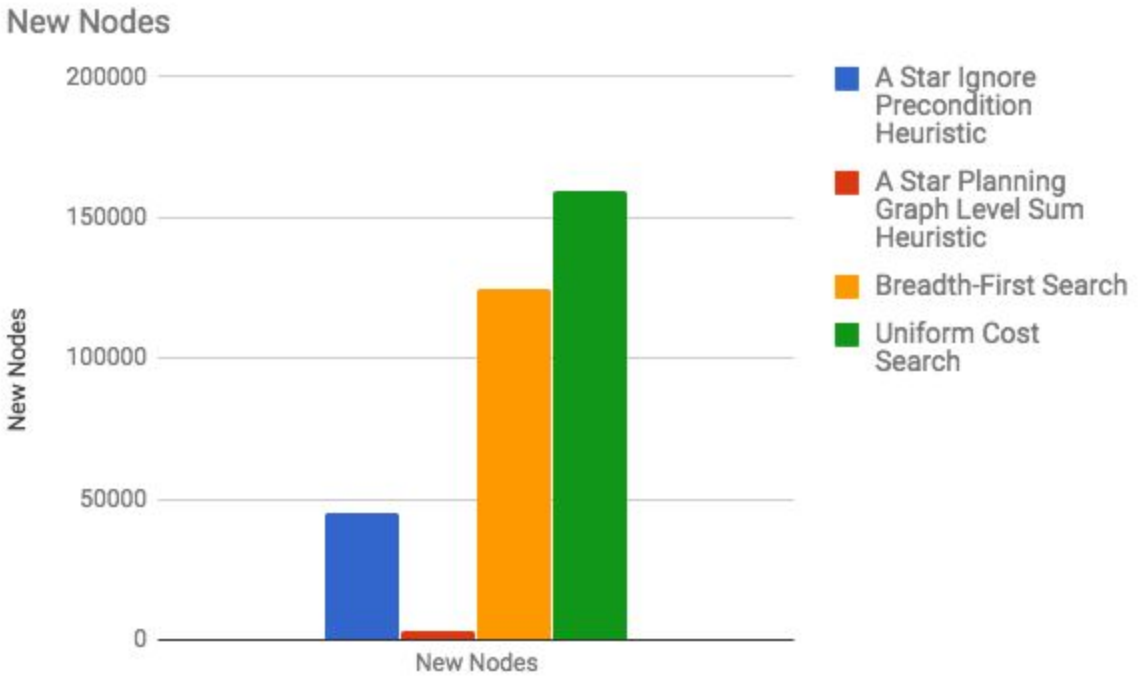
IV. Compare and contrast Non-Heuristic and Heuristic Search

Below table and graph show the statistic number among Heuristic and Non-heuristic Search. DFS performs poorly in all problems and does not find the optimal solution so we did not include it in this comparison. Also, We only compare them with respect to Air Cargo Problem 3:

Figure 7 : Air Cargo Problem 3 - Non-Heuristic vs Heuristic Search

	Plan Length	**Running Time**	**Expansions**	**Goal Tests**	**New Nodes**
A Star Ignore Precondition Heuristic	12	35.17	5040	5042	44944
A Star Planning Graph Level Sum Heuristic	12	278.62	315	317	2902
Breadth-First Search	12	165.63	14120	17673	124926
Uniform Cost Search	12	129.30	18221	18223	159599

Figure 8 : Compare Number of New Nodes Created Among Search Algorithms



For space complexity comparison, we can see from **Figure 8** and **Figure 7** that heuristic searches outperform non-heuristic searches. As we learn that BFS will always expand the next shallowest node in the graph without any direction guiding to the goal. Similarly, UCS also keeps reaching for the smallest cost path without any guidance that if that node would ever lead to the solution. On the other hand, with heuristics guidance, both AIP and AGP are able to find optimal solution with much better space complexity as they expand their operations toward the direction of potential solution.

For running time comparison, as we can see from **Figure 9,** while AIP outperforms all of the non-heuristic searches, AGP's runtime suffers from building and maintaining its internal planning graph data structure. UCS and BFS just basically keep doing forward search until they reach the goal but their operation is much more simpler than AGP. Also, we can see the advantage of having a good admissible heuristic like in the AIP case. Instead of solving big complicated problem, we relaxed all preconditions and combine all of the actions that has effects that satisfied the goal. This technique proves to help running time also.

Figure 9 : Compare Running Times Among Search Algorithms