Emergency Services Router
Team Hermes

Wade Roberts, Riley Hales, Mitch Hadfield

Proposal Leader: Wade Roberts

CE EN 514

Project Objective

The objective of our project is to create a Tethys web application that can be used to map the fastest route to people in an emergency situation, given the location of the emergency respondents and potential obstacles that might block the path of said respondents.

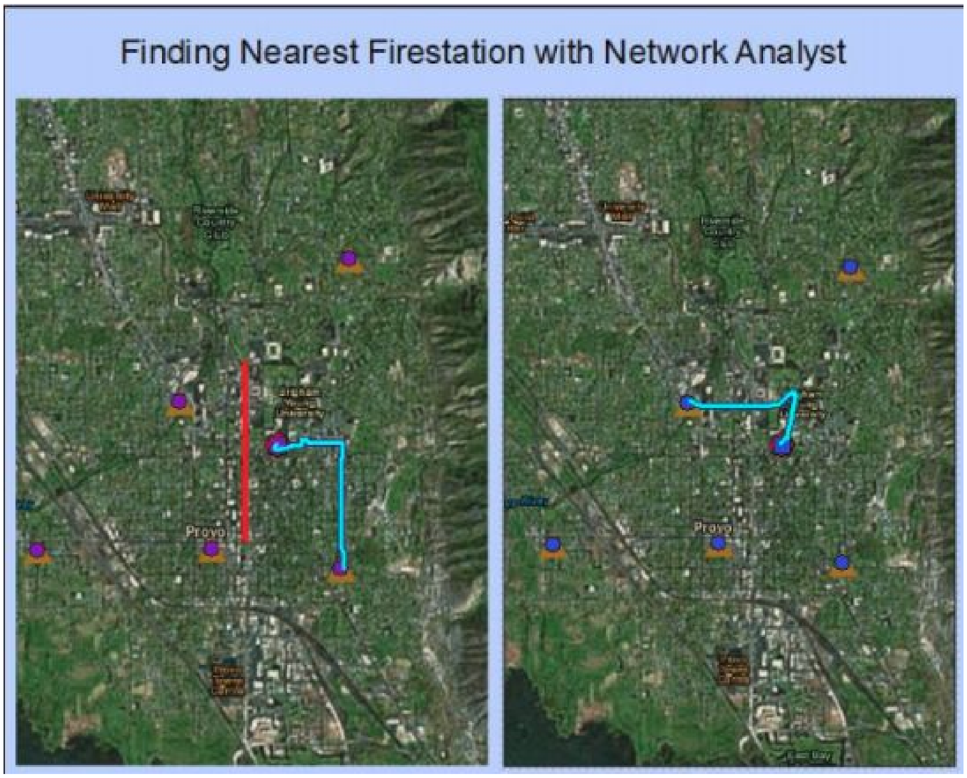

Figure 1: Optimal Route Finding 414 Project

The idea for this project came from a Civil Engineering 414 final project by Tyson Terry, Jason Biesinger, Levi Burr from 2017 called "Optimal Route Finding." Part of their final map can be seen in Figure 1 above.

Data

The application that we are proposing will require a few different types of vector data. All of these datasets will be specific to a certain region (e.g. Provo, UT), which will allow a user to perform analysis in a specific region. These different datasets include the following:

- Location of Roads
- Location of Important Buildings for Emergency Respondents
- User-created obstacles and points
    - User Created Point of Emergency
    - User Created Obstacles
- Location of Recurring Obstacles (e.g. Parades)

The larger datasets above, such as the location of roads, will most likely be stored as a shapefile in a geoserver. This will allow it to be served quickly to users, and allow for scalability with larger datasets with many vertices. Using this approach will allow us to create models of larger areas, such as the city of Salt Lake.

The smaller datasets, such as the location of important buildings and obstacles can be stored as GeoJSON text in a relational database. The database would most likely be something pretty simple, with a list of users information with an associated user ID in one table (which may already be included in the PostgreSQL database in Tethys), and another table with tuples consisting of user ID's and their corresponding obstacle lines stored as GeoJSON text (See Figure 2 below for a diagram of the proposed relational database). Using this type of data storage would allow users to create and save obstacles that could be placed on the map to model different scenarios.

username	**email**	**password**	**userid**
bobsmith	bob@gmail.com	*******	1
janesmith	jane@gmail.com	***********	2
billyjoe	b.joe@gmail.com	****	3

userid	**feature_name**	**GeoJSON feature**
3	Parade	{"type": "Feature", "geometry": etc...}
1	Traffic Jam	{"type": "Feature", "geometry": etc...}
2	Road Block	{"type": "Feature", "geometry": etc...}

Figure 2: A diagram of the layout of the proposed relational database.

User-Interface

The app will have a "front-end" graphical user interface (GUI). The GUI will focus around an interactive map and needs to accomplish four principal functions. The map will let the user draw the locations of interest and allow the datasets to be visualized and manipulated, a series of controls will let the user specify other parameters, and finally, a submit button will package the user's choices and interface with the "back-end" for analysis.

First, the user will need to have the ability to place multiple points on the map. These points are how the user will indicate the points of interest for the multiple route optimization scenarios that could be run with this service. The points will be designated as a start point, stop point or the location of the incident that needs emergency services.

Second, the map will let the user interact with the datasets available for the analysis. The roads and important buildings will be marked with different colors and the user will have the option to toggle on and off the visibility of the various layers. This step will be important for graphical confirmation that the data is imported and analyzed correctly. The user will also have a way to mark if a certain emergency services facility is unavailable and obstructions in the roads. These will be considered additional, case-specific parameters that modify the analysis. As an example, this app was inspired by a case where the fire department needed to get to BYU campus while there was a parade on University Avenue which blocked the most direct routes and used most of the available emergency services closest to the school. In simulating this scenario, the user could indicate that the nearest fire department is unavailable and university avenue is not available for travel.

Third, there will be multiple controls that let the user choose other parameters for the analysis. There will be controls for choosing parameters not indicated on the map. The most important choice will indicate if the user wants to route two points together or route from an emergency building to an incident location. Additional controls might include; format for downloading or saving the results of the analysis, whether to optimize the route by distance or time, or specifying which emergency services need to respond to the problem.

Finally, the user will have a submit button to initiate the processing. First, the user will need to be able to review the selections made. Then, after submitting, a confirmation message will display. The app will then take the user's location marks, modifications to the network, and analysis options and pass it to a "back-end" processing workflow.

Processing Workflow

The app's "back-end" will consist of a database of the shapefiles of vector data, a log of previous submission requests and results, and a function for connecting to the ArcGIS Network Analyst tool.

Processing begins when the user's input is passed from the GUI "front-end" to the processing "back-end." The app will begin by reading the choices made by the user.

It needs to determine the specific type of case, routing two points or a building to an incident and if the user has specified changes to the network. The app will then choose the appropriate data and functions that it will need to use to pass data to the Network Analyst tool. After processing the data, the criteria and response should be added to the list of past analyses performed and then displayed to the current user on the map.

Storage and License

Our code will be stored on Github.com in a free individual account. A free individual account will allow us to share ownership of the code with collaborators. In our case, the collaborators will be our three team members.

Our source code will be licensed under an MIT license. The MIT license is the most popular license for code stored on Github (Balter, 2015). The MIT license is short and will allow us to do anything we need with our app. The following MIT license from Open Source Initiative (OSI) will be included in the source code:

> Copyright <YEAR> <COPYRIGHT HOLDER>
>
> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Responsibilities

The main responsibilities involved in building this web application will most be comprised of the following:

- Design of the GUI
- Acquisition of the vector data we plan to store
- Database setup
- Backend computations via ArcGIS server

Our group plans to all work on the design of the GUI, Mitch Hadfield has been assigned to acquiring the vector data that we will store on the Geoserver and maintaining that aspect of the project. Wade has been assigned to the setup of the database and the maintenance of that aspect of the project. Riley has been assigned to take care of the backend computations that will be performed through the ArcGIS server on campus.

We realize that these tasks may not all require the same amount of effort, so we are willing to help each other out where needed, and we all hope to review one another's work so that we can gain a complete understanding of the project.

References

Balter, Ben (2015-03-09). "Open source license usage on GitHub.com". github.com. Retrieved 2015-11-21. 1 MIT 44.69%, 2 Other 15.68%

"Open Source Initiative OSI – The MIT License: Licensing". Open Source Initiative.