

# Heuristic Analysis

Nathan Findley

March 2017

## Contents

<b>1</b>	<b>Optimal Plan</b>	<b>1</b>
<b>2</b>	<b>Comparisons</b>	<b>1</b>
<b>3</b>	<b>A* Heuristics</b>	<b>2</b>
<b>4</b>	<b>Charts</b>	<b>2</b>

## Abstract

Evaluation of different search algorithms in a planning problem involving movement of cargo between airports via plane.

## 1 Optimal Plan

The order of precedence when evaluating an optimal plan is perhaps debatable but I would consider time to find the path be the most essential constraint followed perhaps by the length of the path in the evaluation of a real world problem. Take for instance the actual planning of shipments, the time taken to plan routes is quite critical when it comes to customer satisfaction, but the actual plan itself could, if unnecessarily long, result in extreme costs of shipping.

Fortunately given the data presented below, there really isn't much debate to be had about the best plan of action: Breadth First Search is a winner. Greedy Best First Graph Search H1 is a tempting choice if the time elapsed for calculation of the plan is extremely important, but given that it produces a route that is nearly twice as long a Breadth First Search it is doubtful that it would actually be useful in a shipping context.

## 2 Comparisons

Breadth First Search and Depth First Search have significant differences in results: DFS is extremely quick when it comes to finding a solution. Unfortunately the path that was found was generally an order of magnitude larger than the path found by BFS, leaving DFS as an unlikely candidate in the real world.

Breadth First Search and Uniform Cost Search appear to be about on par with one another. They are similar algorithms so this shouldn't come as a surprise. On all counts BFS is anywhere from moderately to significantly better

than UCS. While their path lengths are the same, time elapsed makes BFS a clear winner.

### 3 A\* Heuristics

- Compare and contrast heuristic search result metrics using A\* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3. - What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

A\* results contrast significantly depending on whether or not the heuristic involved is "ignore preconditions" or "level-sum". The ignore preconditions heuristic is roughly on par with BFS though it does appear to take more time. Level-Sum, on the other hand, is much more time consuming.

I did not find, unfortunately, that A\* was the best approach. Breadth First Search appears to be a much better algorithm for this type of First Order Logic path finding problem. When I say best, I am indicating that the balance of path found and time elapsed are favorable. I do have to wonder if my implementation might possibly be incorrect since, according to my understanding, A\* is considered to be the best search algorithm available. If, however, you ignore the time taken, then, quite suddenly, A\* using the level-sum heuristic beats everything else available based on the given metrics. I can say this because, while I have not included it specifically in this report, problem 3 will run at about 12.5 minutes for A\* (level-sum) and it will find the 12 step path which means that all other metrics are significantly lower than all other algorithms.

### 4 Charts

Zero value data points indicate that a 10 minute timeout was reached.





