Lab Report

Title: *Lab3: Using the Vehicle Routing Problem to find the most efficient routes*
Notice: Dr. Bryan Runck
Author: Lucas Rosen
Date: 4/1/2021

Project Repository: ***https://github.com/LRosen656/GIS5572.git***

Abstract

Network datasets and inputs can be used to find the most efficient routes. The goal of this assignment is to use depots, orders, and conditions to find the two best routes for delivery. A network dataset was derived from street data and given specific parameters. A depot was added for the start and end point. Orders were added and given time parameters. Two routes were added to the dataset. Barriers accounted for the closure of highways. Finally, the "Vehicle Routing Problem" was used to find both routes. Arc Pro was able to create valid routes. Arc Online was limited and could not account for the barriers or time constraints.

Problem Statement

The purpose of this lab is to use a network analysis to find the best routes for two drivers to various destinations and back to the starting location. The goal is to be follow the road and time constrictions while making the most efficient route possible. It was done in both Arc Pro and Arc GIS Online.

Table 1.The required data is legal routes, time cost, barrier, depot, and orders

#	Requirement	Defined As	Spatial Data	Attribute Data	Dataset	Preparation
1	Legal Routes	One ways and navigable routes obeyed	MN GEO Roads	Oneway	Mn GeoSpatial Commons	Add to Network Dataset
2	Time Cost	Distance * 1/Speed	MN GEO Roads	Shape, Route Speed	Mn GeoSpatial Commons	Add to Network Dataset
3	Barrier	35W and 94	MN GEO Roads	Route ID	Mn GeoSpatial Commons	Add to Vehicle Routing Problem
4	Depot	Starting and end point	Point	Name	Manually Added	Geocoded and Added to Vehicle Routing Problem
5	Orders	Stops for the routes	Points	Earliest and latest start times	Manually Added	Geocoded and Added to Vehicle Routing Problem

Input Data

Three data inputs were used in this assignment. First, Road Centerlines were downloaded from the Minnesota Geospatial Commons. This shapefile is the Geospatial Advisory Council Schema and contains street data from ten Metropolitan Counties. It is in NAD 83 UTM 15 coordinate system. Other input data was the starting depot address and the order addresses. Both were geocoded from excel files with the addresses in them. The output coordinate system is the same as the Road Centerlines.

Table 2. The input Data was Minnesota Roads, Depot, and Orders.

#	Title	Purpose in Analysis	Link to Source
1	Minnesota Roads	Creating a Network Dataset and barriers.	Mn GeoSpatial Commons
2	Depot	Starting and ending address	NA (geocoded)
3	Orders	Stop addresses with restrictions	NA (geocoded)

Methods

Figure 1 shows a flowchart of the methods used in Arc Pro.

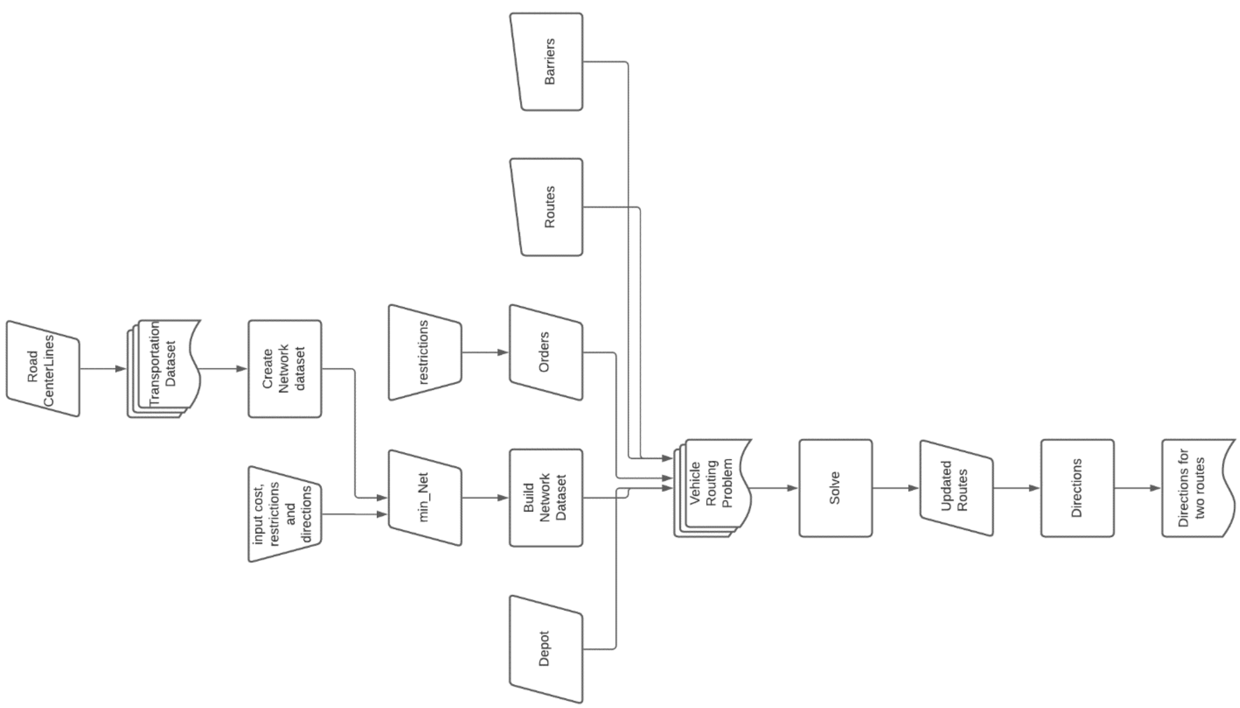

Figure 1 Arc Pro Methods

Creating a Network Dataset

To create a network dataset, I first created a feature class dataset called "Transportation". I then imported the Road Centerlines feature class to the dataset and called the feature "Streets". With the dataset and feature, the "Create Network Dataset" was used. The inputs were the dataset and feature class, and the outputs were a network dataset called "min_net", the junctions feature class (for turning and directions), and the streets feature class.

To create the correct parameters, I right clicked on the network dataset to go to properties. The features that were needed are cost, restrictions, and directions. Each cost and restriction need to be set to "along street" and "against street".

Costs

Two types of cost are needed for the network dataset: Distance and Time. To get the distance, I set the along and against streets distance equal to shape length and the units to meters. To set the time cost, I took the shape length (meters) multiplied by the inverse of route speed (miles per hour) and divided it by the meters per minute conversion unit (~26). The result is the time cost in minutes.

Restrictions

The only restriction necessary for the network was found in the "oneway" attribute. The attribute has four conditions: "B", with both ways routable (could be ignored), "T", one way against street (restriction along), "F", one way along street (restriction against), and "N" which is non routable. To solve, I had along streets restricted from "T" and "N", and against streets restricted from "F" and "N". The result shows the legal routing directions.

Directions

The final setup for the Network Dataset was to input directions. There are three requirements. The prefix direction was set to "ST_PRE_DIR", the prefix type was set to "ST_PRE_TYP, and the base name was set to "ST_NAME".

Once the Network Dataset was ready, I set the travel mode to "Driving Time" and the cost to "Minutes". I then built the Network Dataset.

Vehicle Routing Problem

Once the Network Dataset was built, I was able to use it in the vehicle routing problem. I used the "Create Vehicle Routing Problem" and input the "min_net" dataset. I set the time to minutes and the distance to meters. I also set the default date to December 23rd, 2019.

There are four inputs for the vehicle routing problem: The Depot, which is the start and end point, the Orders, which is where the stops are, routes, which will be the directions, and the barriers, which will add additional restrictions.

Depot

The Depot was geolocated to the address found in the word document. Once located, it was added to the dataset with the name "Warehouse".

Orders

Ten orders were located to the map and added to the dataset with the name as the address and the default start time as 12/23/2019 8:00am. However, two of the addresses (starting with 255 and 3300) needed to have deliveries done between 10:00 and 11:00am. To fix this, I manually changed their start time to 10:00am and their end time to 11:00am in the attribute table.

Routes

To add routes, the "add routes to vehicle routing problem" function was used. The number of routes was set to 2, the starting and end point was set to "Warehouse", and the earliest and latest start time for 8:00am. For the maximum number of deliveries, I set the number to 5. This allowed both drivers to have an equal number of deliveries.

Barrier

The final input for the vehicle routing problem was to input the barriers. I first selected all the streets that had the Route Name equal to highways 94 or 35W. I then erased it with all routes not equal to the highways with a 1-meter buffer. The result shows a barrier with gaps to allow other streets through. While there are some errors such as ramps leading to the highways, it did not affect the result.

Solving Routing Problem

Once all the inputs are ready, I used the function, "Solve Vehicle Routing Problem". I input the depot, orders, routes, and the barrier. I also set the date to 12/23/2019. The result shows routes on the map and I outputted the directions to a text file.

Attempting on Arc GIS Online

To solve the vehicle routing problem online, I first uploaded the depot, orders, routes, and barriers online. The closest function I found to Arc Pro (that worked) was called "Find Routes". I used that function and input the orders, depot, set the number of vehicles to 2, the max orders to 5, the date to 12/23/2019, and 'return to start' equal to "True". This works but the program did not allow me to input the barrier or input the stop times.

Results

Figure 2 shows the map of the routes created in Arc Pro. It shows the Warehouse, stops, and two routes the drivers would take. The directions, notebooks, and layout were uploaded to GitHub. Route 1 takes a total of 2 hours and 25 minutes. It drives 72 miles and has 51 minutes of wait time (from the delivery restrictions). Route 2 Takes 55 minutes. It drives 41 miles and has 0 wait time.

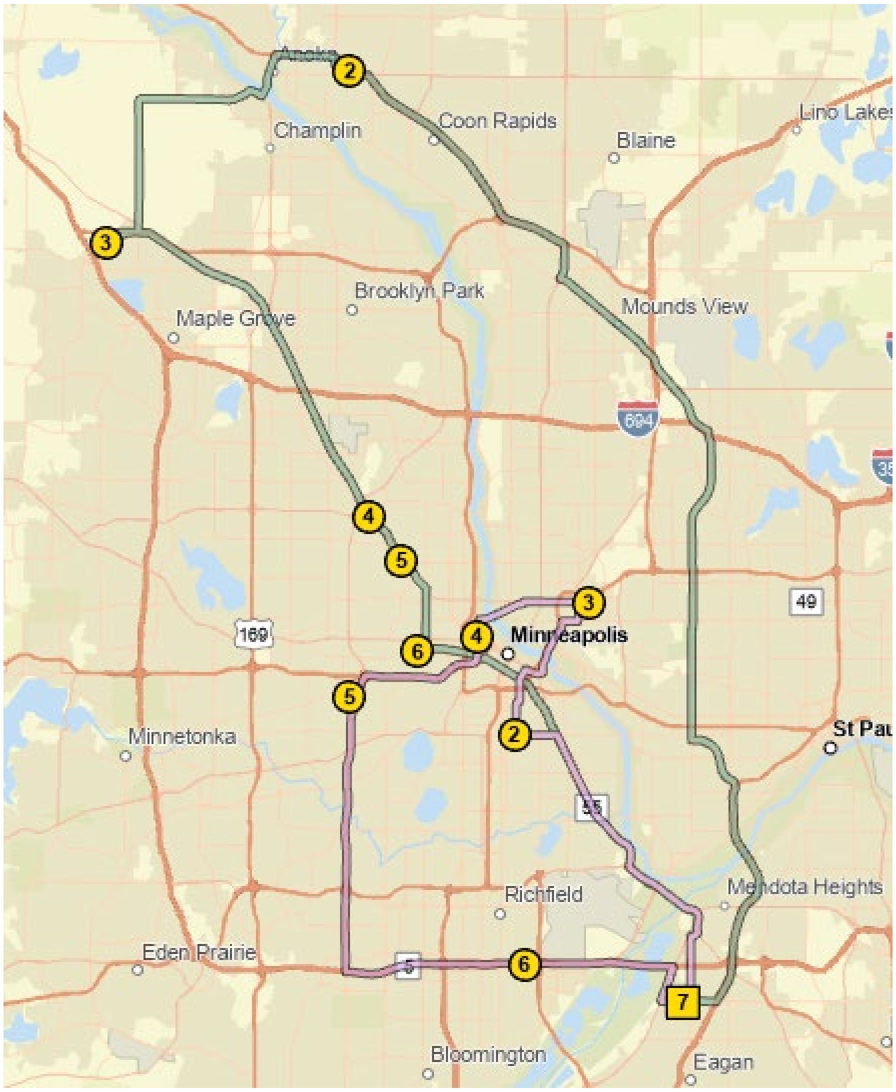

Figure 2 Routes: The Square is the Warehouse and the Circles are the stops.

Results Verification

To verify the results, I looked closer at the directions to make sure all the rules were followed. I first checked that Highways 94 and 35W were not driven on. There was one point in the directions that stated "NB I 35W" in Mounds Park but that is not the same as 35W. I also checked the one ways to make sure they were not going against traffic. One route I checked was on 10th street when it goes across 394. The directions did obey the one way (south) so all rules were obeyed.

While the Arc Online program worked, it was limited and did not obey all the rules; both Highways 94 and 35W appeared in the directions and the time constraints were not followed.

Discussion and Conclusion

Observations

Arc Pro was able to successfully solve a vehicle routing problem with a unique dataset. However, while the routes are technically correct, they may not be the most efficient. For example, if just one person had gone to all the stops, it would result is less driving time. However, since there were two people driving, I programmed the routes to be fairer. Furthermore, traffic was not included which may cause some of the streets driven on to be impractical.

Issues

There are several issues that occurred throughout the project. First, while I was able to get successful routes in the user interface, it only worked when I used the "ready to use tools" to derive routes. While all the code the notebook appears to be correct, the result is a route that has nonsensical times and ignores the barriers. I am unsure why this happened, but it is possible the time issue happened when creating the time cost in the Network Dataset (not sure why it worked in the GUI though).

Arc GIS Online is meant to be more user friendly. Because of this, their default tools do not have as many parameters as Arc Pro, and I could not input the times, nor the barrier. Looking deep into Python, I did find a tool called "Solve Vehicle Routing Problem" that looked the same as Arcpy, but I was unable to get it to work.

Conclusion

Both Arc Pro and Arc GIS Online can successfully solve the Vehicle Routing Problem. Arc Pro was able to apply all the restrictions from the instructions. However, Arc GIS Online has less features than Arc Pro so it would not comply with all parameters. Arc GIS Online is more useful for simple problems with no restrictions.

References

ArcGIS Pro Help. "Create a Network Dataset." Create a Network Dataset-ArcGIS Pro | Documentation, pro.arcgis.com/en/pro-app/latest/help/analysis/networks/how-to-create-a-usable-network-dataset.htm.

ArcGIS Pro Help. "Find the Best Routes to Service Paired Orders." Find the Best Routes to Service Paired Orders-ArcGIS Pro | Documentation, pro.arcgis.com/en/pro-app/latest/help/analysis/networks/find-best-routes-to-service-paired-orders.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	**25**
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	