Traffic Based Pathway Optimization

Michael LeGore
TJHSST
Alexandria, Virginia

June 2, 2009

Abstract

The goal of this project is to make an application capable of finding a path that is optimal in terms of time. This is done by keeping track of travel times at many times of day and weather and using this data to estimate the time traveled.

Keywords: Pathfinding, Traffic Simulation, Data Collection, Distributed Networks

1 Introduction

Many commercial GPS systems include navigators to use in a car to assist the driver in getting to a destination. Many drivers are not concerned about the distance traveled when getting to that destination, rather the time traveled is of primary concern. Often, the time it takes to travel a certain distance can vary wildly throughout the normal traffic cycle, reaching its peak during rush hour. Many human drivers often ignore the directions given by GPS systems during these times, opting for roads that they know personally are faster. Finding better ways to get from point A to point B can be very important for businesses looking to reduce costs, countries looking to reduce emissions, and drivers looking to better use their time and gas.

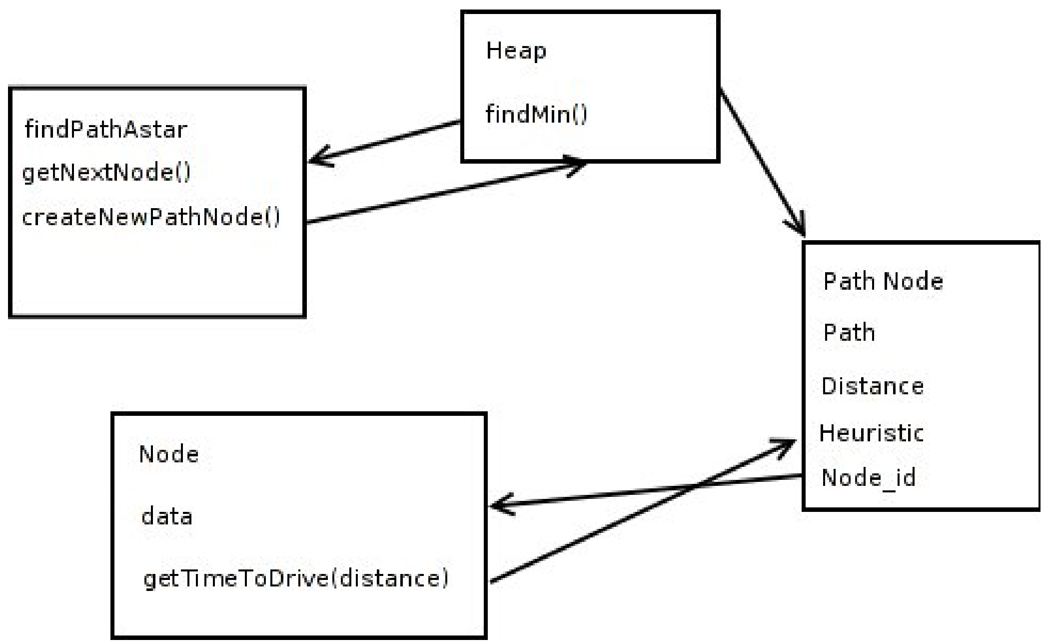

Figure 1: A* search algorithm diagram

2 Algorithm

The algorithm used in finding a path is a modified A* search. The basic algorithm starts by adding a node to a list. The node knows the distance it has traveled from the start and an estimate of the distance to the destination node. The algorithm then iterates over this list removing the node with the minimum value of heuristic and distance. At each node it checks to see whether the node is the destination, if it is, it will return the path associated with that node. Otherwise it will add a new node for each of the other nodes that are connected to it. The only difference between this algorithm and the algorithm used in my model is that instead of dealing with distance, the algorithm deals with time. In addition, it weighs weather and other contextual data in the estimation of time. It does this by querying each segment for the average speed on that segment and using that to determine the amount of time spent on the segment. The segment, when queried, searches for the data that most fits the context given in the query, it then uses this data to return the average speed of travel over that segment.

3 Optimizations

Possible optimizations include travel times based on weather, driver aggressiveness, time of day, and traffic levels. In addition the algorithm can be optimized to prevent excessive amounts of turning at intersections, and to avoid areas where there is a prevalence of car crashes. With the proper data, all of these factors could be taken into account easily by factoring them into the A* heuristic.

4 Development

Development was started by writing an algorithm to use an A* search to find a pathway. I then modified this to work with road segments instead of road nodes. This was done because in a roadway simulation, one moer often thinks about the roads that one travels on rather than the intersections one travels through. After this was accomplished, I started inputting data for the roads. I soon realized that inputting the data for hundreds of roads would be a time consuming process indeed. Because of this, a parser for Census roadway data was implemented. In addition, the A* search was modified to accomodate this different format of roadway data. I then tested the new set of data, comparing the paths generated from my algorithm to those generated by google maps, all the tests return similar or exactly the same path. At this point I had a problem, I did not have a good way of creating the traffic data necessary to find an optimal path. I had three options, look online for data collected by an agency, create the data myself, or create a simulation to run and find the data through that simulation. Initially I tried to create the simulation, but found that it was too ambitious an endevour, so I looked and found some data for Fairfax County, it was not exactly what I needed, but it could work to estimate the travel times. I developed a heuristic equation to estimate the time required to drive on the road based on an equation for the average speed on that road.

5 Traffic Simulation

My original plan was to use a traffic simulation to measure travel times for road segments. This plan had several advantages, including that the data collection would be native, in other words, since the data would be generated directly from a simulation based on the data that I am using, and therefore the datapoints could be correlated directly. In addition, the data collection would be automatic, meaning that I could leave the simulation running and it would learn better paths over time and adapt itself to the effects observed in the simulation. Also, the traffic simulation would give data that is more in line with my original intention, which was a learning algorithm that would adapt itself to data that can be improved over time. Unfortunately there were too many factors that needed to be addressed in the traffic simulation including traffic signals, one way streets, and many more, all of which were not given in the TIGER dataset. In addition the simulation size would be too great for the Computer Systems Lab to handle. So I was forced to compromise on the data collection, in exchange for adaptible data, I had to go with what would work for the final result. I chose to switch to a dataset available online through VDOT that came with its own advantages and limitations.

6 Traffic Data

Upon scrapping the Traffic Simulation, the Traffic data available through VDOT was used. This data did have some limitations however. For example, the traffic data did not include seasonal or time of day changes in traffic volume. In addition, it did not correlate directly with the TIGER data that I used for the algorithm, which meant that one could not use traffic data for the individual segments, but only for the entire road. Some of these limitations one could make up for by factoring weather and time of day into the speed estimate, but it would be a more speculative model and not based on data. The VDOT data provided information on traffic volumes on all roads in my search space for weekdays and on average. In addition it provided data on the ratio of traffic from the design hour (peak hour) to the average. I used the AADT (average annual daily traffic) to determine what type of road a road is. If it has greater that 70000 cars travelling through it each day, it is considered to be a large road and the average base speed is 75MPH. This base speed is then scaled based on several factors. If it is a weekday, it is scaled by the ratio of the AADT to the AAWDT. If it is during rush hour it is scaled by the k factor. This value is then used to determine the time.

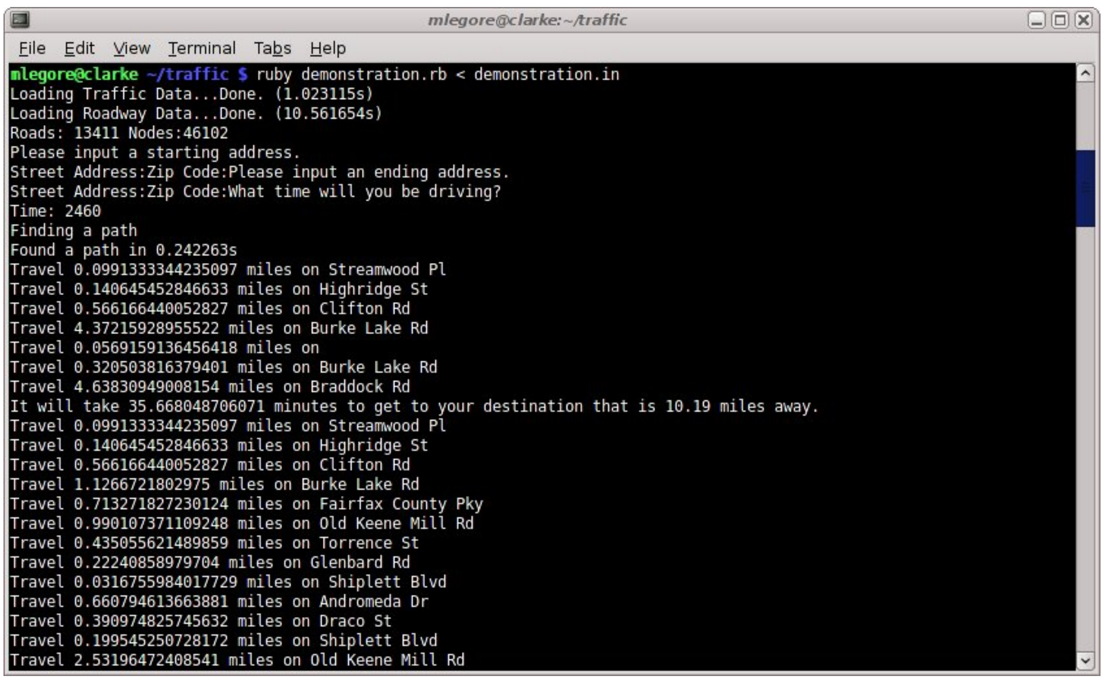

Figure 2: The final program running

7 Final Program

The final product was a program that runs and accepts an address in common language as a starting and ending point. It also accepts natural language as an input for the time of day that the path is being produced. This natural language acceptance is mainly a ruse, in that it only looks for words to identify the time, and so it can be given gibberish and it will try to find a time. It is an example of an artificial intelligence that does not make an attempt to actually be intelligent, but only to work, and to appear intelligent. The program then finds a path both using the distance heuristic and the time heuristic and compares them. When the program run is done it reports to overall results of the various tests.

8 Extensions

One option for the aquisition of the data to be fed into the algorithm is a distributed approach. One could embed a small tranciever in any modern car GPS and have that report back the travel time on specific roads at specific times. This approach would be fairly easy to implement, and if a researcher was given enough nodes in the distibuted data acusition network, it could provide a much more feasible method of collecting data very similar to attaching tags to endangered wildlife to monitor their migration patterns.

9 Applications

Possible applications of this algorithm are in pathfinding for corporations such as shipping companies where delivery times can be of vital importance to the bottom line. In addition, in a large corporation networking all of the vehicles in a fleet would be fairly simple, allowing for companies to improve their pathfinding system as the years go on. In addition embedded GPS systems could keep track of the travel times for the everyday driver by reporting their positions at several times. This could then be applied to commercial GPS devices to provide adaptible paths that learn better ways to travel based on the travel patterns of human "agents" which would provide variablility to the simulation and therefore provide the mutations.

10 Conclusion

Using traffic as a heuristic in finding paths could provide a productivity boost to any consumer that uses it. Given the proper data, such an algorithm would be able to provide a route that is energy and time saving, and a path that helps to relieve congestion in areas.

References

[1] "A* Search Algorithm", *Wikipedia.com* Accessed February 5th, 2009

[2] "A* Search", *http://www.autonlab.org/tutorials/astar.html* Accessed March 11th, 2009

[3] "M. Ebrahim Foulaadvand, Sommayeh Belbaasi, Simulation of traffic flow at a signalised intersection, *Journal of Statistical Mechanics*, September 22, 2008. http://arxiv.org/abs/0809.3591"

[4] "Traffic Simulation", *Wikipedia.com* Accessed March 19th, 2009