Heuristic Analysis for AIND-Planning Project

Air Cargo Problem 1

• Breadth First Search
- Expansions: 43
- Goal Tests: 56
- New Nodes: 180
- Plan Length: 6
- Time (seconds): 0.0411

• Depth First Graph Search
- Expansions: 21
- Goal Tests: 22
- New Nodes: 84
- Plan Length: 20
- Time (seconds): 0.01838

• Uniform Cost Search
- Expansions: 55
- Goal Tests: 57
- New Nodes: 224
- Plan Length: 6
- Time (seconds): 0.0558

• Recursive Best First Search with H1 Heuristic
- Expansions: 4229
- Goal Tests: 4230
- New Nodes: 17023
- Plan Length: 6
- Time (seconds): 3.9187

• A* Search with H1 Heuristic
- Expansions: 55
- Goal Tests: 57
- New Nodes: 224
- Plan Length: 6
- Time (seconds): 0.05799

• A* Search with Ignore Preconditions Heuristic
- Expansions: 41
- Goal Tests: 43
- New Nodes: 170
- Plan Length: 6
- Time (seconds): 0.0620

• A* Search with Planning Graph Level Sum Heuristic
- Expansions: 55
- Goal Tests: 57
- New Nodes: 224
- Plan Length: 6
- Time (seconds): 1.5456

For Problem 1 the ideal solution consisted of 6 steps that reached the goal. A number of search techniques (both uninformed and informed) were able to find the solution in 6 steps. Only Depth First Graph Search took 20 steps to reach the goal. The most optimal technique to solve this problem was Breadth First Search as it took the least amount of expansions, goal tests and time to reach the goal in 6 steps. Strictly amongst the informed heuristic search techniques A* with

Ignore Preconditions heuristic solved the problem in the most efficient way possible. But overall, for simple problems like these with straightforward and small number of goals Breadth First Search is more than adequate for reaching the solution.

Optimal Sequence of Actions for Problem 1:

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

Air Cargo Problem 2

• Breadth First Search
- Expansions: 3343
- Goal Tests: 4609
- New Nodes: 30509
- Plan Length: 9
- Time (seconds): 20.858

• Depth First Graph Search
- Expansions: 624
- Goal Tests: 625
- New Nodes: 5602
- Plan Length: 619
- Time (seconds): 5.385

• Uniform Cost Search
- Expansions: 4835
- Goal Tests: 4837
- New Nodes: 43877
- Plan Length: 9
- Time (seconds): 18.54535

• Recursive Best First Search with H1 Heuristic
This path finding technique was not able to find the Goal in 10 mins for Problem 2 so I terminated the search

• A* Search with H1 Heuristic
- Expansions: 4835
- Goal Tests: 4837
- New Nodes: 43877
- Plan Length: 9
- Time (seconds): 17.2104

• A* Search with Ignore Preconditions Heuristic
- Expansions: 1450
- Goal Tests: 1452
- New Nodes: 13303
- Plan Length: 9
- Time (seconds): 6.09234

• A* Search with Planning Graph Level Sum Heuristic
- Expansions: 4835
- Goal Tests: 4837
- New Nodes: 43877
- Plan Length: 9
- Time (seconds): 696.701

All the search techniques both informed and un-informed took 9 steps to reach the goal except Depth First Graph Search. Depth First again proves to be quite inefficient in solving the problem optimally taking 619 steps to reach the goal but of course its faster than any other search technique. A* informed searches with heuristics also solved the problem in 9 steps. In my opinion, the most optimal search technique was A* Search with Ignore Preconditions Heuristic which solved the problem in the least amount of time (other than Depth First Graph Search) with the least expansion, goal tests and new nodes. It solved the problem in 9 steps.

Optimal Sequence of Actions for Problem 2:

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

Air Cargo Problem 3

• Breadth First Search
- Expansions: 14663
- Goal Tests: 18098
- New Nodes: 129631
- Plan Length: 12
- Time (seconds): 170.510

• Depth First Graph Search
- Expansions: 408
- Goal Tests: 409
- New Nodes: 3364
- Plan Length: 392
- Time (seconds): 3.037

• Uniform Cost Search
- Expansions: 18235
- Goal Tests: 18237
- New Nodes: 159716
- Plan Length: 12
- Time (seconds): 88.0489

• Recursive Best First Search with H1 Heuristic
This path finding technique was not able to find the Goal in 10 mins for Problem 3 so I terminated the search

• A* Search with H1 Heuristic
- Expansions: 18235
- Goal Tests: 18237
- New Nodes: 159716
- Plan Length: 12
- Time (seconds): 81.6087

• A* Search with Ignore Preconditions Heuristic
- Expansions: 5040
- Goal Tests: 5042
- New Nodes: 44944
- Plan Length: 12
- Time (seconds): 25.9162

• A* Search with Planning Graph Level Sum Heuristic
- Expansions: 18235
- Goal Tests: 18237
- New Nodes: 159716
- Plan Length: 12
- Time (seconds): 4614.246

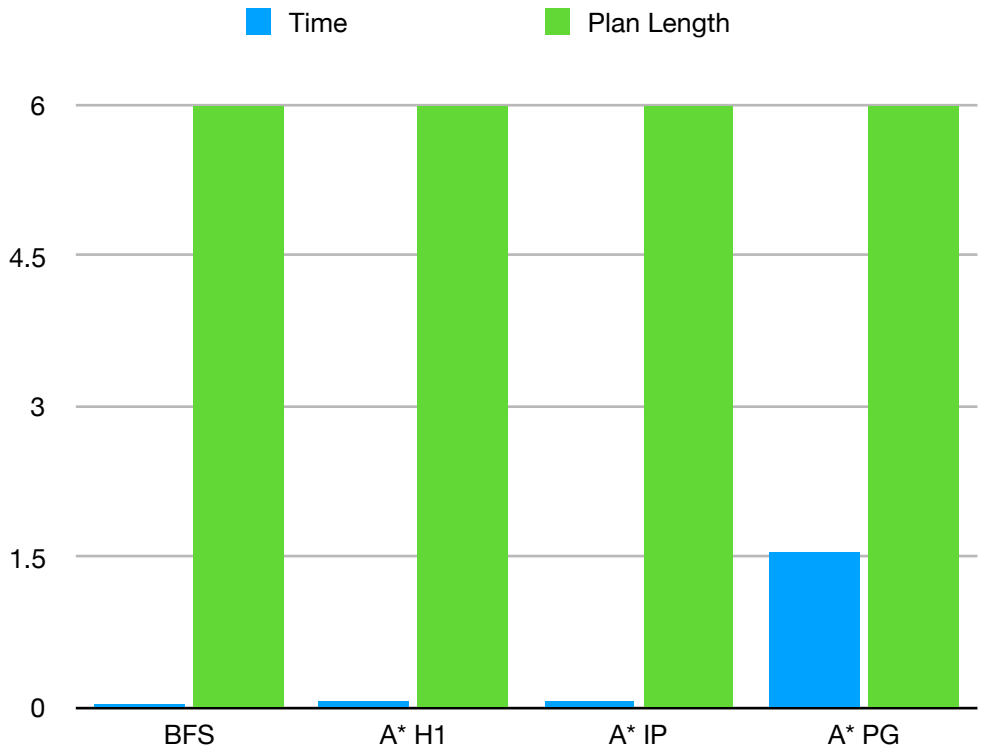
Again, other than Depth First Graph Search which took 392 steps to solve the problem, all the other search techniques took 12 steps to solve the problem. Depth First Graph Search again took the least amount of time to solve the problem which has been the general pattern throughout. A* Search with Ignore Preconditions Heuristic in my opinion is the optimal search algorithm for solving this problem as it took the the least amount of expansions, goal tests, new nodes and time (other than Depth First Graph Search) to arrive at the final goal in 12 steps.

Optimal Sequence of Actions for Problem 3:

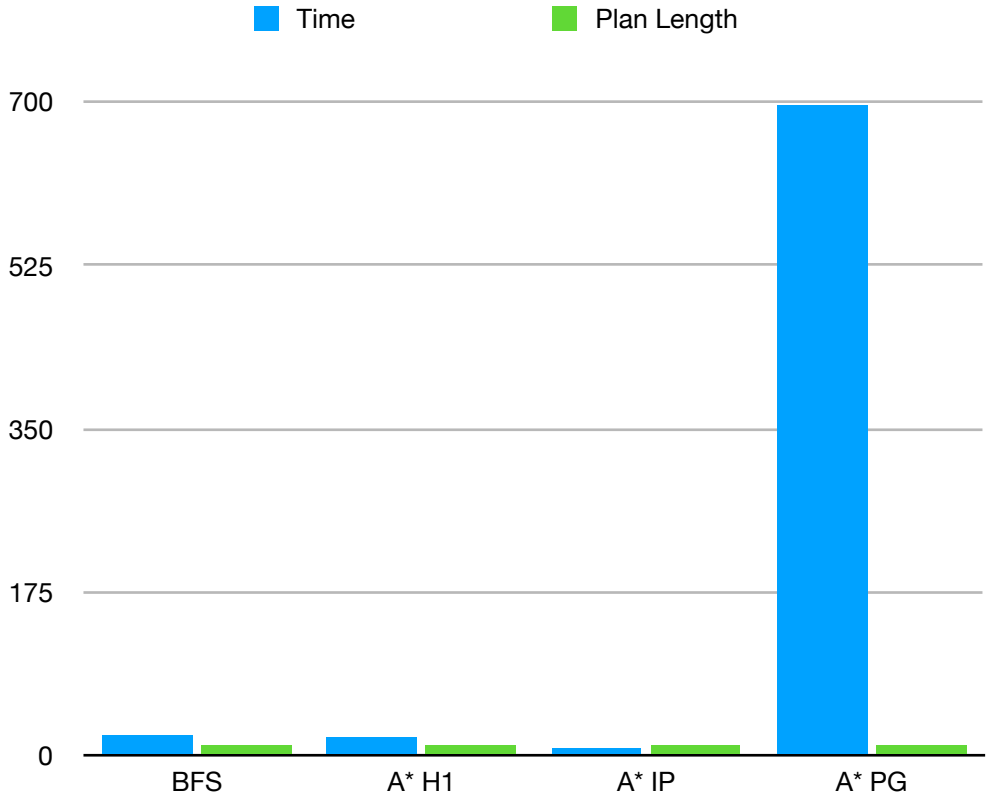
```
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

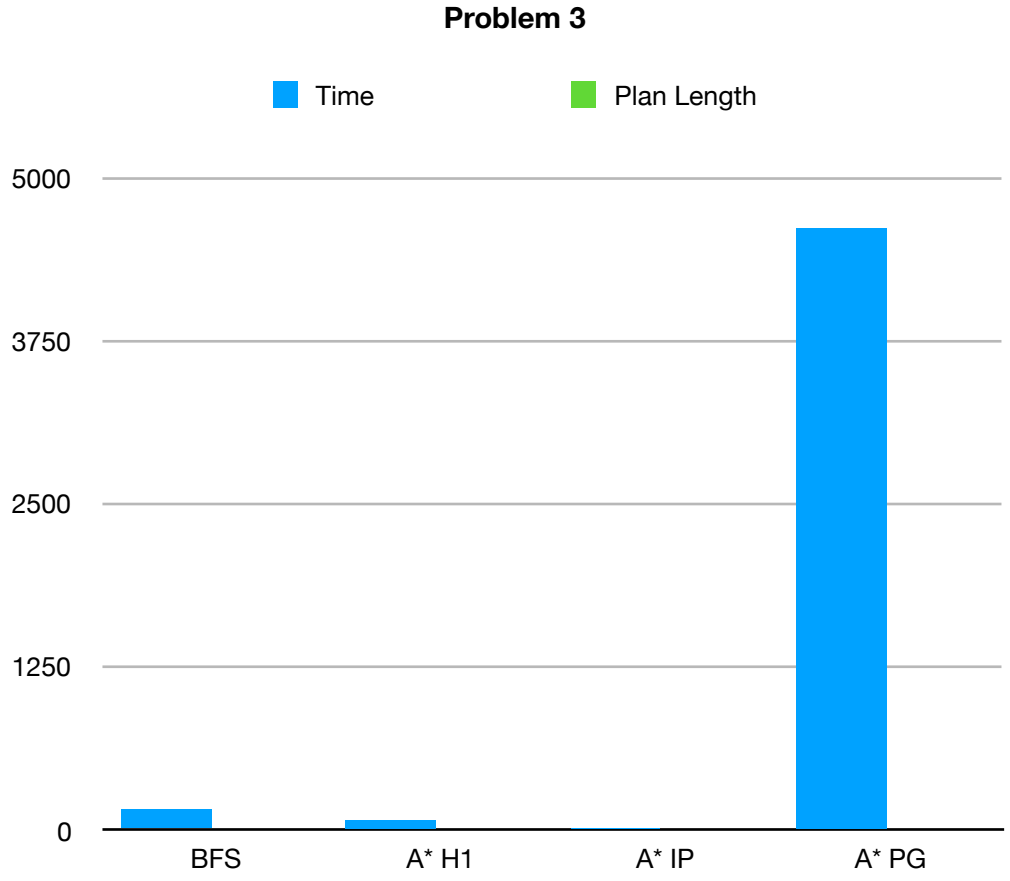
Final Thoughts regarding Planning Graph Level Sum and Ignore Preconditions:
For each and every problem both Planning Graph Level Sum Heuristic and Ignore Preconditions Heuristic in A* Search reached the goal in the same number of steps but Planning Graph was much slower than Ignore Preconditions as it had to expand more nodes and had to perform more goal tests. In my opinion, these problems are not complicated enough that we might need to use Planning Graph and Level Sum Heuristic. Ignore Preconditions does the job for finding the goal. It might be the case for more sophisticated problems that Planning Graph is necessary since we need to see whether the goal is achievable or not.

Problem 1



Problem 2





I have made comparisons between Breadth First Search A* Search with H1, Ignore Preconditions and Planning Graph Level Sum heuristic in the Bar Charts above. They show the time taken to execute the different searches and the number of steps required to reach the goal. As we know the number of expansions and goal tests is directly proportional to the total time taken, the higher the time, the higher number of expansions and goal tests is required.

References:

1. In Lesson 8 Search, video: Search Comparison 1, Peter Norvig explains that Depth First Search due to the technique it employs does not find the optimal solution. This is why I have not selected Depth First Search as the optimal strategy for any problem. https://classroom.udacity.com/nanodegrees/nd889/parts/6be67fd1-9725-4d14-b36e-ae2b5b20804c/modules/f719d723-7ee0-472c-80c1-663f02de94f3/lessons/36fc5b2f-6367-4808-b87c-0faa42744994/concepts/4e06a0da-40b1-48a4-a4d3-a95ceebfb468
2. In video: Quiz: Search Comparison 2, Peter Norvig explains how the Depth First Search makes significant memory savings. https://classroom.udacity.com/nanodegrees/nd889/parts/6be67fd1-9725-4d14-b36e-ae2b5b20804c/modules/f719d723-7ee0-472c-80c1-663f02de94f3/lessons/36fc5b2f-6367-4808-b87c-0faa42744994/concepts/012b8f6c-3403-472a-a318-ba1ab3a2875f
3. “Backtracking search facilitates yet another memory-saving (and time-saving) trick: the idea of generating a successor by *modifying* the current state description directly rather than copying it first.” (AIMA, 3rd Edition, Stuart Russell and Peter Norvig)
4. On the use of **ignore preconditions heuristic** and justifying its optimality “A heuristic function h(s) estimates the distance from a state s to the goal and that if we can derive an admissible heuristic for this distance—one that does not overestimate—then we can use A∗ search to find optimal solutions. An admissible heuristic can be derived by defining a relaxed problem that is easier to solve. The exact cost of a solution to this easier problem then becomes the heuristic for the original problem.“ “the **ignore pre- conditions heuristic** drops all preconditions from actions. Every action becomes applicable in every state, and any single goal fluent can be achieved in one step (if there is an applicable action—if not, the problem is

impossible). This almost implies that the number of steps required to solve the relaxed problem is the number of unsatisfied goals" (AIMA, 3rd Edition, Stuart Russell and Peter Norvig)