

# **DYNAMIC GENERATION OF HURRICANE EVACUATION ROUTES**

-Jon Jones

# Purpose

- ▣ To create a tool that provides a route out of the predicted path of the storm from a given location. The tool will not be given a specified destination, but will try to find the closest destination outside of the path of the hurricane, being constrained by time and route barriers.

- ▣ The route barriers will be identified road sections that would be impassable under certain weather conditions. Weather data will be downloaded and processed from national weather sources. This predictive data describes conditions for set time intervals. The processing will use all these inputs to determine barriers in 3 hour blocks of time. When the model is run a start time will be given and the data from that time will be loaded.

# Motivation

- ▣ - To explore the use of changing data inputs on route creation.
- ▣ - To leverage the power of Network Analyst combined with other geoprocessing tools.
- ▣ - To gain insights on the costs and technical challenges of solving complex routing problems.

# Architecture

- ▣ - ArcGIS Desktop – ArcInfo license
- ▣ - Network Analyst Extension
- ▣ - NAVTEQ NA SDC road network
- ▣ - weather and elevation data
- ▣ - ModelBuilder and Python scripts

# Data -Degrib ([www.noaa.gov](http://www.noaa.gov))

- ▣ - Downloads, reads and converts forecast data from the National Digital Forecast Database
- ▣ - gets current and historical data
- ▣ - saves to shapefile
- ▣ -gives results by region

# Data- NOAA

- ▣ - supply current and historical storm forecast data
- ▣ - includes storm surge and hurricane path
- ▣ - updated every three hours

# Data Downloading

- ▣ NOAA does not seem to currently provide a web service for getting GIS data, it would have to be scraped through a client application.
- ▣ Degrib is an implementation of the MDL GRIB2 decoder, which could possibly be used to create an automated download tool for NDFD data.



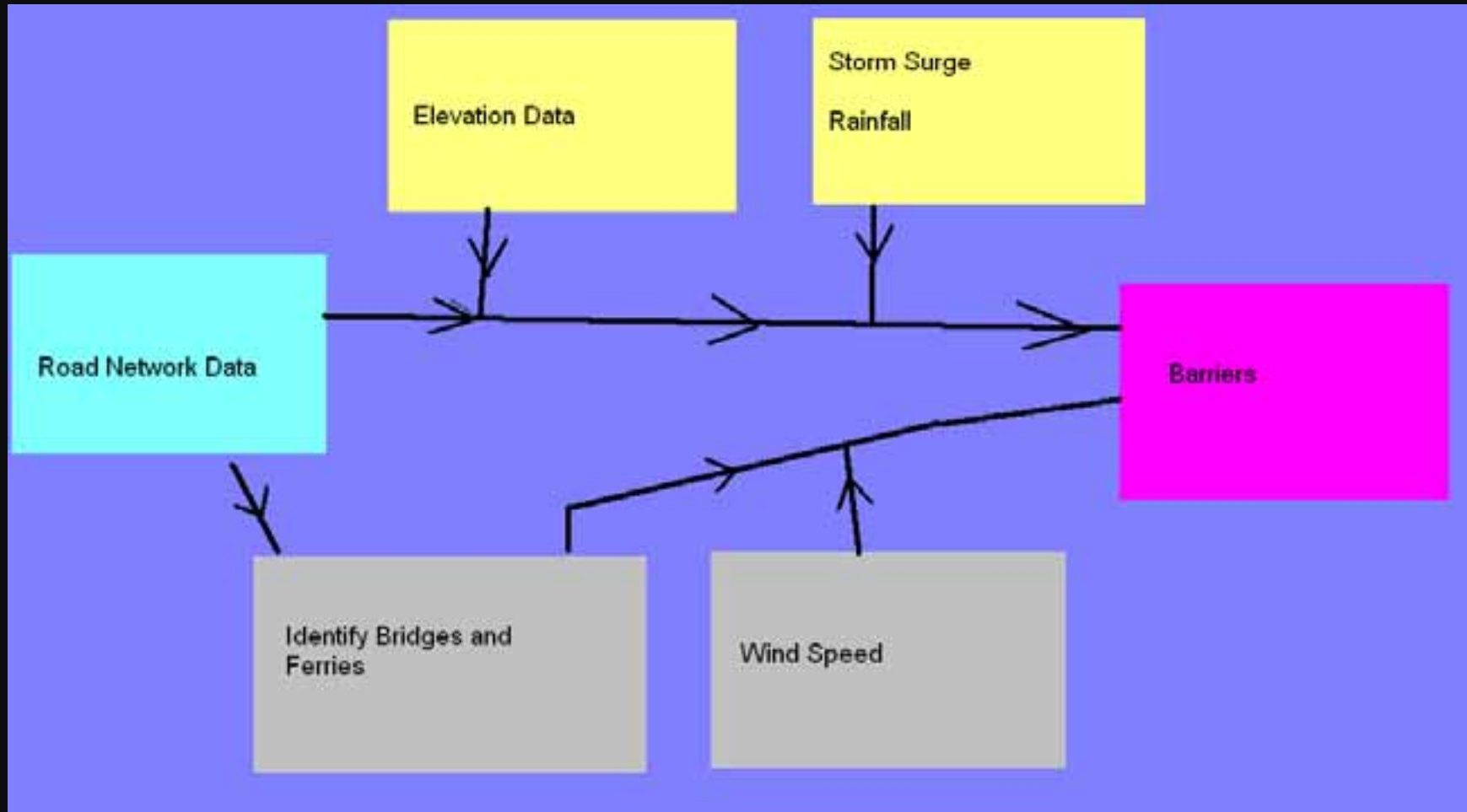
# Data - inputs

- ▣ - wind speed - Degrib
- ▣ - wave height – Degrib
- ▣ - rainfall – Degrib
- ▣ -storm surge – NOAA
- ▣ - hurricane path – NOAA
- ▣ - elevation – NED
- ▣ - vegetation - ?
- ▣ - anything that can block a road in a storm

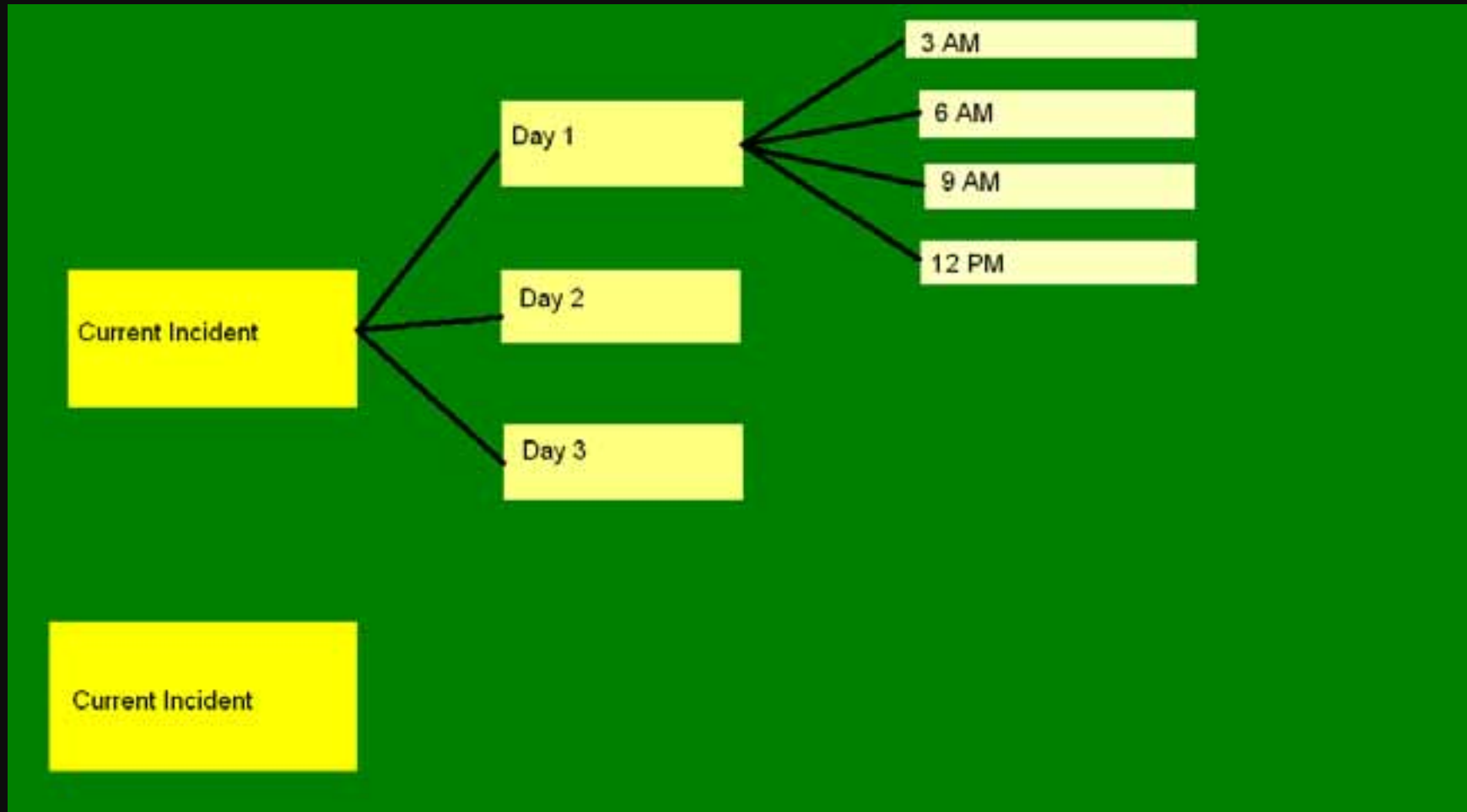
# Data - constraints

- ▣ - as data sources are added, the amount of processing goes up
- ▣ - the potential exists to create so many barriers the model won't run
- ▣ - reliable, meaningful data may be very expensive
- ▣ - must differentiate between factors that *may* block a road and that *will* block a road

# Creating Barriers



# Data Storage

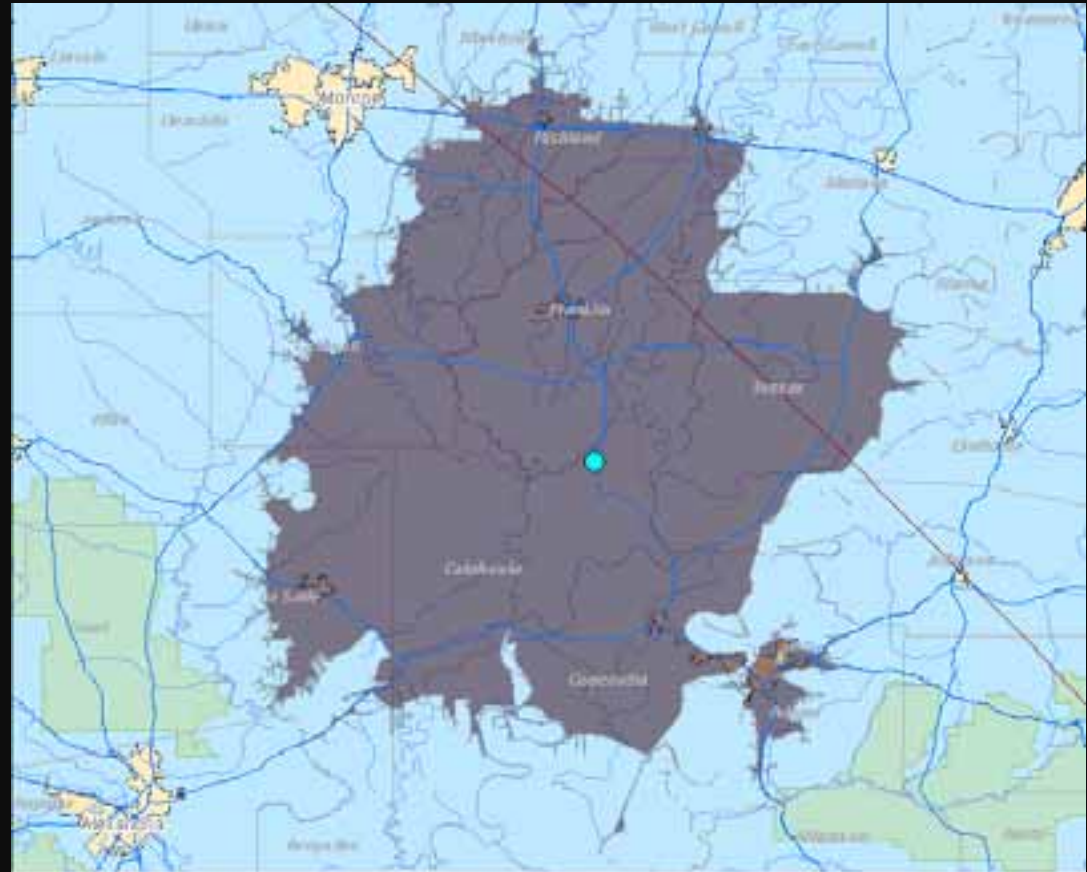


# Route Algorithm

- ▣ The location is added and geocoded, and the start time is given and used to load the appropriate set of barrier points and hurricane path.
- ▣ The process starts by clearing various tables and doing setup procedures.

The initial service area is calculated. In this example the area is conveniently located on the edge of the hurricane path.

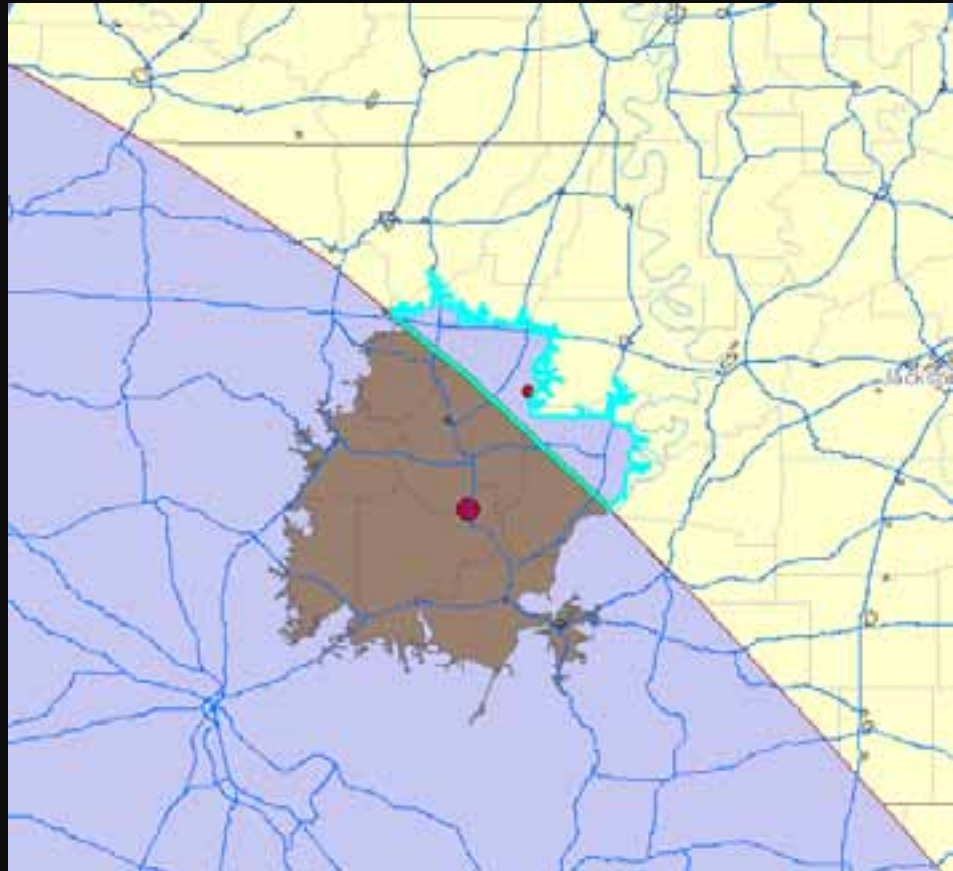
The hurricane path border has been converted to a line feature. The service area is intersected with this line, and if a feature results, it is saved.



If the intersection returns a feature, the process only goes through one iteration. A symmetrical difference tool is run on the area and the path to determine which areas are not shared.



The results of the symmetrical difference is intersected with the line where the service area meets the hurricane path (line), and selects the part that is outside of the path. A centroid point is determined for this area.





A regular route solver is run using the barrier set associated with the departure time. We are out of the path of the hurricane.



## Some things to consider so far...

- ▣ First, what time constraint are we going to use for the service area? The features loaded are valid for a 3 hour window, but that is going to make a huge service area and would only be valid for normal traffic conditions. For this simulation I am using 1 hour constraints to simulate backed-up evacuation traffic.
- ▣ Second, there are more meaningful destination points to choose from than centroid points. If some shelter data could be obtained, then the person could be routed to one or more appropriate spots.

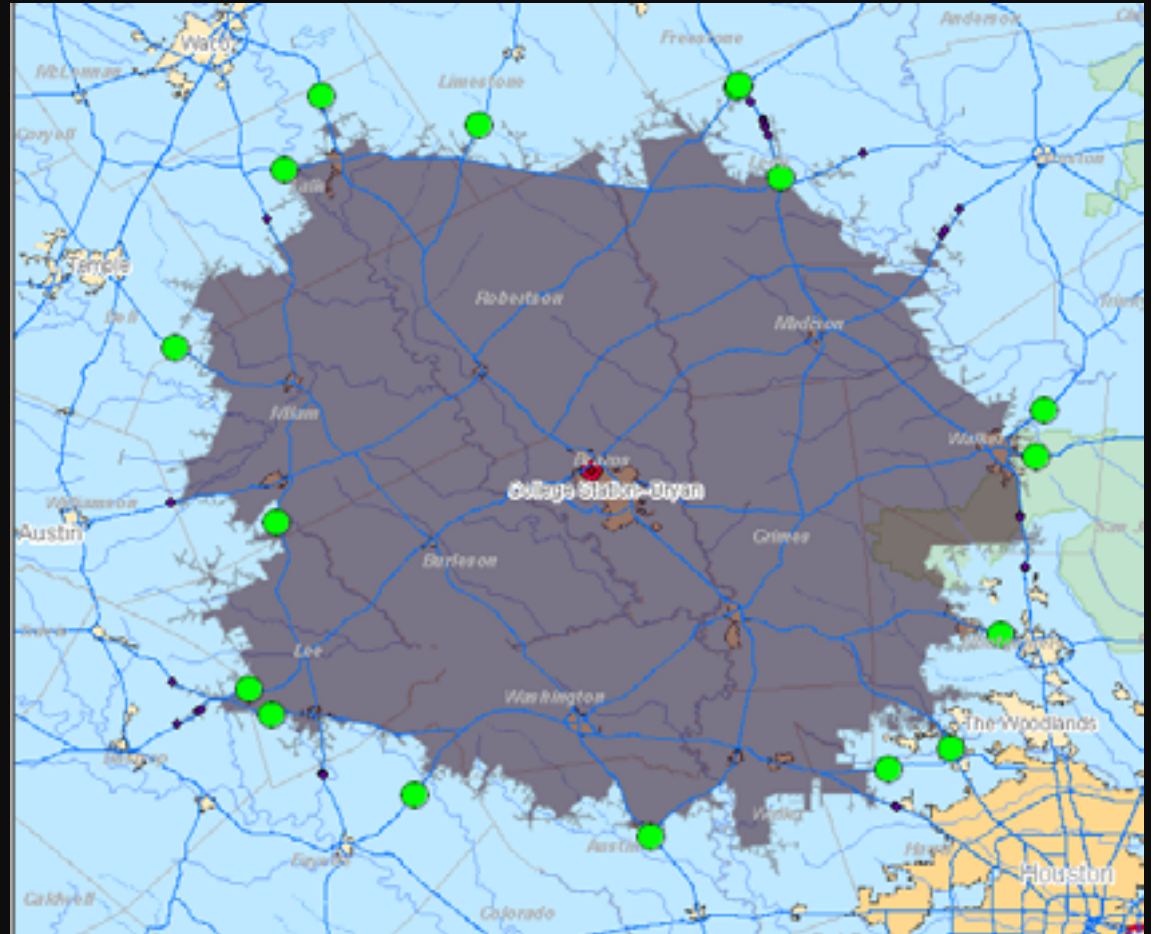
Suppose the hurricane path cannot be escaped in one service area? Here is an example of another start point farther inside the hurricane path. The area is calculated, but no intersection is found. This branches into a new section of the routine.



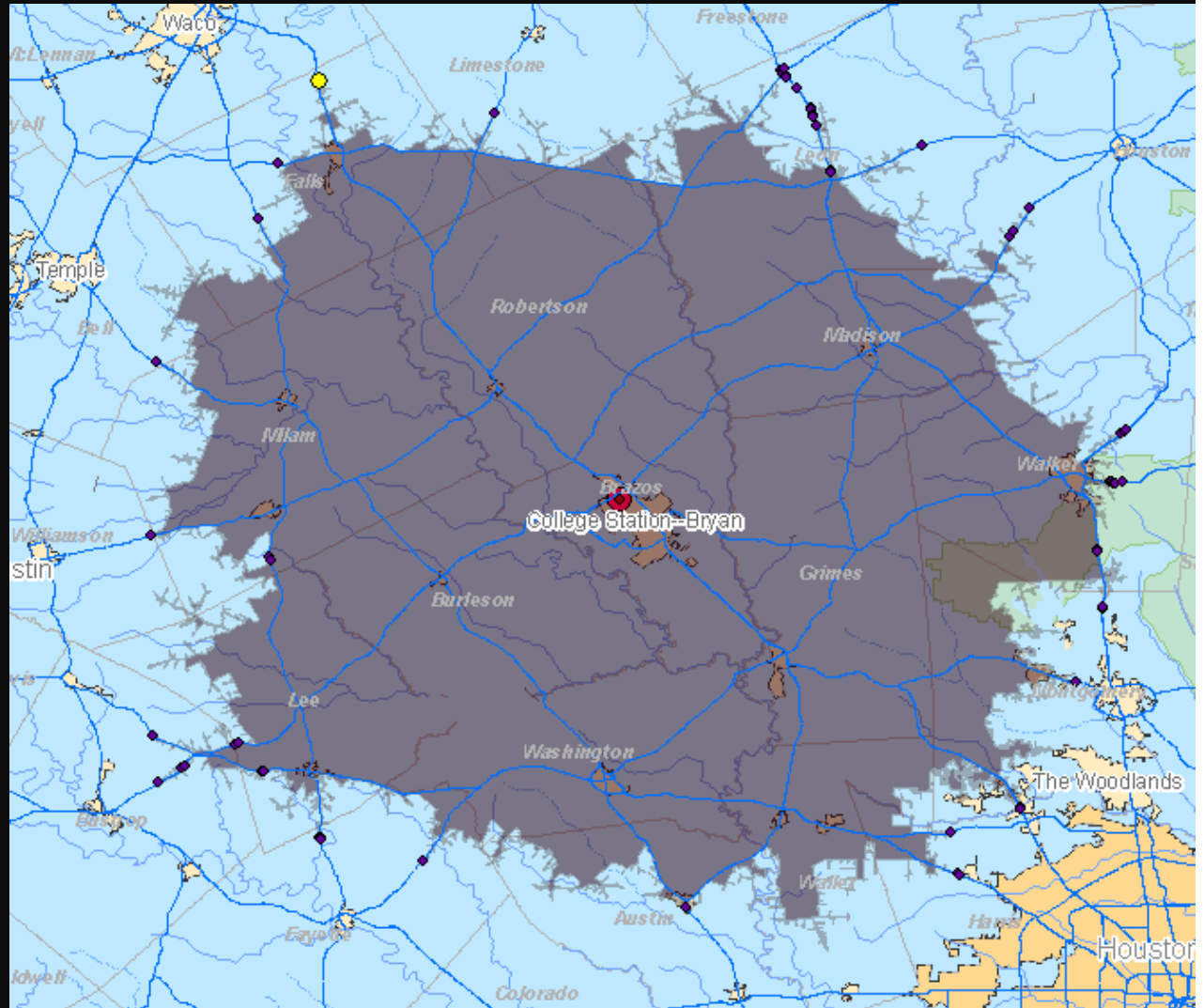
The border of the service area is created as a line, and intersected with major roads. The result should be the starting point for the next round of service areas, but the points tend to cluster where the same road hits the border at several spots.



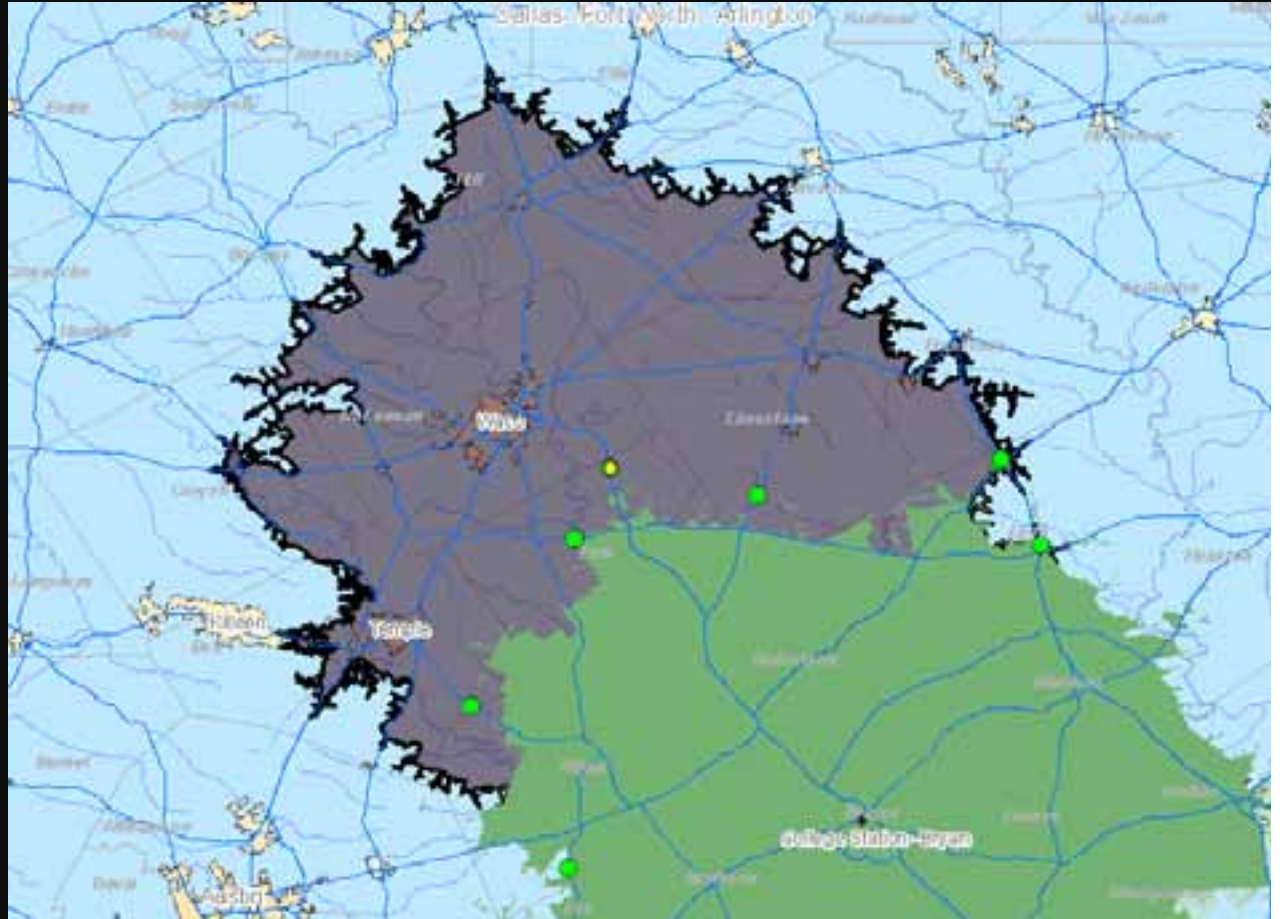
The points are dissolved into multipoint features based on road name, then converted to single points, making sure that the option is checked to make the new point one of the actual original points, and not the centroid of the multipoint feature. Now the set is a bit more manageable.



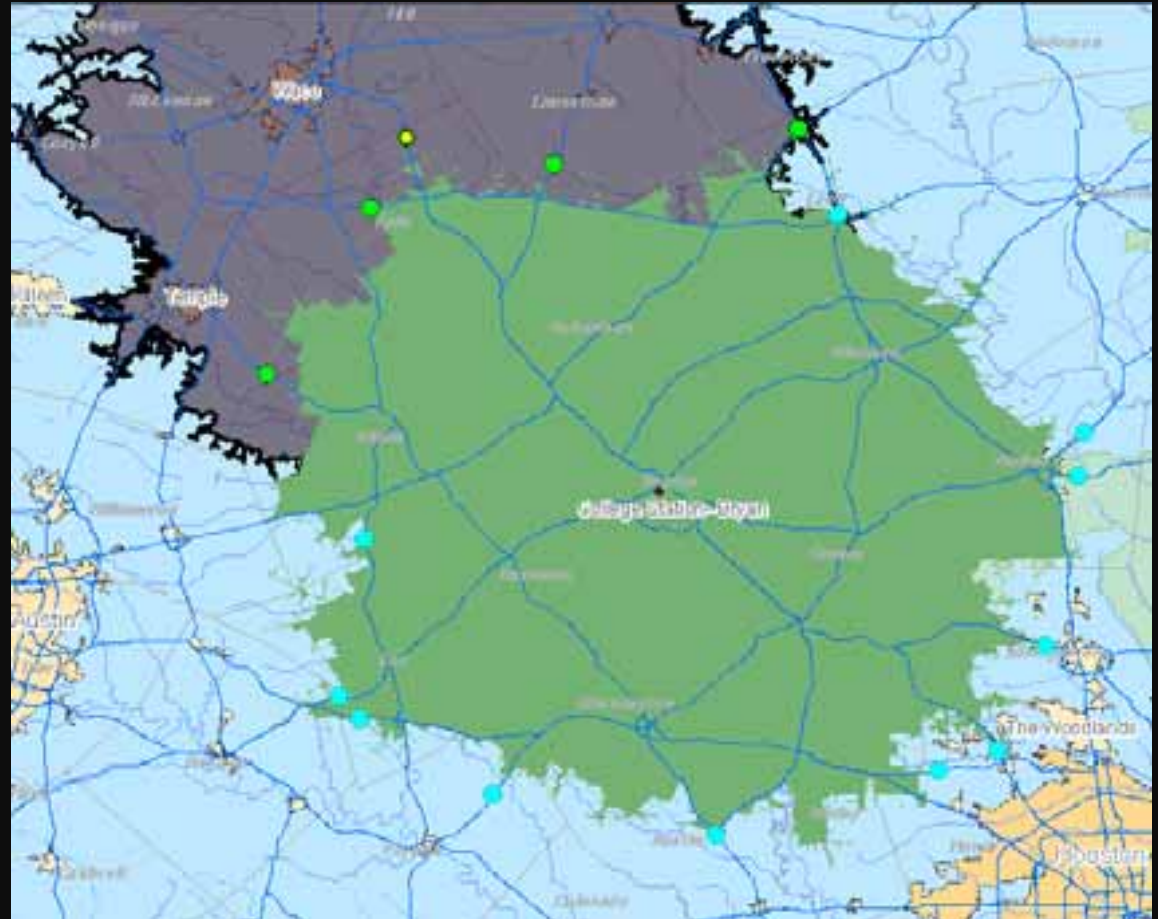
- The road points on the border are analyzed to determine which one is nearest the edge of the hurricane path. Here that point is the yellow one in the top left corner of the service area.



- The selected point is used to run another service area solution using the barriers for the next time block. It is then tested in the same way as before against the hurricane path from the next time block. This one also failed to escape.



- The points that represented the intersection of the original service area are selected based on which ones are completely outside of the new service area.





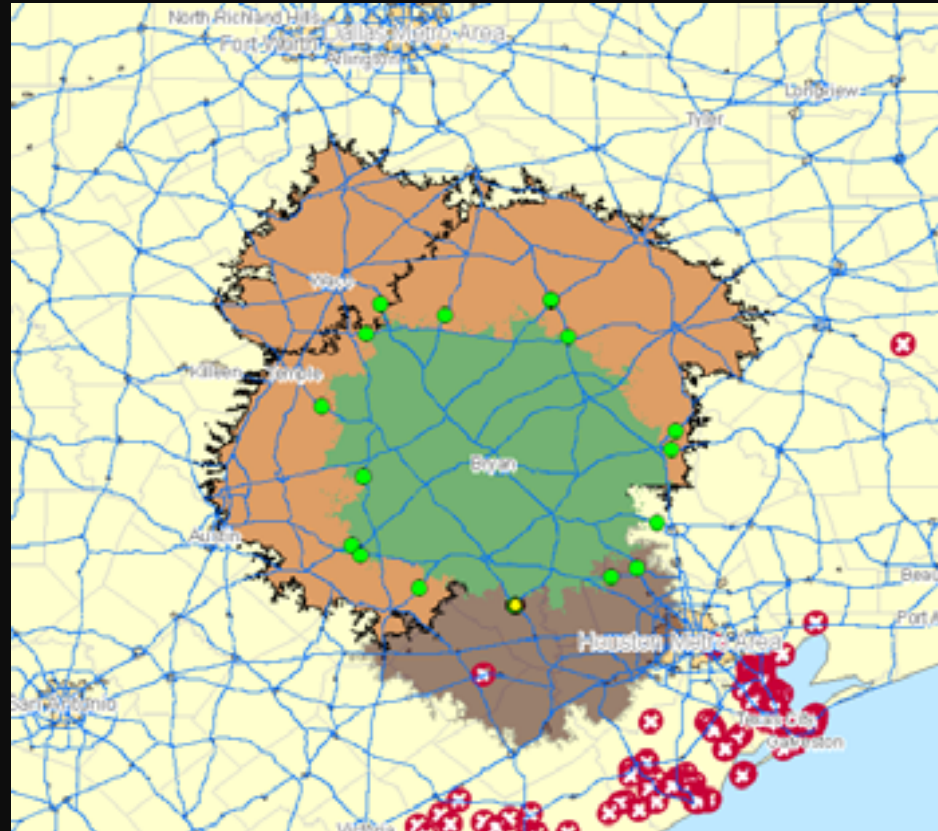
- The 'Near' analysis is run again on the points, but this time to determine which is closest to the service area just created. Also, the service area is appended to a polygon that will hold all the second-generation service areas.



- The 'Near' analysis is run again on the points, but this time to determine which is closest to the service area just created. Also, the service area is appended to a polygon that will hold all the second-generation service areas.

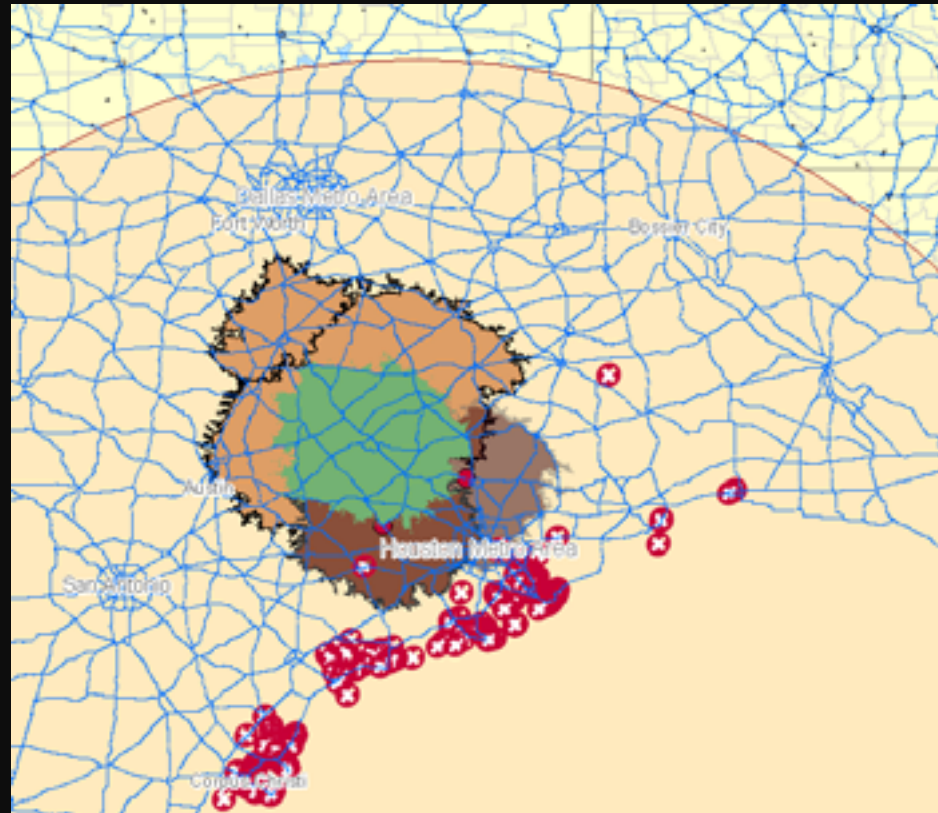


- The algorithm continues to search...





- At this point the algorithm stops without results because none of the service areas ever crossed the hurricane path boundary, and we ran out of available start points.



# Real World Issues

- The scripts and models require supervision to run because of data locking and access issues. A seamlessly running model would require quite a bit of code to manage errors and even then might not overcome the technical hurdles.

# Real World Issues

- ▣ -The choice of elevation points for road flooding was based on untested assumptions. There are reliable sources of this data for sale.
- ▣ -Although the model iterates completely around the first service area, it is an unlikely scenario where the first service area of the second generation would not be the only viable one, since it was chosen by proximity to the hurricane path edge. Possibly there could be a 'quick route' option that only runs the first two service areas, or maybe the first three.

# An Alternate Model

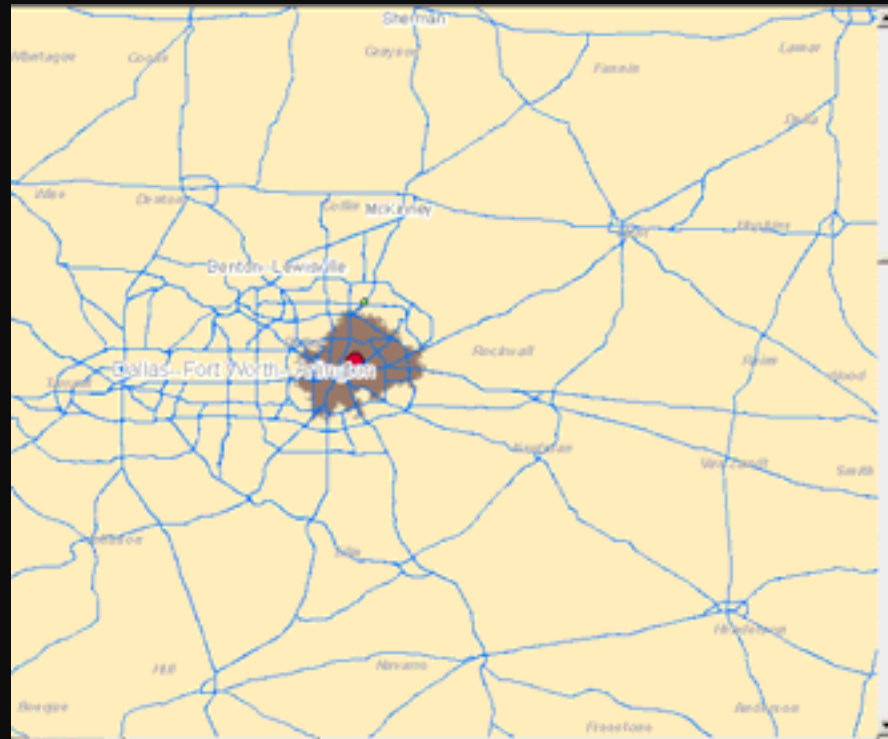
- ▣ Assuming that the point of the initial polygon that is closest to the edge is the only viable choice, perhaps a better route would be to take that point, make another service area, and then find the point on the second generation service area that is closest to the edge of the hurricane path, and generate a third service area.



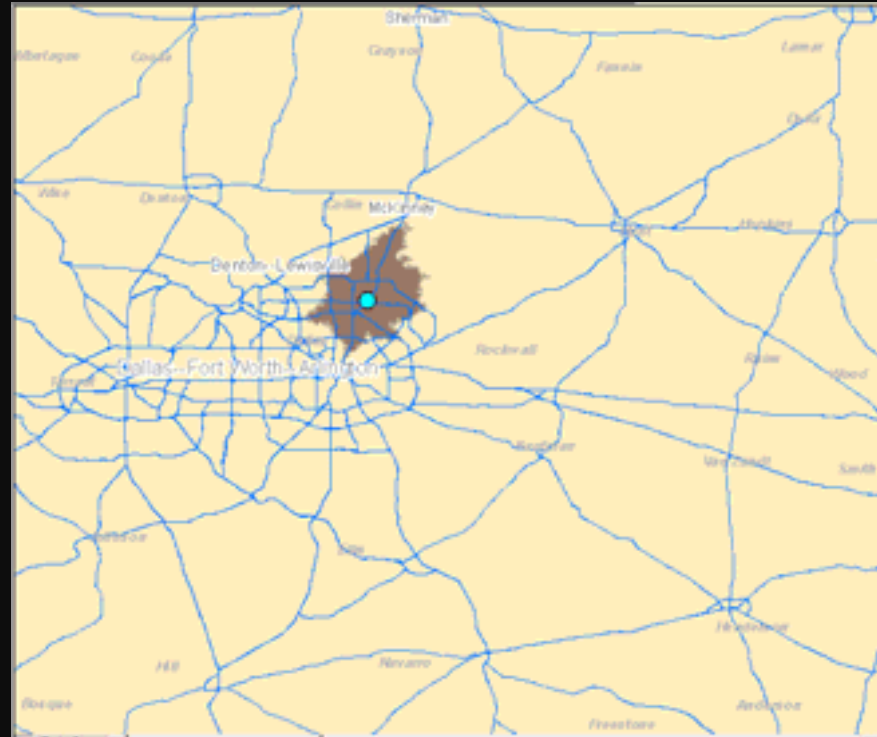
# An Alternate Model

- ▣ This approach would not waste time and resources exploring routes that are not moving towards the closest edge. This would allow the model to traverse greater distances that might not be feasible with the other model.

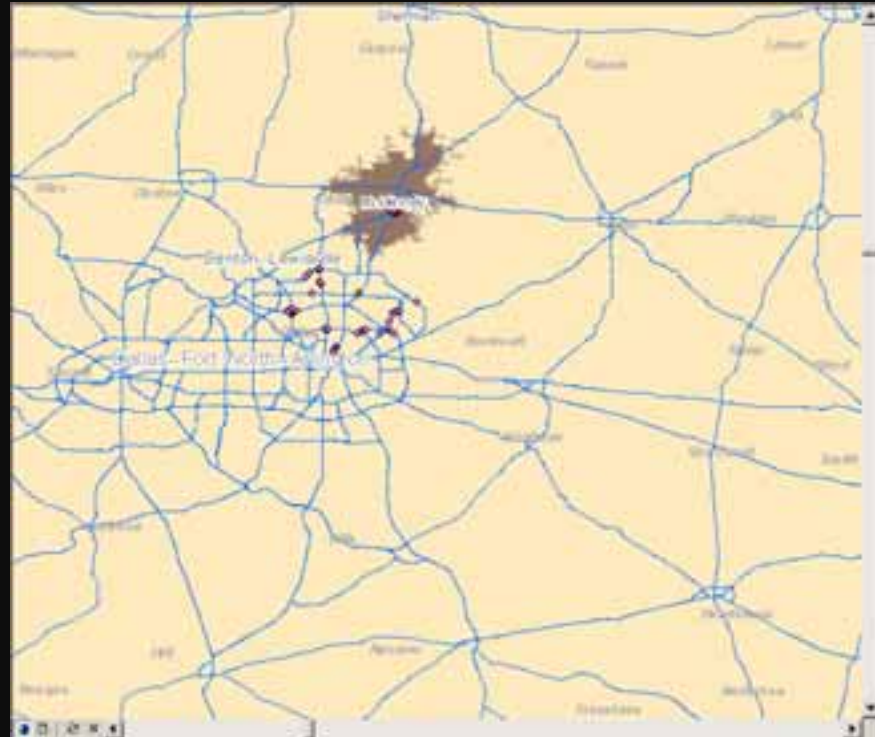
- The initial service area is calculated as before.



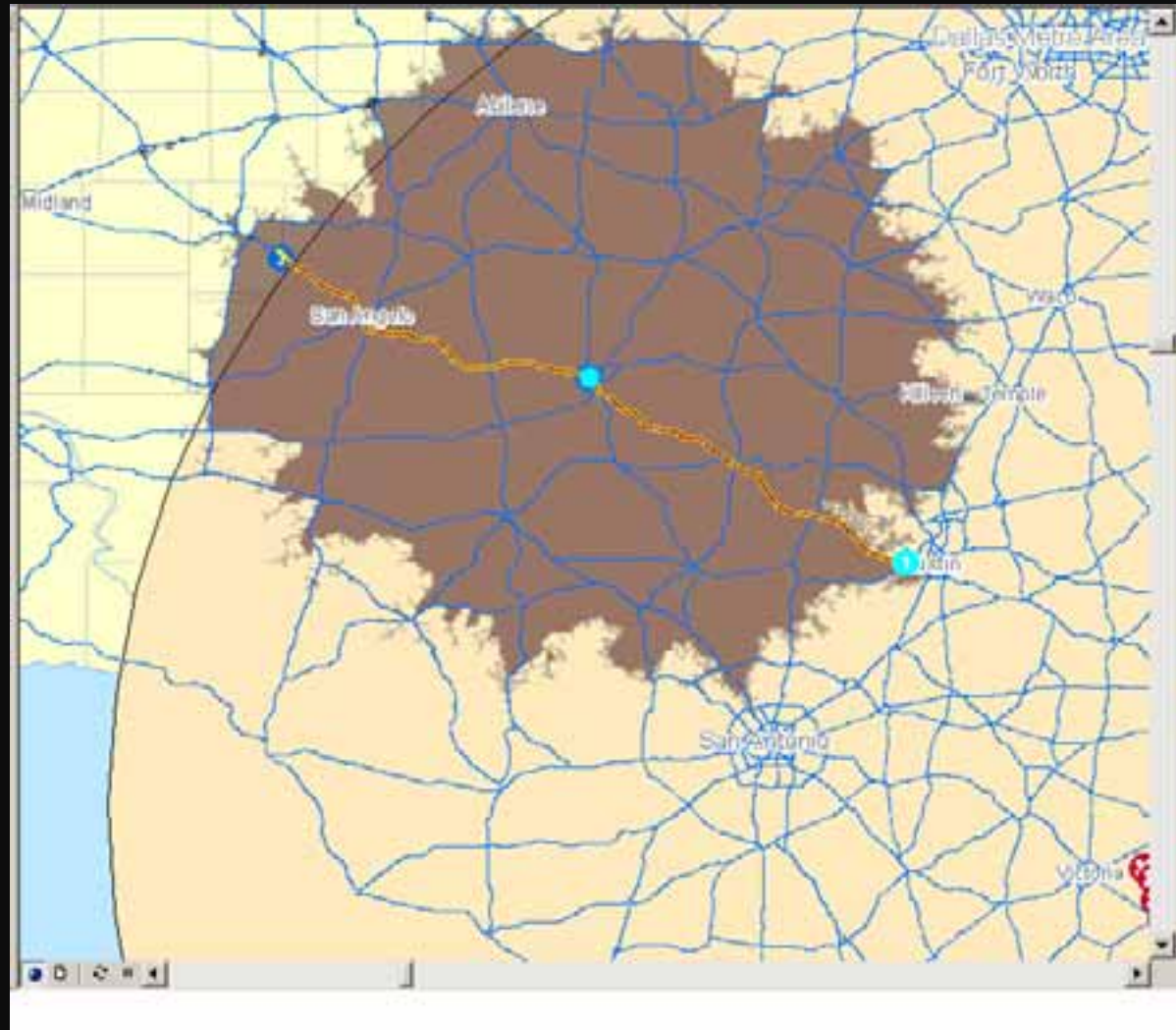
- The second service area is calculated as before. This time the polygon is not saved, only the start point. The new nearest intersection to the hurricane path is calculated.



- The third service area is calculated in the same manner. The algorithm continues to march towards the edge of the storm path until it reaches the edge or a control value shuts it off.

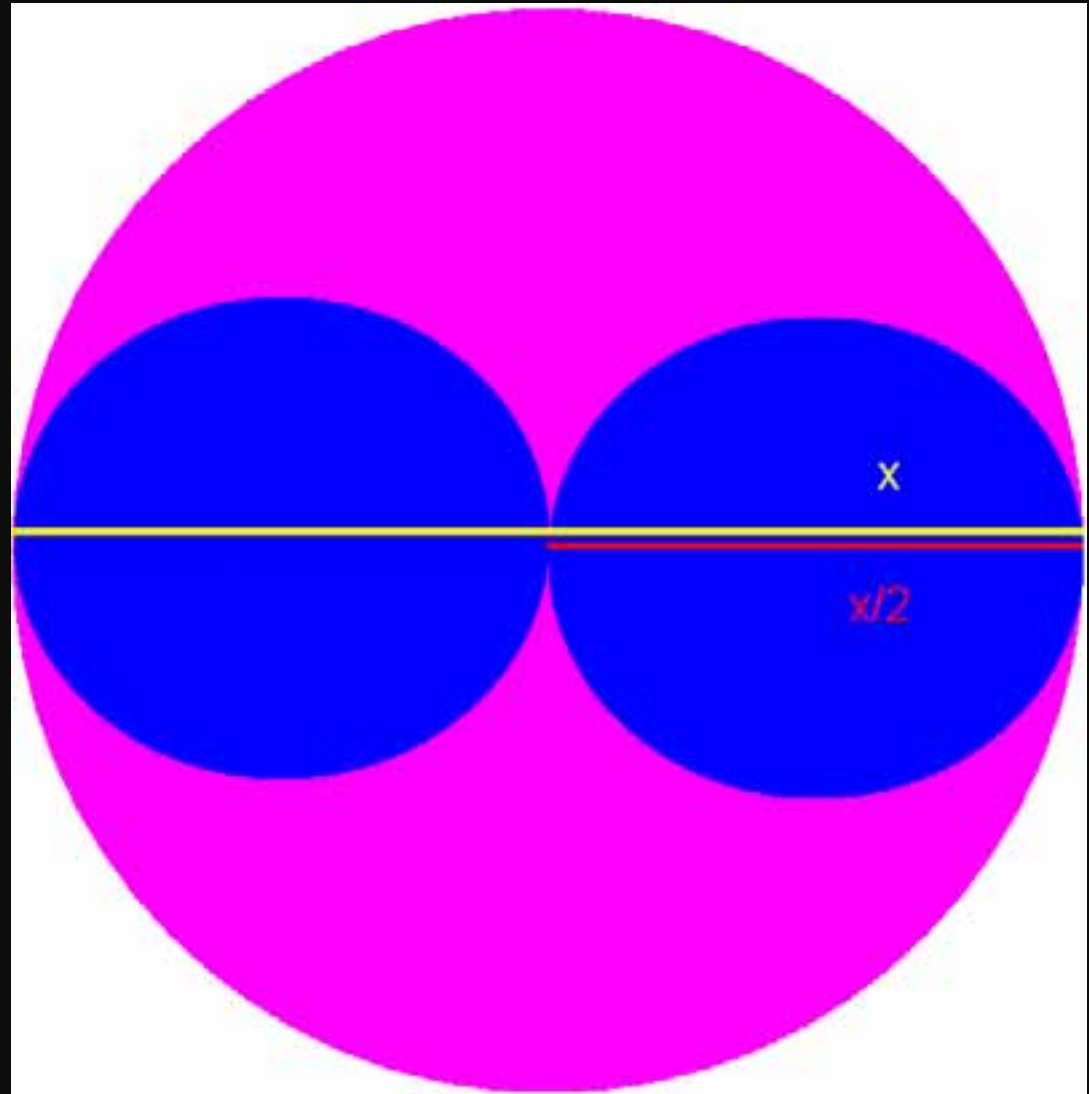


- If the a service area escapes the path, the saved start points are used to create an escape route.



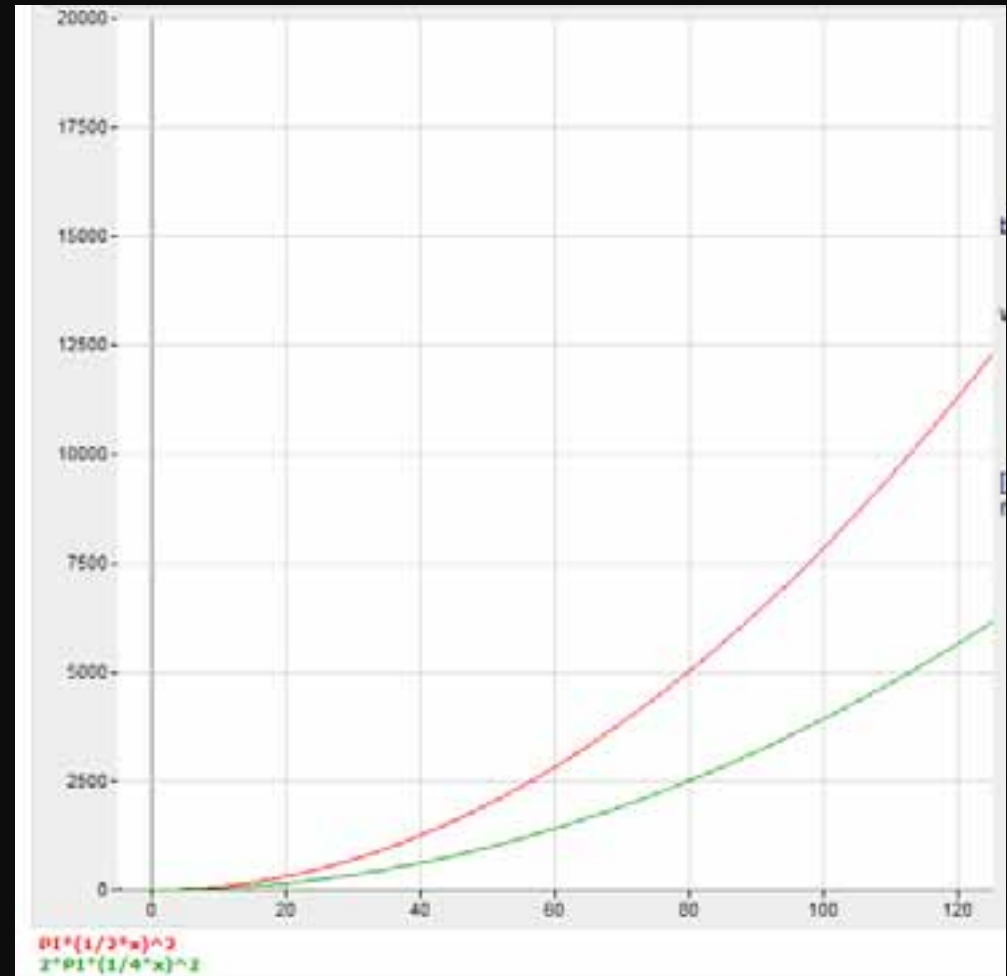
- ▣ A question arises as to how big a service area to run in order to get maximum performance.  
Theoretically, big service areas waste time exploring routes that are not getting nearer the edge of the storm path.
- ▣ I assumed that the data functions were a constant expense, the geoprocessing involving perimeters was a scalar variable, and that the service area solver was quadratic.

Area of pink circle =  
 $\pi(1/2x)^2$   
Area of blue circles =  
 $2\pi(1/4x)^2$



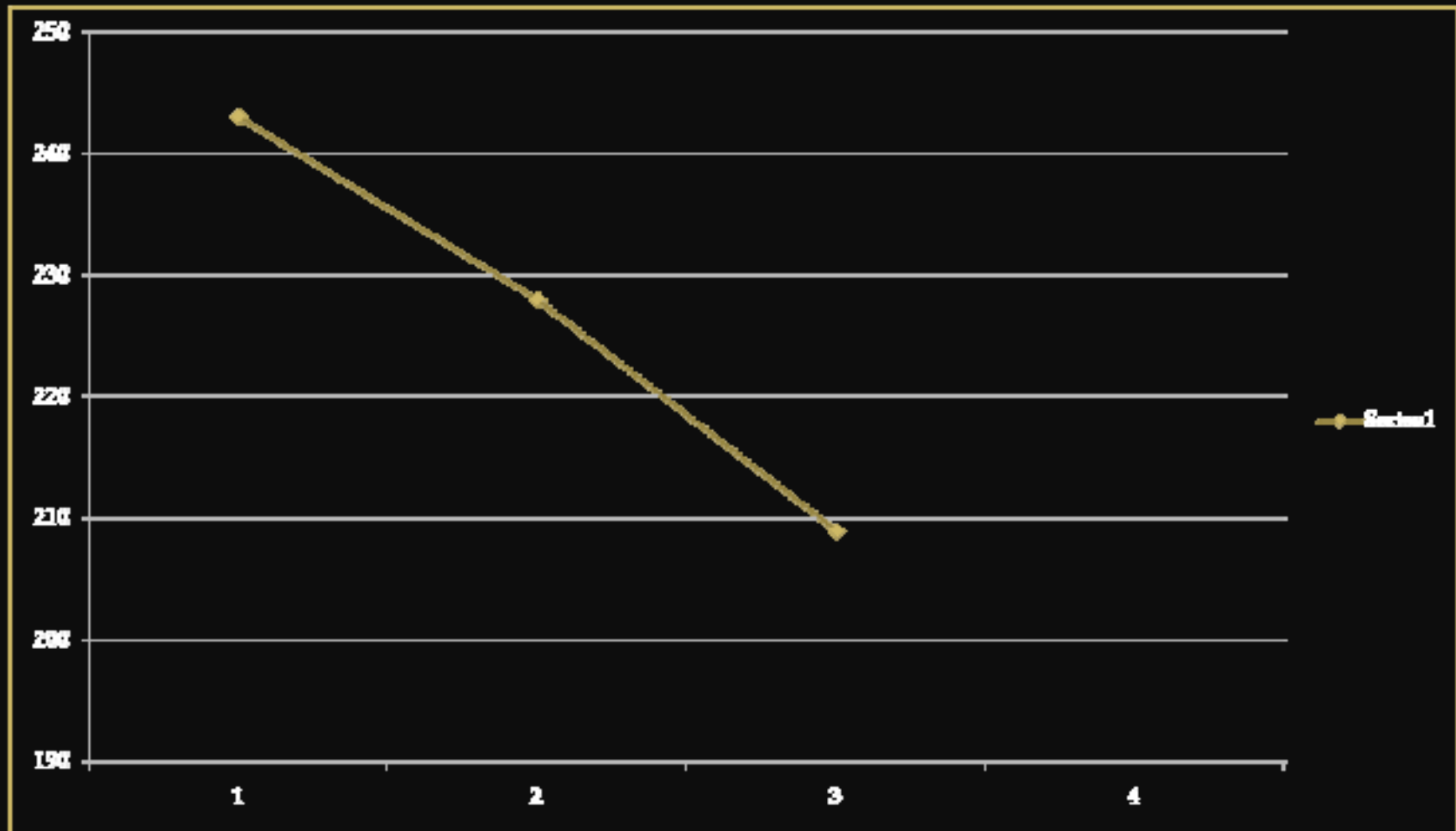
# Predicted Performance

The other variables should become insignificant as the model scales up and the quadratic cost of the solve should make any input cost twice as much using service time  $x$  than using time  $x/2$ .

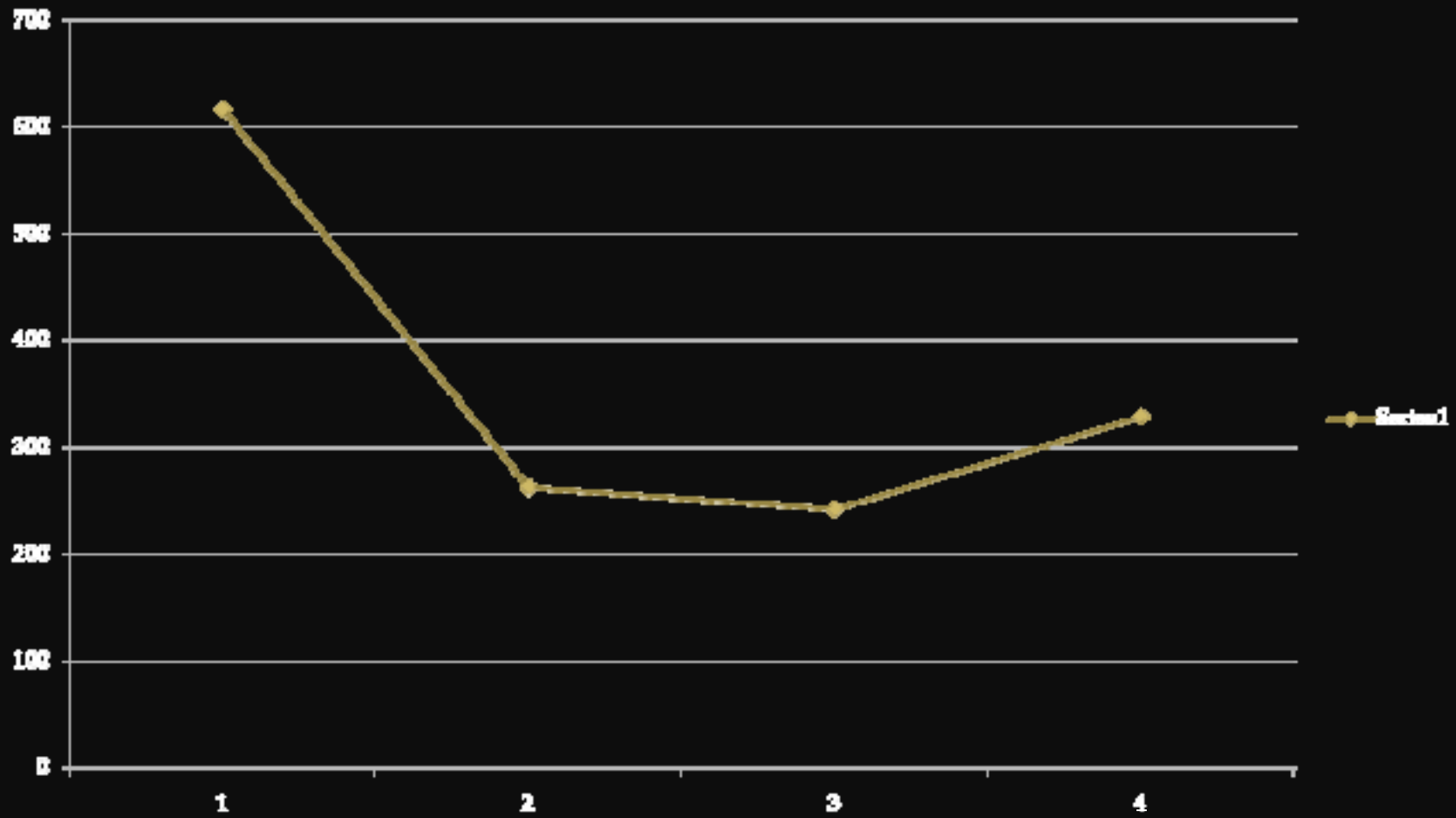




# Performance



# Performance 2



# Explanation

- All processes except for the solve turned out to be a constant cost.
- The solver was not quadratic. It was barely linear. This is probably because most large areas are largely devoid of major road networks. So while a 15 minute service area might not have many segments in its network, most of these segments are not going anywhere complex even when extended out. That is to say, the area of the polygon is increasing greatly, but it is mostly empty.

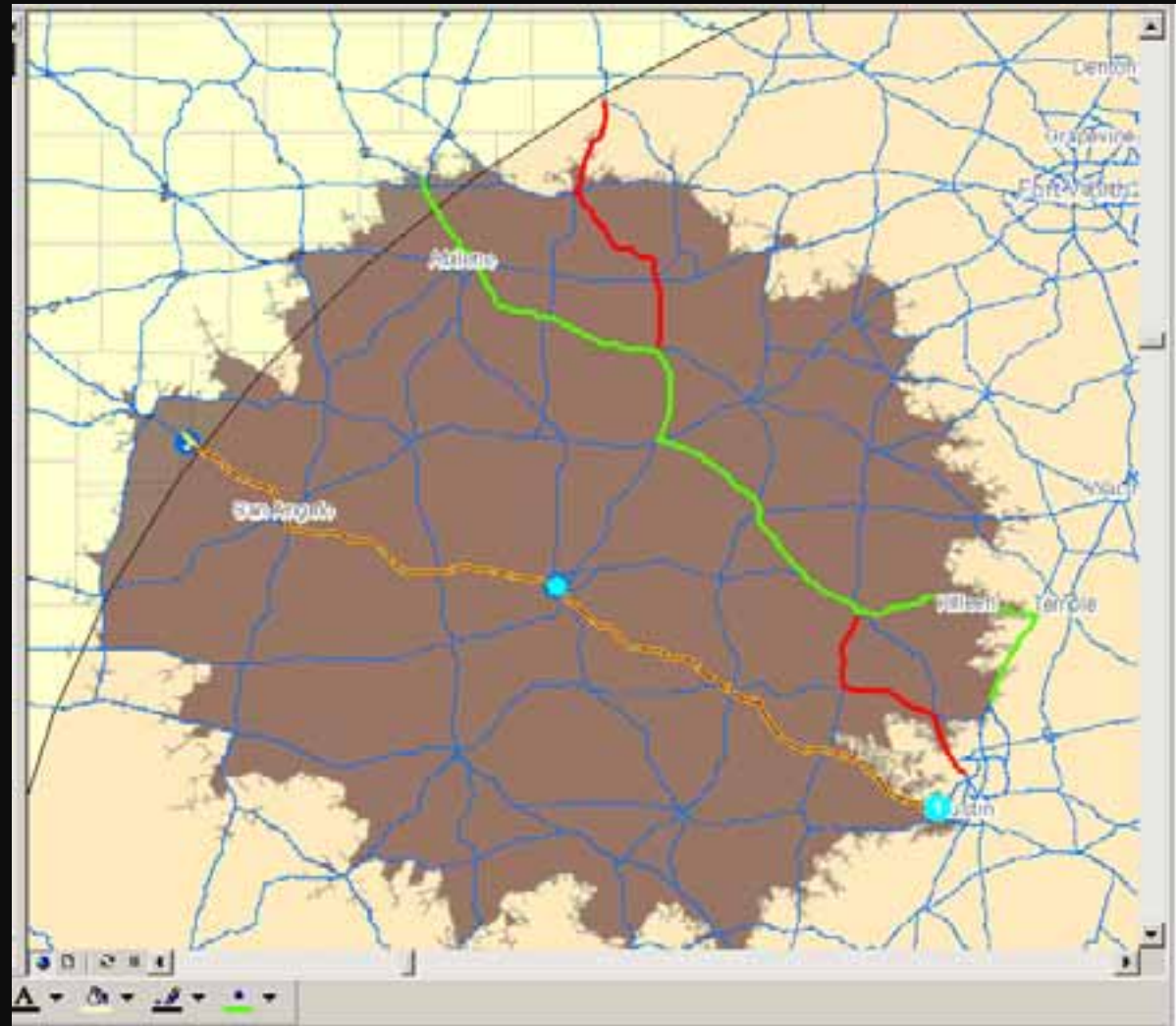
# Explanation

- However, the second test showed that when service areas get large enough, they do encounter other areas of network intensity that are not part of the path leading from the storm.
- -The optimal size for this study area seems to be about one hour. The larger service area also may give a better escape route in the end, because smaller areas tend to meander as they move towards the edge.

• Red – 15 min  
service areas -225  
miles

Green – 30 min  
service areas – 234  
miles

Orange – 60 and  
120 min service  
areas – 200 miles



# Limits

- ▣ - All roads on the network may not necessarily be available because of controlled evacuation routes.
- ▣ - The model has the most application during the time you would least want to be evacuating!
- ▣ - doesn't take into account the congestion that usually accompanies evacuation.

# Results

- ▣ - Experience with and appreciation for the power of Network Analyst.
- ▣ - insight into working with routes that are analyzed by segments.
- ▣ - experience building complex geoprocessing systems.

## Next Step

- ▣ - The model could be made to attempt evacuation to a specific location. The location would take the place of the hurricane path boundary in the proximity analysis.
- ▣ -The performance of the server needs to be tuned to handle real-world extents.
- ▣ - the preprocessing should be automated to the extent possible.



# Acknowledgements

- ▣ - Frank Hardisty, PSU
- ▣ - GeoDecisions Staff
- ▣ - National Oceanic and Atmospheric Administration (NOAA)
- ▣ - Hurricanes Ike and Isabel