

## Lab Report: Semester Project Energy Mobile

Title: Energy Mobile Lowest Energy Route Tracker

Notice: Dr. Bryan Runck

Author: Cole Anderson

Date: 5/13/21

**Project Repository:** <https://github.com/and04671/GIS5572/tree/main/SemProj>

**Time Spent:** 45 hours

### Abstract

The goal for this study was to define and find the least energy route between a start point and end point. The base data utilized for the project are a road centerlines line feature dataset of metro area roads and a 30m DEM of Minnesota. The program first defines slope\*length as energy for all roads in the created network dataset. It creates a route solver with defined point inputs and a manually created travel mode to find the route with the lowest impedance, set as energy value. This route solver outputs a line path that can be placed on a basic base map for every instance, saving it to file and adding the layer to the open ArcPro map. In order to confirm that the output routes are indeed the lowest energy, the path must fit 3 criteria:  $\geq$  length than the shortest route path,  $\leq$  energy than the shortest route path, and prove feasible for actual pedestrians. The examples pass this test except the feasibility of one route. While this program did effectively solve the set problem statement, it could be improved by increasing the accuracy of the calculation for slope and energy, and improving ease of use as a UI. The project made use of network, ETL, and Python skills.

### Problem Statement

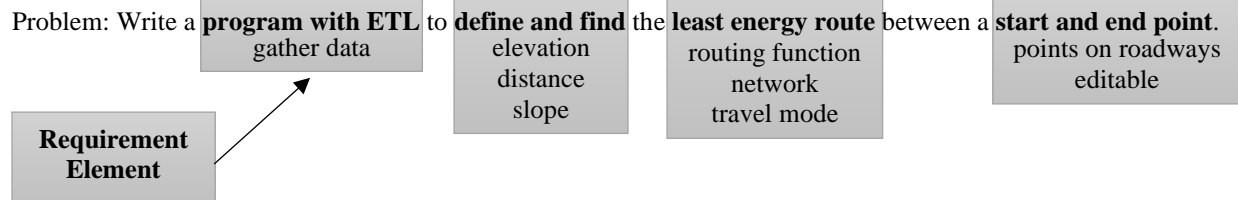


Table 1. Problem Statement Requirements

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	Gather elevation data	Digital elevation model of Minnesota	Elevation	X	<a href="#">MN DEM</a>	Create ETL

2	Gather roadmap data	Network of road lines in metro area	Road centerlines w/ length in m	Road names	<a href="#">Roads_GACS</a>	Create ETL
3	Apply DEM to roadlines	AddSurfaceInformation_3d Find Zmax, Zmin	Elevation, Road centerlines	Road names	X	Roads and DEM in gdb
4	Find Slopes/energy	$(Z_{\max} - Z_{\min}) / \text{Length} = \text{slp.}$ $\text{Slope}(\text{Length}) = \text{energy}$	Length, Elevation	Road Names	X	Unit conversion, new fields
5	Create inputs	-network dataset -inputs layer: takes user given point coordinates in UTM	Network of roads, point coordinates	Road Names	X	User coordinates
6	Travel Mode	-manual creation in ArcPro uses energy as impedance	X	X	X	X
7	Run Route solver	Finds shortest route, can be modified using methods	X	X	X	All other steps

## Input Data

**Road Data:** The program requires road data with lengths; the names are irrelevant until directions are produced in a future update. This particular dataset is an amalgamation of road data from all of Greater Metro counties. The dataset is in UTM 15 NAD83, and the horizontal units are in meters. It is updated continually but was originally published in 2020.

**DEM Data:** The program also requires elevation data in order to calculate slope. This 1:24,000 dataset from USGS captures elevation above sea level in 30 m resolution for the state of Minnesota. The height units are in feet while the horizontal units are meters, which will have to be standardized. It was published in 2004, but is continually updated.

*Table 2. Input Data Sources*

#	Title	Purpose in Analysis	Link to Source
1	Road Centerlines (Geospatial Advisory Council Schema)	Paths for which elevation, length and energy are calculated, the network	<a href="#">Roads_GACS</a>

2	Minnesota Digital Elevation Model - 30 Meter Resolution	Slope and elevation of road centerlines	<a href="#">MN DEM</a>
---	---	---	------------------------

## Methods

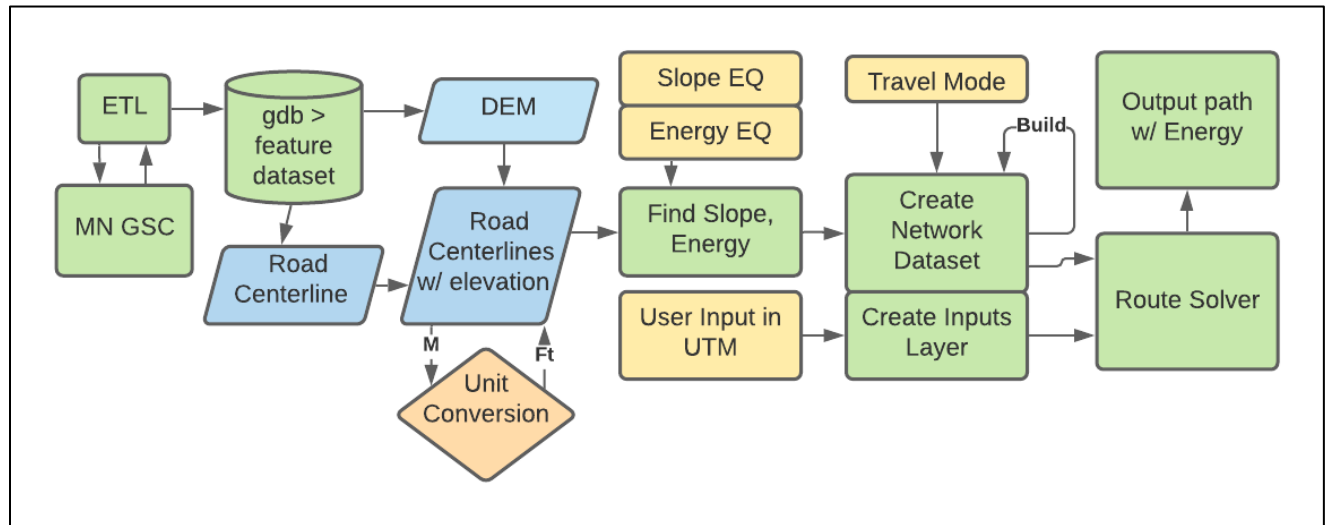


Figure 1: Data Flow Diagram

### Data Acquisition and Upload

- Before the project analysis, a DEM and a roads layer are required. The Extract, Transform, Load or ETL function is used to acquire these datasets. This tool works in four fundamental steps. First, the desired search query is sent to the API. The user can search through the results, and then resources. Under each resource, there is a designated location URL. This URL is then sent a get request, and returns the required files. These files come in ZIP format, and the ZipFile library is used to unzip them to the correct location.
- In order to ensure all layers have the same coordinate system and are located in one place, create a new feature dataset in the project geodatabase (GDB). Find the spatial reference for the RoadCenterline feature class using the describe method.
- Use Arcpy.CreateFeatureDataset\_management to create a feature dataset called 'Network' inside the project geodatabase. Apply the spatial reference gathered earlier.
- Using Arcpy.FeatureClassToGeodatabase\_conversion, bring the RoadCenterline feature class into 'Network' feature dataset, inside the project geodatabase.
- Arcpy.CopyRaster\_management to try to bring raster into 'Network' feature dataset (does this work?).

### Finding Surface Attributes

- Use Arcpy.AddSurface Information\_3d to apply the DEM raster elevations to the RoadCenterline feature class.

- The vertical units from the DEM are feet and the RoadCenterline units are meters. Use `Arcpy.calculate field_management` to convert the RoadCenterline feature class units from M to Ft. Find the road slopes for the RoadCenterline feature class using `Arcpy.calculate field_management`. Use the equation:  $Z_{max} - Z_{min} / LengFT$ .
- Find energy costs for the RoadCenterline feature class using `Arcpy.calculate field_management`. Use the equation  $slope * LengFT$ .

### Network Datasets

- Create a network dataset from the RoadCenterline feature class using `Arcpy.na.CreateNetworkDataset`.
- Create an initial travel mode and cost manually in ArcGIS Pro, making sure to re-build after modifying.
  - create a cost called 'Energy', and set to the calculated energy value
  - create the new travel mode that using the new energy cost as impedance and road length as distance
- Create a temporary ND layer for faster processing environment using `arcpy.nax.MakeNetworkDatasetLayer` (permanent, new).

### Route Solver

- Create a points geometry inputs feature class for the input points. Again, use the spatial reference from original RoadCenterline feature class. Use `arcpy.CreateFeatureclass_management` to create a new feature class that holds point geometry.
- To add stops (Start/Stop) to the inputs layer, the UTM 15 coordinates are inserted into the attribute table using a cursor function. This cursor is deleted after use to unlock the table.
- Initialize the route solver using `arcpy.nax.Route`.
- Set the route solver properties for the best output. Make sure the desired travel mode is selected (manually created mode). Set the accumulate attributes to determine what attribute is counted.
- Load the inputs point geometry layer using the `route.load` method.
- Finally, complete the Route solver using the `route.solve` method

### Results

The figures below show 4 path examples created by running the script in 4 different areas and sets of endpoints. Figure 2 also shows a shortest distance path (red) and attribute accumulation of energy in the least energy path. Figures 3 and 4 run across sections of UMN campus near Washington Ave. Bridge, an area well known for its street layers, strange overpasses, and steep slope. Eventually, the program should be able to navigate this area. It was

a sort of final test for the prototype built here. Figure 5 shows a long walk across the river in order to show the program working in a larger area that the other figures.

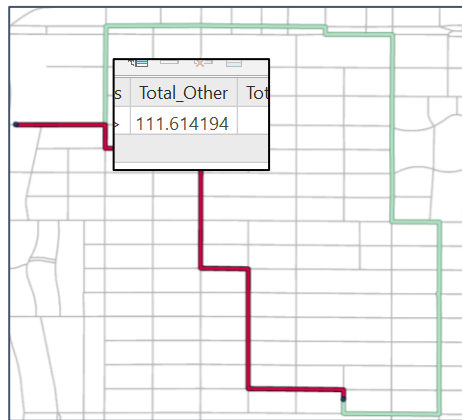


Figure 2: Short Route



Figure 3: West Bank Route

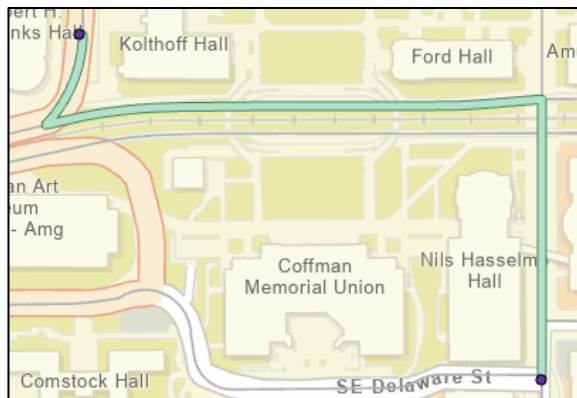


Figure 4: East Bank Route

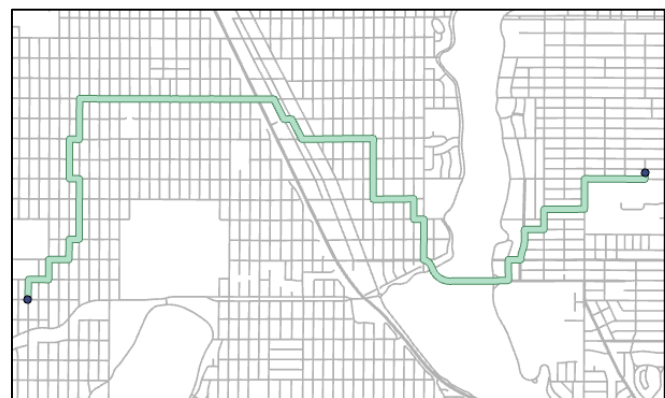


Figure 5: Crosstown Route

## Results Verification

First, the algorithm needs to deliver the least energy route. It is difficult to establish if this is true, because the software can test thousands of possible routes in under a second, while it would take a very long time for a human to double check. However, there are some heuristic tests and approximations that can be used to check for a value close to the correct answer. The simplest test is a simple mental scan. To anyone who knows the mapped area well (like most UMN college students), a visual comparison to mental topography is helpful for verification. Does the route run down steep hills or river valleys, or stay on flat urban terrain? Additionally, we know that the length of the least energy route is greater than or equal to the shortest distance route. We also know the energy required for the least energy route is less than or equal to the shortest distance route. It is possible for the lowest energy and shortest route to be identical, but the above rules are not broken in any case.

The routes returned by the routing solver need to be feasible for pedestrians. This means the path is actually walkable is suggested. For example, if we look at Figure 3, the pathway given coincides with sidewalks for most of

the ways, but the traversal at the West Bank Skyway is not possible for pedestrians without entering a university building to drop a level. The program assumes the point moving along the path is travelling *on the road*, something that will have to be fixed in the next update.

*Table 3. Result Verification*

	Energy $\leq$ Shortest Path Energy	Length $\geq$ Shortest Path Length	Feasibility
Figure 2	Yes	Yes	Yes
Figure 3	Yes	Yes	No
Figure 4	Yes	Yes	Yes
Figure 5	Yes	Yes	Yes

## Discussion and Conclusion

This project script effectively solved the project statement; it gathered data on roadways, applied DEM elevation to it, calculated the slopes, and found the lowest energy path between point set by the user. Granted, the original scope has tightened. A web app version will have to be developed in the future. The project applied both the network skills from Lab 3 and the CKAN ETL from Lab 1. If roads were replaced with pixel cells like in Lab 2, the lowest energy could be calculated across a cost surface instead of a network. It should, overall, be a representative example of many of the skills utilized in this course. There don't seem to be any severe issues with the network data itself; the quality is generally good compared to the ESRI network. There are a few street segments that were not brought into the network because of extremely short length, marked by a thrown warning at creation. The *AddSurfaceInformation* method used early in the script is new; it seems the most efficient way yet to find elevation for vector line of point features.

The results can be improved in several ways given more development time. The current model needs a few accuracy improvements, to start. First, road data is not an adequate proxy for walkable path. There are problems with this around bridges, tunnels, and walkthroughs. The addition of sidewalk data, or being able to mark where walkable path is unavailable, would be valuable. Second, the slope and energy equations are inaccurate. The current slope model does not account for valley and hill shaped areas. In terms of energy, it does not take 0 J to walk along an infinite road at zero slope, like is returned. The energy equation also does not factor in the rate of energy increase for different weights. It does not include friction either. Thirdly, there are areas suggesting walking off an overpass. This suggests an error with vertical connectivity, although this setting is turned on and the data contains connectivity data. The current model only accepts UTM coordinates. UTM coordinates are useless to an everyday user, and really so are DDS. The interface should allow the user to manually click-select a point or enter an address, avoiding coordinates altogether. Eventually, the application should return directions, because many users prefer text directions as an additional option. The current model uses one test travel mode, but ideally the final model would have several for different levels of mobility. Restrictions are not utilized in the current model, but could be used to make pedestrians explicitly avoid slopes over a certain value depending on the walking mode settings. Restrictions would stop pedestrians walking across interstates and other dangerous areas. The final application should have a free

scroll option that allows users to view an extent of streets and receive their slope directly, either via color or weight. Ideally, the final model will be embedded as either a open source web app, or in ArcGIS Online. Finally, the ETL is still painful to use. It takes a lot of manual clicking, scrolling, and number guessing to get the desired data.

## References

- De Neef, M. (2013, May 29). Gradients and cycling: How much harder are STEEPER CLIMBS? Retrieved May 10, 2021, from <http://theclimbingcyclist.com/gradients-and-cycling-how-much-harder-are-steeper-climbs/>
- ESRI. (2021). Create a network dataset. Retrieved May 10, 2021, from <https://pro.arcgis.com/en/pro-app/latest/help/analysis/networks/how-to-create-a-usable-network-dataset.htm>
- ESRI. (2021). Route. Retrieved May 10, 2021, from <https://pro.arcgis.com/en/pro-app/latest/arcpy/network-analyst/route.htm>
- ESRI. (2021). What is the network analyst module (arcpy.nax). Retrieved May 10, 2021, from <https://pro.arcgis.com/en/pro-app/latest/arcpy/network-analyst/what-is-the-network-analyst-module.htm>
- Metropolitan Council. (2020, March 3). Road centerlines (geospatial advisory council schema). Retrieved May 10, 2021, from <https://gisdata.mn.gov/dataset/us-mn-state-metrogis-trans-road-centerlines-gac>
- U.S. Geological Survey. (2004, January 5). Minnesota digital elevation model - 30 meter resolution. Retrieved May 10, 2021, from <https://gisdata.mn.gov/dataset/elev-30m-digital-elevation-model>

## Self-score

*Fill out this rubric for yourself and include it in your lab report. The same rubric will be used to generate a grade in proportion to the points assigned in the syllabus to the assignment.*

Category	Description	Points Possible	Score
<b>Structural Elements</b>	All elements of a lab report are included ( <b>2 points each</b> ): 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	<b>27</b>
<b>Clarity of Content</b>	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 ( <b>12 points</b> ). There is a clear connection from data to results to discussion and conclusion ( <b>12 points</b> ).	24	<b>23</b>
<b>Reproducibility</b>	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	<b>27</b>
<b>Verification</b>	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated ( <b>10 points</b> ), the method of comparison is clearly stated ( <b>5 points</b> ), and the result of verification is clearly stated ( <b>5 points</b> ).	20	<b>19</b>
		100	<b>96</b>