Dijkstra’s Algorithm and Google Maps

Daniel R. Lanning
Department of Math and CS
Valdosta State University
Valdosta, Georgia 31698-0040
1-229-333-5778
drlanning@valdosta.edu

Gregory K. Harrell
Department of Math and CS
Valdosta State University
Valdosta, Georgia 31698-0040
1-229-333-5778
gharrell@valdosta.edu

Jin Wang
Department of Math and CS
Valdosta State University
Valdosta, Georgia 31698-0040
1-229-333-5778
jwang@valdosta.edu

ABSTRACT

Dijkstra’s Algorithm is known as the shortest path source. In this paper, we discuss this Algorithm and the applications that the algorithm has on the modern day. In this study, we provide a pseudo-code of the algorithm. It is the backbone of every navigation system. Google Maps is a typical application of this Algorithm.

Categories and Subject Descriptors

G.4 [**Mathematical Software**]: Algorithm design and analysis, and Efficiency; I.1.2 [**Algorithms**]: Analysis of algorithms

General Terms

Algorithms; Design

Keywords

Dijkstra’s Algorithm ; Google Maps

1. INTRODUCTION

Google Maps welcomed us into the 21st century with the most popular navigation application used amongst the population. It is fast and efficient with directions, which allows us to arrive at our destination at the least amount of travel time, or distance, or with some other constrains. But how does Google Maps operate? Out of all the complexity of Google Maps, it all starts with this basic algorithm: Dijkstra’s Algorithm. It is the beginnings of Google Maps and any navigation system. Dijkstra’s Algorithm is a fairly recent concept. It has only been around for a little over 50 years. Edsger Dijkstra formulated Dijkstra’s Algorithm as a graph search algorithm that will find the shortest path. Dijkstra’s Algorithm is a k-search algorithm that will visit each node of the graph in order to find the shortest path from start to finish. With the help of computers, the algorithm can be computed in a timely matter. From the pseudo-codes of Dijkstra’s Algorithm, personal navigation systems were born. Google Maps were made possible by Dijkstra’s Algorithm. Google Maps has changed the way we look at personal navigation. Google Maps offer navigation on many different types of transportation, and even other planets ([1]), ([4]).

ACM SE '14, Mar 28-29 2014, Kennesaw, GA, USA
ACM 978-1-4503-2923-1/14/03.
http://dx.doi.org/10.1145/2638404.2638494

2. DIJKSTRA’S ALGORITHM

The computer scientist, Edsger Dijkstra, formulated Dijkstra’s Algorithm in 1956 ([1]). As previously stated, It is a graph search algorithm. Nodes and arcs define the graph. Where the nodes are the vertices and the arcs are the ordered pairs of the nodes, or the path between the nodes. This Algorithm can be used to calculate the shortest path from the starting node to the finishing node based on the different path-weight definition. Each graph will have the arcs labeled with weights. The weights are just arbitrary numbers, so they are not always measuring distance and can time and other things. Dijkstra’s Algorithm can only calculate the shortest path based on the path-weight and only weights (lengths) must be nonnegative ([2]).

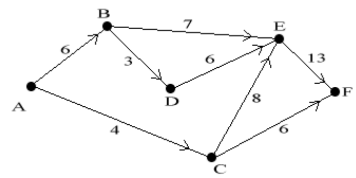

Figure 1: Directed-weighted graph

Figure 1 is an example of Dijkstra’s Algorithm. It is a directed-weighted graph with the goal of finding the shortest path from point A to point F ([3]).

2.1 Algorithm

Below is Dijkstra’s Algorithm, written out with precise instructions. We will need to understand what Dijkstra’s Algorithm is doing when finding the shortest path. The algorithm below will help with our understanding of Dijkstra’s Algorithm and it will allow us to complete problems dealing with Dijkstra’s Algorithm (The algorithm below is from Wikipedia, visited on April 7, 2013).

1. Visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the *unvisited set* is infinity (when planning a Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.

2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the *unvisited set* consisting of all the nodes except the initial node.

3. For the current node, consider all of its unvisited neighbors and calculate their *tentative* distances. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B (through A) will be 6+2=8. If this distance is less than the previously

recorded tentative distance of B, then overwrite that distance. Even though a neighbor has been examined, it is not marked as "visited" at this time, and it remains in the *unvisited set*.

4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again.

5. If the destination node has been marked complete traversal), then stop. The algorithm has finished.

6. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new "current node" then go back to step 3.

2.2 Applications of Dijkstra's Algorithm

As we know, Dijkstra's Algorithm finds the shortest path between two points of a weight graph. The weights are just arbitrary numbers. Money flow, prices, scheduling all can be optimized through Dijkstra's Algorithm. Besides the routing aspect, Dijkstra's Algorithm can be used in currency exchange, telemarketer scheduling, traffic planning, etc. ([2]).

Dijkstra's Algorithm has been useful for many of our navigational luxuries. There are only a few disadvantages to Dijkstra's Algorithm. As previously stated, Dijkstra's Algorithm cannot use negative numbers ([1]). This would have an effect if we were tracking money follow and there was a loss of money, a negative value. Another, Dijkstra's Algorithm is a lengthily process to due by hand. This was more so a problem in the early years of Dijkstra. We now have the capability of using computers to do the work.

2.3 Dijkstra's Algorithm – Applications to Shortest Flight Times Notice

To go along with Dijkstra's Algorithm, we wrote a java code that implements Dijkstra's Algorithm in finding the quickest flight time, solely flight time, to a few cities around the country. For simplicity, the code only included five different airports, but one could always include more airports to the code. Since we are in Valdosta, the code will use the Valdosta airport as the starting point/node. From Valdosta, the code will find the shortest travel time to the following cities: Atlanta, Jacksonville, New York, and Phoenix. To make the problem more interesting flights to and from Valdosta can only go through Atlanta and Jacksonville. The java code is attached to the back of the paper with explanations throughout the code.

3. GOOGLE MAPS

As we must know by now, Dijkstra's Algorithm was influential in modern day navigation systems. Out of all the navigational systems, Google Maps is the most popular of them. Brothers Lars and Jens Rasmussen developed Google Maps as c++ program in 2004. Google Maps has changed the navigation process. From Google Maps alone, we can get directions for many different ways of transportation, from driving to biking and walking. Google Maps has even implemented the best routes for truckers that minimizing the number of right turns. This was all possible due to Dijkstra's Algorithm. Google Maps is going further with mapping. We can now have street views of locations, and even images of the Moon and Mars ([4]).

3.1 The Workings of Google Maps

Understanding the workings of Google Maps is relatively simple, as we will see in the example that follows. Consider on a small scale, that each intersect is the node and each street is the arc. Dijkstra's Algorithm will find the shortest distance from start to finish. In Google Maps, it will tell us the shortest route in distance and time. On a large scale, let us consider that the cities are the nodes and the interstates are the arcs. Dijkstra's Algorithm and Google Maps will do the same thing, finding the shortest path and/or shortest time.

4. EXAMPLE

To understand Dijkstra's Algorithm better, we have the following example. This is a basic example to ease our way into understand the algorithm.

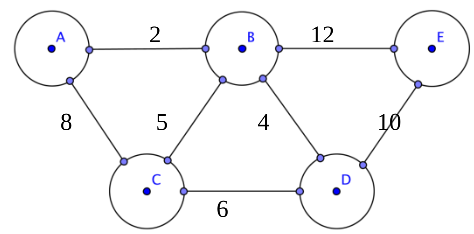

Figure 3: Example problem of Dijkstra's Algorithm, ([6]), ([7])

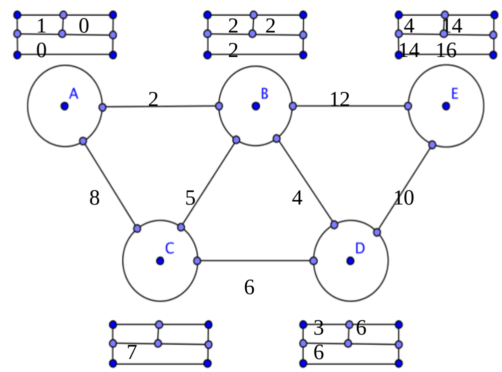

Figure 4: Answer to the example problem of Dijkstra's Algorithm, ([6])

In the following Figure 3, starting from node A, we will go to every node that is connected to node A, which are node B and node C. Next, we put the distance into the working box. Since distance 2 is shorter then distance 8, we choose node B as our next permanent node. We travel to all the nodes that are connected to node B, which are node D and node E. We choose node D our next vertex since distance 6 is shorter than distance 14. Finally, we travel from node D to node E. The distance is 16, but there is a shorter distance to node E of 14. We choose node E as the final permanent node with distance from node A to node E of 14. Now, we will find the shortest path. 14-10=4, therefore the path cannot go from node E to node D. 14-12=2, from node E to node B. 2-2 = 0, from node B to node A. For Dijkstra's Algorithm, we want to take the final shortest answer and minus it from all the paths to the beginning. When we find the path that will end in zero we then have found the shortest path. In this example, the shortest path is from node A to node B to node E ([5]).

5. CONCLUSION

Thanks to Dijkstra's Algorithm, we have the modern day technology of navigation. Dijkstra's Algorithm allows us to find the shortest path from start to finish. Google Maps has taken Dijkstra's Algorithm to create the most popular navigation system. Though Dijkstra's Algorithm can be lengthily, it is not impossible to complete. With computers running the pseudo-code of Dijkstra's Algorithm, the personal navigation system is born. Dijkstra's Algorithm changed the ways of navigation. There is one thing we can work on changing with Dijkstra's Algorithm to better it in the future. If we can implement negative values for the arcs, we can have further uses for the algorithm. This would better Dijkstra's Algorithm in the future and it will have more applications. Google Maps continues to further their navigation system. With every new update, a new feature appears. We can have one day a visual preview of the route we would have to travel. Also, Google Maps continues to better their navigation system by suggesting alternative routes to avoid traffic and delays. Dijkstra's Algorithm and Google Maps has better our society.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/Dijkstra's_algorithm Visited on April 6, 2013

[2] http://www.cs.princeton.edu/~rs/AlgsDS07/15ShortestPaths.pdf Visited on April 7, 2013.

[3] http://www.csse.monash.edu.au/hons/projects/2000/Greg.Rogers/graphdemo.gif Visited on April 6, 2013.

[4] http://en.wikipedia.org/wiki/Google_Maps Visited on April 8, 2013

[5] http://www.youtube.com/watch?v=xT5o1QCeWS8 Visited on April 8,2013.

[6] http:// www.geogegbra.com/cms Visited on April 8,2013.

[7] Winston, W. L. 2004. *Operations Research: Applications and algorithms*, Fourth Edition, Thomson Brooks/Cole, Belmont, CA