Building a Forward Planning Agent

Artificial Intelligence Nanodegree Ebrahim Jakoet

# Results of Problems 1 & 2

The following problems and search algorithms were run for the first run.

## Problems:

1. Air Cargo Problem 1

2. Air Cargo Problem 2

## Search Algorithms:

1. breadth\_first\_search

2. depth\_first\_graph\_search

3. uniform\_cost\_search

4. greedy\_best\_first\_graph\_search h\_unmet\_goals

5. greedy\_best\_first\_graph\_search h\_pg\_levelsum

6. greedy\_best\_first\_graph\_search h\_pg\_maxlevel

7. greedy\_best\_first\_graph\_search h\_pg\_setlevel

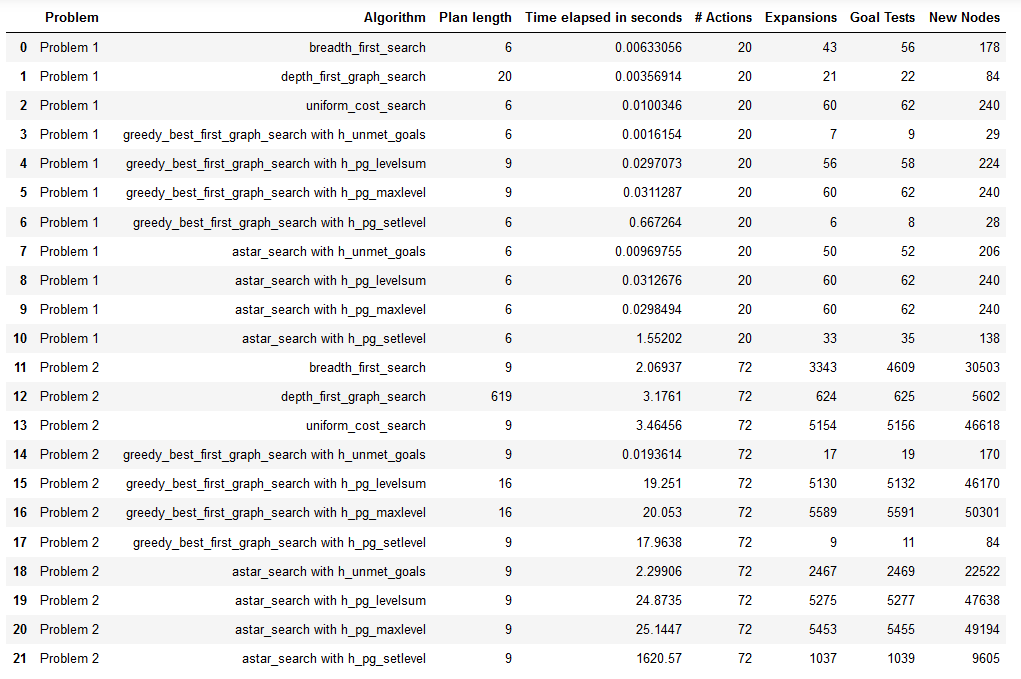
8. astar\_search h\_unmet\_goals

9. astar\_search h\_pg\_levelsum

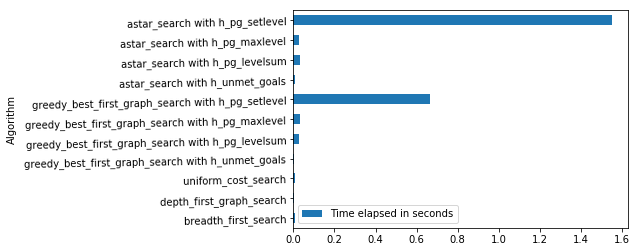
10. astar\_search h\_pg\_maxlevel

11. astar\_search h\_pg\_setlevel

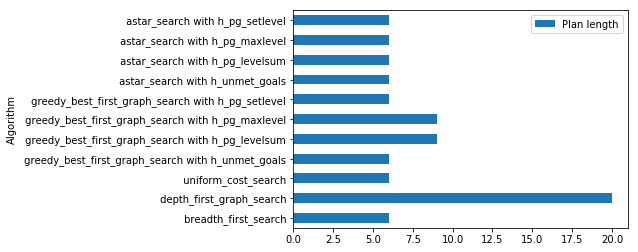
The table below shows the results of these search algorithms on Problems 1 and 2.



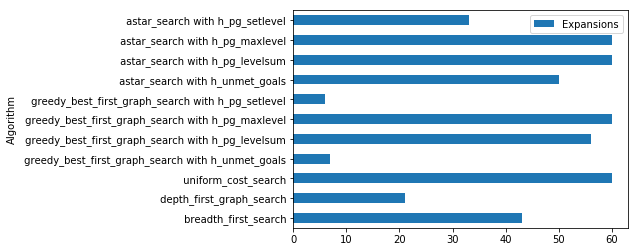
## Problem 1: Time Elapsed per Algorithm



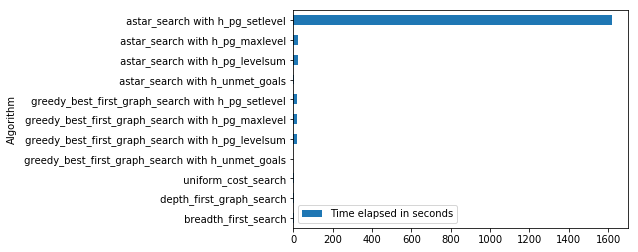
## Problem 1: Plan Length



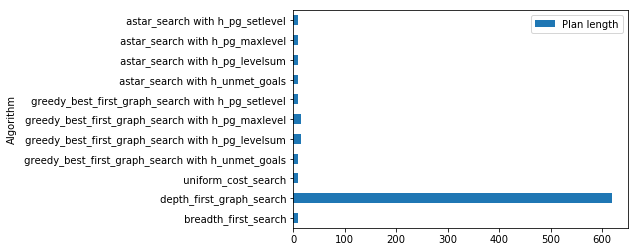
## Problem1: Number of New Node Expansions



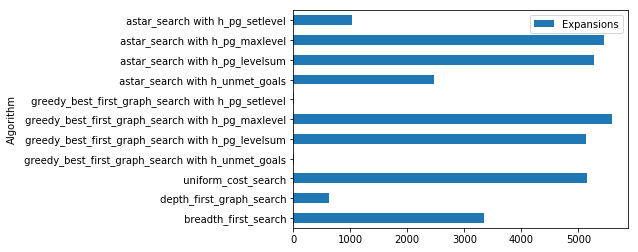
## Problem 2: Time Elapsed per Algorithm



## Problem 2: Plan Length



## Problem 2: Number of New Node Expansions



## Discussion

Based on the results from the table and charts above, we notice that the algorithm with the least amount of elapsed time is the Greedy Best First Graph Search with Unmet Goals Heuristic. This also seems to be the most efficient of the heuristic searches in general. Of the non-heuristic searches for Problem 1, Breadth First Search appears to be the most efficient since Plan Length is 6 with 43 expansions and time elapsed of 0.006s. For Problem 2 we also see the shortest Plan Length and elapsed time, but we notice that the Depth First Search performed poorly with a Plan Length of 619.

# Results of Problem 3 & 4

The following problems and search algorithms were run for the second run.

## Problems:

1. Air Cargo Problem 3

2. Air Cargo Problem 4

## Search Algorithms:

1. breadth\_first\_search

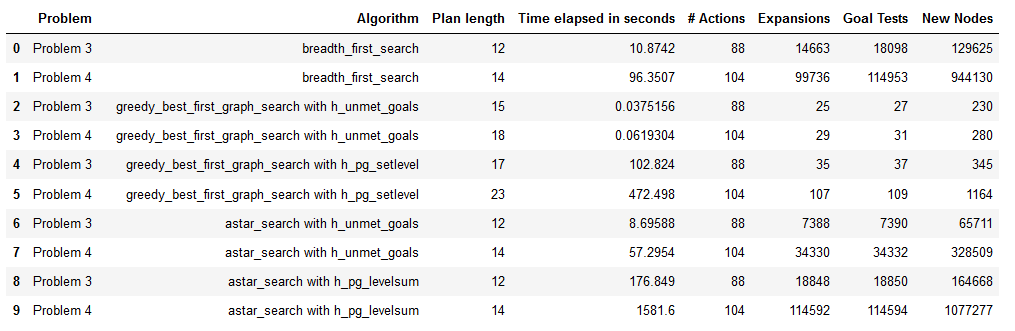
2. greedy\_best\_first\_graph\_search h\_unmet\_goals

3. greedy\_best\_first\_graph\_search h\_pg\_setlevel

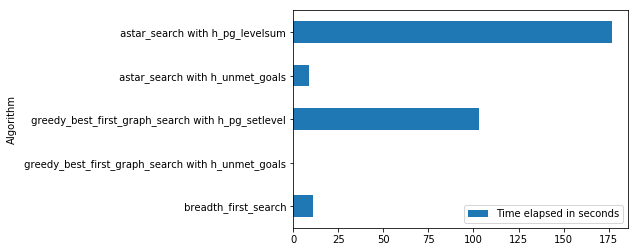
4. astar\_search h\_unmet\_goals

5. astar\_search h\_pg\_levelsum

The table below shows the results of these search algorithms on Problems 3 and 4.



## Problem 3: Time Elapsed per Algorithm



## Problem 3: Plan Length



## Problem 3: Number of New Node Expansions



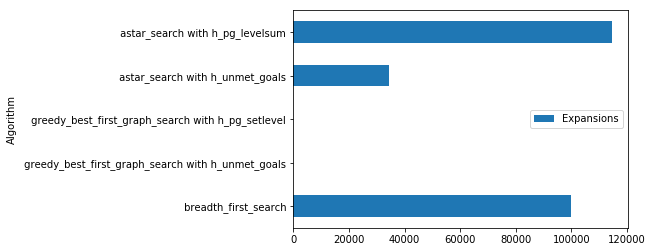
## Problem 4: Time Elapsed per Algorithm



## Problem 4: Plan Length



## Problem 4: Number of New Node Expansions



## Discussion

Based on the table and charts above for Problems 3 and 4, we notice that the Greedy Best First Graph Search with Unmet Goals are the most time efficient, but not the most optimal in terms of Plan Length. The Breadth First Search and heuristic Astar Search with Unmet Goals produced the shortest Plan Length. It’s also interesting to note that the Greedy Best First Graph Algorithms produced significantly lower New Nodes, Goal Tests and Expansions. The most optimal algorithm appears to be the Astar Search with Unmet Goals heuristic that has the lowest plan length, but longer elapsed time compared to the Greedy Best First Graph Algorithms. If time is a factor, the Greedy Best First Graph Algorithms would be a better choice of algorithm for these problems.

Algorithms that would be most appropriate for planning in a very restricted domain are the Greedy Best First Graph algorithms. This is due to the fact that they are fast and have fewer actions which are useful for real time applications. Algorithms more appropriate for planning very large domains would be the Breadth First and Astar search algorithms. We notice that the number of actions, expansions and new nodes increase with the complexity of the problem. Algorithms most suitable to find only optimal solutions are the search algorithms that are the most expensive in time elapsed, expansions and new nodes. The Astar search with level sum heuristic would likely be more suitable for searching for optimal solutions.