**Lab Report**

Title: I’ll Be Home for Christmas: Routing Along Networks

Notice: Dr. Bryan Runck

Author: Cecelia Isaac

Date: 03/31/2021

**Project Repository:** *https://github.com/CeceliaAi/GIS5572/tree/master/Lab3*

**Abstract**

The problem in this lab is to use two different networks to create an optimal route for two delivery drivers. The data used is a list of addresses and a road polyline shapefile. For the first part of the lab, we will make our own network and run the routing tool on that. Next, we will run the same addresses using Esri’s network to get another route output. For both of these networks, we have a series of barriers and restrictions that must be entered. The goal is to create the fastest optimal route. The results of the lab will be two sets of directions, one from each network’s route. The results may be different, and so we cannot compare them against each other to verify the results. Instead, we will have to verify that the restrictions we input are followed. In conclusion, we will compare the strength of our results and their fitness for our two drivers.

**Problem Statement**

The problem in this lab is to use two different networks and platforms to create the shortest route for two delivery drivers between 10 addresses and the origin point. To accomplish this, we will geocode the addresses. We will also use a road dataset to create a network, and to inform our network’s barriers, restrictions, and costs. For the second part of the lab, we will use the same addresses, but utilize Esri’s network to compare the routes. Our goal is to find the fastest route(s) so the drivers will be done as soon as possible, and to output those routes as directions.

*Table 1. Data Requirements*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Requirement** | **Defined As** | **Spatial Data** | **Attribute Data** | **Dataset** | **Preparation** |
| 1 | Road network | Raw input dataset from MNDOT | Road geometry |  | [Mn GeoSpatial Commons](https://gisdata.mn.gov/dataset/trans-roads-mndot-tis) | Clip to study area; find speed limit; create a network |
| 2 | Addresses | Ten stops plus origin/destination point | Point data |  | Address.csv | Geocode table |

**Input Data**

The input datasets include a set of 10 addresses and one origin/destination point, which will need to be geocoded. We will also need to use a roads dataset to create our own network. We will also use this roads data to create restrictions in our route.

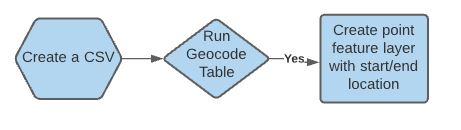
*Table 2. Data Source*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | Minnesota Roads | Raw input dataset for routing analysis from MNDOT | [Mn GeoSpatial Commons](https://gisdata.mn.gov/dataset/trans-roads-mndot-tis) |
| 2 | Address | Destinations for routing analysis | CSV doc |

**Methods**

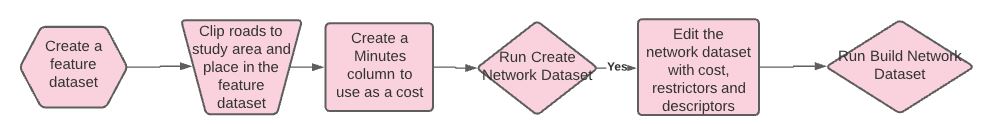
The flow chart below shows steps necessary to both parts of the lab (in blue) and steps only necessary to creating our own network (in pink). To begin the lab, I geocoded the eleven addresses provided. Afterward, I selected the start/end location and made it its own point layer. The remaining ten points stayed in the original layer (Figure 1).

*Figure 1. Importing addresses*



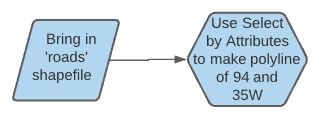
To make my own network, I used a roads centerline shapefile. Before creating the network, I calculated the time cost in a new column by converting the speed limit data into minutes per mile. I started by trying to put everything into meters but this created a cost that was much too high, and did not mesh easily with some other steps in the lab. I eventually put the calculation into miles. I put the shapefile into Create Network Dataset and built the network. Once the shapefile was a network, I opened the properties to edit the costs, including length and the Minutes column I calculated on the shapefile. I also added oneways as a restriction of Avoid (high). I did not include U-turns as a restriction. Once the costs, restrictions, and hierarchy was added, I saved the changes (Figure 2).

*Figure 2. Creating a network*



To create a barrier layer, I selected out the 35W and 94 polylines and but them into their own layer. I then used a buffered version of the original roads layer to erase intersections on the 35W/94 layer. I set this polyline aside for later use (Figure 3). These polylines are labeled in blue because I thought I would be using them for both networks. In the end, I only used them for my network and not with Esri’s.

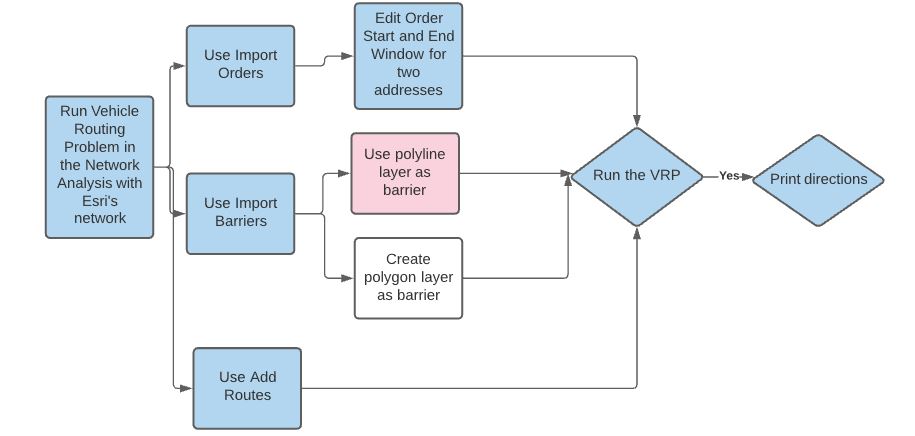
*Figure 3. Making barriers*

**

The next part of the lab was similar for both my network and Esri’s network. I launched the Vehicle Routing Problem. From the toolbar, I imported by origin point as a Depot and my addresses as Orders. I set the delivery times in the attribute table. I also stipulated two drivers in the Routes section, as well as the number of stops they would make (Figure 4).

For my network, the 35W/94 polyline with gaps worked. Initial attempts using a 35W/94 polyline without the gaps at intersections caused the tool to fail, but the insertion of gaps solved the problem for my network. The Esri network failed with the polyline barrier, both with and without gaps. For Esri’s network, I had to create polygon barriers at intersections to prevent the route from using the highways. This was an iterative process that involved making some polygons, running the route, reviewing it for places where the route jumped onto the highway, and adding more polygon features into the barrier class. After the barriers were added, the tool could be run.

*Figure 4. Using the Vehicle Routing Problem*

**

**Results**

The results for a route made with our own network are 2 hours 30 minutes. The number of miles traveled is 89. One driver makes all of the stops. When the maximum number of stops is changed in the VPR tool to five, then the route is more equal but the time increases to 2 hours and 30 minutes for Driver 1, and 1 hour and 5 minutes for Driver 2.

The results for a route made with Esri’s network are: 2 hours and 38 minutes. The number of miles is 93. One driver makes all 10 of the stops. As with the other network, the maximum number can be changed to five, resulting in a time of 2 hours and 35 minutes for Driver 1, and 1 hour and 21 minutes for Driver 2.

The driving instructions are in the folder labeled “Directions” in GitHub, since they are too long to include here.

**Results Verification**

The results can be verified by checking that the final output follows the costs, barriers, and restrictions we input when creating the network and the routes. The two routes are different but that does not necessarily mean one is wrong, so verification has to be done through error-free outputs of the tool, and making sure the restrictions are followed.

**Discussion and Conclusion**

I learned how to use ArcGIS’s Vehicle Routing Problem tool, and the different ways different networks work within those systems. There were a lot of small details that needed to be correct in order to properly run the tool. I spent a lot of time tweaking. The barrier took the most time, partially because at first it was not clear what the error message actually meant. Creating polygon barriers manually did solve the problem, but it was an imprecise and time-consuming way to control the route. Another issue was creating the Minutes column. I had to rebuild my network multiple times after getting errors, or successfully running the tool that produced a route eight months long.

I did use ArcGIS Online’s Plan Routes tool, but did not finish in time to include it in the Methods section. The Plan Routes tool also did not like the long barrier of the two highways, and barriers needed to be structured as polygons. I was able to import my polygons, but did not finish figuring out how to set a delivery time for the two points with specified delivery times.

Whether or not the route is optimal is hard to say. The tool seemed to imply that only one driver was needed. Is Esri facilitating companies who want to reduce their staff and overwork those who remain? That’s beyond the scope of this lab report.

**References**

*arcgis.network.analysis module—Arcgis 1.8.4 documentation*. (n.d.). Retrieved March 25, 2021, from <https://developers.arcgis.com/python/api-reference/arcgis.network.analysis.html#solve-vehicle-routing-problem>

*Create a network dataset—ArcGIS Pro | Documentation*. (n.d.). Retrieved March 14, 2021, from <https://pro.arcgis.com/en/pro-app/latest/help/analysis/networks/how-to-create-a-usable-network-dataset.htm>

*Metadata: Roads, Minnesota, 2012*. (n.d.). Retrieved March 14, 2021, from <https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dot/trans_roads_mndot_tis/metadata/metadata.html>

*Network Analysis: Routing*. (2018, April 3). <https://www.youtube.com/watch?v=fVPUkuCX0Kk>

*Plan Routes—Portal for ArcGIS | Documentation for ArcGIS Enterprise*. (n.d.). Retrieved March 25, 2021, from <https://gis.fema.gov/arcgis/help/en/portal/latest/use/plan-routes.htm>

**Self-score**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Description** | **Points Possible** | **Score** |
| **Structural Elements** | All elements of a lab report are included **(2 points each)**:  Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score | 28 | **28** |
| **Clarity of Content** | Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**. | 24 | **24** |
| **Reproducibility** | Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified. | 28 | **28** |
| **Verification** | Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**. | 20 | **20** |
|  |  | 100 | **100** |