Implementation of a Cost Function to Model Travel Cost for Shortest Path Routing

Author: Carlos Rivera López
Advisor: Dr. Jeffrey Duffany
Computer Engineering & Computer Science

Abstract

As we travel in a region from one place to another we start wondering if there's an optimal travel path. To figure this out, normally, we would seek for details of some of the possible ways of finding such optimal path. However, during our analysis we then start to see things that are subjective such as "steep uphills followed by steep downhills" and we wonder if a path with those characteristics would indeed be the optimal path from point A to B. Through this project we attempt to create a cost function that can help us answer such question. This cost function would take terrain data such as latitude, longitude, elevation, to compute a cost based on constraints subject to the user's interest. With this information we intend to produce a node graph to model a region of interest in a map that shows the optimal path from point A to B.

Introduction

Many times, as we travel in a region we wonder what's the best, optimal path to use to get from a source to a destination. In order to figure this out we need to get some information about the region of interest. Finally, when we have the region information we now start to wonder whether it's best to take a steep uphill and coast downhill across a large distance or just go around the mountainous terrain. Through this project we look forward to model a terrain with latitude, longitude, and elevation in a 3D coordinate system by using a directed node graph, and finally use the Bellman-Ford[1] algorithm to compute a shortest path.

Background

The main reason why this topic was selected to be a project idea, mainly was because there is not much work done in this area. Usually when we search for research topics such as Encryption, Steganography, we find that there a very active academic community contributing to that area. For areas involving "Shortest path", as we attempt to find any existing work we start to see that there are little to no contributions.

In addition to this, the US Department of Energy (DOE) has also conducted some research regarding the fuel efficiency of heavy vehicles [2] based on terrain conditions such as: road grade, travel speed, vehicle weight, among others. A complimentary research to the previous research was conducted by the National Laboratory of Renewable Energy (NREL), as part of the DOE, which consisted of studying the consumption of energy of modern automobiles [3]. For that research, different types of vehicles (i.e. gasoline, electric, High Efficiency Vehicles) were put through simulation of a different series of trips with varying road grades and terrain conditions to determine how these conditions could affect the energy consumption of similar or comparable vehicles.

Problem

The main problem, and challenge, for this project was to figure out a way to model map data in a way that would allow us to compute and visualize a shortest path. In addition to this, designing the cost function to determine a cost from going from one place to another, while also determining which factors will impact the most the travel cost, proved to be a challenge.

Methodology

To solve this we then modeled locations as a 3D coordinate composed of latitude, longitude, and elevation. Since this information is, mostly, publicly available through many sources such as the United States Geological Survey (USGS) or Google we could choose a local region to create, test, and tune our cost function.

Any region that we would choose for our project could be modeled properly by using a directed graph whose vertices would contain: latitude, longitude, elevation, and neighbors; and edges would contain: source, destination, and cost. The cost function would take a source and a destination as inputs and output the computed cost of traveling through that path.

$$costUp(a,b) = i + d\left(1 + \frac{p*s}{m}\right) \qquad (1)$$

$$costDown(a,b) = i + d\left(1 - \frac{b*s}{m}\right) \qquad (2)$$

Because we look forward to make this project scalable for many applications we left some variables in the cost equations for uphill and downhill scenarios. In both equations (1) and (2) we have some terms that are being used to compute a travel cost from point *A* to point *B*, among these terms we have some variables:

- d to represent a distance cost. For this project we computed the Euclidean distance between a, and b.
- p to represent an uphill penalty (e.g. 0.70 for 70%) – used only for uphill cases.
- b to represent a downhill bonus (e.g. 0.25 for 25%) – used only for downhill cases.
- s to represent the slope between two points.
- m to indicate a maximum allowed slope.

With these variables we expect to leave some room for different scenarios such as high-slope or low-slope scenarios where a certain threshold of inclination is possible. To compute the slope between two coordinates in a 3D space we can use vector math to build a right triangle. Since we have coordinates from point *A* to point *B* we only need to compute a coordinate point *C* for the triangle base, as shown in Figure 1. At this point we can obtain the vector $\overrightarrow{AB}$ and vector $\overrightarrow{AC}$, normalize them, and obtain the angle between these two vectors by computing $\cos^{-1}(\overrightarrow{AB} \bullet \overrightarrow{AC})$. For downhill scenarios, the angle would be computed by performing the same operation but with vectors $\overrightarrow{BA}$ and $\overrightarrow{BC}$.

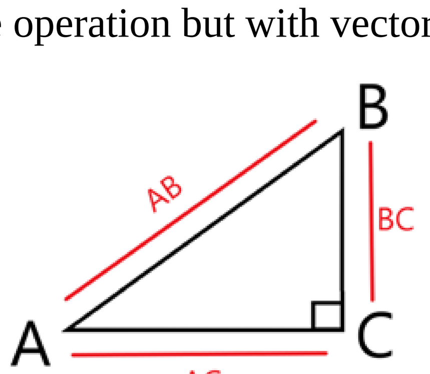

Figure 1: Right triangle made with the start and end coordinates

With the cost function now at hand we can then compute the cost of all the edges in the node graph. After all the edge weights have been computed we can use Bellman-Ford single source shortest path algorithm to determine what is the optimal path from point A to B. In order to visualize the data, along with the computed shortest path based on our cost functions from equations (1) and (2) we used a Python[4] package that could help us plot data and visualize graphs: Matplotlib[5], and NetworkX[6], respectively.

Results and Discussion

First off, we tested this cost function with different scenarios with two algorithms for single source shortest path, namely, Bellman-Ford and Dijkstra's[7] algorithms. The maps that were used to test the cost functions were similar to the map presented in Figure 2; five nodes with one of these placed at a higher elevation to the other four nodes, and six edges to connect the vertices.

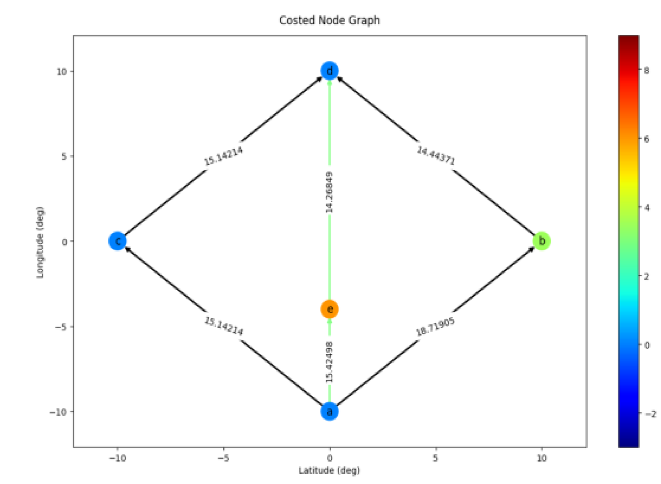

Figure 2: Node Graph with Edge Weights modeled after the Cost Function.

To output the shortest path modeled by the cost function, we then represented the node graph as a 2D plot by having the X axis represent latitudes, and the Y axis represent the longitudes in a 2D coordinate system. The elevation was represented through a Color Map. When a shortest path is computed the edges of the graph are painted in green color to highlight such path, the rest of the paths are plotted in black color; with all edges having their respective weights. This visualization can be observed in Figure 2.

All tests produced results that were consistent to the design of the cost function, and as early as in Figure 2 we can see the some of the results of modeling the edge weights with the cost function. We can see that the cost to travel from node A to node E is nearly the same as travelling from node A to node C, considering that the only difference between these two scenarios is the change in elevation; the former is an uphill scenario while the latter is a plain terrain scenario (i.e. no change in elevation).

As we tested the cost function with different maps we obtained interesting results. Therefore, another scenario that was tested introduced two nodes and three edges to the map; this new map can be observed in Figure 3. The main purpose of testing this map was to show that our cost function can favor different paths as shown in Figure 3. In this map, we can see that all of the nodes in the left side of the graph have the same elevation, allowing us to find a newer, cost-efficient (i.e. energy-efficient) path by travelling around the mountainous path, as presented in Figure 2, through a relatively flat surface.

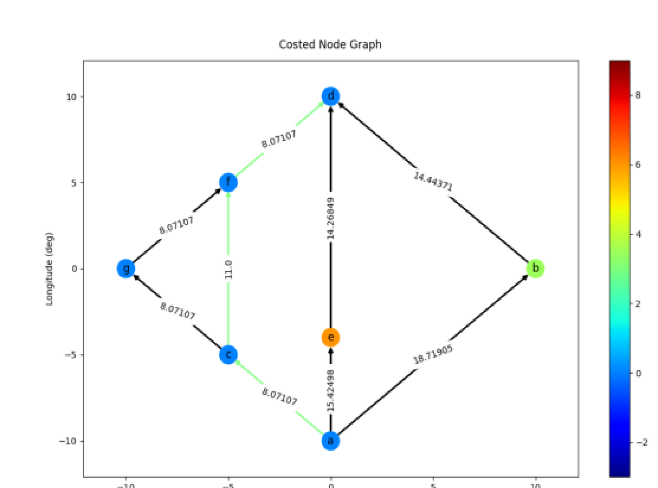

Figure 3: Shortest Path by going through a flat path.

Conclusions

The results that we obtained as part of this work seem to be really promising when thinking about creating a cost function that can accurately model our way of determining optimal travel paths. With the progress that was made as part of this project alone, we can now model optimal travel paths for simple maps such as a town map.

I believe that keeping the uphill and downhill constants as part of the cost function are key computing cost because users may have different limitations, apart from the different cases in which this cost function could be used to model.

Future Work

As part of future work, we could see this project be expanded to take into account more complex elements such as road conditions, maximum speed allowed, physics to observe how the cost could vary depending on the weight of the object that will be travelling through the node graph. Also, because the cost function is may still be "immature", applying this to real world data could still be a challenging milestone.

Another area in which this project can have additional work incorporated is when attempting to discover what are the appropriate values for uphill penalties and downhill bonuses for cases in which the user: rides a bicycle, motorcycle, or an electric vehicle. In the case of electric vehicles, downhills could very well yield significant downhill bonuses since most of these vehicles are designed to recharge their batteries when going downhills..

References

[1] Bang-Jensen, Jørgen; Gutin, Gregory. "Section 2.3.4: The Bellman-Ford-Moore algorithm", Digraphs: Theory, Algorithms and Applications (First ed.), 2000, ISBN 978-1-84800-997-4.
[2] Oscar Franzese, Diane Davidson, "Effect of Weight and Roadway Grade on the Fuel Economy of Class-8 Freight Trucks", 2011, ORNL/TM-2011/471 Available: https://info.ornl.gov/sites/publications/files/Pub33386.pdf
[3] Eric Wood, Evan Burton, Adam Duran, and Jeff Gonder, "Contribution of Road Grade to the Energy Use of Modern Automobiles Across Large Datasets of Real-World Drive Cycles", 2014. SAE World Congress 2014 Detroit, Michigan. Available: https://www.nrel.gov/docs/fy14osti/ 61108.pdf
[4] Python Software Foundation [Computer Software], 2019. Retrieved from: https://www.python.org
[5] John D. Hunter, "Matplotlib: A 2D Graphics Environment", 2007. 21-9615/07/$25.00 © 2007 IEE.
[6] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, Gäel Varoquaux, Travis Vaught, and Jarrod Millmam, "Exploring network structure, dynamics, and function using NetworkX", Proceedings of the 7th Python in Science Conference (SciPy2008) (Pasadena, CA USA), 2008, pp. 11–15, Aug 2008
[7] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford, "Section 24.3: Dijkstra's algorithm", Introduction to Algorithms, 2001, (Second ed.). MIT Press and McGraw–Hill. pp. 595–601. ISBN 0-262-03293-7